PONTIFICIA UNIVERSIDAD
CATÓLICA DEL PERÚ

Escuela de Posgrado

Aprendizaje Estadístico Supervisado con Máquina de Soporte Vectorial

Tesis para obtener el grado académico de Maestro en Estadística que presenta:

Sergio Daniel Falcón Cisneros

Asesor:

Carlos Nilberto Véliz Capuñay

Lima, 2024

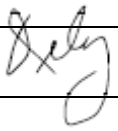
Informe de Similitud

Yo, CARLOS VÉLIZ CAPUÑAY, docente de la Escuela de Posgrado de la Pontificia Universidad Católica del Perú, asesor(a) de la tesis/el trabajo de investigación titulado “APRENDIZAJE ESTADÍSTICO SUPERVISADO CON MÁQUINA DE SOPORTE VECTORIAL”, del/de la autor(a) / de los(as) autores(as)

SERGIO DANIEL FALCÓN CISNEROS
dejo constancia de lo siguiente:

- El mencionado documento tiene un índice de puntuación de similitud de 5%. Así lo consigna el reporte de similitud emitido por el software *Turnitin* el 27/11/2023.
- He revisado con detalle dicho reporte y la Tesis o Trabajo de Suficiencia Profesional, y no se advierte indicios de plagio.
- Las citas a otros autores y sus respectivas referencias cumplen con las pautas académicas.

Lugar y fecha:
Lima, 27 de noviembre de 2023

Apellidos y nombres del asesor / de la asesora: Véliz Capuñay, Carlos	
DNI: 07911208	Firma 
ORCID: 0009-0005-1529-080X	

Dedicatoria

Dedico este trabajo de tesis a mi esposa Aixa, el amor de mi vida, por enseñarme que la felicidad y el amor verdadero existen. A mis hijas Alessia y Alondra, que iluminan mis días y son el motor que me empuja a ser mejor cada día. A mis padres Sergio y Celmira, por todo el amor, esmero y la educación que he recibido y sigo recibiendo de parte de ellos. Y a mis hermanos Luis y Javier, por su cariño y apoyo.

Agradecimentos

A mi esposa Aixa, por su amor, paciencia y comprensión, y por darme fuerza en los momentos difíciles.

A mis padres, por su amor, cariño y por el esmero que pusieron en mi educación.

A mi asesor de tesis, el Dr. Carlos Veliz Capuñay, por el apoyo recibido durante el desarrollo de este trabajo.

A los profesores de la maestría en estadística por todo lo aprendido durante el programa de estudios.

A Shen, Yanett y Leda, por todo el apoyo y comprensión brindados en el centro laboral para completar el trabajo tesis.

Resumen

Actualmente las organizaciones recolectan datos en grandes volúmenes y de fuentes muy variadas. Para dar sentido y convertir los datos en información útil es necesario utilizar técnicas que permitan encontrar y entender las relaciones ocultas en los datos. Generalmente, la relación que nos interesa estudiar es cómo predecir un evento utilizando un conjunto de variables. Sin embargo, muchas veces la relación entre los datos es muy compleja y no puede ser analizada adecuadamente usando las técnicas más conocidas, dado que éstas suelen tener supuestos que no necesariamente se cumplen. Por ello, es importante conocer técnicas de análisis más complejas y flexibles.

Esta tesis busca ser un instrumento de ayuda en el aprendizaje y uso de nuevas técnicas para estudiar los datos, lo cual es relevante sobre todo en el medio local en el que este tema es poco conocido. Con este objetivo, presenta una revisión introductoria de la teoría del aprendizaje estadístico, la cual provee del marco teórico para que distintos métodos utilicen los datos para aprender, y usando este conocimiento puedan hacer predicciones sobre datos nuevos o diferentes. Luego se centra en un estudio exhaustivo del método de aprendizaje de Máquinas de Soporte Vectorial (SVM por sus siglas en inglés), introduciendo y aplicando las funciones Kernel. Este método se puede entender como una representación de los datos como puntos en el espacio, asignados de tal forma que exista una brecha grande que separe a los elementos diferentes.

Finalmente se pone en práctica la teoría estudiada aplicando el método SVM a datos de clientes de una entidad financiera. Esta entidad financiera usa predominantemente técnicas de aprendizaje estadístico simples y con varios supuestos; particularmente usa una de estas técnicas en un modelo que predice la propensión a la compra y persistencia del producto Seguro de Protección de Tarjetas. Por ello, la presente tesis se centra en aplicar el método SVM para construir una alternativa a este modelo.

Palabras-clave: Máquinas de Soporte Vectorial, Hiperplanos Separadores, Hiperplano Óptimo, Hiperplano Óptimo Generalizado, Producto Interno, Funciones Kernel, Datos Desbalanceados, Aprendizaje Estadístico, Clasificación Binaria, Modelos de Propensión.

Abstract

Nowadays organizations collect data in big volumes and from multiple sources. In order to give meaning and convert the data into useful information it is necessary to use techniques that find and understand the relationships hidden within the data. Usually, the relationship we want to study is how to predict an event using a set of variables. However, many times the relationship between the data is very complex and can not be studied properly using the most well known techniques, for they usually have assumptions that are not necessarily met. Thus, it is important to know more complex and flexible analysis techniques.

This thesis seeks to be a helping instrument in the learning and usage of new techniques to study data, subject that is relevant in the local media in which the subject in not well known. To this end, it presents an introductory review of statistical learning theory, which provides the theoretical framework that enables different methods to use the data to learn, and using that knowledge to make predictions on new or different data. Then it focuses on studying the Support Vector Machines (SVM) learning method, introducing and applying the Kernel functions. This method can be understood as a representation of the data as dots in the space, mapped in such a way that there exists a large gap that separates the different elements.

Finally we put in practice the learned theory by applying the SVM method to the client data from a financial institution. This financial institution uses mostly statistical learning techniques that are simple and have various assumptions; in particular it uses one of theses techniques in a model that predicts the propensity to buying and maintaining the product Card Protection Insurance. Therefore, this thesis focuses con applying the SVM method to build an alternative to this model.

Keywords: Support Vector Machines, Separating Hiperplanes, Optimal Hiperplanes, Generalized Optimal Hiperplanes, Dot Product, Kernel Functions, Unbalanced Data, Statistical Learning, Binary Classification, Propensity Models.

Índice general

Índice de figuras

Índice de cuadros

Capítulo 1

Introducción

El aprendizaje estadístico es el marco teórico de una serie de herramientas y técnicas para estudiar y comprender los datos. Éstas nos permiten encontrar información importante oculta en los datos (patrones, tendencias, predicciones, etc.) y descartar lo irrelavante.

Existen dos categorías principales para clasificar los problemas de aprendizaje estadístico: supervisado, y no supervisado. El presente documento se enfoca en el estudio del aprendizaje estadístico supervisado, mediante el cual se busca construir modelos que relacionen una o más variables de entrada con una variable respuesta, a fin de predecir o estimar esta respuesta en nuevas observaciones, o para comprender mejor la relación entre la respuesta y los predictores. Así, por ejemplo, podemos plantear un modelo que permita predecir las ventas de una compañía tomando como datos de entrada los presupuestos de publicidad; o predecir la probabilidad que un cliente caiga en default (deje de pagar un crédito) en base a variables sociodemográficas y/o económicas.

El aprendizaje estadístico supervisado busca encontrar funciones que relacionen una variable respuesta o de salida Y, con p variables de entrada o predictores $X = (X_1, X_2, ..., X_p)$ (James et al., 2013). Esta relación tiene la siguiente forma:

$$Y = f(X) + \varepsilon \, ,$$

donde f es una función fija pero desconocida (una caja negra) y ε es un término de error aleatorio independiente de X. Esta función f representa la información sistemática que X proporciona acerca de Y. El aprendizaje estadístico supervisado nos provee de técnicas y metodologías para estimar $\hat{f}$; es decir, estimar una aproximación a la función real f.

Según el tipo de la variable respuesta Y, el proceso de estimación de f recibe nombres distintos. Si Y es de tipo numérica o continua, el problema de aprendizaje es llamado regresión. Por otro lado, si Y es de tipo categórica el problema es llamado clasificación.

Dentro del aprendizaje estadístico podemos distinguir claramente dos enfoques para estimar la función f:

1. Enfoque Paramétrico. Este enfoque asume la forma de la función f. Por ejemplo: que existe una relación lineal entre X e Y. La principal ventaja de este enfoque es que, al

asumir la forma de la función f, el problema se reduce a estimar un conjunto de parámetros de dicha función, lo cual facilita también su interpretación. Sin embargo, el modelo escogido no necesariamente calzará con la forma verdadera y desconocida de f. Así, si un modelo hace una mala emulación de la realidad, las conclusiones y predicciones que obtengamos de éste podrían ser equivocadas (Breiman, 2001). Podemos remediar este problema usando modelos flexibles que adopten varias formas funcionales generales, pero esta alternativa eleva la complejidad del modelo haciendo más difícil su interpretación, e incrementa la cantidad de parámetros lo cual podría generar problemas de sobreajuste si hay pocos datos. Ejemplos de métodos que siguen el enfoque paramétrico son la Regresión Lineal, Regresión Logística, Análisis Discriminante Lineal, etc.

2. Enfoque No paramétrico. Este enfoque no hace suposiciones explícitas sobre la forma funcional de f. Más bien, se busca una estimación de f que se acerque a los datos. Los métodos que siguen este enfoque tienen el potencial de ajustar mejor a un rango más amplio de formas para f. Sin embargo, generalmente requieren de una gran cantidad de observaciones (mucho más que en el caso paramétrico) para obtener una buena estimación de f, y la interpretación del modelo suele ser más complicada. Ejemplos de métodos que siguen este enfoque son las Máquinas de Soporte Vectorial, Redes Neuronales, Árboles de Decisión y Regresión, etc.

Sin importar el enfoque usado se hace necesario evaluar el desempeño de los métodos de aprendizaje estadístico. Para ello necesitamos alguna forma de medir qué tan buenas son las predicciones realizadas por el modelo; es decir, qué tan cercano son los valores predichos por $\hat{f}(X)$ respecto a los valores verdaderos Y:

- Para los problemas de regresión suele utilizarse el Error Cuadrático Medio (MSE por sus siglas en inglés), el cual se define como:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}(x_i))^2$$

- Para los problemas de clasificación, suele usarse el ratio de error (ER), es decir, la proporción de errores cometidos al aplicar la función $\hat{f}$:

$$ER = \frac{1}{n}\sum_{i=1}^{n} I(y_i \neq \hat{f}(x_i))$$

  donde I es la función indicadora.

Cuando se aplica un método de aprendizaje estadístico generalmente interesa dividir los datos en por lo menos dos subconjuntos: los datos de entrenamiento, y los datos de prueba. Los datos de entrenamiento, como el nombre lo sugiere, se utilizan para entrenar o estimar la función $\hat{f}$. Los datos de prueba se utilizan para medir el desempeño o bondad de ajuste del modelo. Esto se hace para evitar problemas de sobreajuste del modelo, es decir, evitar

que el modelo identifique relaciones aparentes en los datos de entrenamiento, pero que no se cumplen en general.

En el aprendizaje estadístico no existe un único método superior que presente mejores resultados que el resto para todos los posibles conjuntos de datos. Por el contrario, es importante decidir qué método o métodos de aprendizaje son los más adecuados para cada conjunto de datos. Esta selección del mejor enfoque puede ser una de las tareas más complicadas en el aprendizaje estadístico (James et al., 2013).

Luego, es importante conocer una amplia gama de técnicas y métodos estadísticos con los cuales podremos estudiar los datos y elegir los que sean más apropiados para el problema en estudio, teniendo en cuenta las particularidades de los datos y los objetivos y restricciones del problema (por ejemplo: se busca explicar las relaciones entre los datos, o tener el mejor poder de predicción). Esto se vuelve más importante hoy, cuando las organizaciones recolectan cada vez más grandes volúmenes de datos de fuentes muy variadas como: Datawarehouse o Datalake, redes sociales, correos electrónicos, logs de aplicaciones y sistemas, Dataenrichment, etc.

El análisis de fuentes de datos variados y enormes (Big Data) representa un desafío importante y puede proveer a las organizaciones de información muy valiosa que de otra forma permanecería oculta. Algunas veces las relaciones ocultas en los datos son muy complejas y no pueden ser modeladas adecuadamente asumiendo supuestos restrictivos (por ejemplo, linealidad de los datos). Por ello, es poco lo que se puede lograr si, como sucede muchas veces, las organizaciones sólo conocen los modelos paramétricos más tradicionales.

Por este motivo, en el presente desarrollo de tesis estudiaremos el aprendizaje estadístico mediante la técnica no paramétrica Máquinas de Soporte Vectorial (SVM por sus siglas en inglés), el cual ha mostrado tener un buen desempeño en una gran variedad de escenarios, y es incluso considerado uno de los mejores métodos para resolver problemas de clasificación (James et al., 2013). En particular, se busca aplicar esta técnica para resolver un problema de clasificación binaria utilizando los datos de una entidad financiera local. Esta entidad financiera utiliza predominantemente métodos de aprendizaje paramétricos para construir varios tipos de modelos. Particularmente utiliza el método paramétrico de regresión logística en la implementación de un modelo de propensión a la compra y persistencia del producto Seguro de Protección de Tarjetas. Se busca implementar una alternativa a este modelo usando las Máquinas de Soporte Vectorial.

El método de aprendizaje SVM es una generalización de un método de clasificación binaria llamado Clasificador de Máximo Margen. Este método plantea la idea de un hiperplano (es decir, un subespacio $p-1$ dimensional) que parte en 2 un espacio p dimensional. Dado que hay infinitos hiperplanos que pueden separar el espacio, se plantea un método óptimo que maximiza la distancia del hiperplano al punto más cercano de cada clase (el margen M). Al maximizar el margen en los datos de entrenamiento también se produce un mejor desempeño en los datos de prueba, dado que habrá una mayor separación entre ambas clases (Hastie et al., 2009). El clasificador de Máximo Margen que se obtenga sólo tiene sentido cuando los

datos son perfectamente separables linealmente, lo cual no siempre es posible.

El método Máquinas de Soporte Vectorial generaliza el método de clasificación de Máximo Margen permitiendo su implementación en casos no separables (cuando las clases se sobreponen) creando fronteras no lineales en el espacio de origen mediante la construcción de fronteras lineales en una versión transformada (de mayor dimensión) del espacio de los datos (Hastie et al., 2009), para lo cual introduce 2 conceptos:

- El hiperplano no necesita separar perfectamente a las 2 clases, a fin de lograr mayor robustez y una mejor clasificación de la mayor parte de los datos. Así, se permite que algunas observaciones estén en el lado incorrecto del margen, o incluso en el lado incorrecto del hiperplano. Estas observaciones son las únicas que influyen y determinan el clasificador, y son llamados vectores de soporte (James et al., 2013).
- El uso de funciones Kernel $K(u, v)$ que aumentan la dimensión de los datos para trabajar de forma computacionalmente eficiente, de forma que sea posible construir una frontera no lineal que los separe (James et al., 2013). Algunas opciones comunes que se usan como funciones Kernel son: polinomios de grado d, bases radiales, y redes neuronales (Hastie et al., 2009).

También existe una extensión llamada Regresión de Soporte Vectorial (SVR, por sus siglas en inglés) que permite aplicar este método a un problema de regresión. Esta técnica es una generalización no lineal del modelo de regresión que utiliza una medida de error "*epsilon* insensible", que ignora los errores de tamaño menor a un *epsilon*. Haciendo una analogía con el método de clasificación, se ignora las observaciones en el lado correcto del hiperplano y las que estén lejos de éste (Hastie et al., 2009).

1.1. Objetivos

El objetivo general de la tesis es constituir un instrumento de aprendizaje de nuevas técnicas estadísticos para analizar los datos, mediante el estudio y aplicación de la teoría del aprendizaje estadístico en la técnica Máquinas de Soporte Vectorial (SVM), y desarrollando una aplicación en un conjunto de datos. De manera específica:

- Revisar la literatura y presentar un resumen introductorio de la teoría de aprendizaje estadístico.
- Estudiar la teoría, propiedades, consideraciones prácticas, y formas de implementación de la técnica de aprendizaje estadístico Máquinas de Soporte Vectorial (SVM).
- Aplicar e implementar la técnica de aprendizaje estadístico SVM usando un software estadístico para resolver un problema de clasificación binaria en un conjunto de datos de una entidad financiera. Específicamente, se busca desarrollar un modelo de propensión a la compra y persistencia de seguros.

- Comparar los resultados del modelo de propensión obtenido con la técnica SVM vs un modelo de regresión logística, en términos de desempeño de predicción y facilidad de implementación.

1.2. Organización del Trabajo

En el Capítulo 2 presentamos conceptos relacionados al aprendizaje estadístico.

En el Capítulo 3 presentamos el método de aprendizaje Máquinas de Soporte Vectorial (SVM), enfocándonos en sus aspectos teóricos.

En el Capítulo 4 estudiamos la aplicación del modelo SVM para resolver un problema de clasificación binaria, encontrando un modelo de propensión para la compra y persistencia de seguros.

Finalmente, en el Capítulo 5 discutimos las conclusiones obtenidas en este trabajo. Analizamos la ventajas y desventajas del método SVM, y proponemos ideas para investigaciones futuras.

En el anexo se presenta las variables analizadas y los programas utilizadas en las aplicaciones al conjunto de datos reales.

Capítulo 2

Aprendizaje Estadístico

En este capítulo presentamos los conceptos básicos del aprendizaje estadístico que sirven de sustento para desarrollar, en el próximo capítulo, el método de aprendizaje Máquinas de Soporte Vectorial.

2.1. Aprendizaje Estadístico

El aprendizaje estadístico juega un rol clave en muchas áreas de la ciencia, finanzas y la industria. Tiene un rol importante en campos de estadística, minería de datos e inteligencia artificial, intersectándose además con áreas de ingeniería y otras disciplinas (Hastie et al., 2009). Algunos ejemplos de sus aplicaciones son:

- Predecir si un paciente hospitalizado por un ataque al corazón tendrá un segundo ataque, en base a datos demográficos, dieta, y mediciones clínicas.
- Predecir el precio que tendrá una acción en 6 meses, basado en el desempeño de la compañía y en datos económicos.
- Identificar caracteres escritos a mano en una imagen digitalizada.
- Identificar el riesgo de cáncer de próstata en un paciente usando variables clínicas y demográficas.

Durante el presente documento se trabajará con variables aleatorias, refiriéndonos a ellas tanto de forma general como a sus valores observados. A fin de facilitar la lectura se definió como notación usar letras en mayúscula, como X, para referirse a aspectos generales de una variable o grupo de variables. Por otro lado se usa letras en minúscula, como x, para referirse a valores observados de una variable o grupo de variables. El detalle de la notación empleada se encuentra en el apéndice A.

Supongamos un escenario en el que tenemos una variable respuesta que deseamos predecir, que en adelante llamaremos Y. Para efectuar la predicción usaremos un conjunto de p características o variables predictoras, que en adelante llamaremos $X = (X_1, X_2, ..., X_p)$. Luego, se asume que existe una relación entre Y y X, que puede ser escrita de forma general como (James et al., 2013):

$$Y = f(X) + \varepsilon$$

donde f es una función fija pero desconocida de X, y ε es un error aleatorio independiente de X con media cero. En esta formulación, f representa la información sistemática que X provee acerca de Y, y ε el error irreductible.

En esencia, el aprendizaje estadístico consta de un conjunto de enfoques para estimar la función f (James et al., 2013) que permitirá predecir el valor de la variable respuesta para datos no observados (Hastie et al., 2009).

2.1.1. Aprendizaje Supervisado y No Supervisado

La mayoría de los problemas de aprendizaje cae en una de dos categorías: supervisado, y no supervisado.

En el aprendizaje supervisado, para cada observación de las variables predictoras x_i existe una variable respuesta asociada y_i. Buscamos encontrar un modelo que relacione la respuesta a los predictores, con el objetivo de predecir la respuesta para observaciones futuras (predicción) o para entender mejor la relación entre la respuesta y los predictores (inferencia). En esta categoría caen muchos de los métodos estadísticos clásicos, como la regresión lineal y la regresión logística, así como otros enfoques más modernos como Modelos Aditivos Generalizados (GAM, por sus siglas en inglés), redes neuronales, y máquinas de soporte vectorial (James et al., 2013). El enfoque de la presente tesis es el estudio del aprendizaje estadístico supervisado, centrado en el estudio de las máquinas de soporte vectorial.

Por el contrario, el aprendizaje no supervisado describe escenarios en los que, para un vector de variables x_i no existe una respuesta asociada y_i. Es decir, se puede decir que en cierto sentido se trabaja a ciegas, y de allí el nombre no supervisado. En estas situaciones, se buscar entender las relaciones entre las variables o entre las observaciones (James et al., 2013). Por ejemplo:

- El análisis cluster, que permite encontrar subgrupos o *clusters* en los datos. Es decir, se busca particionar los datos en grupos distintos de forma que las observaciones de un grupo sean similares entre sí, y observaciones de grupos diferentes sean distintas (James et al., 2013).
- El análisis de componentes principales, que permite resumir un conjunto de datos con variables correlacionadas, mediante un conjunto más pequeño de variables que, en conjunto, explican la mayor parte de la variabilidad en los datos originales (James et al., 2013).

2.1.2. Regresión y Clasificación

Las variables pueden clasificarse como *cuantitativas* o *cualitativas* (también llamadas categóricas). Las variables cuantitativas toman valores numéricos, por ejemplo: la edad, altura o ingresos de una persona, el precio de una acción, el nivel de glucosa en la sangre, etc. Por otro lado, las variables cualitativas toman como valores una de k clases o categorías posibles, por ejemplo: el sexo de una persona (masculino o femenino), si un cliente deja de pagar un

crédito (sí o no), un diagnóstico de cáncer (tiene o no tiene cáncer), etc (James et al., 2013). Las variables cualitativas tienen además un subgrupo llamado *categóricas ordinales*, en donde hay un orden natural para las categorías pero no existe una métrica apropiada para medir una distancia entre ellas (Hastie et al., 2009). Por ejemplo: el coeficiente intelectual de una persona, nivel socioeconómico (A1, A2, B1, B2, C, D, E), grado de instrucción (primaria, secundaria, superior), etc.

Existe una distinción para los problemas de aprendizaje estadístico según el tipo de la variable respuesta. Los problemas con variables respuesta cuantitativas o numéricas son conocidos como problemas de regresión, mientras que los problemas con variables respuesta cualitativas son llamados problemas de clasificación. Particularmente cuando la variable respuesta tiene sólo dos categorías el problema es llamado de clasificación binaria (James et al., 2013). A pesar de las distinciones, ambos tipos de problemas tienen mucho en común y pueden entenderse como la tarea de encontrar una función que estime el valor verdadero de la variable respuesta (Hastie et al., 2009).

2.1.3. Datos de Entrenamiento, Validación y Prueba

Podemos dividir los problemas de aprendizaje estadístico en dos etapas:

- Selección del modelo: se estima el desempeño de diferentes modelos para elegir el mejor.
- Valoración del modelo: habiendo elegido un modelo final, se estimar su error de predicción en datos nuevos (error de generalización).

El mejor enfoque para resolver ambas etapas del problema, al menos en un escenario en el que tenemos suficientes datos, es dividir aleatoriamente los datos en tres partes: datos de entrenamiento, datos de validación, y datos de prueba (Hastie et al., 2009):

- Los datos de entrenamiento se usan para entrenar el modelo y estimar $\hat{f}$.
- Los datos de validación se usan para estimar el error de predicción en los $\hat{f}$ estimados, y así elegir el mejor modelo.
- Los datos de prueba se usan para estimar el error de generalización del modelo final $\hat{f}$.

Idealmente, los datos de prueba debe estar "ocultos", y sólo deben utilizarse al final del análisis. De lo contrario, corremos el riesgo de subestimar el verdadero error de prueba.

2.2. Enfoque de los Modelos de Aprendizaje Estadístico

Al aplicar un método de aprendizaje estadístico, nuestra meta es estimar la función desconocida f que relaciona las variables predictoras con la variable respuesta. Es decir, queremos encontrar una función $\hat{f}$ tal que $Y \approx \hat{f}(X)$ para cualquier observación (X, Y).

De forma general, la mayoría de métodos de aprendizaje estadístico pueden ser clasificados como paramétricos o no paramétricos.

2.2.1. Modelos Paramétricos

Los métodos paramétricos asumen que la función f tiene una forma conocida. Por ejemplo, una asunción muy simple es que f es lineal en X:

$$f(X) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_p X_p$$

Una vez que se asume la forma de f el problema de estimar esta función se simplifica considerablemente, dado que es mucho más sencillo estimar un conjunto de parámetros (como los β_i en el modelo lineal) que ajustar una función f totalmente arbitraria (James et al., 2013).

La potencial desventaja de este enfoque es que el modelo que escojamos generalmente no se ajustará a la forma real y desconocida de f. Por lo que si el modelo elegido es muy distinto a la forma real de f, entonces las estimaciones que obtengamos serán malas (James et al., 2013).

Podemos intentar resolver este problema escogiendo modelos flexibles que ajusten a varias formas funcionales de f. Pero en general ajustar a un modelo más flexible requiere estimar más parámetros. Estos modelos más complejos pueden llevar a problemas de sobreajuste de los datos; es decir, que el modelo encuentre relaciones de predicción *aparentes* en los datos de entrenamiento, pero que en realidad son causadas por ruido o error aleatorio y no por las propiedades de la función f (James et al., 2013).

2.2.2. Modelos No Paramétricos

Los métodos no paramétricos no hacen asunciones explícitas sobre la forma de f. En cambio buscan una estimación de f que se aproxime tanto como sea posible a los datos, de forma suave (es decir, sin ser demasiado brusca ni demasiado sinuosa) (James et al., 2013).

Este enfoque tiene una gran ventaja sobre los modelos paramétricos: todo método paramétrico implica la posibilidad de que la forma funcional asumida para f sea muy distinta a su forma real. Por el contrario, los métodos no paramétricos evitan completamente este riesgo, dado que no hay ninguna asunción sobre la forma de f, y tienen el potencial de ajustar a un rango más amplio de formas para f (James et al., 2013).

Sin embargo, los métodos no paramétricos sufren de una importante desventaja: requieren una gran cantidad de datos para obtener una buena aproximación de la función f. Esto es debido a que el modelo los métodos no paramétricos debe estimar la forma de la función f usando los datos, lo cual es más complejo que estimar un número relativamente pequeño de parámetro para una forma conocida, como en los modelos paramétricos (James et al., 2013).

2.2.3. Balance entre Bondad de Predicción e Interpretabilidad del Modelo

Algunos métodos de aprendizaje estadístico son menos flexibles que otros, en el sentido que sólo funcionan en un rango pequeño de formas funcionales de f. Por ejemplo el modelo de regresión lineal es un enfoque relativamente inflexible, dado que sólo genera funciones

lineales. Por el contrario, otros métodos como los *splines* son más flexibles debido a que pueden adaptarse a un rango más amplio de formas para estimar a f (James et al., 2013).

Entonces ¿cuándo es conveniente usar un modelo restrictivo en lugar de uno flexible? Mucho dependerá del uso que se necesite para el modelo. Si se busca explicar el papel de las variables predictoras en la relación con la variable respuesta entonces es conveniente usar modelos más restrictivos, dado que son más fáciles de interpretar. Por ejemplo, en un modelo paramétrico como la regresión lineal es mucho más fácil entender la relación entre las variables predictoras X_i y la variable respuesta Y, que en un modelo no paramétrico como los *splines* o las redes neuronales (James et al., 2013).

En algunos casos, sin embargo, sólo interesa la predicción y no la interpretabilidad del modelo. Por ejemplo si se busca un algoritmo para predecir el precio de una acción sólo interesa que la predicción sea precisa, para lo cual un modelo flexible generalmente tendrá un mejor desempeño. Sin embargo no siempre es apropiado usar el modelo más flexible, dado que podríamos enfrentar un problema de sobreajuste (James et al., 2013), es decir, que el modelo identifique relaciones aparentes en los datos de entrenamiento, pero que no se cumplen en general.

2.3. Evaluación de la Precisión de un Modelo

En el aprendizaje estadístico no existe un método que domine al resto en todos los conjuntos de datos posibles. Luego, es importante decidir qué método produce mejores resultados en cada caso. Esa selección del mejor método es una de las tareas más desafiantes del aprendizaje estadístico.

Para evaluar el desempeño de un método de aprendizaje estadístico en un conjunto de datos necesitamos medir que tan cercanas son las predicciones calculadas respecto a los datos observados de la variable respuesta.

En un problema de regresión, la medición más usada es el Error Cuadrático Medio (MSE por sus siglas en inglés), dado por la siguiente fórmula:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}(x_i))^2$$

El MSE es pequeño si los valores predecidos están muy cercanos a los valores reales de la variable respuesta, y será grande si hay diferencias sustanciales entre éstos.

Por otro lado en un problema de clasificación la forma más común para medir la precisión de $\hat{f}$ es la tasa de error (ER): la proporción de errores que se comete al aplicar el $\hat{f}$ estimado en los datos de entrenamiento:

$$ER = \frac{1}{n}\sum_{i=1}^{n} I(y_i \neq \hat{f}(x_i))$$

donde $I(y_i \neq \hat{f}(x_i))$ es una función indicadora que es igual a 1 cuando $y_i \neq \hat{f}(x_i)$, y cero cuando $y_i = \hat{f}(x_i)$

Otra forma de ver el desempeño de un método de aprendizaje en un escenario de clasificación es mediante la Matriz de confusión. Esta consiste en una tabla de doble entrada (filas y columnas), donde las columnas representan las clases reales, mientras que las filas representan las clases predichas por el modelo (o vice-versa). El nombre se origina del hecho que es sencillo ver si un modelo están confundiendo dos o más clases. El cuadro 2.1 muestra un esquema de matriz de confusión para clasificación binaria.

Cuadro 2.1: Esquema de Matriz de confusión para clasificación binaria

		Real		
		0	**1**	**Total**
Predicción	**0**	Verdaderos negativos	Falsos negativos	Total predicciones negativas
	1	Falsos positivos	Verdaderos positivos	Total predicciones positivas
	Total	Total positivos reales	Total negativos reales	Total muestra

En los casos de clasificación binaria se tiene además las siguientes medidas de precisión:

- Sensitividad: es el porcentaje de verdaderos positivos que son identificados correctamente, respecto al total de positivos reales.
- Especificidad: es el porcentaje de verdaderos negativos que son identificados correctamente, respecto al total de negativos reales.
- Valor de Precisión Balanceada: mide el promedio entre la Sensitividad y la Especificidad.
- Valor de Predicción Positivo: es el porcentaje de verdaderos positivos respecto al total de predicciones positivas.
- Valor de Predicción Negativa: es el porcentaje de verdaderos negativos respecto al total de predicciones negativas.
- Curva ROC (Receiver Operating Characteristics): es un gráfico de sensitividad versus 1 - especificidad a medida que varían los parámetros de la regla de clasificación, tal como se observa en la figura 2.1.
- Área bajo la curva ROC (AUC): es una medida cuantitativa de resumen para la curva ROC.

Las medidas de bondad de ajuste son calculadas en los datos de entrenamiento. Sin embargo nos interesa medir la precisión de las predicciones cuando aplicamos el método

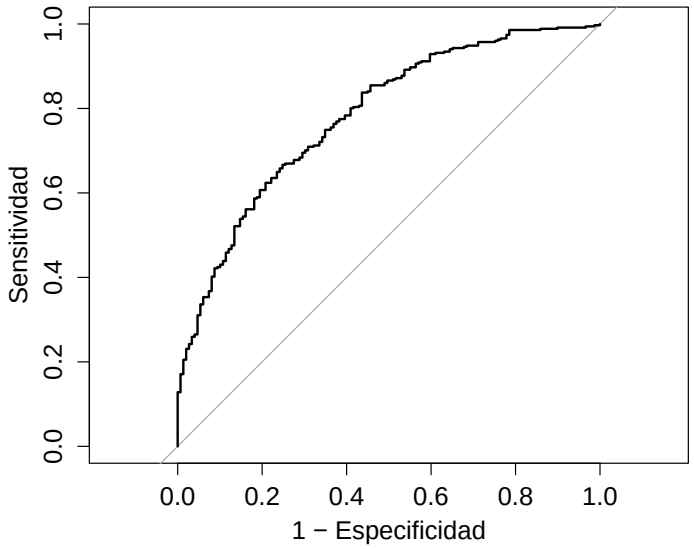


Figura 2.1: La curva ROC grafica la sensitividad versus la especificidad. El área bajo la curva se usa como una medida de resumen de la bondad de predicción.

en datos de prueba previamente desconocidos. Es decir, buscamos elegir un método que generalice bien: que resulte en la mejor medición de performance de ajuste en los datos de prueba, en lugar de sólo el mejor desempeño en los datos de entrenamiento (James et al., 2013).

2.3.1. Balance entre Sesgo y Varianza

Dada una observación de prueba x_0 es posible descomponer el MSE esperado, es decir, el MSE promedio para la observación de prueba x_0 que se obtendría si estimamos $\hat{f}$ muchas veces usando una gran cantidad de datos de entrenamiento, como la suma de 2 cantidades fundamentales: la *varianza* de $\hat{f}(x_0)$, y el *sesgo* al cuadrado de $\hat{f}(x_0)$. Es decir:

$$E(y_0 - \hat{f}(x_0))^2 = \text{Var}(\hat{f}(x_0)) + (\text{Sesgo}(\hat{f}(x)))^2$$

Esta ecuación sugiere que para minimizar el error esperado en los datos de prueba necesitamos seleccionar un método de aprendizaje que simultáneamente alcance una baja varianza y un bajo sesgo.

La varianza se refiere a cuánto varía $\hat{f}$ si la estimamos usando otro conjunto de datos de entrenamiento. Dado que usamos los datos para entrenar un método de aprendizaje estadístico, diferentes datos nos darán diferentes $\hat{f}$. Idealmente la estimación de $\hat{f}$ no debería variar significativamente si se usa datos de entrenamiento diferentes. Sin embargo, en un método con varianza alta cambios pequeños en los datos de entrenamiento pueden resultar en grandes variaciones de $\hat{f}$. En general, los métodos de aprendizaje más flexibles tienen mayor varianza (James et al., 2013).

Por otro lado, el sesgo se refiere al error producido por aproximar un problema real, que puede ser extremadamente complicado, con un modelo más simple. Por ejemplo, la regresión lineal asume una relación lineal entre Y y X; es poco probable que en un problema real exista una perfecta relación lineal entre los datos, por lo tanto este método indudablemente resultará en un sesgo en la estimación de $\hat{f}$. En general, los métodos de aprendizaje más flexibles tienen menor sesgo (James et al., 2013).

Como regla general, al usar modelos más flexibles la varianza aumenta y el sesgo disminuye. Por ejemplo, en la figura 2.2, el gráfico a la izquierda muestra un modelo muy simple que tiene varianza nula pero un sesgo muy alto, mientras que el gráfico al centro muestra un modelo muy complejo que minimiza el sesgo pero tiene una varianza muy alta. La tasa de cambio relativo de ambas cantidades determina si la medida de bondad de ajuste aumenta o disminuye en los datos de entrenamiento. A medida que usamos métodos más flexibles, el sesgo tiende a decrecer más rápido que la varianza. Sin embargo, en un punto el incremento en flexibilidad tiene poco impacto en el sesgo, pero empieza a impactar significativamente la varianza. Cuando esto sucede, la bondad de ajuste disminuye (James et al., 2013).

Tener un buen desempeño de predicción en los datos de prueba depende entonces de que el método de aprendizaje tenga poca varianza y poco sesgo. A esto se le conoce como balance, porque es muy fácil obtener un método con bajo sesgo pero alta varianza (por ejemplo, dibujando una línea que pase por todos los puntos de los datos de entrenamiento, como en el gráfico al centro de la figura 2.2) o un método con alto sesgo y baja varianza (por ejemplo, ajustando mediante una línea horizontal, como en el gráfico a la izquierda de la figura 2.2). El desafío yace en encontrar un método en el que tanto la varianza como el sesgo sean pequeños (James et al., 2013), como sugiere el gráfico a la derecha de la figura 2.2.

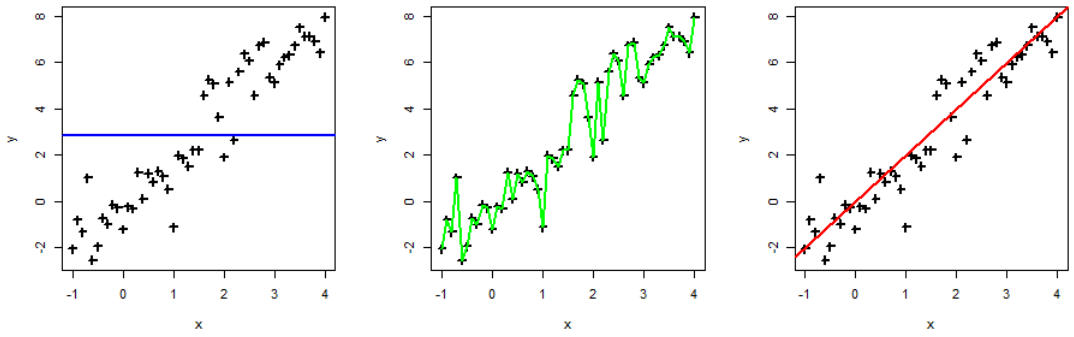


Figura 2.2: Sesgo y Varianza para tres modelos de aprendizaje estadístico en el mismo conjunto de datos. Primero, la figura de la izquierda muestra un modelo muy simple: una línea horizontal en la media de la variable respuesta. Este modelo tiene sesgo muy alto pero una varianza nula. Luego, la figura del centro muestra un modelo muy complejo que pasa por todos los datos de entrenamiento. Este modelo tiene sesgo nulo pero una varianza muy alta. Finalmente la figura de la derecha muestra un modelo de complejidad media que sigue la tendencia de los datos de entrenamiento. Este modelo alcanza un equilibrio entre sesgo y varianza.

2.4. Funciones de Pérdida y Riesgo

Un requisito inmediato en cualquier problema de aprendizaje es especificar exactamente lo que queremos lograr, minimizar, acotar, o aproximar. Es decir, necesitamos encontrar un critero que mida la calidad de la estimación de Y mediante el $\hat{f}(X)$ que obtenemos de los datos (Scholkopf y Smola, 2002). Algunos de los conceptos que se introducen para este objetivo son la función de pérdida y la función de riesgo.

Esta pregunta no es trivial. Incluso en los problemas de clasificación binaria existen múltiples opciones: la fracción de datos correctamente clasificados, o tomar en cuenta el hecho que las pérdidas no son simétricas entre las dos clases. Los escenarios de clasificación multinomial añaden aún más complejidad a este problema puesto que cada forma de mala clasificación podría generar un tipo de pérdida diferente (Scholkopf y Smola, 2002).

2.4.1. Función de Pérdida

Sea (x, y) un par de vectores de variables predictoras y respuesta que pertenecen a un espacio χ y Υ, respectivamente. Luego se tiene la triada $(x, y, f(x)) \in \chi \times \Upsilon \times \Upsilon$, donde $f(x)$ es la predicción hecha por el método de aprendizaje. Entonces introducimos la función de pérdida no negativa $c : \chi \times \Upsilon \times \Upsilon \to [0, \infty[$, con la propiedad que $c(x, y, y) = 0$; es decir, c es igual a 0 cuando la predicción $f(x)$ es correcta.

A continuación se presenta las principales funciones de pérdida para problemas de clasificación binaria (Scholkopf y Smola, 2002). En estos escenarios, el rango Υ de la variable respuesta y toma sólo dos valores posibles: $\Upsilon = -1, +1$.

- Pérdida 0-1 (Tasa de Error, o Error de Clasificación): Es la función de pérdida más simple. Cuando se comete un error de clasificación incurrimos en una pérdida de 1; caso contrario la pérdida es 0:

$$c(x, y, f(x)) = \begin{Bmatrix} 0 & , & \text{si } y = f(x) \\ 1 & , & \text{caso contrario} \end{Bmatrix} \tag{2.1}$$

  Esta definición de c no distingue entre diferentes clases y tipos de errores (falsos positivos o negativos).

- Pérdida de Margen Suave o Pérdida Bisagra: Esta función de pérdida mide la calidad de la estimación mediante el producto $yf(x)$:

$$c(x, y, f(x)) = \text{máx}\{0, 1 - yf(x)\} = \begin{Bmatrix} 0 & , & \text{si } yf(x) \geq 1 \\ 1 - yf(x) & , & \text{caso contrario} \end{Bmatrix} \tag{2.2}$$

  En esta función de pérdida la salida de la función de predicción $f(x)$ puede ser cualquier número real, pero la etiqueta de predicción estará dada por signo$(f(x))$. Además el valor absoluto de $|f(x)|$ mide la confianza de la predicción (a mayor valor mayor confianza en la predicción, aunque cuando esta pueda estar equivocada). Podemos profundizar

esta interpretación:

- Si y y $f(x)$ tienen el mismo signo (es decir, si $f(x)$ predice la clase correcta) y $|f(x)| >= 1$ entonces la función de pérdida será 0. Esto se interpreta como $f(x)$ predice la clase correcta y y estamos muy seguros de esta predicción, por lo tanto no hay penalización.
- Si y y $f(x)$ tienen el mismo signo (es decir, si $f(x)$ predice la clase correcta) pero $|f(x)| < 1$ entonces la función de pérdida disminurá lineamente con el valor de $|f(x)|$. Esto se interpreta como $f(x)$ predice la clase correcta y pero no tenemos mucha certeza de dicha predicción, por lo tanto se incurre en una penalización.
- Si y y $f(x)$ tienen signos opuestos, la función de pérdida incrementa linealmente con el valor de $|f(x)|$. En este caso se interpreta como una penalización en la función de pérdida porque $f(x)$ se equivoca en la predicción de y, y la penalización será mayor en cuanto mayor sea el valor absoluto $|f(x)|$ (estamos más seguros de la predicción de $f(x)$, pero esta predicción es equivocada).

- Pérdida de Margen Suave Cuadrada: Es una versión al cuadrado de la Pérdida de Margen Suave o Bisagra que se utiliza en algunos casos debido a que es más sencilla de minimizar:

$$c(x, y, f(x)) = \text{máx}\{0, 1 - yf(x)\}^2 \tag{2.3}$$

En el caso de clasificación multinomial el problema es bastante más complejo que en el caso binario. Cada forma de mala clasificación puede potencialmente incurrir en una pérdida distinta, llevando a una matriz $K \times K$ de valores de pérdida (donde K es el número de clases de la variable respuesta).

En el caso de un problema de regresión generalmente nos interesa medir el tamaño de la diferencia $\xi = f(x) - y$. Es decir, se busca cuantificar qué tan alejada está la predicción respecto al valor real, en lugar del producto $yf(x)$. Con lo cual las funciones de pérdida generalmente tienen la forma:

$$c(x, y, f(x)) = \widetilde{c}(f(x) - y) = \widetilde{c}(\xi) \tag{2.4}$$

donde $\widetilde{c}$ indica cualquier función no negativa: $\widetilde{c} : \mathbb{R} \to [0, \infty[$, con la propiedad que $\widetilde{c}(0) = 0$.

Para los escenarios de regresión se tiene las siguientes funciones de pérdida (Scholkopf y Smola, 2002):

- Pérdida Cuadrática: La opción más popular es minimizar la suma de los cuadrados de los residuales o errores $f(x) - y$. Es decir se busca minimizar:

$$c(x, y, f(x)) = \widetilde{c}(f(x) - y) = (f(x) - y)^2 \tag{2.5}$$

- Pérdida *epsilon* insensible: Es una extensión de la pérdida de margen suave para pro-

blemas de regresión. Se obtiene de la siguiente forma:

$$\widetilde{c}(\xi) = |\xi|_\epsilon = \text{máx}(|\xi| - \epsilon, 0) \tag{2.6}$$

La idea de esta función es que las desviaciones menores a un valor ϵ no deben ser penalizadas, y cualquier desviación mayor sólo debe incurrir en una penalidad lineal. Colocando un valor $\epsilon = 0$ construimos la pérdida l_1, que busca la minimización de la suma de los valores absolutos de las desviaciones.

Es importante que las funciones de pérdida satisfagan algunas características para lograr implementaciones eficientes de los métodos de aprendizaje, como (Scholkopf y Smola, 2002):

- Ser fáciles de calcular.
- Que la primera derivada tenga pocas o ninguna discontinuidad.
- Ser convexas para asegurar la unicidad de la solución.

2.4.2. Riesgo Esperado

La función de pérdida determina cómo penalizar los errores en instancias específicas de $(x, y, f(x))$. Ahora debemos encontrar un método para combinar estas penalizaciones individuales, a fin de evaluar la función f estimada.

En adelante asumiremos que existe una función de probabilidad $P(x, y)$ en $\chi \times \Upsilon$ que gobierna la generación de los datos y su dependencia funcional subyacente f. Además, identificaremos como $P(y|x)$ a la distribución condicional de y dado x, y como $dP(x, y)$ y $dP(y|x)$ las densidades de las distribuciones $P(x, y)$ y $P(y|x)$, respectivamente. Asumimos además que los datos (x, y) son independientes e idénticamente distribuidos en $P(x, y)$ (Scholkopf y Smola, 2002).

Supongamos que conocemos los datos de entrenamiento $\{(x_1, y_1), ..., (x_n, y_n)\}$, así como los datos de prueba $\{(x'_1, y'_1), ..., (x'_m, y'_m)\}$. Entonces nos interesa minimizar la función de riesgo para este conjunto de datos de prueba en particular (Scholkopf y Smola, 2002):

$$R_{\text{test}}[f] = \frac{1}{m} \sum_{i=1}^{m} \int_{\Upsilon} c(x'_i, y, f(x'_i)) dP(y|x'_i) \tag{2.7}$$

Este problema es bastante difícil de resolver tanto computacional como conceptualmente. Entonces nos enfocaremos en estudiar la minimización de la función de riesgo para los casos en que no conozcamos los datos de prueba (o bien los conocemos y decidimos ignorar esta información). En este escenario debemos minimizar el error esperado sobre todos los posibles datos de prueba. Es decir debemos encontrar una función f que minimice el error total esperado ($E[R_{\text{test}}[f]]$) respecto a la distribución P y a la función de pérdida c (Scholkopf y

Smola, 2002):

$$R[f] = E[R_{\text{test}}[f]] = E[c(x, y, f(x))] = \int_{\chi \times \Upsilon} c(x, y, f(x)) dP(x, y) \tag{2.8}$$

Este problema tampoco tiene solución dado que no conocemos $P(x, y)$ explícitamente. Sin embargo sí conocemos los datos de entrenamiento (x_i, y_i), los cuales nos permiten reemplazar la distribución $P(x, y)$ desconocida por su estimación empírica.

Luego, para estudiar la relación entre las funciones de pérdida y de densidad es conveniente asumir que existe una función de densidad $p(x, y)$ para $P(x, y)$, tal que: $\int dP(x, y) = \int p(x, y) dx dy$.

Sin embargo esto aún no resuelve el problema. Nuevamente, sólo tenemos a nuestra disposición los datos de entrenamiento. Entonces, podemos reemplazar $p(x, y)$ por su *densidad empírica* ρ (Scholkopf y Smola, 2002):

$$\rho_{\text{emp}} = \frac{1}{n} \sum_{i=1}^{n} \delta_{x_i}(x) \delta_{y_i}(y) \tag{2.9}$$

donde $\delta_{x_i}(x)$ denota una distribución delta de Dirac δ, que satisface: $\int \delta_{x'}(x) f(x) dx = f(x')$. La intención es que, reemplazando $p(x, y)$ por ρ_{emp}, obtendremos una cantidad razonablemente cercana al riesgo esperado, lo cual se cumplirá si el rango de soluciones posibles f es suficientemente limitado (Scholkopf y Smola, 2002).

2.4.3. Riesgo Empírico

El Riesgo Empírico resulta de reemplazar la densidad empírica (2.9) en la ecuación del Riesgo Esperado (2.8):

$$R_{\text{emp}}[f] = \int_{\chi \times \Upsilon} c(x, y, f(x)) \rho_{\text{emp}}(x, y) dx dy = \frac{1}{n} \sum_{i=1}^{n} c(x_i, y_i, f(x_i)) \tag{2.10}$$

Esta cantidad tiene la ventaja de que, dados los datos de entrenamiento, podemos calcularla y minimizarla fácilmente.

Entonces, la ecuación 2.10 parecería ser la solución al problema, sólo quedando pendiente encontrar una clase de funciones $\mathcal{F} \ni f$ que minimice $R_{\text{emp}}[f]$ respecto a $\mathcal{F}$.

Lamentablemente es bastante difícil encontrar a $\mathcal{F}$ y la minimización de $R_{\text{emp}}[f]$ puede llevar a un problema mal planteado. Por ejemplo, si la familia de funciones $\mathcal{F}$ es muy extensa la diferencia entre el riesgo empírico y el riesgo esperado puede ser grande (Scholkopf y Smola, 2002).

Incluso si $\mathcal{F}$ contiene una función f que predice correctamente todos los valores en los datos de entrenamiento, no podemos asegurar que esto se repetirá en datos no observados (datos de prueba). Sin embargo podemos superar este problema agregando un término de

regularización al Riesgo Empírico.

2.4.4. Función de Riesgo Regularizada

Minimizar el riesgo empírico de un método de aprendizaje f puede llevar a un modelo con pobre desempeño de generalización (es decir, bajo poder de predicción en datos nuevos, diferentes a los de entrenamiento). Para evitar este problema se introduce un término de regularización para esta función.

La idea clave en la regularización es restringir la clase de funciones posibles F (donde $f \in F$) que minimicen la función de riesgo empírico $R_{\text{emp}}[f]$, tal que F sea un conjunto compacto.

Se asumirá que la función $R_{\text{emp}}[f]$ es continua en f. Esto debido a que, sin esta restricción, el problema de minimización de la función de riesgo es muy complejo de resolver, incluso cuando se trata de buscar soluciones aproximadas (Scholkopf y Smola, 2002).

Es facil cumplir con este requisito en problemas de regresión, por ejemplo usando la función de pérdida cuadrática, o la función de pérdida ϵ insensible (2.4.1). Sin embargo no todas las funciones de pérdida para problemas de clasificación binaria cumplen con este requisito de continuidad (por ejemplo, la pérdida 0-1 no lo cumple). En esta situación, se prefiere aproximar el problema usando funciones de pérdida continuas, como la pérdida de margen suave (Scholkopf y Smola, 2002).

Para plantear el problema de regularización no es necesario especificar directamente un conjunto compacto F, dado que esto puede resultar en problemas complejos de resolver. En cambio se agrega un término de estabilización o regularización $\Omega[f]$ a la función de riesgo original. Entonces la función de riesgo regularizada tiene la siguiente forma (Scholkopf y Smola, 2002):

$$R_{\text{reg}}[f] = R_{\text{emp}}[f] + \lambda\Omega[f] = \frac{1}{n}\sum_{i=1}^{n} c(x_i, y_i, f(x_i)) + \lambda\Omega[f] \tag{2.11}$$

donde $\lambda > 0$ es un parámetro de regularización que especifica el balance entre la minimización del riesgo empírico ($R_{\text{emp}}[f]$) y la simplicidad de la función f, la cual es forzada por un término de regularización $\lambda\Omega[f]$ pequeño. Usualmente se busca que $\Omega[f]$ sea convexa dado que esto asegura que exista un único mínimo global, siempre y cuando el término de Riesgo Empírico sea también convexo (Scholkopf y Smola, 2002).

Capítulo 3

Máquinas de Soporte Vectorial

En este capítulo se presenta el método de clasificación binaria Máquinas de Soporte Vectorial. Este método nace a partir del método de clasificación del hiperplano óptimo, el cual funciona cuando se tienen dos clases perfectamente separables en forma lineal. El método Máquinas de Soporte Vectorial extiende el caso anterior a casos no separables mediante el uso de "funciones kernel", sumergiendo el problema en un espacio de dimensión mayor que la del problema original.

3.1. Antecedentes Teóricos

Es necesario introducir algunas ideas como las siguientes:

3.1.1. Espacio Vectorial

Es una estructura matemática formada por un conjunto de "vectores" en donde están definidas dos operaciones: adición de vectores, y multiplicación de vectores por un escalar. Estas operaciones gozan de propiedades que se pueden revisar en el libro Grossman y Flores (2012).

3.1.2. Espacio Vectorial Normado

Es un espacio vectorial donde está definida una norma. Este concepto extiende la idea de longitud o magnitud de un vector para un espacio de dimensión n.

3.1.3. Espacio Producto Interno

Es un espacio vectorial con una estructura adicional llamada producto interno o producto escalar de vectores.

El producto interno es una operación algebraica que toma 2 vectores del mismo espacio y retorna un número escalar. Esta operación se puede definir de forma algebraica o geométrica. Así, por ejemplo dados dos vectores x y x', se puede definir el producto interno euclídeo de un espacio de dimensión n:

- Algebraicamente, como la suma del producto de los elementos de los vectores. Es decir:

$$x \cdot x' = \sum_{i=1}^{N} x_i x'_i$$

  donde: x_i es el i-ésimo elemento del vector x, y x'_i es el i-ésimo elemento del vector x'

- Geométricamente, como el producto de la norma o longitud de los vectores por el coseno del ángulo θ entre éstos:

$$x \cdot x' = \|x\| \left\|x'\right\| \cos(\theta)$$

El producto interno euclídeo también puede usarse para definir la norma o magnitud de un vector x:

$$\|x\| = \sqrt{x \cdot x}$$

Adicionalmente, el producto interno euclídeo puede usarse como medida de cercanía o similaridad entre dos vectores. Esto se nota al revisar la definición geométrica del producto interno euclídeo, dado que interviene el coseno del ángulo entre los vectores x y x'. Así, si estos vectores son paralelos (en el mismo sentido o en el sentido opuesto) la magnitud del producto interno euclídeo será máxima (máxima similaridad). Por el contrario, si los vectores son ortogonales, el producto interno euclídeo será 0, indicando similaridad nula.

3.1.4. Espacio Métrico

Es un espacio vectorial en el cual se ha definido la distancia entre todos sus elementos:

$$d(x, x') = \left\|x - x'\right\|$$

3.1.5. Sucesión de Cauchy

Es una sucesión $\{x_n\}_{n \in N}$ definida en un espacio métrico tal que, para cualquier distancia ϵ siempre se puede encontrar un término $n_0 \in \mathbb{N}$ tal que la distancia entre dos términos posteriores cualquiera x_m, x_n es menor que ϵ; es decir:

$$|x_m - x_n| < \epsilon \ , \ \forall n, m \geq n_0$$

Esto implica que los términos de la sucesión se van acercando unos a otros.

3.1.6. Espacio Métrico Completo

Es un espacio métrico M en el que cualquier sucesión de puntos de Cauchy tiene un límite también en este espacio M. Más informalmente podemos entenderlo como un espacio métrico en el que no hay "puntos perdidos". Por ejemplo: los números racionales no son un conjunto métrico completo, porque le faltan los números irracionales; en cambio, los números reales sí son un conjunto métrico completo.

3.1.7. Espacio de Hilbert

Es un espacio métrico de producto interno que es completo. Generaliza la idea de espacio euclidiano a espacios de dimensión n, extendiendo los métodos de álgebra vectorial y cálculo.

El desarrollo de esta tesis se realiza en un espacio $\mathbb{R}^n$, el cual es un espacio de Hilbert y por lo tanto tiene habilitado el producto interno euclídeo anteriormente descrito.

3.1.8. El Método de Multiplicadores de Lagrange

Es un método para encontrar los máximos y mínimos locales de una función sujeta a restricciones de igualdad. Por ejemplo se puede planter el siguiente problema:

$$\begin{aligned} \text{maximizar:} \quad & f(x,y) \\ \text{sujeto a:} \quad & g(x,y) = c \end{aligned}$$

para lo cual es necesario que las funciones f y g tengan primera derivada parcial continua.

Este método introduce una variable nueva λ llamada multiplicador de Lagrange, y busca los valores óptimos de (x, y) maximizando la funcion de Lagrange (o Lagrangiano) definida como:

$$\Lambda(x, y, \lambda) = f(x, y) + \lambda(g(x, y) - c)$$

3.2. Clasificación mediante Hiperplano Óptimo

Si un conjunto de datos que pertenece a dos clases $+1$ y -1 es "linealmente separable", entonces es posible trazar una cantidad infinita de hiperplanos que separe a los elementos según su clase, como sugiere la figura 3.1. Sin embargo, sólo existirá un hiperplano óptimo que "separe mejor" a estos elementos (Hastie et al., 2009).

El método del Hiperplano Óptimo, o Hiperplano de Máximo Margen trata de encontrar este hiperplano óptimo que "separa mejor" los elementos, donde "separar mejor" se refiere a que este hiperplano tenga el margen M mayor; es decir, la mayor distancia M del hiperplano al vector más cercano en ambas clases, como indica la figura 3.2. Estos vectores más cercanos en ambas clases son los llamados Vectores de Soporte (SV por sus siglas en inglés: *Support Vectors*) y, como se verá a detalle en la subsección 3.2.1, son los únicos vectores de datos que intervienen en la definición del hiperplano óptimo.

A continuación se define formalmente el método del Hiperplano Óptimo. Se tiene los siguientes datos:

$$(x_1, y_1), ..., (x_n, y_n), \quad x_i \in R^n, \quad y_i \in \{+1, -1\}$$

que pueden ser separados por un hiperplano:

$$w' \cdot x + b' = 0$$

Por definición, el hiperplano debe separar los datos según pertenezcan a la clase $y = +1$ o $y = -1$. Formalmente podemos expresar esta condición de la siguiente forma:

$$\begin{aligned} w' \cdot x_i + b' &\geq M \quad , \quad \text{si} \quad y_i = +1 \\ w' \cdot x_i + b' &\leq -M \quad , \quad \text{si} \quad y_i = -1 \end{aligned} \tag{3.1}$$

donde M es el margen o distancia del hiperplano al vector más cercano.

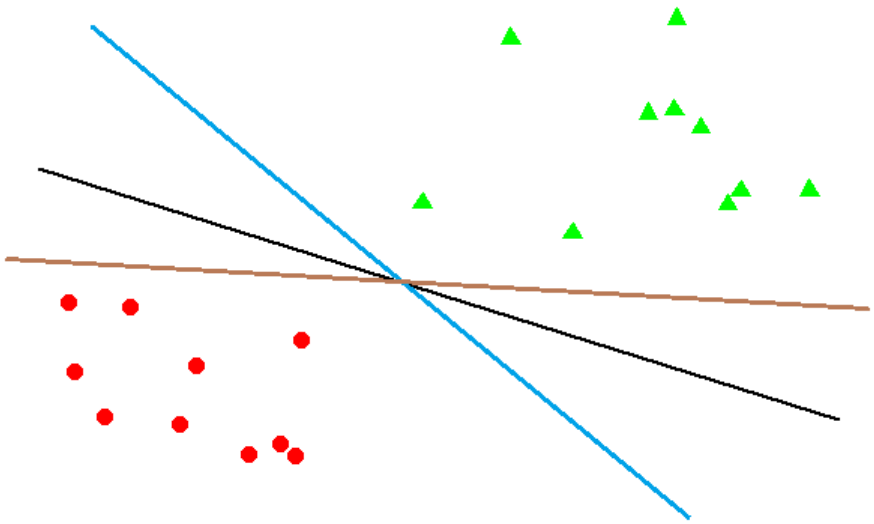


Figura 3.1: Dado un conjunto de datos separable linealmente, es posible trazar infinitos hiperplanos que separen los datos.

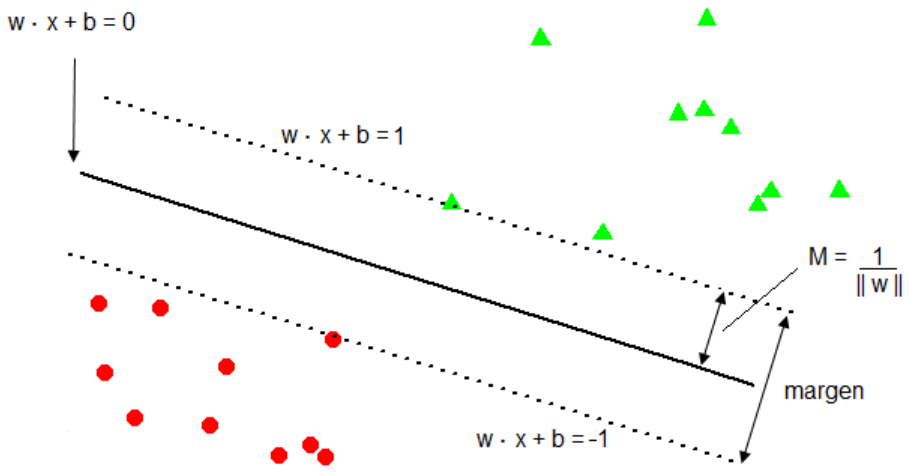


Figura 3.2: El hiperplano óptimo es aquel que "separa mejor" los datos de las dos clases. Esto se logra encontrando el hiperplano que maximiza el margen M; es decir, que maximiza la distancia M del hiperplano al vector más cercano en ambas clases.

Además se introduce una restricción sobre el módulo del vector w':

$$\left\|w'\right\| = 1 \tag{3.2}$$

mediante la cual, la distancia D_i del punto x_i al hiperplano es (James et al., 2013):

$$D_i = y_i \left(w' \cdot x_i + b'\right)$$

De manera equivalente, la condición 3.1 indicada anteriormente puede expresarse de la siguiente manera:

$$y_i \left(w' \cdot x_i + b'\right) \geq M, \quad i = 1, ..., n \tag{3.3}$$

Luego el Hiperplano Óptimo es aquel que maximiza este margen M.

Este problema de optimización puede ser replanteado de forma más conveniente si nos deshacemos de la restricción 3.2 sobre el módulo del vector w'. Para ello, podemos convertir la condición 3.3 en:

$$\frac{1}{M} y_i \left(w' \cdot x_i + b'\right) \geq 1, \quad i = 1, ..., n \tag{3.4}$$

o equivalentemente:

$$y_i \left(\frac{w'}{M} \cdot x_i + \frac{b'}{M}\right) \geq 1, \quad i = 1, ..., n \tag{3.5}$$

Podemos definir las variables $w = \frac{w'}{M}$ y $b = \frac{b'}{M}$ y replantear la restricción en base a estas nuevas variables:

$$y_i \left(w \cdot x_i + b\right) \geq 1, \quad i = 1, ..., n \tag{3.6}$$

En esta reparametrización del problema, el margen es igual a la inversa del módulo del vector w que define el hiperplano. Es decir:

$$\|w\| = \frac{1}{M} \tag{3.7}$$

Con lo cual el problema de optimización se convertiría en la minimización de $\|w\|$, el módulo del vector w que define el hiperplano. Sin embargo, a fin de simplificar cálculos posteriores podemos reemplazar esta expresión por otra equivalente. Así, dado que $\|w\|$ por definición es una cantidad positiva, entonces este problema es equivalente a minimizar $\|w\|^2$, debido a que la función cuadrática es monótona creciente para valores positivos. Adicionalmente, el problema de optimización tampoco varía al multiplicar esta expresión por una constante positiva.

Finalmente, el problema de optimización se convierte en la minimización del siguiente funcional (donde funcional se refiere a una función que toma un vector como parámetro de

entrada, y retorna un escalar):

$$\Phi(w) = \frac{1}{2}\|w\|^2 = \frac{1}{2}(w \cdot w) = \frac{1}{2}\frac{1}{M^2} \tag{3.8}$$

Respecto a la siguiente restricción de desigualdad:

$$y_i\,(w \cdot x_i + b) \geq 1, \quad i = 1, ..., n \tag{3.9}$$

De la ecuación 3.9, se puede deducir que hay puntos que satisfacen la restricción de igualdad; es decir, que cumplen:

$$y_i\,(w \cdot x_i + b) = 1, \quad i = 1, ..., n \tag{3.10}$$

Se puede comprobar que los puntos que cumplen la igualdad 3.10 se ubican en hiperplanos paralelos al hiperplano óptimo, a una distancia M de éste. Es decir, la ecuación 3.10 corresponde a los llamados hiperplanos de soporte, donde se encuentran los vectores de soporte (Hamel, 2009).

3.2.1. Cálculo del Hiperplano Óptimo

Dados los datos de entrenamiento $(x_1, y_1), ...(x_n, y_n)$, donde x_i son las variables predictoras y $y_i \in \{+1, -1\}$ es la variable respuesta, para encontrar el Hiperplano Óptimo debemos minimizar el módulo del hiperplano:

$$\Phi(w) = \frac{1}{2}\|w\|^2 = \frac{1}{2}(w \cdot w) \tag{3.11}$$

bajo la restricción de desigualdad:

$$y_i\,(w \cdot x_i + b) \geq 1, \quad i = 1, 2, ..., n \tag{3.12}$$

La solución a esta problema de optimización se encuentra en el punto silla del funcional de Lagrange o Lagrangiano; es decir, se encuentra minimizando el Lagrangiano respecto a w y b, y maximizándolo respecto a los multiplicadores de Lagrange α_i:

$$L(w, b, \alpha) = \frac{1}{2}(w \cdot w) - \sum_{i=1}^{n} \alpha_i\,((w \cdot x_i + b)\,y_i - 1) \tag{3.13}$$

donde los multiplicadores de Lagrange $\alpha_i \geq 0$.

Sean $w = w_0$, $b = b_0$, y $\alpha = \alpha_0$ la solución al problema de optimización (es decir, minimizan el Lagrangiano 3.13), entonces deben satisfacer las siguientes condiciones en el

punto silla (Vapnik, 1998):

$$\frac{\partial L(w,b,\alpha)}{\partial b} = \sum_{i=1}^{n} \alpha_i y_i = 0$$

$$\frac{\partial L(w,b,\alpha)}{\partial w} = w - \sum_{i=1}^{n} \alpha_i y_i x_i = 0$$

Al reescribir estas ecuaciones se obtiene las siguientes propiedades del Hiperplano Óptimo:

1. Los coeficientes α_i^0 del hiperplano óptimo deben satisfacer las siguientes restricciones:

$$\sum_{i=1}^{n} \alpha_i^0 y_i = 0, \quad \alpha_i^0 \geq 0, \quad i = 1, ..., n \tag{3.14}$$

2. El vector w_0 que define el hiperplano óptimo es una combinación lineal de los vectores de entrenamiento:

$$w_0 = \sum_{i=1}^{n} \alpha_i^0 y_i x_i = 0, \quad \alpha_i^0 \geq 0, \quad i = 1, ..., n \tag{3.15}$$

3. Únicamente los vectores de soporte (o SV por sus siglas en inglés) tienen coeficientes α_i^0 diferentes a 0 en la expresión de w_0 (3.15). Estos vectores de soporte son los que logran la condición de igualdad en la ecuación 3.12. Es decir, cumplen la condición:

$$y_i \left(w \cdot x_i + b\right) = 1, \quad i = 1, 2, ..., n \tag{3.16}$$

   Con ello se obtiene la expresión para w_0 (Vapnik, 1998):

$$w_0 = \sum_{SV} \alpha_i^0 y_i x_i, \quad \alpha_i^0 \geq 0 \tag{3.17}$$

   donde SV indica los vectores de soporte.

   Esto se deriva del teorema de Kühn-Tucker, el cual indica que la condición suficiente y necesaria para el hiperplano óptimo es que éste cumpla con la siguiente restricción (Vapnik, 1998):

$$\alpha_i^0 \left(\left(w_0 \cdot x_i + b_0\right) y_i - 1\right) = 0, \quad i = 1, ..., n, \tag{3.18}$$

Al colocar la expresión para w_0 en (3.17), y teniendo en cuenta las restricciones del teorema de Kühn-Tucker y la restricción (3.14) en el Lagrangiano (3.13), se obtiene el siguiente funcional:

$$W(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) \tag{3.19}$$

El cual debe ser maximizado bajo las siguientes restricciones:

$$\alpha_i \geq 0 \tag{3.20}$$

$$\sum_{i=1}^{n} \alpha_i y_i = 0 \tag{3.21}$$

Si $\alpha_0 = (\alpha_1^0, ..., \alpha_n^0)$ es una solución al problema de optimización entonces la norma del vector w_0 que corresponde al hiperplano óptimo es igual a:

$$\|w_0\|^2 = 2W(\alpha_0) = \sum_{SV} \alpha_i^0 \alpha_j^0 (x_i \cdot x_j) y_i y_j \tag{3.22}$$

Finalmente, según la definición de w_0 en 3.17 la regla de separación según el hiperplano óptimo es la función:

$$f(x) = \text{signo}\left(\sum_{SV} \alpha_i^0 y_i (x_i \cdot x) + b_0\right) \tag{3.23}$$

donde: x_i son los vectores de soporte, α_i son los multiplicadores de Lagrange y b_0 es la constante de desplazamiento.

Para hallar el valor de la constante de desplazamiento se usa la restricción 3.16, la cual debe cumplirse para cualquier vector de soporte. Así, sea x_{SV}^+ un vector de soporte de la clase $y = +1$ elegido arbitrariamente, entonces se cumple:

$$w_0 \cdot x_{SV}^+ + b_0 = 1$$

de donde se deduce el valor de b_0:

$$b_0 = 1 - w \cdot x_{SV}^+ = 1 - \sum_{SV} \alpha_i y_i (x_i \cdot x_{SV}^+) \tag{3.24}$$

3.3. Hiperplano Óptimo para el Caso No Separable

El Hiperplano Óptimo sólo funciona cuando los datos son perfectamente separables linealmente. Naturalmente esta característica no suele suceder en la realidad, por lo tanto introducimos el concepto del Hiperplano Óptimo Generalizado para los casos no separables linealmente.

Así, a fin de lograr mayor robustez y una mejor clasificación de la mayor parte de los datos (James et al., 2013) se permite que la clasificación mediante el hiperplano cometa errores de clasificación. Para ello, como indica la figura 3.3, se introduce variables de holgura $\xi_1, ..., \xi_n$ que se usan en las restricciones de separación:

$$y_i \left(w \cdot x_i + b\right) \geq 1 - \xi_i \tag{3.25}$$

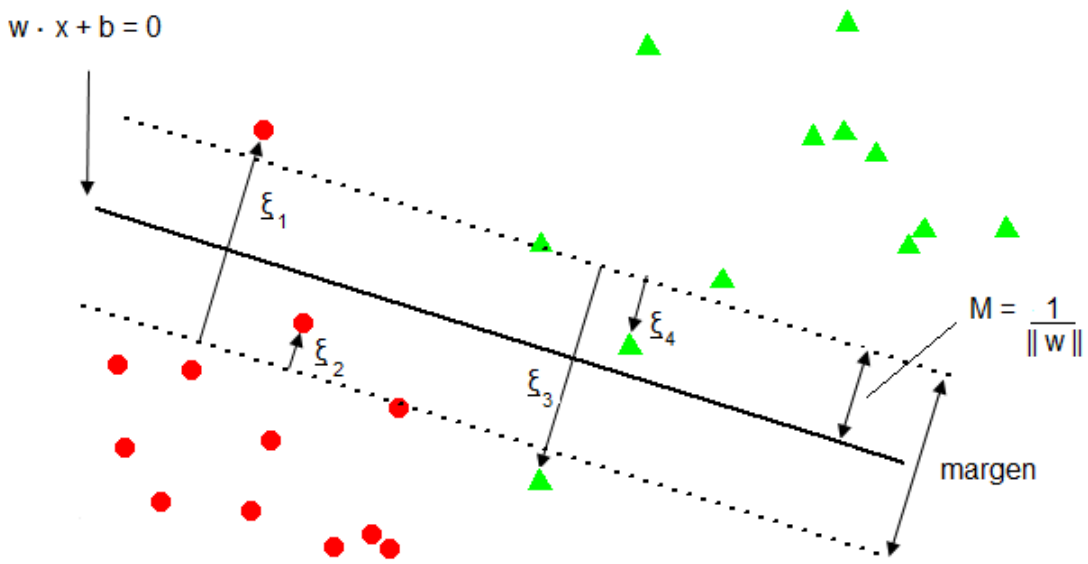


Figura 3.3: Hiperplano Óptimo Generalizado. Mediante la introducción de variables de holgura $\xi_1, ..., \xi_n$ se permite que la clasificación mediante el hiperplano no sea perfecta y cometa errores de clasificación.

Estas variables de holgura ξ_i representan la distancia al hiperplano de las observaciones que caen dentro del margen, o que están mal clasificadas por el hiperplano. Es decir, las variables de holgura miden la magnitud del error cometido por el clasificador usando el hiperplano.

Así, se tiene que ξ_i puede tomar los siguientes valores:

- Si $\xi_i = 0$, entonces x_i está correctamente clasificada y está fuera del margen.
- Si $0 < \xi_i \leq 1$, entonces x_i está correctamente clasificada, pero está dentro del margen.
- Si $\xi_i > 1$, entonces x_i está mal clasificada (está en el lado incorrecto del hiperplano).

3.3.1. Cálculo del Hiperplano Óptimo Generalizado

Encontrar el hiperplano con la menor cantidad de errores de entrenamiento requiere minimizar la siguiente expresión:

$$\Theta(\xi) = \sum_{i=1}^{n} \theta(\xi_i)$$

sujeto a las restricciones:

$$y_i \left(w \cdot x_i + b\right) \geq 1 - \xi_i, \quad i = 1, ..., n, \quad \xi_i \geq 0 \tag{3.26}$$

$$(w \cdot w) \leq A^2 \tag{3.27}$$

donde A es una restricción arbitraria sobre el valor máximo para el módulo del vector w que define el hiperplano, y por ende también constituye una restricción sobre el valor mínimo que puede tomar el margen. Además:

$$\theta(\xi) = 0 \quad , \quad \text{si } \xi = 0$$
$$\theta(\xi) = 1 \quad , \quad \text{si } \xi > 0$$

Este problema de optimización no es directamente resoluble por lo que se estudia una aproximación que consiste en maximizar la siguiente función (Vapnik, 1998):

$$\Theta(\xi) \approx \sum_{i=1}^{n} \xi_i^{\sigma}$$

bajo las restricciones indicadas en 3.26 y 3.27, y donde $\sigma \geq 0$ es un valor pequeño. Por un tema computacional se elige un valor de $\sigma = 1$, el σ más pequeño que lleva a un problema de optimización simple (Vapnik, 1998). Por lo tanto, el problema de optimización se reduce a minimizar el siguiente funcional:

$$\Theta(\xi) \approx \sum_{i=1}^{n} \xi_i \tag{3.28}$$

sujeto a las restricciones (3.26 y 3.27).

Para resolver este problema, tal como se hizo en el caso separable, se debe encontrar el punto silla del Lagrangiano:

$$L(w, b, \alpha, \beta, \gamma) = \sum_{i=1}^{n} \xi_i - \frac{1}{2}\gamma(A^2 - w \cdot w) - \sum_{i=1}^{n} \alpha_i \left(y_i \left(w \cdot x_i + b\right) - 1 + \xi_i\right) - \sum_{i=1}^{n} \beta_i \xi_i \tag{3.29}$$

es decir, se debe encontrar el mínimo del Lagrangiano respecto a w, b, ξ_i y su máximo respecto a los multiplicadores de Lagrange no negativos α_i, β_i, γ. Los parámetros que minimizan el Lagrangiano cumplen con las siguientes condiciones:

$$\frac{\partial L(w, b, \xi, \alpha, \beta, \gamma)}{\partial w} = \gamma w - \sum_{i=1}^{n} \alpha_i y_i x_i = 0$$

$$\frac{\partial L(w, b, \xi, \alpha, \beta, \gamma)}{\partial b} = -\sum_{i=1}^{n} \alpha_i y_i = 0$$

$$\frac{\partial L(w, b, \xi, \alpha, \beta, \gamma)}{\partial \xi_i} = 1 - \alpha_i - \beta_i = 0$$

de las cuales se obtiene que:

$$w = \frac{1}{\gamma} \sum_{i=1}^{n} \alpha_i y_i x_i \tag{3.30}$$

$$\sum_{i=1}^{n} \alpha_i y_i = 0 \tag{3.31}$$

$$\alpha_i + \beta_i = 1 \tag{3.32}$$

Luego, sustituyendo el valor de w en 3.30 en el Lagrangiano 3.29 y tomando en consideración las ecuaciones 3.31 y 3.32 se obtiene el siguiente funcional:

$$W(\alpha, \gamma) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2\gamma} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) - \frac{\gamma A^2}{2} \tag{3.33}$$

que debe ser maximizado respecto a las siguientes restricciones:

$$\begin{gathered} \textstyle\sum_{i=1}^{n} \alpha_i y_i = 0 \\ 0 \leq \alpha_i \leq 1 \\ \gamma \geq 0 \end{gathered}$$

El valor del parámetro γ que maximiza el funcional $W(\alpha, \gamma)$ y que en adelante llamaremos γ_0 cumple con la siguiente condición:

$$\frac{\partial W(w, \gamma)}{\partial \gamma} = \frac{1}{2\gamma^2} \sum_{i=1}^{n} \sum_{i=1}^{n} \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) - \frac{A^2}{2} = 0$$

de donde se deduce que el valor óptimo γ_0 es igual a (Vapnik, 1998):

$$\gamma_0 = \frac{\sqrt{\sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j (x_i \cdot x_j)}}{A} \tag{3.34}$$

con lo cual la expresión para calcular el hiperplano queda como la maximización del siguiente funcional:

$$W(\alpha) = \sum_{i=1}^{n} \alpha_i - A \sqrt{\sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j (x_i \cdot x_j)} \tag{3.35}$$

sujeto a las restricciones:

$$\begin{gathered} \textstyle\sum_{i=1}^{n} \alpha_i y_i = 0 \\ 0 \leq \alpha_i \leq 1 \end{gathered}$$

Luego, el vector de parámetros $\alpha_0 = (\alpha_1^0, ..., \alpha_n^0)$ que maximiza el funcional $W(\alpha)$, define

por 3.30 y 3.34 la regla de decisión del hiperplano óptimo generalizado de la siguiente forma:

$$f(x) = \text{signo}\left(\frac{A}{\sqrt{\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i^0\alpha_j^0 y_i y_j (x_i \cdot x_j)}}\sum_{i=1}^{n}\alpha_i^0 y_i (x \cdot x_i) + b_0\right) \tag{3.36}$$

Para calcular el valor de la constante de desplazamiento b_0 se usa la condición de Kühn-Tucker para el caso no separable (Hamel, 2009):

$$\alpha_i^0\left(y_i\left(w_0 \cdot x_i + b_0\right) + \xi_i^0 - 1\right) = 0 \tag{3.37}$$

donde dada la observación x_i se tiene que: α_i^0 es el valor óptimo del multiplicador de Lagrange, y_i es el valor de la variable respuesta, w_0 es el vector que define el hiperplano óptimo, y ξ_i^0 es el valor óptimo de la variable de holgura. De esta ecuación se deduce que, para cualquier vector de soporte x_{SV} elegido arbitrariamente, debe cumplirse la siguiente condición:

$$y_{SV}\left(w_0 \cdot x_{SV} + b_0\right) + \xi_{SV}^0 - 1 = 0 \tag{3.38}$$

donde, dado el vector de soporte x_{SV}, se tiene que: α_{SV} es el multiplicador de Lagrange, y_{SV} es la variable respuesta y, y ξ_{SV} es la variable de holgura ξ.

Finalmente, se elige un vector de soporte y se despeja el valor de b_0 en la ecuación 3.38. Sin embargo, a diferencia del caso separable, esta elección no es totalmente arbitraria y tiene una restricción: el valor de variable de holgura ξ_{SV} debe ser igual a 0, puesto que esto indica que el vector de soporte se encuentra en el lado correcto del hiperplano y del margen (Hamel, 2009). Así, sea x_{SV}^+ un vector de soporte que pertenece a la clase $y = +1$ y con variable de holgura $\xi_{SV} = 0$, entonces el valor de la constante de desplazamiento b_0 es:

$$b_0 = 1 - w_0 \cdot x_{SV}^+ = 1 - \frac{A}{\sqrt{\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i^0\alpha_j^0 y_i y_j (x_i \cdot x_j)}}\sum_{i=1}^{n}\alpha_i^0 y_i (x_i \cdot x_{Sv}^+) \tag{3.39}$$

3.3.2. Simplificación del Cálculo del Hiperplano Óptimo Generalizado

Para simplificar los cálculos computaciones se introduce una versión modificada del hiperplano óptimo generalizado. Este hiperplano está determinado por el vector w que minimiza el funcional:

$$\Theta(w, \xi) = \frac{1}{2}(w \cdot w) + C(\sum_{i=1}^{n}\xi_i) \tag{3.40}$$

sujeto a la restricción:

$$y_i\left(x_i \cdot w + b\right) \geq 1 - \xi_i, \quad i = 1, ..., n \tag{3.41}$$

donde C es una constante positiva que determina el balance entre dos metas en conflicto: minimizar el error de entrenamiento (es decir, minimizar la suma de las variables de holgura

ξ_i), y maximizar el margen $\frac{1}{\|w\|}$ (Hamel, 2009).

El valor de la constante C, también llamada costo, indica la tolerancia a los errores de clasificación. Así, valores grandes de C configuran errores de clasificación costosos; por lo tanto, la solución óptima tendrá pocas variables de holgura ξ_i distintas a cero. Más precisamente, un valor grande de C forzará la búsqueda de hiperplanos con márgenes pequeños.

Por otro lado valores pequeños de C definen errores de clasificación "económicos", por lo que el uso de las variables de holgura ξ_i estará menos penalizado y es posible encontrar soluciones con un margen grande (Hamel, 2009).

La solución a este problema de optimización es similar a la encontrada para el caso separable (Vapnik, 1998); es decir, se encuentra en el mínimo del Lagrangiano respecto a w, b, ξ, y en el máximo respecto a los multiplicadores de Lagrange α, β:

$$L(w, b, \alpha, \beta, \xi) = \frac{1}{2} w \cdot w + C \sum_{i=1}^{n} \xi_i - \sum_{i=1}^{n} \alpha_i \left(y_i (w \cdot x_i + b) + \xi_i - 1 \right) - \sum_{i=1}^{n} \beta_i \xi_i \tag{3.42}$$

Para encontrar los vectores w que definen el hiperplano óptimo generalizado,

$$w = \sum_{i=1}^{n} \alpha_i y_i x_i$$

se debe encontrar los parámetros $\alpha = \alpha_1, ..., \alpha_n$ que maximicen la misma forma cuadrática que en el caso separable (Vapnik, 2000):

$$W(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j (x_i \cdot x_j) \tag{3.43}$$

bajo las restricciones:

$$0 \leq \alpha_i \leq C, \quad i = 1, ..., n \tag{3.44}$$

$$\sum_{i=1}^{n} \alpha_i y_i = 0 \tag{3.45}$$

De la misma forma que en el caso separable, solo algunos de los coeficientes $\alpha_i^0 = \alpha_1^0, ..., \alpha_n^0$ son diferentes de cero y corresponden a los vectores de soporte que determinan el hiperplano óptimo generalizado:

$$\sum_{i=1}^{n} \alpha_i^0 y_i (x_i \cdot x) + b_0 = 0 \tag{3.46}$$

Debemos notar que cuando el coeficiente C en el funcional $\Theta(w, \xi)$ (3.40) es igual al valor óptimo del parámetro γ_0 en la maximización del funcional $W(\alpha, \gamma)$ (3.33), entonces la solución para ambos problemas de optimización coincide (Vapnik, 1998). Es decir: $C = \gamma_0$.

Finalmente, la regla de decisión para este clasificador es de la forma:

$$f(x) = \text{signo}\left(\sum_{SV} \alpha_i^0 y_i (x_i \cdot x) + b_0\right) \tag{3.47}$$

donde b_0 se calcula en base a la condición de Kühn-Tucker para el caso no separable 3.38:

$$b_0 = 1 - w_0 \cdot x_{SV}^+ = 1 - \sum_{SV} \alpha_i^0 y_i (x_i \cdot x_{SV}^+) \tag{3.48}$$

donde x_{SV}^+ es un vector de soporte de la clase $y = +1$ con valor de la variable de holgura ξ igual a 0.

3.4. Máquinas de Soporte Vectorial

La Máquina de Soporte Vectorial es un método de aprendizaje estadístico que se basa en la transformación del espacio de origen de los datos a un espacio de mayor dimensión en el cual es posible hacer una separación lineal.

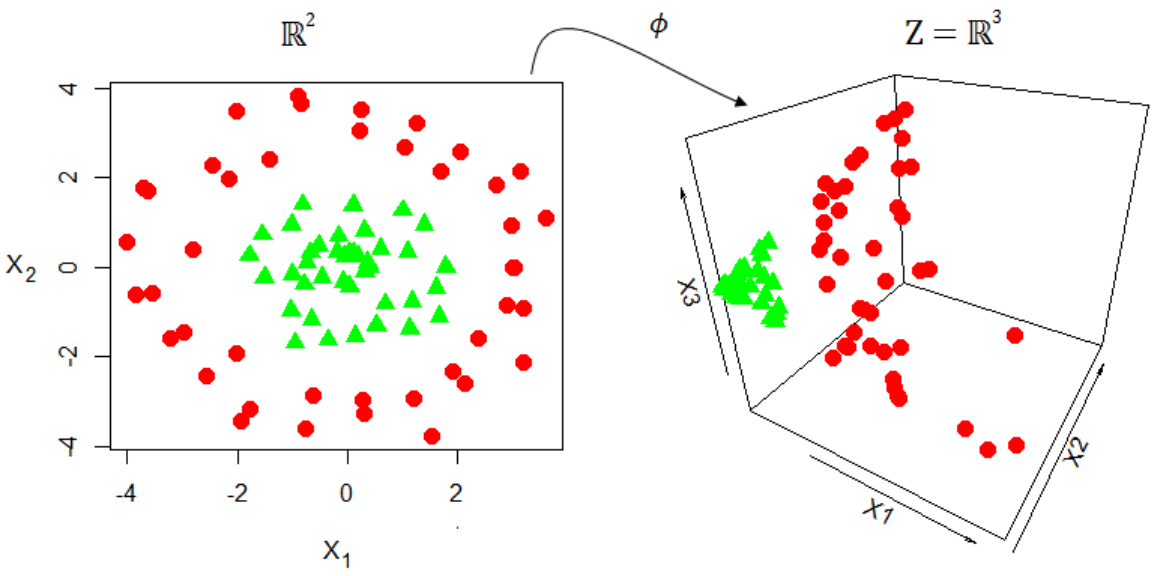


Figura 3.4: Mediante una función no lineal ϕ se lleva los datos de un espacio de características origen ($\mathbb{R}^2$) a un espacio de características más grande ($Z = \mathbb{R}^3$). En este espacio más grande es posible realizar una separación lineal mediante hiperplanos.

Examinemos el caso propuesto en la imagen 3.4. En el gráfico de la izquierda se tiene un conjunto de datos en $\mathbb{R}^2$ que es imposible de separar linealmente usando los métodos aquí estudiados (Hiperplano Óptimo o Hiperplano Óptimo Generalizado).

Para solucionar este problema, como sugiere el gráfico a la derecha, no se trabaja con los datos en el espacio de origen $\mathbb{R}^2$. En cambio se tiene una función ϕ que lleva los datos a un espacio de mayor dimensión Z, en este caso $\mathbb{R}^3$, en el cual sí es posible separar los datos

linealmente mediante hiperplanos.

Es decir, dado un vector $x = (x_a, x_b)$ en un espacio de origen $\mathbb{R}^2$, se busca una función ϕ que lleve este vector x a un vector z de mayor dimensión. Por ejemplo, una forma de definir a ϕ podría ser:

$$z = \phi(x) = (x_a^2, x_b^2, x_a x_b)$$

Las Máquinas de Soporte Vectorial utilizan el producto interno en el espacio vectorial extendido (de mayor dimensión) como una medida de similaridad para definir la regla de separación, de forma similar a los métodos del Hiperplano Óptimo y el Hiperplano Óptimo Generalizado.

3.4.1. Mapeo del Producto Interno en un Espacio de Mayor Dimensión

Las Máquinas de Soporte Vectorial buscan un nuevo espacio Z de mayor dimensión que el espacio de origen de los datos. En este espacio Z se usa el producto interno como medida de similaridad para comparar vectores.

Sin embargo, aún si se garantiza la existencia de un hiperplano que separe los datos en el espacio extendido Z y que generalice bien para datos desconocidos, queda pendiente cómo trabajar el problema de forma eficiente.

Como solución a este problema, la construcción de un hiperplano en el espacio Z no requiere que se trabaje en este espacio de forma explícita, y tampoco se requiere conocer la función ϕ. Sólo hace falta calcular los productos internos entre los vectores de soporte, y entre los vectores de soporte y los vectores en el espacio extendido.

Así, sean dos vectores x_1 y x_2 que pertenecen al espacio de origen $\mathbb{R}^m$, y sus respectivas imágenes z_1 y z_2 en el espacio extendido $Z = \mathbb{R}^n$ (donde $n > m$) definidas a través de una función ϕ; es decir:

$$x_1 \longrightarrow z_1 = \phi(x_1) = (\phi_1(x_1), \phi_2(x_1), ..., \phi_n(x_1))$$
$$x_2 \longrightarrow z_2 = \phi(x_2) = (\phi_1(x_2), \phi_2(x_2), ..., \phi_n(x_2))$$

De acuerdo a la teoría de Hilbert-Schmidt, el producto interno en un espacio $\mathbb{R}^n$ (el cual es un espacio de Hilbert) se puede representar como (Vapnik, 2000):

$$(z_1 \cdot z_2) = \sum_{r=1}^{n} \phi_r(x_1)\phi_r(x_2) = K(x_1, x_2) \tag{3.49}$$

si y sólo si se cumple la condición del teorema de Mercer: para que la función Kernel K

$$K(x_1, x_2) = \sum_{r=1}^{n} \phi_r(x_1)\phi_r(x_2)$$

describa un producto interno en el espacio Z, es suficiente y necesario que la condición:

$$\int\int K(u,v)g(u)g(v)dudv \geq 0$$

sea válida para toda función $g \neq 0$ que cumpla con la condición (Vapnik, 2000):

$$\int g^2(u)du \leq \infty$$

Es decir, el teorema de Mercer garantiza que existe la función ϕ aún cuando no es necesario conocerla explícitamente, y que la función Kernel $K(x_1, x_2)$ mapea al producto interno en el espacio extendido Z que genera ϕ, si y sólo si la función Kernel K es una función simétrica semidefinida positiva (Vapnik, 2000).

3.4.2. Construcción de la Máquina de Soporte Vectorial

En los métodos de clasificación revisados en el presente desarrollo (hiperplano óptimo e hiperplano óptimo generalizado) se define una regla de decisión que depende del producto interno de los vectores de entrenamiento de la forma:

$$f(x) = \text{signo}\left(\sum_{SV} y_i \alpha_i^0 (x_i \cdot x) + b_0\right)$$

En las Máquinas de Soporte Vectorial se requiere plantear una regla de decisión:

$$f(x) = \text{signo}\left(\sum_{SV} y_i \alpha_i^0 \left(\phi(x_i) \cdot \phi(x_i)\right) + b_0\right) \tag{3.50}$$

de forma que se use el producto interno de las imágenes de los vectores de entrenamiento en el espacio extendido Z. Sin embargo en vez de encontrar una función ϕ que lleve explícitamente los vectores a este espacio, usaremos las funciones Kernel que, como vimos en la sección anterior, define este producto interno mediante operaciones en el espacio de origen. Así, la regla de decisión para las Máquinas de Soporte Vectorial tiene la forma:

$$f(x) = \text{signo}\left(\sum_{SV} y_i \alpha_i^0 K(x_i, x) + b_0\right) \tag{3.51}$$

Por lo tanto para construir la función de decisión 3.51 podemos usar los métodos expuestos para el Hiperplano Óptimo (3.2.1) y para el Hiperplano Óptimo Generalizado (3.3.1). Así, tenemos los siguientes casos:

1. En el caso separable debemos encontrar los coeficientes α_i que definen los vectores de soporte. Estos son los vectores que se encuentran exactamente en el margen, es decir,

cumplen la condición:

$$y_i \sum_{SV} y_i \alpha_i^0 K(x_i, x) = 1$$

Para hallar los coeficientes α_i se encuentra el máximo de la función $W(\alpha)$:

$$W(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j K(x_i, x_j) \tag{3.52}$$

sujeto a las siguientes restricciones:

$$\begin{aligned} &\textstyle\sum_{i=1}^{n} \alpha_i y_i = 0 \\ &\alpha_i \geq 0, \qquad i = 1, ..., n \end{aligned}$$

2. En el caso no separable, se busca maximizar $W(\alpha)$ bajo las siguientes restricciones:

$$\begin{aligned} &\textstyle\sum_{i=1}^{n} \alpha_i y_i = 0 \\ &0 \leq \alpha_i \leq C, \quad i = 1, ..., n \end{aligned}$$

3.4.3. Máquinas de Soporte Vectorial para datos desbalanceados

Un problema de clasificación binaria es desbalanceado cuando una de las clases (generalmente la negativa $y = -1$) es más numerosa que la otra (generalmente la positiva $y = +1$). Existen muchas aplicaciones reales con datos con esta característica de desbalance, por ejemplo: detección de fraudes, identificación de potenciales clientes que caerían en default (mora) en un crédito, identificación de clientes con propensión a la compra de un producto, etc. En estas aplicaciones la proporción de elementos de la clase positiva respecto a la negativa puede ser muy variada, tomando valores moderados (por ejemplo, de 20 a 80) a extremos (por ejemplo, de 1 a $1,000$ o más).

La implementación estudiada para Máquinas de Soporte Vectorial en casos no separables (3.4.2 y 3.3.1) asigna un mismo costo C a los errores de clasificación de las dos clases. En caso de desbalance esto podría ocasionar que el hiperplano que divide las clases esté más cerca a la clase minoritaria, con lo que el método de aprendizaje tendería a clasificar los datos en la clase mayoritaria (Batuwita y Palade, 2013). Naturalmente, esto no produce un modelo óptimo.

Una forma de solucionar este problema es mediante la introducción de costos diferentes para los errores de clasificación de cada clase. Así, en lugar de tener un costo C de clasificación, se tiene a C^+ como el costo de error de clasificación para la clase positiva ($y = +1$), y a C^- como el costo de error de clasificación para la clase negativa ($y = -1$) (Chang y Lin, 2011).

Luego, el problema de encontrar el hiperplano óptimo w se convierte en el problema de minimizar la función:

$$\Theta(w, \xi) = \frac{1}{2}(w \cdot w) + C^+ \sum_{i|y_i=+1} \xi_i + C^- \sum_{i|y_i=-1} \xi_i \tag{3.53}$$

sujeto a la restricción:

$$y_i \left(w \cdot \Phi(x_i) + b \right) \geq 1 - \xi_i \tag{3.54}$$

donde Φ se refiere al mapeo (no necesariamente conocido) que lleva el vector x_i a un espacio de mayor dimensión. Según el teorema de Mercer, el producto interno en este espacio extendido se puede calcular mediante una función Kernel K que trabaja en el espacio de origen.

Aplicando el método del Lagrangiano indicado en la sección 3.3.2, el problema de encontrar el hiperplano óptimo $w = \sum_{i=1}^{n} \alpha_i y_i x_i$ se convierte en la maximización de la función $W(\alpha)$:

$$W(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j K(x_i, x_j) \tag{3.55}$$

sujeto a la restricciones:

$$\begin{aligned} &\textstyle\sum_{i=1}^{n} y_i \alpha_i = 0 \\ &0 \leq \alpha_i \leq C^+ \quad \text{si } y_i = +1 \\ &0 \leq \alpha_i \leq C^- \quad \text{si } y_i = -1 \end{aligned} \tag{3.56}$$

Esta versión modificada del SVM puede reducir el efecto del desbalance de las clases, asignando un costo de error de clasificación mayor para la clase minoritaria. Esto reduce el sesgo del clasificador al alejar el hiperplano de la clase minoritaria. Como regla general, se obtiene buenos resultados si el ratio C^-/C^+ es igual a la proporción de elementos de la clase minoritaria respecto a clase la mayoritaria (Batuwita y Palade, 2013).

3.4.4. Consideraciones Prácticas en la Implementación de SVM

Antes de aplicar el método Máquinas de Soporte Vectorial (SVM) a un conjunto de datos, es conveniente tomar en cuenta las siguientes consideraciones:

- Los SVM requieren que las variables predictoras de cada fila de datos sean representadas como un vector de números reales debido a que el producto interno, que se usa como medida de similaridad, sólo tiene sentido con datos numéricos. Por lo tanto si hay variables predictoras categóricas primero deben ser convertidas en datos numéricos. La forma usual de lograr esta conversión es usar el método de variables *dummy*; es decir, convertir una variable con k categorías en $k-1$ variables numéricas que sólo toman dos valores posibles: 1 y 0 (Hsu et al., 2000). Por ejemplo, una variable con tres categorías como $\{A, B, C\}$ puede representarse con 2 variables numéricas con valores $(1, 0)$, $(0, 1)$, y $(0, 0)$. Esta codificación no afecta significativamente a las variables dicotómicas, dado que no es necesario agregar variables *dummy* adicionales, sino sólo basta asegurarse que estén codificadas como 1 y 0.

- Es recomendable escalar los datos antes de aplicar el método SVM. Esto sirve para evitar que variables con rangos numéricos muy altos dominen a variables con rangos numéricos más pequeños (situación que se daría, por ejemplo, si una de las variables predictoras fuera el ingreso, y otra fuera la edad). La forma usual y recomendada de

escalamiento es estandarizar los datos; es decir, llevarlos a media 0 y varianza 1 (Hsu et al., 2000).

La mayoría de paquetes estadísticos, particularmente las librerías en R revisadas en este trabajo, implementan estas dos consideraciones de forma automática.

3.5. Tipos de Funciones Kernel

En esta sección se desarrolla las funciones Kernel más usadas para construir Máquinas de Soporte Vectorial. En la literatura no existe una regla que indique explícitamente qué tipo de Kernel usar en cada problema. Por lo tanto, al implementar Máquinas de Soporte Vectorial se recomienda aplicar distintos tipos de funciones Kernel, probando diferentes configuraciones de hiperparámetros, a fin de encontrar la opción que genera mejores resultados.

Por otro lado, se considera que el Kernel de Base Radial funciona bien como función de propósito general, y como tal puede ser una buena opción de Kernel por defecto.

3.5.1. Kernel de Base Radial

Los kernel de base radial, o RBF por sus siglas en inglés (Radial Basis Functions) dependen de la distancia entre dos vectores dados u y v, y de un hiperparámetro γ. Tienen la siguiente forma:

$$K_\gamma(u, v) = \exp(-\gamma \left|u - v\right|^2) \tag{3.57}$$

La función de base radial K_γ es una función simétrica semidefinida positiva para cualquier valor de γ, por lo tanto siempre cumple la condición del teorema de Mercer (Vapnik, 2000).

Los kernel de base radial mapean la siguiente regla de decisión en una Máquina de Soporte Vectorial:

$$f(x) = \text{signo}\left(\sum_{i=1}^{n} y_i \alpha_i \exp(-\gamma \left|x - x_i\right|^2) - b\right) \tag{3.58}$$

3.5.2. Kernel Polinomial

Los kernel polinomiales entre dos vectores u y v tienen la forma:

$$K_d(u, v) = (\gamma(u \cdot v) + \delta)^d \tag{3.59}$$

donde:

- d es el grado del polinomio
- γ es un factor de escala que acota el valor del kernel a un intervalo. Su valor depende de la dimensión de los datos y del rango de valores que éstos puedan tomar.
- δ es un valor de desplazamiento respecto al origen

La función kernel K_d es una función simétrica semidefinida positiva para cualquier valor de los parámetros d, γyδ, por lo tanto siempre satisface las condiciones del teorema de Mercer (Vapnik, 2000).

Los kernel polinomiales generan Máquinas de Soporte Vectorial con la siguiente regla de decisión:

$$f(x) = \text{signo}\left(\sum_{SV} y_i \alpha_i \left(\gamma(x \cdot x_i) + \delta\right)^d + b\right) \tag{3.60}$$

3.5.3. Kernel Sigmoidal

Los kernel sigmoidales aplicados a dos vectores u y v tienen la siguiente forma:

$$K_S(u, v) = S\left(\gamma(u \cdot v) + \delta\right) \tag{3.61}$$

donde $S(z)$ es una función sigmoidal. Para este tipo de kernel suele utilizarse la función tangente hiperbólica, con lo cual el kernel queda de la siguiente forma (Vapnik, 2000):

$$K_S(u, v) = \tanh\left(\gamma(u \cdot v) + \delta\right) \tag{3.62}$$

Los kernel sigmoidales generan Máquinas de Soporte Vectorial con la siguiente regla de decisión:

$$f(x) = \text{signo}\left(\sum_{SV} y_i \alpha_i S\left(\gamma(x \cdot x_i) + \delta\right) + b\right) \tag{3.63}$$

3.5.4. Kernel Lineal

Los kernel lineales mapean directamente el producto interno en el espacio de origen; es decir, para dos vectores u y v:

$$K_l(u, v) = u \cdot v \tag{3.64}$$

Este tipo de Kernel cumple la condición del teorema Mercer. Esta condición se evalúa para asegurar que la función Kernel sea equivalente al producto interno de dos vectores en un espacio diferente, sin necesidad de conocer el mapeo que traduzca el espacio de origen al nuevo espacio. Dado que el Kernel lineal define el producto interno en el espacio de origen, por definición siempre cumple con la condición del teorema de Mercer.

Los kernel lineales generan una regla de decisión equivalente a la del Hiperplano Óptimo, o Hiperplano Óptimo Generalizado:

$$f(x) = \text{signo}\left(\sum_{SV} y_i \alpha_i (x \cdot x_i) + b\right) \tag{3.65}$$

3.6. Clasificación para el caso multinomial

El desarrollo anterior se ha enfocado en resolver problemas de clasificación binaria mediante el método de aprendizaje Máquinas de Soporte Vectorial. A continuación se presenta la idea general de cómo extender este método para problemas de clasificación multinomial.

Existen varias propuestas para extender las Máquinas de Soporte Vectorial (SVM) a problemas de clasificación con k clases. Sin embargo 2 de ellas son las más populares: el enfoque uno-versus-uno y el enfoque uno-versus-todo (James et al., 2013):

- El enfoque uno-versus-uno construye $\binom{k}{2}$ SVMs binarios, cada uno de los cuales evalúa una de las combinaciones por pares de las k clases. Luego, la regla de decisión de este enfoque clasifica una observación usando cada uno de los $\binom{k}{2}$ SVM, y luego elige la clase más votada por los SVM.

- El enfoque uno-versus-todo construye k SVMs, cada uno de los cuales compara cada una de las k clases (la cual se codifica como $y = +1$) versus las $k - 1$ clases restantes (las cuales se codifican colectivamente como $y = -1$). Si w_i es el vector que define el hiperplano separador resultado de ajustar un SVM comparando la clase i versus el resto, y x^* es una observación que se busca clasificar, entonces se asigna esta observación a la clase i con el valor $(w_i \cdot x^*)$ más alto (James et al., 2013).

  El valor $(w_i \cdot x^*)$ puede interpretarse como la distancia de la observación x^* al hiperplano del $i-$ésimo SVM. Entonces, un valor positivo de $(w_i \cdot x^*)$ indica que el punto x^* se clasifica en la clase $y = +1$; es decir, que pertence a la clase i. Luego, mientras más alto sea este valor, la observación x^* estará más alejada del $i-$ésimo hiperplano, con lo cual tendremos más confianza que la observación x^* pertenece a la clase i (Hamel, 2009).

3.7. Regresión de Soporte Vectorial

En el desarrollo anterior se ha expuesto el uso del método de aprendizaje Máquinas de Soporte Vectorial (SVM) para resolver problemas de clasificación. Adicionalmente a ello, los SVM pueden adaptarse para resolver problemas de regresión. Este tema escapa del alcance de la presente tesis; sin embargo, presentaremos algunos conceptos introductorios al tema.

Primero tengamos en cuenta un modelo de regresión lineal:

$$f(x) = x^T w + b_0$$

Para estimar w, necesitamos minimizar la función de riesgo regularizada (Hastie et al., 2009):

$$H(w, b_0) = \sum_{i=1}^{n} c(x_i, y_i, f(x_i)) + \frac{\lambda}{2} \|w\|^2$$

donde c es una función de pérdida para el caso de regresión (2.4.1), y λ es un parámetro de

regularización que indica el balance entre la minimización de los errores y la simplicidad de la función f. Típicamente se usa la función de pérdida ϵ insensible, que ignora los errores de magnitud menor a ϵ. Haciendo una analogía con el escenario de clasificación, los puntos con "errores pequeños"serían equivalentes a los puntos en el lado correcto del hiperplano (Hastie et al., 2009).

Si $\hat{w}$ y $\hat{w}_0$ minimizan a H, entonces la solución tiene la forma (Hastie et al., 2009):

$$\hat{w} = \sum_{i=1}^{n} (\hat{\beta}_i - \hat{\alpha}_i) x_i$$

$$f(x) = \sum_{i=1}^{n} (\hat{\beta}_i - \hat{\alpha}_i)(x \cdot x_i) + \hat{b}_0$$

donde $\hat{\alpha}_i, \hat{\beta}_i$ son positivos y minimizan la función $W(\alpha, \beta, \epsilon)$:

$$W(\alpha, \beta, \epsilon) = \sum_{i=1}^{n} (\beta_i + \alpha_i) - \sum_{i=1}^{n} y_i(\beta_i - \alpha_i) + \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} (\beta_i - \alpha_i)(\beta_j - \alpha_j)(x_i \cdot x_j)$$

sujeto a las restricciones:

$$0 \leq \alpha_i, \quad \beta_i \leq \tfrac{1}{\lambda}$$
$$\textstyle\sum_{i=1}^{n} (\beta_i - \alpha_i) = 0$$
$$\alpha_i \beta_i = 0$$

Al igual que en el caso de clasificación, la solución depende del producto interno de los vectores de entrenamiento. Luego, podemos generalizar este método de regresión para resolver problemas no lineales mediante el uso de funciones Kernel (Hastie et al., 2009).

Capítulo 4

Aplicación de SVM a un problema de clasificación

4.1. Objetivo

El presente trabajo de aplicación tiene los siguientes objetivos:

- Aplicar los conceptos teóricos revisados en los capítulos 2 y 3 y probar la técnica SVM en un problema real de clasificación binaria.
- Comparar el desempeño de predicción del modelo SVM versus un modelo de regresión logística sobre los mismos datos.

4.2. Descripción de la Aplicación

Actualmente la entidad financiera "EF" tiene un problema de baja permanencia en el producto Seguro de Protección de Tarjetas. Aproximadamente la mitad de los clientes que adquiere este producto lo abandona antes del sexto mes. Esta situación genera un problema serio para la entidad financiera, dado que la rentabilidad del producto está directamente ligada al tiempo de permanencia: un cliente que compra el producto y lo mantiene por menos de 6 meses genera pérdidas en lugar de rentabildad.

Como solución a este problema se plantea aplicar el método de aprendizaje estadístico Máquinas de Soporte Vectorial (SVM) a los datos de clientes de la entidad financiera, para resolver el problema de predecir los clientes más propensos no sólo a comprar el producto, sino también a mantenerlo por lo menos durante seis meses. Para ello, se define el evento de interés sobre una variable respuesta Y de la siguiente forma:

- Si $Y = +1$, entonces el cliente compró el producto Seguro de Protección de Tarjeta y lo mantuvo activo por lo menos durante 6 meses.
- Si $Y = -1$, entonces el cliente no compró el producto Seguro de Protección de Tarjeta; o compró el producto pero lo abandonó antes del sexto mes.

Todo el trabajo de aplicación se realizó usando el *software* estadístico R, mediante librerías que implementan las Máquinas de Soporte Vectorial. Como parte de la aplicación se buscó encontrar la configuración que optimiza el poder de predicción de los modelos SVM: el tipo de

kernel a utilizar, los hiperparámetros del kernel, y el valor de la constante de costo C que mide la tolerancia a los errores de clasificación (esta última es una constante en la formulación del problema pero es tratada como un parámetro en las librerías en R que implementan SVM).

Para encontrar la configuración adecuada del modelo se entrenó los modelos SVM probando diferentes parámetros, y se elegió el modelo con el mejor desempeño de predicción medido con el área bajo la curva ROC. Esta selección se realizó midiendo el desempeño de predicción en una muestra de validación diferente a la de entrenamiento. Luego, se midió el poder de predicción del modelo seleccionado en una muestra de prueba final.

Finalmente se comparó el desempeño del modelo obtenido mediante las Máquinas de Soporte Vectorial versus el modelo paramétrico de regresión logística o Logit. Se elige el modelo de regresión logística debido a que es bastante conocido, y a que la entidad financiera ya lo tiene implementado para esta aplicación.

4.3. Descripción de los Datos

Los datos para la aplicación tienen 2'617,267 registros (clientes) con 680 variables (incluyendo la variable respuesta y variables para identificar al cliente). El volumen de datos supera los 6 GB en un archivo plano de tipo CSV.

La tasa de respuesta, es decir, la proporción de clientes que presenta el evento de interés es muy baja: aproximadamente el 0,6 % del total de la base compra el producto y lo mantiene por 6 meses o más. Esto, sumado al alto volumen de los datos, introduce una complejidad importante en la aplicación.

Las variables predictoras se pueden resumir de la siguiente forma:

- Variables sociodemográficas del cliente (edad, estado civil, ubicación, etc.)
- Variables de Montos, Saldos, y Tenencia en diversos productos: CTS, Depósito a Plazo, Fondos Mutuos, Créditos Personales, Créditos Hipotecarios, Tarjetas de Crédito, Productos Pasivos (Cuentas de Ahorro, Cuentas Corrientes), Seguros, Tarjeta de Débito. Incluye: valores por moneda, valores del último mes, valores promedio de los últimos 6 o 12 meses, evolución, tendencias, etc.
- Variables de Montos y Saldos en el Sistema Financiero (SBS): por tipo y estado de la deuda, por Entidad Financiera, etc.
- Variables de Reclamos por Producto.
- Variables de Montos y Tenencia de Transacciones por Canal, Rubro, y Programa de Beneficio.
- Variables de Montos de Pago y Tenencia de Utilities y Servicios.
- Variables de Campañas, por Tipo de Campaña, Estado, Canal.

4.4. Consideraciones Computacionales

4.4.1. Lectura de los Datos

Debido al alto volumen de los datos no es factible cargarlos en memoria usando las funciones tradicionales de R *read.csv* o *read.table*, ni trabajarlos con la clase *data.frame*. Usando estas funciones no es posible cargar los archivo de datos completo dado que consume más del 100 % de la memoria física disponible (16 GB en este caso); sólo permiten cargar los datos por partes.

Luego, para levantar los datos al ambiente del *software* estadístico R se encontró las siguientes dos alternativas:

- La función *fread* de la librería *data.table* tiene un buen desempeño en la lectura de archivos planos, así como en operaciones rápidas con grandes volúmenes de datos, por ejemplo resúmenes, agrupaciones, cruce de tablas y otros. La sintaxis de trabajo es muy similar a la de la clase tradicional *data.frame* (Dowle et al., 2014).
- La función *read.csv.ffdf* de la libreria *ff* provee estructuras de datos que son almacenadas en disco pero se comportan como si estuvieran en memoria RAM, asignando de forma transparente sólo una sección de los datos (tamaño de página) en la memoria principal (Adler et al., 2014).

Finalmente se decidió hacer la lectura de los datos con la función *fread* de la librería *data.table* debido a que ésta provee un mejor desempeño general, al trabajar los datos directamente en memoria.

4.4.2. Muestreo de los Datos

Los tiempos de entrenamiento en los modelos SVM "escalan" de forma cúbica con la cantidad de registros que se evalúa (Tsang et al., 2005); es decir, a medida que la cantidad de registros aumenta, el tiempo de entrenamiento del modelo aumenta proporcionalmente al cubo del incremento en los datos. Dado que se tiene más de 2,6 millones de registros en total, se hace necesario tomar muestras más pequeñas para entrenar el modelo.

Sin embargo, dado que la proporción de la población que presenta el evento de interés es muy baja (alrededor de 0,6 %) no es conveniente tomar una muestra aleatoria simple de la base de datos para entrenar el modelo, dado que aún con una cantidad relativamente grande de datos se tendría muy poca información sobre este evento. Por ejemplo: si se toma una muestra aleatoria simple de 100,000 registros se esperaría tener apenas alrededor de 600 eventos.

Por lo tanto es necesario hacer un remuestreo que asegure que la proporción del evento en los datos de entrenamiento sea mayor respecto a la proporción real. Idealmente, esta proporción sería del 50 %; sin embargo, si consideramos esta proporción podríamos tener muy poca información sobre los clientes que no presentan el evento. Por ello, se remuestreó

sobre los datos de entrenamiento logrando que la proporción del evento de interés sea del 25 %. Esta proporción nos da suficiente información sobre el evento de interés, además de darnos más información sobre los clientes que no presentan el evento, y mantiene una desbalance menor entre los clientes que tienen el evento y los que no.

Así, se utilizó el siguiente esquema de muestreo:

- Se tomó una muestra aleatoria de 250,000 registros como muestra de validación.
- Se tomó una muestra aleatoria de 750,000 registros como muestra de prueba.
- Se tomó una muestra de 47,800 registros como muestra de entrenamiento de la siguiente forma:
  - 11,950 casos de clientes que presentan el evento de interés ($Y = +1$); es decir, el 25 % de la muestra.
  - 35,850 casos de clientes que no presentan el evento de interés ($Y = -1$); es decir, el 75 % restante de la muestra.

4.4.3. Implementaciones de SVM en el Software Estadístico R

Se evaluaron 3 librerías en R que implementan Support Vector Machines:

- Librería *e1071*: es la librería estándard de R para SVM. Provee una interfaz a la implementación en C++ de la librería *libsvm* (Meyer et al., 2014; Chang y Lin, 2011).
- Librería *kernlab*: provee métodos de aprendizaje estadístico basados en Kernels para R. Es más flexible que la librería *e1071* en el sentido que implementa una mayor variedad de kernels, e incluso permite utilizar kernels definidos por el usuario (Karatzoglou et al., 2004).
- Librería *rpud*: implementa una funcionalidad equivalente a la de la librería *e1071*, con la misma sintaxis y parámetros. Además promete acelerar significativamente los tiempos de entrenamiento y predicción al hacer uso del GPU (Graphical Processing Unit) para procesar los datos. Sin embargo, esta librería sólo funciona con tarjetas gráficas NVidia.

Debido a la gran cantidad de datos fue necesario escoger una implementación de Support Vector Machine que sea eficiente y rápida. Para ello se evaluaron los tiempos de entrenamiento de las tres librerías para un mismo conjunto de datos. Así, se entrenó el modelo SVM con 10 muestras aleatorias pequeñas ($10,000$ registros) tomadas de la muestra de entrenamiento.

Como resultado de este análisis se obtuvo los tiempos mostrados en el cuadro 4.1. Con lo cual se eligió la librería *rpud*, dado que sus tiempos de entrenamiento son mucho mejores que los del resto de librerías.

Además de la librería *rpud*, en el presente trabajo se utilizaron las siguientes librerías en R para distintas tareas:

Cuadro 4.1: Tiempos de entrenamiento del modelo SVM para 10 muestras usando librerías R

Librería	**Tiempo Total**	**Tiempo Promedio**	**Desv. Est.**
e1071	1324.53	132.45	17.94
kernlab	803.76	80.38	1.08
rpud	166.12	16.61	0.75

- La librería *kernlab* no sólo provee funciones para entrenar modelos SVM. Una de las funciones utilizadas en el presente desarrollo es la función *sigest*, la cual provee un método heurístico para encontrar un rango de valores adecuados para el parámetros gamma (γ) en los Kernel de Base Radial.
- La librería *caret* (Classification and Regresion Training) provee, entre otras, funciones para facilitar el pre procesamiento de los datos. Por ejemplo, para encontrar predictores con nula o poca varianza, o para encontrar grupos de variables correlacionadas.
- La librería *caTools* provee la función *colAUC* para calcular el área bajo la curva ROC de forma rápida y eficiente, incluso cuando se evalúa un volumen grande de datos.
- La librería *xtable* provee funciones para facilitar exportar datos y tablas en R a formato Latex.

4.5. Modelo SVM para Clasificación Binaria

A continuación se describen los pasos seguidos para entrenar el método de aprendizaje de Máquinas de Soporte Vectorial.

4.5.1. Pre-procesamiento de los datos

Los datos a evaluar tienen un total de 680 posibles variables predictoras. Naturalmente se busca reducir la cantidad de predictores para tener un modelo más robusto, eliminar redundancias (variables altamente correlacionadas), y en general tener un modelo más simple y eficiente. Para ello se aplicaron las siguientes técnicas al total de los datos:

- Se eliminaron 459 variables que tenían varianza cero y varianza "casi cero", mediante el uso de la función *nearZeroVar* de la librería *caret* (Kuhn et al., 2015). Es decir, se eliminó variables que en realidad son constantes y tienen en mismo valor para todos los casos, y variables con muy poca variabilidad y que podrían generar problemas en caso se incluyan en el modelo (por ejemplo: variables categóricas en las que el 99 % de los datos se concentren en un único valor). Se consideró que una variable tiene varianza casi cero cuando cumple con las siguientes dos características (Kuhn et al., 2015):
  - Tiene pocos valores únicos relativos al total de registros en los datos: el porcentaje de valores únicos dividido entre la cantidad de registros es menor a 10 %.
  - La frecuencia del valor más común respecto a la frecuencia del segundo valor más común es grande. Concretamente, si este ratio de frecuencias es mayor que 95 a 5.

El detalle de las variables resultantes que quedaron después de aplicar este paso se encuentra en el apéndice B.

- Se identificaron grupos de variables numéricas con correlaciones muy altas (mayores o igual a 0,75):
  - Se encontraron 30 variables correlacionadas en pares. Es decir, 15 casos en los que una variable está correlacionada únicamente con otra variable. En cada caso se descartó una de las variables.
  - Se aplicó componentes principales a 45 variables correlacionadas, obteniéndose 13 factores que reemplazaron a las variables originales.

  Finalmente, se consiguió reducir en 47 la cantidad de variables predictoras (60 variables descartadas y 13 factores creados por el análisis factorial). El detalle de las variables descartadas y de los factores creados se encuentra en el apéndice C.
- Se eliminaron 4 variables categóricas por tener más de 30 categorías: *STR_PROVINCIA*, *COD_DISTRITO*, *COD_ACTECONOMICA*, y *COD_PROFESION*.

En resumen, con la aplicación de estas técnicas el número de variables predictoras se redujo a 154.

4.5.2. Aplicación del Modelo SVM

Las Máquinas de Soporte Vectorial tienen parámetros que, según su configuración, pueden afectar el poder de predicción del modelo. Sin embargo, la mayoría de estos parámetros no tiene una forma automática de estimación; por ello es necesario entrenar los modelos SVM probando distintas configuraciones de estos parámetros, a fin de encontrar los que producen el mejor desempeño. Este proceso de búsqueda se conoce como *grid search*.

En la búsqueda del mejor modelo y usando la librería *rpud* los SVM fueron entrenados en 2 escenarios de prueba:

- Escenario 1: entrenar los modelos SVM y Logit usando los factores obtenidos en el análisis de componentes principales más el resto de las variables numéricas. En total, se utilizan 90 variables.
- Escenario 2: entrenar los modelos SVM y Logit usando los factores obtenidos en el análisis de componentes principales más el resto de las variables numéricas y las variables categóricas; es decir, todas las variables disponibles (154 variables en total). Dado que estos métodos de aprendizaje no aceptan variables categóricas, éstas se transformaron usando la técnicas de variables *dummy*.

El detalle de las variables utilizadas en cada escenario se encuentra en el apéndice D.

En cada escenario de prueba se entrenó un modelo Logit y tres modelos SVM, cada uno con una función Kernel diferente: Base Radial, Polinomial y Lineal. Para estos Kernel se usó

los hiperparámetros indicados en el cuadro 4.2. Cada Kernel además se entrenó probando 4 valores de la constante C que mide la tolerancia a los errores de clasificación: 1; 2,5; 5; 10.

Cuadro 4.2: Configuración de Hiperparámetros según el tipo de función Kernel en la construcción de los modelos SVM. En los Kernel de Base Radial, el parámetro γ se estima mediante una función llamada *sigest* de la librería *kernlab*. Por otro lado, en los Kernel Polinomial se toma el valor por defecto del parámetro γ, el cual es igual a: 1/(número de dimensiones). Finalmente, los Kernel Lineal no utilizan hiperparámetros.

Kernel	**gamma (γ)**	**grado (d)**	**delta (δ)**
Radial	Función *sigest*	-	-
Polinomial	Valor por defecto	3	0

Entre los parámetros usados para la librería *rpud* se aplicaron los pesos de clase, con la finalidad de convertir la constante C en dos valores C^+ y C^- que indican el costo de los errores de clasificación para cada clase. El parámetro de pesos de clase se configuró proporcional al desbalance de las clases (Batuwita y Palade, 2013); es decir, 0,25 para la clase $Y = -1$, y 0,75 para la clase $Y = +1$, permitiendo que la clase $Y = -1$ tenga un costo de error de clasificación de $(0{,}25)C$, y la clase $Y = +1$ un costo de $(0{,}75)C$.

Se evaluó el desempeño de predicción de los modelos SVM entrenados en cada escenario usando el indicador AUC (Área bajo la Curva ROC) en la muestra de validación, obtiendo los resultados mostrados en el cuadro 4.3. Se observa que el mejor desempeño de predicción se obtuvo en el Escenario 2, usando un Kernel Polinomial y valor del costo de error de clasificación $C = 5$.

El Kernel Polinomial produjo los mejores resultados medidos en la muestra de validación en el escenario 2, y en algunas configuraciones del escenario 1 (sin embargo, el desempeño de los otros Kernels no es muy diferente). Por otro lado, el valor óptimo de la constante C que indica el costo de error de clasificación depende de las variables usadas para entrenar el modelo. Así, el valor óptimo de $C = 5$ se obtiene cuando se utilizan todas las variables numéricas y categóricas transformadas mediante variables *dummy*; es decir, para el escenario 2. Por otro lado, cuando se entrena el modelo usando sólo 1las variables numéricas (escenario 1) un valor adecuado para esta constante es $C = 10$, dado que tuvo mejor desempeño de predicción en general.

Por otro lado, el modelo Logit con mejor desempeño de predicción se obtiene en el escenario 1, con un AUC de 0,7184. El escenario 2 no produjo un resultado para este modelo debido a que con el algoritmo de la función *glm* tuvo problemas de convergencia al utilizar todas las variables disponbiles (numéricas y categóricas).

Finalmente, se definió la regla de corte (*cutoff*) para el mejor modelo Logit y para el mejor modelo SVM usando los resultados de la muestra de validación.

Para el modelo Logit se buscó el punto de corte que optimiza el balance entre predicciones positivas y negativas; es decir, el punto donde se cruzan los valores de sensibilidad y especificidad. Para ello, se calculó los valores de sensitividad y especificidad para varios puntos en el intervalo $]0, 1[$, y se seleccionó el punto de intersección: 0,2406647.

Cuadro 4.3: Desempeño de predicción de los modelos SVM medida con el AUC (área bajo la curva ROC) en la muestra de validación para todas las configuraciones de parámetros probadas. Se evaluó el desempeño en 2 escenarios de variables (escenario 1 usando sólo las variables numéricas, y escenario 2 usando variables numéricas y categóricas), con 4 valores para la constante de costo C (1; 2,5; 5; 10) y 3 Kernel diferentes (Radial, Polinomial, y Lineal).

Escenario	**C**	**Kernel Radial**	**Kernel Polinomial**	**Kernel Lineal**
Escenario 1	1.0	0.7162	0.7118	0.7119
	2.5	0.7192	0.7170	0.7122
	5.0	0.7196	0.7192	0.7121
	10.0	0.7197	0.7203	0.7124
Escenario 2	1.0	0.7200	0.7178	0.7164
	2.5	0.7182	0.7221	0.7163
	5.0	0.7036	0.7238	0.7162
	10.0	0.6805	0.7209	0.7164

Por otro lado, para el modelo SVM se usó la regla estándard: la etiqueta de clase quedó definida por el signo de la función de clasificación: $f(x) = \sum_{SV} \alpha_i y_i K(x, x_i)$.

4.5.3. Análisis de resultados

El desempeño de predicción del modelo SVM, medido mediante el AUC (Área bajo la Curva ROC) es ligeramente mejor que el del modelo Logit tanto en la muestra de validación (donde se hizo la selección del modelo) como en la muestra de prueba (donde se hizo la medición final). La mejora en el AUC del modelo SVM es menor a 1 % respecto al modelo Logit, como se observa en el cuadro 4.4 y en la figura 4.1.

Cuadro 4.4: Medición de desempeño de predicción en las muestras de validación y prueba para los modelos SVM y Logit. Se observa que el modelo SVM tiene mejor desempeño de predicción, medido mediante el área bajo la curva ROC (AUC) en ambas muestras. Sin embargo, la diferencia no es muy grande: alrededor de 1 %.

Modelo	**AUC Validación**	**AUC Test**
SVM Polinomial	0,7238	0,7257
Logit	0,7184	0,7181

Usando las reglas de corte definidas en la sección anterior se obtuvo las matrices de confusión para el modelo Máquinas de Soporte Vectorial (4.5) y para el modelo de Regresión Logística (4.6), y el cuadro resumen de indicadores de desempeño de ambos modelos (4.7).

Cuadro 4.5: Matriz de confusión en los datos de prueba para el modelo SVM

		Real		
		0	**1**	**Total**
Predicción	**0**	327,772	975	328,747
	1	169,191	2062	171,253
	Total	496,963	3037	500,000

El modelo SVM tiene un valor de Sensitividad de 0,6790, mientras que el modelo Logit

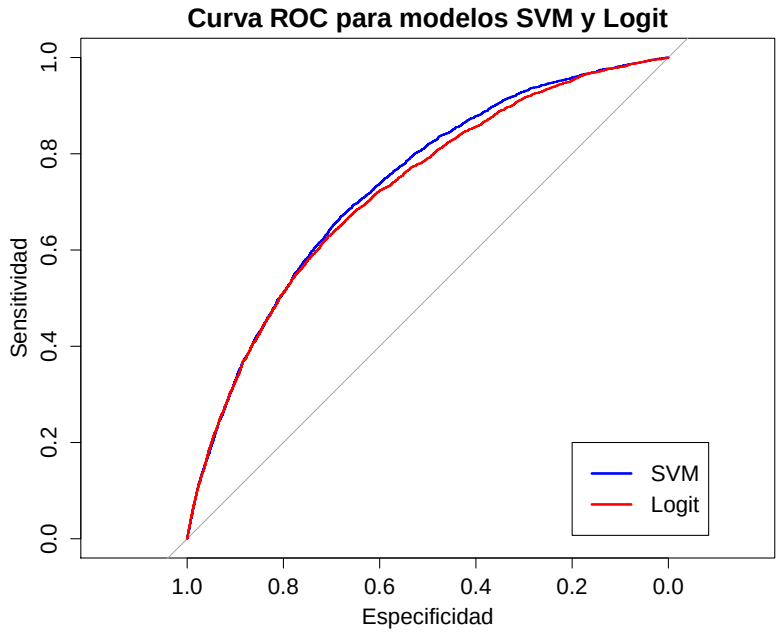


Figura 4.1: Curvas ROC para los modelos SVM Polinomial y Logit, medidas en la muestra de prueba.

Cuadro 4.6: Matriz de confusión en los datos de prueba para el modelo Logit

		Real		
		0	**1**	**Total**
Predicción	**0**	319,064	959	320,023
	1	177,899	2078	179,977
	Total	496,963	3037	500,000

Cuadro 4.7: Indicadores de desempeño de los modelos SVM y Logit medidos en los datos de prueba. Se observa que el modelo SVM tiene mejores valores de Especificidad, Precisión Balanceada y Valor Predicción Positiva. Por otro lado, el modelo Logit tiene mejores valores de Sensitividad. Ambos modelos están empatados en Valor Predicción Negativa. En resumen, el desempeño del modelo SVM es ligeramente superior al del modelo Logit.

Indicador	**SVM Polinomial**	**Logit**
Sensitividad	0,6790	0,6842
Especificidad	0,6596	0,6420
Precisión Balanceada	0,6691	0,6631
Valor Predicción Positiva	0,0120	0,0115
Valor Predicción Negativa	0,9970	0,9970

tiene un valor de 0,6842. Es decir, el modelo Logit tiene un valor de Sensitividad 0,78 % mayor que el modelo SVM. Esto quiere decir que el modelo Logit clasifica un 0,78 % más verdaderos positivos que el modelo SVM. Por otro lado, el modelo SVM tiene un valor de Especificidad de 0,6596, mientras que el modelo Logit tiene un valor de 0.6420. Es decir, el modelo SVM tiene un valor de Especificidad 2,66 % mayor que el modelo Logit. Esto significa que el modelo SVM clasifica un 2,66 % más verdaderos negativos.

El valor de Precisión Balanceada mide el promedio entre la Sensitividad y la Especificidad

para el modelo; es decir, es un indicador que toma en cuenta el balance entre ambas medidas. Se observa que el modelo SVM tiene un valor de Predicción Balanceada ligeramente mayor que el modelo Logit (casi 1 % de mejora).

El modelo SVM tiene un Valor de Predicción Positivo (es decir, el Número de Verdaderos Positivos entre el Número de Predicciones Positivas) de 0,0120, mientras que para el modelo Logit este valor es de 0,0115. Esto significa que un valor predicho como positivo por el modelo SVM presentará el evento de interés en el 1,20 % de los casos, mientras que en el modelo Logit lo hará en el 1,15 % de los casos. Es decir, el modelo SVM tiene un 4 % más de probabilidades de predecir un verdadero positivo. Este valor es muy bajo en ambos modelos, lo cual se nota también al observar las matrices de confusión (4.5 y 4.6): ambos modelos predicen una gran cantidad de registros como "1" (alrededor del 34 % de los datos para el modelo SVM, y 36 % para el modelo Logit), cuando la proporción real de "1" es mucho menor (alrededor del 0,6 % de los datos).

Dada la importancia de identificar correctamente a los verdaderos positivos se buscó medir el efecto que tienen los parámetros del modelo SVM en las predicciones posivitas. Así, se evaluaron los resultados de falsos positivos, verdaderos positivos, y el valor de predicción positivo para 4 diferentes valores de la constante de costo C en la muestra de prueba para los 3 tipos de kernel estudiados, según se muestra en el cuadro 4.8. Se observa que incrementar el valor de C mejora ligeramente el valor de predicción positiva debido a que reduce la cantidad de falsos positivos, aunque esto también tiene el efecto de reducir, en menor proporción, la cantidad de verdaderos positivos. Sin embargo, la mejora no es muy significativa. En el mejor caso, el de kernel Polinomial, sólo se logra una mejora de 1,7 % en el Valor de Predicción Positiva al cambiar el valor de C de 5 a 10. Pero esta mejora viene acompañada de una reducción de 0,5 % en el área bajo la curva ROC (AUC).

Cuadro 4.8: Resultados de falsos positivos, verdaderos positivos, total de predicciones positivas, y valor de predicción positiva en la muestra de prueba para los 3 tipos de kernel estudiados (Radial, Polinomial, y Lineal) según diferentes valores de la constante de costo C (1, 2,5, 5, y 10). En general, para valores mayores de C mejora el valor de predicción positiva debido a que disminuye la cantidad de falsos positivos, aunque también lo hace la cantidad de verdaderos positivos, pero en menor medida. Sin embargo, la mejora no es muy significativa y además se reduce el área bajo la curva ROC.

Kernel	C	**AUC**	**Falsos Positivos**	**Verdaderos Positivos**	**Predicciones Positivas**	**Valor Predicción Positiva**
Radial	1	0,7238	174,829	2071	176,900	0,0117
	2.5	0,7210	155,473	1948	157,421	0,0124
	5	0,7053	128,938	1620	130,558	0,0124
	10	0,6826	104,167	1301	105,468	0,0123
Polinomial	1	0,7217	178,963	2073	181,036	0,0115
	2.5	0,7255	173,734	2078	175,812	0,0118
	5	0,7257	169,191	2062	171,253	0,0120
	10	0,7218	161,646	2001	163,647	0,0122
Lineal	1	0,7219	165,372	2015	167,387	0,0120
	2.5	0,7217	165,066	2013	167,079	0,0120
	5	0,7216	165,081	2012	167,093	0,0120
	10	0,7217	164,995	2013	167,008	0,0121

Así, se comparó el resultado obtenido por el modelo SVM seleccionado, el cual se configuró usando pesos proporcionales al desbalance de las clases (0,25 para la clase Y=-1 y 0,75 para la clase Y=+1), versus no usar dicho parámetro (equivalente a tener un peso de 0,50 para cada clase), y versus usar otras configuraciones de pesos, tal como se indica en el cuadro 4.9. Se observa que el mejor resultado del valor de predicción positivo se presenta cuando no se selecciona el parámetro, dado que el número de falsos positivos baja considerablemente. Sin embargo, también baja el número de verdaderos positivos identificados por el modelo, y el desempeño global de éste (medido mediante el AUC). Con todo, si bien el valor de predicción positivo sube más de 80 %, sigue siendo bajo (alrededor de 2,25 %).

Cuadro 4.9: Resultados de falsos positivos, verdaderos positivos, total de predicciones positivas, y valor de predicción positiva en la muestra de prueba para el modelo SVM con mejor desempeño de predicción (kernel Polinomial con constante de costo $C = 5$).

Pesos -1;+1	**AUC**	**Falsos Positivos**	**Verdaderos Positivos**	**Predicciones Positivas**	**Valor Predicción Positiva**
25-75	0,7257	169,191	2062	171,253	0,0120
-	0,6900	20,081	463	20,544	0,0225
10-90	0,6895	403,317	2939	406,256	0,0072
40-60	0,7149	51,724	957	52,681	0,0182
75-25	0,6406	1	0	1	0,0000

Con esto, todo parece indicar que el bajo valor de predicción positivo podría originarse en el hecho que el evento de interés es muy poco frecuente en sí, con lo cual la información del evento que se dispone en el entrenamiento es escasa. También podría indicar que las variables de entrada no son suficientemente buenas para predecir el evento con mayor precisión. Estas ideas se refuerzan al comprobar que el alto número de falsos positivos también está presente en los resultados del modelo de regresión logística.

Por otro lado, el Valor de Predicción Negativa (es decir, el Número de Verdaderos Negativos entre el Número de Predicciones Negativas) es bastante alto en ambos modelos, y tienen valores muy similares alrededor de 0,9970. Esto quiere decir que, casi con total certeza, un valor predicho como negativo no presentará el evento de interés.

Finalmente, entrenar un modelo SVM es bastante más lento que entrenar un modelo de regresión logística. Además, generalmente no basta entrenar el modelo SVM una sola vez, sino es necesario probar distintas configuraciones de parámetros (tipo de kernel, hiperparámetros del kernel, constante de costo C) a fin de encontrar la opción más adecuada. Todo esto incrementa aún más el tiempo de entrenamiento del modelo SVM, y hace más complejo y difícil encontrar el modelo óptimo. El cuadro 4.10 muestra un resumen de los tiempos de entrenamiento de los modelos SVM y Logit.

En conclusión, el modelo SVM tiene un desempeño ligeramente mejor que el modelo Logit para los datos de la aplicación. Sin embargo, el modelo SVM fue más difícil y lento de entrenar.

Cuadro 4.10: Tiempos de entrenamiento para los modelos SVM y Logit. Se muestra el tiempo promedio que tomó entrenar un modelo SVM con 3 tipos de Kernel (Radial, Polinomial, y Lineal), y un modelo de regresión logística.

Modelo	**Tiempo Promedio**	**Desv. Est.**
SVM Kernel Radial	77.49	6.52
SVM Kernel Polinomial	76.40	6.17
SVM Kernel Lineal	436.45	357.27
Logit	2.47	0.43

Capítulo 5

Conclusiones

5.1. Conclusiones

A partir de los temas desarrollados en la presente tesis y a la aplicación realizada se puede concluir lo siguiente:

- El aprendizaje estadístico provee el marco teórico para encontrar funciones que permitan predecir una respuesta o evento a través de otras variables. Este marco teórico además nos ayuda a encontrar la mejor función para predecir una respuesta según el objetivo del problema, buscando un equilibro entre la complejidad de la función y su poder de predicción.
- Las Máquinas de Soporte Vectorial (SVM) son un método de aprendizaje estadístico para resolver problemas de clasificación binaria. Este método trabaja en una versión extendida (de mayor dimensión) del espacio de datos, utilizando funciones Kernel para mapear el producto interno como medida de similaridad en este espacio extendido sin tener que trabajar directamente en él.
- Las Máquinas de Soporte Vectorial (SVM) pueden presentar problemas al enfrentarse a problemas de clasificación binaria donde una de las clases es mucho más numerosa que la otra. Por ello se plantea una modificación a la formulación original de los SVM que asigna costos diferentes a los errores de clasificación en cada clase. Esto hace que los SVM sean robustos para clasificar datos desbalanceados, incluso si el desbalance en los datos es muy alto.
- En este trabajo se ha desarrollado un método de clasificación binaria para encontrar un modelo de propensión a la compra y persistencia del producto Seguro de Protección de Tarjetas para los clientes de una entidad financiera. Este modelo se entrenó usando el método de aprendizaje Máquinas de Soporte Vectorial (SVM) y se comparó con el modelo de Regresión Logística (Logit). Se observó que, en este conjunto de datos, el modelo entrenado con SVM tuvo mejor desempeño de predicción que el modelo entrenado con Logit. Sin embargo, la ganancia del modelo SVM respecto al modelo Logit fue pequeña (alrededor de 1 %).

- El método Máquinas de Soporte Vectorial (SVM) obtuvo mejores resultados que el método de Regresión Logística al ser entrenado con una cantidad grande de variables predictoras, sobre todo al incluir las variables categóricas. Así, los modelos entrenados con SVM no tuvieron problemas cuando se usaron todas las variables disponibles (numéricas y categóricas) y produjeron resultados óptimos. Por otro lado, cuando se intentó entrenar el modelo de Regresión Logística usando todas las variables disponibles (numéricas y categóricas) no se logró convergencia.

- Los modelos de Máquinas de Soporte Vectorial (SVM) son más difíciles de entrenar que los modelos de Regresión Logística. Esto se debe a que un modelo SVM debe ser entrenado con muchas configuraciones diferentes de parámetros a fin de encontrar la más óptima, dado que la formulación del problema no tiene una forma automática para encontrar la configuración óptima. Así, para un modelo se puede evaluar distintos tipos de función Kernel, los hiperparámetros que afinan la función Kernel, o el costo de los errores de clasificación.

- Los tiempos de entrenamiento de los modelos que utilizan Máquinas de Soporte Vectorial (SVM) son bastante lentos cuando se aplican sobre bases de datos grandes, sobre todo si se comparan con los tiempos de entrenamiento de los modelos de Regresión Logística. En la aplicación desarrollada, el entrenamiento del modelo SVM tomó 30 veces más que el modelo Logit. Esto se debe a que, según la formulación de los SVM, los tiempos de entrenamiento son proporcionales al cubo de la cantidad de datos.

- En la aplicación desarrollada en este trabajo de tesis, el modelo que tuvo un mejor desempeño general fue el modelo de regresión logística. Así, en este problema en particular, no se justifica usar un modelo más complejo y con tiempos de entrenamiento mucho más lentos, considerando que la ganancia en poder de predicción es pequeña. Por otro lado, esto no significa un veto general sobre el modelo SVM. Siempre es necesario conocer los métodos de aprendizaje estadístico a fin de estudiar cada problema de forma individual, y elegir el método de aprendizaje más adecuado.

5.2. Sugerencias para investigaciones futuras

Luego de revisar la teoría de Máquinas de Soporte Vectorial, se recomienda profundizar en:

- El estudio de Máquinas de Soporte Vectorial para resolver problemas de Clasificación Multinomial. Se propone estudiar los dos enfoques más conocidos: uno-versus-uno, y uno-versus-todo, comparando robustez, bondad de ajuste, y tiempos de entrenamiento.

- El estudio de Máquinas de Soporte Vectorial para resolver problemas de Regresión. Se propone estudiar el método de aprendizaje Regresión de Soporte Vectorial (SVR por sus siglas en inglés: Support Vector Regression) mediante la revisión de la literatura y una aplicación.

Adicionalmente, después de haber realizado este estudio se recomienda investigar los siguientes temas:

- El estudio de adecuaciones a las Máquinas de Soporte Vectorial para resolver problemas de clasificación y regresión con gran cantidad de datos de forma computacionalmente más eficiente, a fin de reducir los tiempos de entrenamieto de los modelos. Actualmente existen en la literatura algunas propuestas para lograr este objetivo, entre las cuales figuran: Core Vector Machine, un enfoque que encuentra subconjuntos de los datos (conjuntos núcleo o *core sets*) en los cuales se puede resolver el problema de optimización como una buena aproximación, en lugar de hacerlo en los datos completos (Tsang et al., 2005); Reduced Support Vector Machines (RSVM), el cual genera una superficie separadora usando sólo el 1 % de los datos (Yuh-jye y L. Mangasarian, 2001); y Fast Support Vector Machines, una aproximación que descompone el problema original en problemas más pequeños mediante árboles de decisión cuyos nodos consisten en SVM lineales (Fehr et al., 2007). Se propone investigar a fondo al menos alguna de estas propuestas u otra con propiedades similares, construir una implementación computacional en el *software* estadístico R, y comparar el desempeño de predicción y los tiempos de entrenamiento respecto a los modelos SVM tradicionales.

- El estudio de métodos para la estimación automática de parámetros usados en las Máquinas de Soporte Vectorial, como los hiperparámetros de los Kernel o la constante de costo C. Tradicionalmente, los SVM requieren que un modelo se entrene bajo distintas configuraciones de parámetros (*grid search*), lo cual incrementa el tiempo y la complejidad del proceso de entrenamiento. Luego, se propone investigar la literatura y estudiar al menos un algoritmo o metodología para automatizar la estimación de estos parámetros, construir una implementación computacional en el *software* estadístico R, y comparar el desempeño de predicción y los tiempos de entrenamiento versus el SVM usual. Un ejemplo que resalta en la literatura para este fin es el algoritmo colonia de hormigas (ACO por sus siglas en inglés) (Basim Alwan y Ruhana Ku-Mahamud, 2013), el cual consiste en simular el comportamiento de hormigas buscando comida: las hormigas caminan aleatoriamente y cuando encuentra comida, regresan a su colonia dejando un rastro de feromonas. Si otras hormigas encuentran ese rastro probablemente lo sigan, reforzándolo. Con el tiempo el rastro de feromonas se evapora, perdiendo su fuerza de atracción. Pero si el camino es corto y recorrido con frecuencia, tendrá una densidad de feromonas más grande que otros caminos más largos. Así, cuando una hormiga encuentra un buen camino entre la colonia y la comida, hay más probabilidades que otras hormigas lo sigan y, con retroalimentación positiva, todas las hormigan vayan por un solo camino. Las hormigas simuladas por al algoritmo ACO imitan este comportamiento caminando a través de un grafo que simula el problema en cuestión.

- El estudio de Métodos automáticos para la selección de variables predictoras en las Máquinas de Soporte Vectorial. En la formulación de los SVM no existe una forma directa para elegir qué variables incluir en el modelo y cuales descartar, tal que se obtenga un modelo sencillo sin sacrificar poder de predicción; este es un proceso que

debe realizarse de forma manual mediante prueba y error. Por ello, se propone estudiar al menos un algoritmo que automatice la tarea de selección de variables, por ejemplo: algoritmos genéticos, eliminación recursiva de variables, selección por filtros, algoritmos de recocido simulado (simulated annealing), etc.

- El estudio de la medición de cómo influyen las variables predictoras en la clasificación cuando se utiliza Máquinas de Soporte Vectorial. Los SVM no permiten medir o explicar esta influencia de forma nativa. Por lo tanto, se propone investigar métodos para medirla, por ejemplo, a través de estudios de simulación, o definir medidas de importancia para las variables predictoras.

Apéndice A

Listado de Símbolos

X	Conjunto de variables aleatorias predictoras. Indica los aspectos generales de las variables aleatorias predictoras.
Y	Variable aleatoria respuesta. Indica los aspectos generales de la variable aleatoria respuesta.
n	Número de observaciones en una muestra de datos.
p	Número de variables predictoras.
x	Matriz $n \times p$ de variables predictoras. Indica los valores observados de las variables predictoras. Según el contexto, también puede referirse a una observación de dichas variables predictoras.
y	Vector de variables respuesta. Indica los valores observados de la variable respuesta.
x_i	i-ésimo vector de la matriz x de variables predictoras. Cada x_i tiene tamaño p.
y_i	i-elemento del vector de variables respuesta y.
K	Función Kernel (función simétrica semidefinida positiva).
$u \cdot v$	Producto interno entre los vectores u y v.
w	Vector de pesos que define el hiperplano óptimo.
w_0	Solución al vector de pesos que define el hiperplano óptimo.
b	Constante de desplazamiento en el hiperplano óptimo.
b_0	Solución a la constante de desplazamiento en el hiperplano óptimo.
α_i, β_i	Multiplicadores de Lagrange en la formulación del hiperplano óptimo.
α, β	Vectores de los multiplicadores de Lagrange α_i y β_i, respectivamente, en la formulación del hiperplano óptimo.
α_i^0, β_i^0	Solución a los multiplicadores de Lagrange en la formulación del hiperplano óptimo.
α_0, β_0	Vectores de soluciones a los multiplicadores de Lagrange α_i^0 y β_i^0, respectivamente, en la formulación del hiperplano óptimo.
ξ_i	Variables de holgura en la formulación del método del hiperplano óptimo generalizado.
ξ	Vector de variables de holgura ξ_i en la formulación del método del hiperplano óptimo generalizado.

Apéndice B

Listado de Variables

Cuadro B.1: Variables Predictoras en la base de datos de clientes, excluyendo las variables

Variable	**Descripción**	**Tipo Dato**
CODCLAVECIC	Código de cliente	Numérica
CODMES	Año y mes (AAAAMM)	Numérica
STR_TIPOBANCA	Tipo de segmento al que pertenece	Categórica
STR_MACROZONALIMA	Macrozona de Lima	Categórica
STR_PROVINCIA	Provincia	Categórica
STR_ESTCIVIL	Estado Civil	Categórica
NUM_EDAD	Edad del cliente	Numérica
NUM_FEC_APERTURA	Antigüedad del cliente en años	Numérica
IND_SEXO	Sexo	Categórica
IND_NIVSOCIOECO	Indicador de nivel sociodemográfico calculado como el número del grupo A, B, C, D o E con mayor porcentaje de población en ese distrito.	Categórica
IMP_INGRESOEST_MED	Importe medio de los ingresos estimados durante los últimos 6 meses	Numérica
IMP_INGRESOESTFAM_MED	Importe medio de los ingresos familiares estimados durante los últimos 6 meses	Numérica
IND_SEGMENTOBEX	Indicador de segmento BEX	Categórica
IND_SEGMENTOPDH	indicador de segmento PdH	Categórica

Cuadro B.1

IMP_RENTABILIDAD_MED	Rentabilidad media del cliente durante los últimos 6 meses	Numérica
PCT_SHAREWALLET_MED	Share Wallet medio en los últimos 6 meses	Numérica
NUM_MES_ULTOPE_CAH	Número de meses desde la última operación en Ctas Ahorro	Numérica
IMP_OPEHAB_CVIS_MED	Media del número mensual de operaciones al haber en Ctas Vista en los últimos 6 meses	Numérica
IMP_OPEDEB_CVIS_MED	Media del número mensual de operaciones al debe en Ctas Vista en los últimos 6 meses	Numérica
NUM_OPEHAB_CVIS_MED	Media del número mensual de operaciones al haber en Ctas Vista en los últimos 6 meses	Numérica
NUM_OPEDEB_CVIS_MED	Media del número mensual de operaciones al debe en Ctas Vista en los últimos 6 meses	Numérica
NUM_MES_ULTOPE_CVIS	Número de meses desde la última operación en Ctas Vista	Numérica
IMP_MED_CAH_MED	Media del saldo medio mensual en Ctas Ahorro en los últimos 6 meses	Numérica
IMP_MED_CSU_MED	Media del saldo medio mensual en Ctas Sueldo en los últimos 6 meses	Numérica
IMP_MED_CVIS_MED	Media del saldo medio mensual en Ctas Vista en los últimos 6 meses	Numérica
IMP_MED_CVIS_MED_EVOL	Diferencia relativa (sobre la media del primer semestre) de medias semestrales de saldos medios mensuales en Ctas Vista	Numérica
IMP_MED_CVIS_TEND	Indicador de tendencia creciente/decreciente del saldo medio mensual en Ctas Vista en los últimos 6 meses (basado en un contraste sobre la pendiente de la recta de regresión)	Numérica
IND_CAH	Indicador de tenencia de Ctas Ahorro	Categórica
IND_CSU	Indicador de tenencia de Ctas Sueldo	Categórica
IND_CVIS	Indicador de tenencia de Ctas Vista	Categórica

Cuadro B.1

NUM_CVIS	Número de Ctas Vista vigentes	Numérica
NUM_CTACANC_CVIS	Número de cuentas canceladas en Ctas Vista	Numérica
IND_CANCREC_CVIS	Indicador de cancelación reciente en Ctas Vista	Categórica
IND_APEREC_CAH	Indicador de apertura reciente en Ctas Ahorro	Categórica
IND_APEREC_CSU	Indicador de apertura reciente en Ctas Sueldo	Categórica
IND_APEREC_CVIS	Indicador de apertura reciente en Ctas Vista	Categórica
NUM_MES_APEREC_CVIS_LN	Log(1+ Número de meses desde la cancelación más reciente en Ctas Vista)	Numérica
NUM_MES_APEANT_CVIS_LN	Log(1+ Número de meses desde la cancelación más reciente en Ctas Vista)	Numérica
IMP_SUELDO_CSU_MED	Media del importe del pago mensual de sueldo en los últimos 6 meses	Numérica
IMP_INTANG_CTS1	Importe intangible en CTS	Numérica
IMP_INTANG_SOL_CTS	Importe intangible en soles en CTS	Numérica
IMP_CTS	Importe en CTS	Numérica
IMP_SOL_CTS	Importe en soles en CTS	Numérica
IND_CTS	Indicador de tenencia de CTS	Categórica
NUM_MES_APEREC_CTS_LN	Numero de meses desde apertura más reciente de CTS	Numérica
NUM_MES_ULTREC_CTS_LN	Numero de meses desde la última cancelación de CTS	Numérica
IND_REC_APER_CTS	Indicador de reciente apertura de CTS	Categórica
IND_REC_CANC_CTS	Indicador de cancelación reciente de CTS	Categórica
NUM_CTS	Número de cuentas CTS vigentes	Numérica
NUM_CANCEL_CTS	Número de cuentas CTS canceladas	Numérica
IMP_CTS_MED	Importe medio en CTS	Numérica
IMP_CTS_MED_EVOL	Diferencia relativa (sobre la media del primer semestre) de medias semestrales de importes s en CTS	Numérica

Cuadro B.1

IMP_SOL_CTS_MED	Importe medio en soles en CTS	Numérica
IMP_SOL_CTS_MED_EVOL	Diferencia relativa (sobre la media del primer semestre) de medias semestrales de importes en soles en CTS	Numérica
IMP_INTANG_CTS_MED	Importe medio intangible en CTS	Numérica
IMP_INTANG_CTS_MED_EVOL	Diferencia relativa (sobre la media del primer semestre) de medias semestrales de importes intangibles en CTS	Numérica
IMP_INTANG_SOL_CTS_MED	Importe medio intangible en soles en CTS	Numérica
IMP_INTANG_SOL_CTS_MED_EVOL	Diferencia relativa (sobre la media del primer semestre) de medias semestrales de importes intangibles en soles en CTS	Numérica
PCT_IMP_INTANG_CTS_MED	Porcentaje medio de importe intangible en CTS en los últimos 6 meses	Numérica
PCT_IMP_SOL_CTS_MED	Porcentaje medio de importe en soles en CTS en los últimos 6 meses	Numérica
PCT_IMP_DOL_CTS_MED	Porcentaje medio de importe en dolares solarizados en CTS en los últimos 6 meses	Numérica
PCT_IMP_INTANG_SOL_CTS_MED	Porcentaje medio de importe intangible en soles en CTS en los últimos 6 meses	Numérica
PCT_IMP_INTANG_DOL_CTS_MED	Porcentaje medio de importe intangible en dolares solarizados en CTS en los últimos 6 meses	Numérica
IMP_ACTIVO_TOTAL_ACTUAL	Importe de la deuda actual en PPE e HIP	Numérica
IMP_ACTIVO_MED	Importe medio de deuda en PPE e HIP en los últimos 6 meses	Numérica
IMP_ACTIVO_MED_EVOL	Diferencia relativa (sobre la media del primer semestre) de medias semestrales de deudas en PPE e HIP	Numérica
PCT_ACTIVO_ACTUAL_MAX12	Porcentaje de deuda actual sobre el máximo de los últimos 12 meses	Numérica

Cuadro B.1

IMP_DEUDA_PPE_MED	Importe medio de deuda en PPE en los últimos 6 meses	Numérica
IND_PEFE	Indicador de tenencia de Préstamo Efectivo (2: tiene - 1: no tiene pero tuvo en los últimos 12 meses - 0: no ha tenido los últimos 12 meses)	Categórica
IND_PPE	Indicador de tenencia de PPE (2: tiene - 1: no tiene pero tuvo en los últimos 12 meses - 0: no ha tenido los últimos 12 meses)	Categórica
NUM_PPE	Número de contratos de PPE en vigor	Numérica
NUM_CTACANC_PPE	Número de contratos cancelados de PPE en la historia del cliente	Numérica
NUM_CTATOT_PPE	Número de contratos de PPE en la historia del cliente	Numérica
PCT_VISTA_PASIVO_TOTAL_ACTUAL	Porcentaje actual entre el importe en cta vista y el pasivo total (cts + cta vista + dap + fmu)	Numérica
IMP_PASIVO_ACTUAL	Importe pasivo en mes actual (cts + cta vista + dap + fmu)	Numérica
PCT_VISTA_PASIVO_TOTAL_MED	Porcentaje medio entre el importe en cta vista y el pasivo total (cts + cta vista + dap + fmu) en los últimos 6 meses	Numérica
IMP_PASIVO_MED	Importe medio pasivo en ultimos 6 meses (cts + cta vista + dap + fmu)	Numérica
IMP_VISTA_MED	Importe medio cta vista en ultimos 6 meses	Numérica
IMP_PASIVO_TEND	Tendencia del importe medio en pasivos durante los últimos 6 meses	Numérica
IMP_PASIVO_MED_EVOL	Diferencia relativa (sobre la media del primer semestre) de medias semestrales de importes en pasivo	Numérica
IND_TC	Indicador de tenencia de tarjetas de crédito en el último mes	Categórica
IND_TC_PREMIUM	Indicador de tenencia de tarjetas de crédito Premium en el último mes	Categórica

Cuadro B.1

NUM_TC	Número de plásticos de tarjetas de crédito en el último mes	Numérica
NUM_CANCEL_TC	Número de cancelaciones de plásticos de tarjetas de crédito en el último mes	Numérica
IMP_DEUDA_TC_MED	Importe medio de deuda en tarjetas de crédito en los últimos 6 meses	Numérica
IMP_DISPONIBLE_TC_MED	Importe medio del disponible en tarjetas de crédito en los últimos 6 meses	Numérica
IMP_DISPONIBLE_TC_MED_EVOL	Evolución de la media semestral del disponible en operaciones de tarjeta de crédito durante los últimos 12 meses	Numérica
PCT_IMP_OPE_SOL_TC_MED	Porcentaje del importe medio semestral realizado en soles respecto al importe total	Numérica
PCT_IMP_DISPUESTO_TC_MED	Porcentaje del importe dispuesto medio en tarjetas de crédito respecto del importe total	Numérica
IMP_OPE_TD	Importe de las operaciones en tarjeta de débito en el último mes	Numérica
NUM_OPE_TD	Número de operaciones en tarjetas de débito en el último mes	Numérica
IND_TD	Indicador de tenencia de tarjetas de débito en el último mes	Categórica
NUM_TD	Número de plásticos de tarjetas de débito en el último mes	Numérica
IMP_OPE_TD_MED	Importe medio de operaciones de tarjetas de débito en los últimos 6 meses	Numérica
IMP_OPE_TD_MED_EVOL	Evolución de la media semestral del importe de operacionesen tarjetas de débito durante los últimos 12 meses	Numérica
IMP_MED_OPE_TD_MED	Media semestral del importe medio mensual de las operaciones realizadas	Numérica
NUM_OPE_TD_MED	Número medio de operaciones de tarjetas de débito en los últimos 6 meses	Numérica

Cuadro B.1

NUM_OPE_TD_MED_EVOL	Evolución de la media semestral del número de operaciones en tarjetas de débito durante los últimos 12 meses	Numérica
NUM_MESES_OPERA_TD	Número de meses en los que se registran operaciones en tarjetas de débito a lo largo de los 6 últimos	Numérica
PCT_IMP_OPE_POS_TD_MED	Media semestral de los porcentajes de importes de operaciones en POS respecto al importe total mensual	Numérica
NUM_OPE_IN_TNS	Número de operaciones cobradas por transaccionalidad en el último mes	Numérica
IND_REC_USO_IN_TRA	Indicador de cobro en los últimos 6 meses de una transferencia	Categórica
IND_REC_USO_OUT_TRA	Indicador de pago en los 6 últimos meses de una transferencia	Categórica
NUM_MESES_OPERA_TRA	Número de meses en los que se registran operaciones en transferencias a lo largo de los 6 últimos	Numérica
IMP_OPE_IN_TNS_MED	Importe medio de los cobros por transaccionalidad en los últimos 6 meses	Numérica
IMP_OPE_IN_TNS_MED2	Importe medio de los cobros por transaccionalidad en el penúltimo semestre	Numérica
IMP_OPE_IN_TNS_MED_EVOL	Evolución de la media semestral del importe de los cobros en transaccionalidad durante los últimos 12 meses	Numérica
IMP_OPE_OUT_TNS_MED_EVOL	Evolución de la media semestral del importe de los pagos en transaccionalidad durante los últimos 12 meses	Numérica
NUM_OPE_IN_TNS_MED	Número medio de cobros por transaccionalidad en los últimos 6 meses	Numérica
IMP_OPE_MP	Importe de operaciones en el último mes de la ventana de análisis	Numérica

Cuadro B.1

NUM_OPE_MP	Número de operaciones en el último mes de la ventana de análisis	Numérica
IND_MP	Indicador de tenencia de algún medio de pago durante los últimos 12 meses de la ventana de análisis	Categórica
NUM_MP	Número de tarjetas vigentes en el último mes de la ventana de análisis	Numérica
IMP_OPE_MP_MED	Importe medio de operaciones en medios de pago durante los 6 últimos meses de la ventana de análisis	Numérica
IMP_OPE_MP_MED_EVOL	Evolución del Importe medio de operaciones en medios de pago durante el útlimo año de la ventana de análisis	Numérica
NUM_OPE_MP_MED	Número medio de operaciones en medios de pago en los últimos 6 meses de la ventana de análisis	Numérica
NUM_MESES_OPERA_MP	Número de meses en los que se registran operaciones en los últimos 6 meses de la ventana de análisis	Numérica
IND_SEG	Indicador de tenencia de seguros vida retorno, accidentes retorno y múltiple durante los últimos 12 meses de la ventana de análisis	Categórica
IMP_RCC_VIGENTE	Importe de la deuda vigente en RCC	Numérica
IMP_RCC_VIGENTE_MED	Importe medio de la deuda vigente en RCC durante los últimos 6 meses	Numérica
IMP_RCC_VIGENTE_MED_EVOL	Evolución de la media semestral del importe de la deuda media vigente durante los últimos 12 meses	Numérica
IMP_RCC_VIGENTE_TEND	Tendencia de la deuda media vigente en RCC durante los últimos 6 meses	Numérica
IMP_RCC_TC	Importe de la deuda vigente en tarjetas de crédito en RCC en el último mes	Numérica

Cuadro B.1

IMP_RCC_PTM	Importe de la deuda vigente en préstamos personales en RCC en el último mes	Numérica
IMP_MAX_12M_CREDITOS	Importe máximo de la deuda vigente en los 12 meses anteriores	Numérica
IMP_MAX_12M_CREDITOS_PRESTAMOS	Importe máximo de la deuda vigente en tarjetas de crédito y préstamos personales en los 12 meses anteriores	Numérica
PCT_RCC_EF_MED	Porcentaje medio de la deuda en la entidad financiera respecto del total de la deuda en RCC durante los últimos 6 meses	Numérica
PCT_RCC_BCO_MED	Porcentaje medio de la deuda en bancos principales respecto del total de la deuda en RCC durante los últimos 6 meses	Numérica
PCT_RCC_RET_MED	Porcentaje medio de la deuda en retailers respecto del total de la deuda en RCC durante los últimos 6 meses	Numérica
PCT_RCC_OTR_MED	Porcentaje medio de la deuda en otras entidades respecto del total de la deuda en RCC durante los últimos 6 meses	Numérica
RAT_RCC_4A_CONSUMO_VS_TOTAL	Porcentaje deuda histórica (¿4a) en TC y préstamos respecto al total	Numérica
IMP_LIM_CRED_TOTAL	Límite total de crédito en todo el sistema financiero	Numérica
NUM_OPE_GRUPO1_MED	Número medio de operaciones en construcción e inmobiliario en los 6 últimos meses	Numérica
NUM_OPE_GRUPO6_MED	Número medio de operaciones en servicios financieros en los 6 últimos meses	Numérica
IMP_OPE_GRUPO1_MED	Importe de operaciones en construcción e inmobiliario en los 6 últimos meses	Numérica
IND_PROGRAMA_PDH	Indicador de pertenencia a programas PdH en el último mes de la ventana de análisis	Categórica
IND_PROGRAMA_LANPASS	Indicador de pertenencia a progrmas LANPass en el último mes de la ventana de análisis	Categórica

Cuadro B.1

PCT_USO_CANAL_ATM	Porcentaje de uso del canal cajero (ATM) respecto al número total de operaciones en el último mes	Numérica
PCT_USO_CANAL_VENT	Porcentaje de uso del canal ventanilla respecto al número total de operaciones en el último mes	Numérica
RAT_pasivo_total_Med	Ratio medio del pasivo del cliente respecto a la media del microsegmento en los últimos 6 meses	Numérica
RAT_pasivo_total_Med_Evol	Evolución del ratio medio del pasivo del cliente respecto a la media del microsegmento en los últimos 6 meses	Numérica
RAT_pasivo_total_Med2	Ratio medio del pasivo del cliente respecto a la media del microsegmento en el penúltimo semestre	Numérica
RAT_ahorro_vista_Med	Ratio medio del ahorro del cliente respecto a la media del microsegmento en los últimos 6 meses	Numérica
RAT_ahorro_vista_Med_Evol	Evolución del ratio medio del ahorro del cliente respecto a la media del microsegmento en los últimos 6 meses	Numérica
RAT_ahorro_vista_Med2	Ratio medio del ahorro del cliente respecto a la media del microsegmento en el penúltimo semestre	Numérica
RAT_fmudap_Med	Ratio medio del ahorro a plazo - fondos mutuos del cliente respecto a la media del microsegmento en los últimos 6 meses	Numérica
RAT_fmudap_Med_Evol	Evolución del ratio medio del ahorro a plazo - fondos mutuos del cliente respecto a la media del microsegmento en los últimos 6 meses	Numérica
RAT_fmudap_Med2	Ratio medio del ahorro a plazo - fondos mutuos del cliente respecto a la media del microsegmento en el penúltimo semestre	Numérica

Cuadro B.1

RAT_activo_total_Med	Ratio medio del activo del cliente respecto a la media del microsegmento en los últimos 6 meses	Numérica
RAT_activo_total_Med_Evol	Evolución del ratio medio del activo del cliente respecto a la media del microsegmento en los últimos 6 meses	Numérica
RAT_activo_total_Med2	Ratio medio del activo del cliente respecto a la media del microsegmento en el penúltimo semestre	Numérica
IMP_INGREG_01	Importe medio de ingresos en el último semestre de la ventana de análisis	Numérica
IMP_INGREG_02	Importe medio de ingresos en el penúltimo semestre de la ventana de análisis	Numérica
IMP_GASTREG_01	Importe medio de gastos en el último semestre de la ventana de análisis	Numérica
IMP_GASTREG_02	Importe medio de gastos en el penúltimo semestre de la ventana de análisis	Numérica
IMP_INGREG_CV	Coeficiente de variación de los ingresos regulares en el último año de la ventana de análisis	Numérica
IMP_GASTREG_CV	Coeficiente de variación de los gastos regulares en el último año de la ventana de análisis	Numérica
IND_INGREG	Indicador de ingresos regulares	Categórica
IND_GASTREG	Indicador de gastos regulares	Categórica
NUM_CAMP_ENVIADAS	Numero de campañas distintas enviadas recibidas en los ultimos 6 meses	Numérica
NUM_CAMP_CONTACT	Numero de campañas con contacto efectivo recibidas en los ultimos 6 meses	Numérica
NUM_CAMP_EXITO	Numero de campañas aceptadas (respuestas positivas) en los ultimos 12 meses	Numérica
NUM_CAMP_EXITO_INC_TASAS	Numero de campañas aceptadas con incentivo tasas en los ultimos 12 meses	Numérica

Cuadro B.1

IND_CAMP_EXITO_CANAL_AGEN_REA	Indicaddor de alguna campaña aceptada por el canal Agente/sucursal REACTIVO en los últimos 12 meses	Categórica
NUM_CAMP_SOLI_INC_TASAS	Numero de campañas solicitadas con incentivo tasas en los ultimos 12 meses	Numérica
IND_CAMP_SOLI_CANAL_AGEN_REA	Indicaddor de alguna campaña solicitadas por el canal Agente/sucursal REACTIVO en los últimos 12 meses	Categórica
IMP_RCC_TC_DISPEFECT	Suma del importe de la deuda vigente en tarjetas de crédito por disponibilidad de efectivo en RCC en los últimos 6 meses.	Numérica
IMP_RCC_TC_COMPRA	Suma del importe de la deuda vigente en tarjetas de crédito por compra en RCC en los últimos 6 meses.	Numérica
IND_REVOLVING_6M	Indicador de si el cliente ha tenido revolving en los 6 últimos meses	Categórica
IMP_REVOLVING_MED	Media del importe de revolving en los 6 últimos meses	Numérica
IND_EDAD_MED	Media ponderada de los intervalos de edad por manzana	Categórica
IND_NSE_MED	Media ponderada del nivel socioeconómico de la manzana	Categórica
GRP_TIPO_VIVIENDA_MAX	Máximo - Tipo de vivienda por manzana	Categórica
GRP_VIVIENDA_MAX	Máximo - Tenencia de vivienda por manzana	Categórica
GRP_PARED_MAX	Máximo - contrucción predominante de las paredes (pared1,, pared8)	Categórica
GRP_PISO_MAX	Máximo - Contrucción predominante del piso (piso_1, , piso_7)	Categórica
GRP_TIPO_ABASTECIMIENTO_MAX	Máximo - Abastecimiento de agua (red, otros)	Categórica
PCT_AGUA_DIARIO	Porcentaje de servicio de agua diario	Numérica

Cuadro B.1

GRP_SERV_HIGIENICOS_MAX	Máximo de los 3 tipos de servicios higienicos (pozo, red, no tiene)	Categórica
PCT_ALUMBRADO	Porcentaje de alumbrado por manzana	Numérica
GRP_ENERGIA_MAX	Máximo de la energía utilizada por manzana (sin agrupar)	Categórica
GRP_LENGUA_MAX	Máximo de las lenguas utilizadas por manzana (nativo, castellano, extranjero)	Categórica
IND_NIVEL_ESTUDIOS_MED	Media ponderada del nivel de estudios de la manzana	Categórica
GRP_ESTADO_CIVIL_MAX	Máximo del estado civil por manzana (pareja, expareja, solteros)	Categórica
IND_DENSI_FARM_DECIL	Densidad de farmacias por manzanas toma valores entre 0 y 10	Categórica
IND_DENSI_REST_DECIL	Densidad de restaurantes por manzanas toma valores entre 0 y 10	Categórica
SEGM	Número de segmento en que se ha clasificado al cliente por renta - edad	Categórica
IND_PAFG_CAH	Indicador de población de análisis del modelo de Fuga de Cuenta Ahorro	Categórica
IND_PAXS_CSU	Indicador de población de análisis del modelo de Cross-selling de Cuenta Sueldo	Categórica
IND_PAFG_CSU	Indicador de población de análisis del modelo de Fuga de Cuenta Sueldo	Categórica
IND_PAXS_CTS	Indicador de población de análisis del modelo de Cross-selling de CTS	Categórica
IND_PAFG_CTS	Indicador de población de análisis del modelo de Fuga de Cuenta CTS	Categórica
IND_PAXS_HIP_AyB	Indicador de población de análisis del modelo de Cross-selling de Hipotecas del nivel socioeconómica A y B	Categórica

Cuadro B.1

IND_PAXS_HIP_CyD	Indicador de población de análisis del modelo de Cross-selling de Hipotecas del nivel socioeconómica C y D	Categórica
IND_PAXS_PPE	Indicador de población de análisis del modelo de Cross-selling de Préstamos Personales	Categórica
IND_PAUP_PPE	Indicador de población de análisis del modelo de Upselling de Préstamos personales	Categórica
IND_PAFG_PPE	Indicador de población de análisis del modelo de Fuga de Préstamos Personales	Categórica
IND_PAXS_TC	Indicador de población de análisis del modelo de Cross-selling de Tarjeta de Crédito	Categórica

Apéndice C

Análisis de Correlaciones y Componentes Principales

Cuadro C.1: Pares de variables con correlaciones muy altas ($R^2 \geq 0{,}75$). Para reducir la cantidad de variables predictoras del modelo, se decidió trabajar únicamente con una de las variables del par (columna "Variable Seleccionada") y descartar la otra.

Variable Seleccionada	**Variable Descartada**	R^2
PCT_SHAREWALLET_MED	PCT_RCC_BCP_MED	0,916 404
NUM_PPE	PCT_ACTIVO_ACTUAL_MAX12	0,826 111
NUM_MES_APEREC_CVIS_LN	NUM_MES_APEANT_CVIS_LN	0,763 652
IMP_PASIVO_MED	IMP_PASIVO_ACTUAL	0,971 360
NUM_OPE_IN_TNS_MED	NUM_OPE_IN_TNS	0,871 715
IMP_INGREG_02	IMP_GASTREG_02	0,988 036
PCT_AGUA_DIARIO	PCT_ALUMBRADO	0,967 517
NUM_TD	NUM_MP	0,989 723
NUM_CAMP_SOLI_INC_TASAS	NUM_CAMP_EXITO_INC_TASAS	0,957 359
NUM_OPEHAB_CVIS_MED	NUM_OPEDEB_CVIS_MED	0,879 387
PCT_IMP_DOL_CTS_MED	PCT_IMP_INTANG_DOL_CTS_MED	0,997 646
RAT_ahorro_vista_Med	RAT_ahorro_vista_Med2	0,767 051
IMP_CTS_MED_EVOL	IMP_INTANG_CTS_MED_EVOL	0,999 961
IMP_SOL_CTS_MED_EVOL	IMP_INTANG_SOL_CTS_MED_EVOL	0,999 996
IMP_MED_CVIS_MED_EVOL	RAT_ahorro_vista_Med_Evol	0,955 160

Cuadro C.2: Factores resultantes del análisis de componentes principales entre las variables predictoras. Para reducir la cantidad de variables predictoras del modelo, se aplicó el anàlisis de componentes principales a los grupos de variables (más de 2 miembros) con altas correlaciones entre sí ($R^2 \geq 0{,}75$). Luego, estos grupos fueron descartados del modelo y reemplazados por los factores indicados.

Factor	**Descripción**	**% Var Explicada**
FR_OPE_TD_MP	Factor resultado de análisis factorial de variables: IMP_OPE_TD_MED, IMP_OPE_TD, IMP_OPE_MP	0,840
FR_NUM_OPE_TD_MP	Factor resultado de análisis factorial de variables: NUM_OPE_TD_MED, NUM_OPE_TD, NUM_OPE_MP	0,904
FR_IMP_CTS_1 FR_IMP_CTS_2	Factores resultado de análisis factorial de variables: IMP_INTANG_CTS_MED, IMP_INTANG_CTS1, IMP_CTS_MED, IMP_INTANG_SOL_CTS_MED, IMP_INTANG_SOL_CTS, IMP_SOL_CTS_MED, IMP_CTS, IMP_SOL_CTS	0,978
FR_IMP_RCC	Factor resultado de análisis factorial de variables: IMP_MAX_12M_CREDITOS, IMP_RCC_VIGENTE, IMP_RCC_VIGENTE_MED	0,985
FR_IMP_VISTA_CAH	Factor resultado de análisis factorial de variables: IMP_VISTA_MED, IMP_MED_CAH_MED, IMP_MED_CVIS_MED	0,864
FR_NUM_PPE	Factor resultado de análisis factorial de variables: NUM_CTATOT_PPE, NUM_PPE, NUM_CTACANC_PPE	0,773

FR_ACTIVO	Factor resultado de análisis factorial de variables: IMP_ACTIVO_MED, IMP_ACTIVO_TOTAL_ACTUAL, RAT_activo_total_Med, RAT_activo_total_Med2	0,814
FR_CVIS_INGGAST	Factor resultado de análisis factorial de variables: IMP_OPEHAB_CVIS_MED, IMP_OPEDEB_CVIS_MED, IMP_INGREG_01, IMP_GASTREG_01	0,903
FR_PCT_CTS	Factor resultado de análisis factorial de variables: PCT_IMP_INTANG_CTS_MED, NUM_CTS, PCT_IMP_SOL_CTS_MED, PCT_IMP_INTANG_SOL_CTS_MED	0,748
FR_RAT_PASIVO_FMU	Factor resultado de análisis factorial de variables: RAT_pasivo_total_Med2, RAT_fmudap_Med2, RAT_fmudap_Med, RAT_pasivo_total_Med	0,808
FR_RAT_PASIVO_FMU_EVOL	Factor resultado de análisis factorial de variables: RAT_pasivo_total_Med_Evol, IMP_PASIVO_MED_EVOL, RAT_fmudap_Med_Evol	0,997

Apéndice D

Listado de Variables Finales por Escenario

Cuadro D.1: Variables finales por Escenario para los modelos de clasificación

Variable	**Escenario 1**	**Escenario 2**
NUM_EDAD	S	S
NUM_FEC_APERTURA	S	S
IMP_INGRESOEST_MED	S	S
IMP_INGRESOESTFAM_MED	S	S
IMP_RENTABILIDAD_MED	S	S
PCT_SHAREWALLET_MED	S	S
NUM_MES_ULTOPE_CAH	S	S
NUM_OPEHAB_CVIS_MED	S	S
NUM_MES_ULTOPE_CVIS	S	S
IMP_MED_CSU_MED	S	S
IMP_MED_CVIS_MED_EVOL	S	S
IMP_MED_CVIS_TEND	S	S
NUM_CVIS	S	S
NUM_CTACANC_CVIS	S	S
NUM_MES_APEREC_CVIS_LN	S	S
IMP_SUELDO_CSU_MED	S	S
NUM_MES_APEREC_CTS_LN	S	S
NUM_MES_ULTREC_CTS_LN	S	S
NUM_CANCEL_CTS	S	S
IMP_CTS_MED_EVOL	S	S
IMP_SOL_CTS_MED_EVOL	S	S
PCT_IMP_DOL_CTS_MED	S	S
IMP_ACTIVO_MED_EVOL	S	S
IMP_DEUDA_PPE_MED	S	S
PCT_VISTA_PASIVO_TOTAL_ACTUAL	S	S
PCT_VISTA_PASIVO_TOTAL_MED	S	S
IMP_PASIVO_MED	S	S
IMP_PASIVO_TEND	S	S

Cuadro D.1

NUM_TC	S	S
NUM_CANCEL_TC	S	S
IMP_DISPONIBLE_TC_MED	S	S
IMP_DISPONIBLE_TC_MED_EVOL	S	S
PCT_IMP_OPE_SOL_TC_MED	S	S
PCT_IMP_DISPUESTO_TC_MED	S	S
NUM_TD	S	S
IMP_OPE_TD_MED_EVOL	S	S
IMP_MED_OPE_TD_MED	S	S
NUM_OPE_TD_MED_EVOL	S	S
NUM_MESES_OPERA_TD	S	S
PCT_IMP_OPE_POS_TD_MED	S	S
NUM_MESES_OPERA_TRA	S	S
IMP_OPE_IN_TNS_MED	S	S
IMP_OPE_IN_TNS_MED2	S	S
IMP_OPE_IN_TNS_MED_EVOL	S	S
IMP_OPE_OUT_TNS_MED_EVOL	S	S
NUM_OPE_IN_TNS_MED	S	S
IMP_OPE_MP_MED	S	S
IMP_OPE_MP_MED_EVOL	S	S
NUM_OPE_MP_MED	S	S
NUM_MESES_OPERA_MP	S	S
IMP_RCC_VIGENTE_MED_EVOL	S	S
IMP_RCC_VIGENTE_TEND	S	S
IMP_RCC_PTM	S	S
IMP_MAX_12M_CREDITOS_PRESTAMOS	S	S
PCT_RCC_BCO_MED	S	S
PCT_RCC_RET_MED	S	S
PCT_RCC_OTR_MED	S	S
RAT_RCC_4A_CONSUMO_VS_TOTAL	S	S
IMP_LIM_CRED_TOTAL	S	S
NUM_OPE_GRUPO1_MED	S	S
NUM_OPE_GRUPO6_MED	S	S
IMP_OPE_GRUPO1_MED	S	S
PCT_USO_CANAL_ATM	S	S
PCT_USO_CANAL_VENT	S	S
RAT_ahorro_vista_Med	S	S
RAT_activo_total_Med_Evol	S	S
IMP_INGREG_02	S	S
IMP_INGREG_CV	S	S
IMP_GASTREG_CV	S	S

Cuadro D.1

NUM_CAMP_ENVIADAS	S	S
NUM_CAMP_CONTACT	S	S
NUM_CAMP_EXITO	S	S
NUM_CAMP_SOLI_INC_TASAS	S	S
IMP_RCC_TC_DISPEFECT	S	S
IMP_REVOLVING_MED	S	S
PCT_AGUA_DIARIO	S	S
NUM_PRODUCTOS	S	S
STR_TIPOBANCA		S
STR_MACROZONALIMA		S
STR_ESTCIVIL		S
IND_SEXO		S
IND_NIVSOCIOECO		S
IND_SEGMENTOBEX		S
IND_SEGMENTOPDH		S
IND_CAH		S
IND_CSU		S
IND_CVIS		S
IND_CANCREC_CVIS		S
IND_APEREC_CAH		S
IND_APEREC_CSU		S
IND_APEREC_CVIS		S
IND_CTS		S
IND_REC_APER_CTS		S
IND_REC_CANC_CTS		S
IND_PEFE		S
IND_PPE		S
IND_TC		S
IND_TC_PREMIUM		S
IND_TD		S
IND_REC_USO_IN_TRA		S
IND_REC_USO_OUT_TRA		S
IND_MP		S
IND_SEG		S
IND_PROGRAMA_PDH		S
IND_PROGRAMA_LANPASS		S
IND_INGREG		S
IND_GASTREG		S
IND_CAMP_EXITO_CANAL_AGEN_REA		S
IND_CAMP_SOLI_CANAL_AGEN_REA		S
IND_REVOLVING_6M		S

Cuadro D.1

IND_EDAD_MED		S
IND_NSE_MED		S
GRP_TIPO_VIVIENDA_MAX		S
GRP_VIVIENDA_MAX		S
GRP_PARED_MAX		S
GRP_PISO_MAX		S
GRP_TIPO_ABASTECIMIENTO_MAX		S
GRP_SERV_HIGIENICOS_MAX		S
GRP_ENERGIA_MAX		S
GRP_LENGUA_MAX		S
IND_NIVEL_ESTUDIOS_MED		S
GRP_ESTADO_CIVIL_MAX		S
IND_DENSI_FARM_DECIL		S
IND_DENSI_REST_DECIL		S
SEGM		S
IND_PAFG_CAH		S
IND_PAXS_CSU		S
IND_PAFG_CSU		S
IND_PAXS_CTS		S
IND_PAFG_CTS		S
IND_PAXS_HIP_AyB		S
IND_PAXS_HIP_CyD		S
IND_PAXS_PPE		S
IND_PAUP_PPE		S
IND_PAFG_PPE		S
IND_PAXS_TC		S
IND_PAXS_BT		S
IND_PAXS_EP		S
IND_PAUP_REV		S
COD_DEPARTAMENTO		S
RCC_TIPCLASIFRIESGO		S
FR_OPE_TD_MP	S	S
FR_NUM_OPE_TD_MP	S	S
FR_IMP_CTS_1	S	S
FR_IMP_CTS_2	S	S
FR_IMP_RCC_TC	S	S
FR_IMP_RCC	S	S
FR_IMP_VISTA_CAH	S	S
FR_NUM_PPE	S	S
FR_ACTIVO	S	S
FR_CVIS_INGGAST	S	S

Cuadro D.1

FR_PCT_CTS	S	S
FR_RAT_PASIVO_FMU	S	S
FR_RAT_PASIVO_FMU_EVOL	S	S

Apéndice E

Código Fuente en R

```
#Lectura de archivo con variables predictoras, y archivo con variable respuesta. Se
    hace el match entre ambos usando el campo CODCLAVECIC (identificador del cliente)
setwd("C://Users//Sergio//Documents//Cursos//PUCP - Maestría Estadística//2014.2 -
    Ciclo 4//1 - Investigación en Estadística//Aplicación//Datos//Versión 1")
library(data.table)

pt = proc.time()
seguros.nba = fread("nba_tablon_201403.csv", header=T, sep=",", stringsAsFactors=T)
proc.time() - pt
setkey(seguros.nba, CODCLAVECIC)
head(seguros.nba)

pt = proc.time()
seguros.nba.Y = fread("seg_tablon_15.csv", header=T, sep=",", stringsAsFactors=T)
proc.time() - pt
setkey(seguros.nba.Y, CODCLAVECIC)
head(seguros.nba.Y)

nrow(seguros.nba)
nrow(seguros.nba.Y)

memory.limit(32000)
pt = proc.time()
seguros.nba = merge(x=seguros.nba, y=seguros.nba.Y)
proc.time() - pt
gc()
head(seguros.nba)

table(seguros.nba$SB_FLGVTA)
table(seguros.nba$SB_FLGVTA_PERSISTENCIA6M)

pt = proc.time()
write.table(seguros.nba, "SegurosNBA_1.csv", sep = ",", col.names=T, row.names=F)
proc.time() - pt
```

: ./scripts/01_Union_X_Y.R

```
#Evaluar predictores con Varianza 0, o Varianza casi 0
setwd("C://Users//Sergio//Documents//Cursos//PUCP - Maestría Estadística//2014.2 -
    Ciclo 4//1 - Investigación en Estadística//Aplicación//Datos//Versión 3")
library(data.table)
library(caret)

memory.limit(40000)
pt = proc.time()
```

```
seguros.nba = fread("SegurosNBA_1.csv", header=T, sep=",", stringsAsFactors=T)
proc.time() - pt

columnasTotal = colnames(seguros.nba)
columnasNum = c("IMP_INGRESOEST","NUM_EDAD","NUM_FEC_APERTURA","IMP_INGRESOEST_MED","
    IMP_INGRESOESTFAM_MED","IMP_RENTABILIDAD_MED","NUM_MAXBEHAVIORSCORE","PCT_
    SHAREWALLET_MED","NUM_MES_ULTOPE_CCT","NUM_MESVCTO_LSOBREG_CCT","NUM_MES_ULTOPE_
    CAH","NUM_MES_ULTOPE_CSU","NUM_MES_ULTSUELDO_CSU","IMP_MED_LSOBREG_CCT","IMP_
    OPEHAB_CVIS_MED","IMP_OPEDEB_CVIS_MED","NUM_OPEHAB_CVIS_MED","NUM_OPEDEB_CVIS_MED"
    ,"NUM_MES_ULTOPE_CVIS","IMP_MED_CCT_MED",
                "IMP_MED_CAH_MED","IMP_MED_CSU_MED","IMP_MED_CVIS_MED","IMP_MED_CVIS_
                    MED_EVOL","IMP_MED_CVIS_TEND","PCT_IMP_MED_CVIS_DOL_MED","NUM_CVIS
                    ","NUM_CTACANC_CVIS","NUM_MES_APEREC_CCT_LN","NUM_MES_APEREC_CAH_
                    LN","NUM_MES_APEREC_CSU_LN","NUM_MES_APEREC_CVIS_LN","NUM_MES_
                    APEANT_CCT_LN","NUM_MES_APEANT_CAH_LN","NUM_MES_APEANT_CSU_LN","
                    NUM_MES_APEANT_CVIS_LN","NUM_MES_CANCREC_CCT_LN","NUM_MES_CANCREC_
                    CAH_LN","NUM_MES_CANCREC_CSU_LN","NUM_MES_CANCREC_CVIS_LN",
                "IMP_MED_LSOBREG_CCT_MAX","IMP_LIM_LSOBREG_CCT_MED","NUM_DIAS_SOBREG_
                    CCT_MAX","NUM_MES_SOBREGREC_CCT","IMP_SUELDO_CSU_MED","IMP_TANG_
                    CTS","IMP_TANG_SOL_CTS","IMP_TANG_DOL_CTS","IMP_INTANG_CTS1","IMP_
                    INTANG_SOL_CTS","IMP_INTANG_DOL_CTS","IMP_CTS","IMP_SOL_CTS","IMP_
                    DOL_CTS","NUM_MES_APEANT_CTS_LN","NUM_MES_APEREC_CTS_LN","NUM_MES_
                    CIEREC_CTS_LN","NUM_MES_ULTREC_CTS_LN","NUM_CTS","NUM_CANCEL_CTS",
                "IMP_CTS_MED","IMP_CTS_MED_EVOL","IMP_CTS_TEND","IMP_SOL_CTS_MED","IMP
                    _SOL_CTS_MED_EVOL","IMP_SOL_CTS_TEND","IMP_DOL_CTS_MED","IMP_DOL_
                    CTS_MED_EVOL","IMP_DOL_CTS_TEND","IMP_TANG_CTS_MED","IMP_TANG_CTS_
                    MED_EVOL","IMP_TANG_CTS_TEND","IMP_TANG_SOL_CTS_MED","IMP_TANG_SOL
                    _CTS_MED_EVOL","IMP_TANG_SOL_CTS_TEND","IMP_TANG_DOL_CTS_MED","IMP
                    _TANG_DOL_CTS_MED_EVOL","IMP_TANG_DOL_CTS_TEND","IMP_INTANG_CTS_
                    MED","IMP_INTANG_CTS_MED_EVOL",
                "IMP_INTANG_CTS_TEND","IMP_INTANG_SOL_CTS_MED","IMP_INTANG_SOL_CTS_MED
                    _EVOL","IMP_INTANG_SOL_CTS_TEND","IMP_INTANG_DOL_CTS_MED","IMP_
                    INTANG_DOL_CTS_MED_EVOL","IMP_INTANG_DOL_CTS_TEND","PCT_IMP_TANG_
                    CTS_MED","PCT_IMP_INTANG_CTS_MED","PCT_IMP_SOL_CTS_MED","PCT_IMP_
                    DOL_CTS_MED","PCT_IMP_TANG_SOL_CTS_MED","PCT_IMP_TANG_DOL_CTS_MED"
                    ,"PCT_IMP_INTANG_SOL_CTS_MED","PCT_IMP_INTANG_DOL_CTS_MED","
                    CTDMESAPERTURAANTIGUADPLZ","CTDMESAPERTURARECIENTEDPLZ","
                    CTDMESCIERRERECIENTEDPLZ","IMP_DAP","IMP_SOL_DAP",
                "IMP_DOL_DAP","IMP_NORED_DAP","IMP_NORED_SOL_DAP","IMP_NORED_DOL_DAP",
                    "NUM_MES_APEANT_DAP_LN","NUM_MES_APEREC_DAP_LN","NUM_MES_CIEREC_
                    DAP_LN","NUM_DAP","NUM_CANCEL_DAP","IMP_DAP_MED","IMP_DAP_MED_EVOL
                    ","IMP_DAP_TEND","IMP_SOL_DAP_MED","IMP_SOL_DAP_MED_EVOL","IMP_SOL
                    _DAP_TEND","IMP_DOL_DAP_MED","IMP_DOL_DAP_MED_EVOL","IMP_DOL_DAP_
                    TEND","IMP_NORED_DAP_MED","IMP_NORED_DAP_MED_EVOL",
                "IMP_NORED_DAP_TEND","IMP_NORED_SOL_DAP_MED","IMP_NORED_SOL_DAP_MED_
                    EVOL","IMP_NORED_SOL_DAP_TEND","IMP_NORED_DOL_DAP_MED","IMP_NORED_
                    DOL_DAP_MED_EVOL","IMP_NORED_DOL_DAP_TEND","PCT_PONDINTER_SOL_DAP_
                    MED","PCT_PONDINTER_DOL_DAP_MED","PCT_IMP_SOL_DAP_MED","PCT_IMP_
                    DOL_DAP_MED","PCT_IMP_NORED_SOL_DAP_MED","PCT_IMP_NORED_DOL_DAP_
                    MED","IMP_SAL_FMU_RF","IMP_SAL_FMU_MO","IMP_SAL_FMU_AG","IMP_SAL_
                    FMU","NUM_FMU","NUM_CANCEL_FMU","NUM_MES_APEREC_FMU_LN",
                "NUM_MES_APEANT_FMU_LN","NUM_MES_CANCREC_FMU_LN","NUM_AP_FMU_MED","NUM
                    _RES_FMU_MED","NUM_PART_FMU_MED","NUM_MES_AP_FMU_OPERA","NUM_MES_
                    RES_FMU_OPERA","NUM_MES_PART_FMU_OPERA","IMP_SAL_FMU_MED","IMP_SAL
                    _FMU_MED_EVOL","IMP_SAL_FMU_TEND","IMP_SAL_FMU_RF_MED","IMP_SAL_
                    FMU_MO_MED","IMP_SAL_FMU_AG_MED","PCT_IMP_SAL_DOL_FMU_MED","PCT_
                    IMP_SAL_FMU_RF_MED","PCT_IMP_SAL_FMU_MO_MED","PCT_IMP_SAL_FMU_AG_
                    MED","IMP_ACTIVO_TOTAL_ACTUAL","IMP_ACTIVO_MED",
                "IMP_ACTIVO_MED_EVOL","IMP_ACTIVO_TEND","PCT_ACTIVO_ACTUAL_MAX12","IMP
                    _AMORT_PPE_MAX","IMP_AMORT_PPE_MED","IMP_DEUDAACT_PPE","IMP_DEUDA_
```

```
        PPE_MAX","IMP_DEUDA_PPE_MED","IMP_DEUDA_PPE_MED_EVOL","IMP_DEUDA_
        PPE_TEND","NUM_DIAS_MAXMORA_PPE_MAX","NUM_MES_MAXVCTO_PPE_LN","NUM
        _PPE","NUM_CTACANC_PPE","NUM_CTATOT_PPE","NUM_MES_APEREC_PPE_LN","
        NUM_MES_APEANT_PPE_LN","NUM_MES_CANCREC_PPE_LN","IMP_DEUDAACT_HIP"
        ,"IMP_AMORT_HIP_MAX",
21      "IMP_AMORT_HIP_MED","IMP_DEUDAINI_HIP_MAX","IMP_DEUDAINI_HIP_MED","IMP
        _DEUDA_HIP_MAX","IMP_DEUDA_HIP_MED","NUM_DIAS_MAXMORA_HIP_MAX","
        NUM_MES_MAXVCTO_HIP_LN","NUM_HIP","NUM_CTACANC_HIP","NUM_CTATOT_
        HIP","NUM_MES_APEREC_HIP_LN","NUM_MES_APEANT_HIP_LN","NUM_MES_
        CANCREC_HIP_LN","IMP_OPE_BT","NUM_OPE_BT","NUM_OPE_SOL_BT","NUM_
        OPE_DOL_BT","NUM_MESES_ANT_BT_LN","NUM_MESES_REC_BT_LN","IMP_OPE_
        BT_MED",
22      "IMP_OPE_BT_MED_EVOL","NUM_OPE_BT_MED","NUM_MESES_BT_OPERA","NUM_OPE_
        SOL_BT_MED","NUM_MESES_SOL_BT_OPERA","NUM_OPE_DOL_BT_MED","NUM_
        MESES_DOL_BT_OPERA","PCT_IMP_CITI_BT_MED","PCT_IMP_INTER_BT_MED","
        PCT_IMP_SCOTIA_BT_MED","PCT_IMP_OTR_BT_MED","PCT_IMP_SAGA_BT_MED",
        "PCT_IMP_RIPLEY_BT_MED","PCT_OPE_CITI_BT_MED","PCT_OPE_INTER_BT_
        MED","PCT_OPE_SCOTIA_BT_MED","PCT_OPE_OTR_BT_MED","PCT_OPE_SAGA_BT
        _MED","PCT_OPE_RIPLEY_BT_MED","PCT_OPE_SOL_BT_MED",
23      "PCT_OPE_DOL_BT_MED","PCT_IMP_REVOLV_BT_MED","PCT_IMP_CUOTAS_BT_MED","
        IMP_OPE_EP","IMP_OPE_SOL_EP","IMP_OPE_DOL_EP","NUM_OPE_EP","NUM_
        OPE_SOL_EP","NUM_OPE_DOL_EP","NUM_MESES_ANT_EP_LN","NUM_MESES_REC_
        EP_LN","IMP_OPE_EP_MED","IMP_OPE_SOL_EP_MED","IMP_OPE_DOL_EP_MED",
        "NUM_OPE_EP_MED","NUM_MESES_EP_OPERA","NUM_OPE_SOL_EP_MED","NUM_
        MESES_SOL_EP_OPERA","NUM_OPE_DOL_EP_MED","NUM_MESES_DOL_EP_OPERA",
24      "PCT_IMP_SOL_EP_MED","PCT_IMP_DOL_EP_MED","PCT_OPE_SOL_EP_MED","PCT_
        OPE_DOL_EP_MED","PCT_VISTA_PASIVO_TOTAL_ACTUAL","IMP_PASIVO_ACTUAL
        ","PCT_VISTA_PASIVO_TOTAL_MED","IMP_PASIVO_MED","IMP_VISTA_MED","
        IMP_FMUDAP_MED","PCT_FMUDAP_PASIVO_TOTAL_MED","IMP_PASIVO_TEND","
        IMP_PASIVO_MED_EVOL","IMP_OPE_TC","IMP_DEUDA_TC","IMP_AMORT_TC","
        NUM_OPE_TC","IMP_LIMITE_TC","IMP_DISPONIBLE_TC","NUM_TC",
25      "NUM_CANCEL_TC","NUM_MESES_APER_TC","IMP_OPE_TC_MED","IMP_OPE_TC_MED_
        EVOL","IMP_OPE_TC_TEND","IMP_DEUDA_TC_MED","IMP_DEUDA_TC_MED_EVOL"
        ,"IMP_DEUDA_TC_TEND","IMP_AMORT_TC_MED","IMP_DISPONIBLE_TC_MED","
        IMP_DISPONIBLE_TC_MED_EVOL","IMP_DISPONIBLE_TC_TEND","IMP_LIMITE_
        TC_MED","IMP_LIMITE_TC_MED_EVOL","IMP_LIMITE_TC_TEND","IMP_MED_OPE
        _TC_MED","NUM_OPE_TC_MED","NUM_OPE_TC_TEND","NUM_MESES_OPERA_TC","
        PCT_IMP_OPE_BENEF_TC_MED",
26      "PCT_IMP_OPE_SOL_TC_MED","PCT_IMP_DISPUESTO_TC_MED","PCT_NUM_OPE_BENEF
        _TC_MED","PCT_IMP_OPE_POS_TC_MED","IMP_OPE_TD","NUM_OPE_TD","NUM_
        TD","IMP_OPE_TD_MED","IMP_OPE_TD_MED_EVOL","IMP_OPE_TD_TEND","IMP_
        MED_OPE_TD_MED","NUM_OPE_TD_MED","NUM_OPE_TD_MED_EVOL","NUM_MESES_
        OPERA_TD","PCT_IMP_OPE_POS_TD_MED","IMP_OPE_IN_TNS","IMP_OPE_OUT_
        TNS","NUM_OPE_IN_TNS","NUM_OPE_OUT_TNS","NUM_MESES_OPERA_GNA",
27      "NUM_MESES_OPERA_GIN","NUM_MESES_OPERA_REM","NUM_MESES_OPERA_TRA","IMP
        _OPE_IN_TNS_MED","IMP_OPE_IN_TNS_MED2","IMP_OPE_IN_TNS_MED_EVOL","
        IMP_OPE_IN_TNS_TEND","IMP_OPE_OUT_TNS_MED","IMP_OPE_OUT_TNS_MED2",
        "IMP_OPE_OUT_TNS_MED_EVOL","IMP_OPE_OUT_TNS_TEND","NUM_OPE_IN_TNS_
        MED","NUM_OPE_IN_TNS_TEND","NUM_OPE_OUT_TNS_MED","NUM_OPE_OUT_TNS_
        TEND","RAT_IMP_OPE_IN_TNS_MED","IMP_OPE_MP","NUM_OPE_MP","NUM_MP",
        "IMP_OPE_MP_MED",
28      "IMP_OPE_MP_MED_EVOL","IMP_OPE_MP_TEND","NUM_OPE_MP_MED","NUM_OPE_MP_
        TEND","NUM_MESES_OPERA_MP","PCT_IMP_OPE_CREDITO_MP_MED","PCT_NUM_
        OPE_CREDITO_MP_MED","NUM_SVR","NUM_CANCEL_SVR","NUM_MESES_APER_SVR
        ","IMP_CUOTA_SVR_MED","IMP_CUOTA_SVR_MED_EVOL","NUM_SEG","NUM_
        CANCEL_SEG","NUM_MESES_APER_SEG","IMP_CUOTA_SEG_MED","IMP_CUOTA_
        SEG_MED_EVOL","IMP_RCC_VIGENTE","IMP_RCC_REFINANCIADA","IMP_RCC_
        VENCIDA",
29      "IMP_RCC_JUDICIAL","NUM_MES_ULT_HIPOTECA","NUM_MES_ULT_PRESTAMO","IMP_
        RCC_VIGENTE_MED","IMP_RCC_VIGENTE_MED_EVOL","IMP_RCC_VIGENTE_TEND"
```

```
                    ,"IMP_RCC_VENCIDA_MED","IMP_RCC_VENCIDA_MED_EVOL","IMP_RCC_VENCIDA
                    _TEND","IMP_RCC_REFINANCIADA_MED","IMP_RCC_JUDICIAL_MED","IMP_RCC_
                    TC","IMP_RCC_PTM","IMP_RCC_HIP","IMP_MAX_12M_CREDITOS","IMP_MAX_12
                    M_CREDITOS_PRESTAMOS","PCT_RCC_REFINANCIADA_MED","PCT_RCC_JUDICIAL
                    _MED","PCT_RCC_DOL_MED","PCT_RCC_BCP_MED",
30              "PCT_RCC_BCO_MED","PCT_RCC_CAJ_MED","PCT_RCC_RET_MED","PCT_RCC_OTR_MED
                    ","RAT_RCC_4A_CONSUMO_VS_TOTAL","NUM_ACREEDORES","IMP_LIM_CRED_
                    TOTAL","NUM_OPE_GRUPO1_MED","NUM_OPE_GRUPO2_MED","NUM_OPE_GRUPO3_
                    MED","NUM_OPE_GRUPO4_MED","NUM_OPE_GRUPO5_MED","NUM_OPE_GRUPO6_MED
                    ","IMP_OPE_GRUPO1_MED","IMP_OPE_GRUPO2_MED","IMP_OPE_GRUPO3_MED","
                    IMP_OPE_GRUPO4_MED","IMP_OPE_GRUPO5_MED","IMP_OPE_GRUPO6_MED","IMP
                    _REDIMIDO_MILTRA_12M",
31              "PCT_USO_CANAL_ATM","PCT_USO_CANAL_TLF","PCT_USO_CANAL_INT","PCT_USO_
                    CANAL_PLAT","PCT_USO_CANAL_POS","PCT_USO_CANAL_SDO","PCT_USO_CANAL
                    _VENT","NUM_RECLAMOS_6M_AHORRO","NUM_RECLAMOS_6M_CTS","NUM_
                    RECLAMOS_6M_DEPOSITOS","NUM_RECLAMOS_6M_FONDOS","NUM_RECLAMOS_6M_
                    PRESTAMOS","NUM_RECLAMOS_6M_HIPOTECAS","NUM_RECLAMOS_6M_TC","NUM_
                    RECLAMOS_6M_TD","NUM_RECLAMOS_6M_SEG_RETORNO","NUM_RECLAMOS_6M_SEG
                    _OTROS","NUM_RECLAMOS_6M_OTROS","NUM_RECLAMOS","RAT_pasivo_total_
                    Med",
32              "RAT_pasivo_total_Med_Evol","RAT_pasivo_total_Med2","RAT_ahorro_vista_
                    Med","RAT_ahorro_vista_Med_Evol","RAT_ahorro_vista_Med2","RAT_
                    fmudap_Med","RAT_fmudap_Med_Evol","RAT_fmudap_Med2","RAT_activo_
                    total_Med","RAT_activo_total_Med_Evol","RAT_activo_total_Med2","
                    IMP_INGREG_01","IMP_INGREG_02","IMP_GASTREG_01","IMP_GASTREG_02","
                    IMP_INGREG_CV","IMP_GASTREG_CV","NUM_D_UTILITIES","VAR_D_UTILITIES
                    ","IMP_D_UTILITIES_AGUA_MED",
33              "IMP_D_UTILITIES_TELECO_MED","IMP_D_LIFES_SEGCLINICAS_MED","IMP_D_
                    LIFES_CLUB_MED","IMP_D_LIFES_BELLEZA_MED","IMP_D_LIFES_EDUC_MED","
                    IMP_D_LIFES_INSTITUC_EMPR_MED","IMP_D_IMPTO_MUNICIPSAT_SUM","NUM_
                    UTILITIES","VAR_UTILITIES","IMP_UTILITIES_AGUA_MED","IMP_UTILITIES
                    _TELECO_MED","IMP_UTILITIES_LUZ_MED","IMP_UTILITIES_GAS_MED","IMP_
                    LIFES_SEGCLINICAS_MED","IMP_LIFES_CLUB_MED","IMP_LIFES_BELLEZA_MED
                    ","IMP_LIFES_EDUC_MED","IMP_LIFES_INSTITUC_EMPR_MED","IMP_LIFES_
                    AUTO_MED","IMP_LIFES_FINANAFP_MED",
34              "IMP_LIFES_CONLINE_MED","IMP_IMPTO_MUNICIPSAT_SUM","NUM_CAMP_ENVIADAS"
                    ,"NUM_CAMP_ENVIADAS_CANALINTR","NUM_CAMP_CONTACT","NUM_CAMP_
                    CONTACT_CANALINTR","NUM_CAMP_EXITO","NUM_CAMP_EXITO_INC_VIAJES","
                    NUM_CAMP_EXITO_INC_PUNTOS","NUM_CAMP_EXITO_INC_TASAS","NUM_CAMP_
                    EXITO_INC_AMPLIACION","NUM_CAMP_EXITO_INC_COMPDEUDA","NUM_CAMP_
                    EXITO_INC_NO","NUM_CAMP_RECHA_PROD_AHORRO","NUM_CAMP_RECHA_PROD_
                    DEUDA","NUM_CAMP_RECHA_PROD_EF","NUM_CAMP_RECHA_PROD_CP","NUM_CAMP
                    _RECHA_PROD_CH","NUM_CAMP_RECHA_PROD_SEGUROS","NUM_CAMP_RECHA_PROD
                    _TARJETA",
35              "NUM_CAMP_RECHA_PROD_DATOS","NUM_CAMP_SOLI_INC_VIAJES","NUM_CAMP_SOLI_
                    INC_PUNTOS","NUM_CAMP_SOLI_INC_TASAS","NUM_CAMP_SOLI_INC_
                    AMPLIACION","NUM_CAMP_SOLI_INC_COMPDEUDA","NUM_CAMP_SOLI_INC_NO","
                    RCC_DIAMOROSIDAD","IMP_RCC_TC_DISPEFECT","IMP_RCC_TC_COMPRA","IMP_
                    RCC_TC_OTROSCONC","NUM_REVOLVING_6M","IMP_REVOLVING_MED","IMP_
                    RENTAS_INF_MED","NUM_NHOGARES_MANZANA","NUM_NPERSONAS_MANZANA","
                    PCT_MUJERES","PCT_EXTRANJEROS","PCT_ALQUILER",
36              "PCT_AGUA_DIARIO","PCT_ALUMBRADO","PCT_TV_CABLE","PCT_SIN_SERVICIOS","
                    PCT_ESTUDIANTES","PCT_PARADOS","PCT_DISCAPACITADOS","PCT_OCUP_
                    PATRON","PCT_OCUP_FAMILIA","PCT_RELIGION_EVAN","PCT_AFILIADOS","
                    PCT_DNI","PCT_ANALFABETISMO","NUM_PRODUCTOS")
37 columnasCat = c("FLGCLIENTENEGATIVO","STR_TIPOBANCA","STR_MACROZONALIMA","STR_
    PROVINCIA","STR_ESTCIVIL","IND_SEXO","IND_SITLABORAL","IND_EMPTRABAJADORBCP","IND_
    RESIDENTE","IND_NIVSOCIOECO","IND_SEGMENTOBEX","IND_SEGMENTOPDH","IND_CCT","IND_
    CAH","IND_CSU","IND_CVIS","IND_CANCREC_CCT","IND_CANCREC_CAH","IND_CANCREC_CSU","
    IND_CANCREC_CVIS",
```

```
38                "IND_APEREC_CCT","IND_APEREC_CAH","IND_APEREC_CSU","IND_APEREC_CVIS","
                      IND_CTS_MIG_IN","IND_CTS_MIG_IN_W","IND_CTS_MIG_OUT","IND_CTS_MIG_
                      OUT_W","IND_CTS","IND_REC_APER_CTS","IND_REC_CANC_CTS","IND_DAP_
                      PREF","IND_DAP","IND_DAP_NORED","IND_REC_APER_DAP","IND_REC_CANC_
                      DAP","IND_FMU","IND_REC_APER_FMU","IND_REC_CANC_FMU","IND_PEFE",
39                "IND_PVEH","IND_POTR","IND_PPE","IND_CTAVEND_PPE","IND_CANCREC_PPE","
                      IND_APEREC_PPE","IND_HTRA","IND_HMIV","IND_HIP","IND_CTAVEND_HIP",
                      "IND_CANCREC_HIP","IND_APEREC_HIP","IND_TC","IND_TC_PREMIUM","IND_
                      REC_APER_TC","IND_REC_CANC_TC","IND_TD","IND_REC_USO_IN_GNA","IND_
                      REC_USO_OUT_GNA","IND_REC_USO_IN_GIN",
40                "IND_REC_USO_OUT_GIN","IND_REC_USO_IN_REM","IND_REC_USO_IN_TRA","IND_
                      REC_USO_OUT_TRA","IND_MP","IND_SVR","IND_REC_APER_SVR","IND_REC_
                      CANC_SVR","IND_SEG","IND_REC_APER_SEG","IND_REC_CANC_SEG","IND_
                      PROGRAMA_PDH","IND_PROGRAMA_PRIMAX","IND_PROGRAMA_MILTRA","IND_
                      PROGRAMA_LANPASS","IND_RECLAMOS","IND_INGREG","IND_GASTREG","IND_D
                      _UTILITIES_AGUA","IND_D_UTILITIES_TELECO",
41                "IND_D_UTILITIES","IND_D_LIFES_SEGCLINICAS","IND_D_LIFES_CLUB","IND_D_
                      LIFES_BELLEZA","IND_D_LIFES_EDUC","IND_D_LIFES_INSTITUC_EMPR","IND
                      _D_IMPTO_MUNICIPSAT","IND_UTILITIES_AGUA","IND_UTILITIES_TELECO","
                      IND_UTILITIES_LUZ","IND_UTILITIES_GAS","IND_UTILITIES","IND_LIFES_
                      SEGCLINICAS","IND_LIFES_CLUB","IND_LIFES_BELLEZA","IND_LIFES_EDUC"
                      ,"IND_LIFES_INSTITUC_EMPR","IND_LIFES_AUTO","IND_LIFES_FINANAFP","
                      IND_LIFES_CONLINE",
42                "IND_IMPTO_MUNICIPSAT","IND_CAMP_EXITO_CANAL_TELEMKT_OUT","IND_CAMP_
                      EXITO_CANAL_TELEMKT_IN","IND_CAMP_EXITO_CANAL_AGEN_PRO","IND_CAMP_
                      EXITO_CANAL_AGEN_REA","IND_CAMP_EXITO_CANAL_MAIL","IND_CAMP_EXITO_
                      CANAL_EMAIL","IND_CAMP_EXITO_CANAL_INTERNET","IND_CAMP_EXITO_CANAL
                      _OTROS","IND_CAMP_SOLI_CANAL_TELEMKT_OUT","IND_CAMP_SOLI_CANAL_
                      TELEMKT_IN","IND_CAMP_SOLI_CANAL_AGEN_PRO","IND_CAMP_SOLI_CANAL_
                      AGEN_REA","IND_CAMP_SOLI_CANAL_MAIL","IND_CAMP_SOLI_CANAL_EMAIL","
                      IND_CAMP_SOLI_CANAL_INTERNET","IND_CAMP_SOLI_CANAL_OTROS","IND_
                      CAMP_SOLI_PROD_AHORRO","IND_CAMP_SOLI_PROD_DEUDA","IND_CAMP_SOLI_
                      PROD_EF",
43                "IND_CAMP_SOLI_PROD_CP","IND_CAMP_SOLI_PROD_CH","IND_CAMP_SOLI_PROD_
                      SEGUROS","IND_CAMP_SOLI_PROD_TARJETA","IND_CAMP_SOLI_PROD_DATOS","
                      IND_REVOLVING_6M","IND_EDAD_MED","IND_NSE_MED","GRP_TIPO_VIVIENDA_
                      MAX","GRP_VIVIENDA_MAX","GRP_PARED_MAX","GRP_PISO_MAX","GRP_TIPO_
                      ABASTECIMIENTO_MAX","GRP_SERV_HIGIENICOS_MAX","GRP_ENERGIA_MAX","
                      GRP_LENGUA_MAX","IND_NIVEL_ESTUDIOS_MED","GRP_ESTADO_CIVIL_MAX","
                      IND_DENSI_SUPE_DECIL","IND_DENSI_BANC_DECIL","IND_DENSI_FARM_DECIL
                      ",
44                "IND_DENSI_REST_DECIL","SEGM","IND_PAFG_CAH","IND_PAXS_CSU","IND_PAFG_
                      CSU","IND_PAXS_CTS","IND_PAFG_CTS","IND_PAXS_DAP","IND_PAUP_DAP","
                      IND_PAFG_DAP","IND_PAXS_FMU","IND_PAUP_FMU","IND_PAFG_FMU","IND_
                      PAXS_HIP_AyB","IND_PAXS_HIP_CyD","IND_PAFG_HIP","IND_PAXS_PPE","
                      IND_PAUP_PPE","IND_PAFG_PPE","IND_PAXS_TC",
45                "IND_PAUP_TC","IND_PAFG_TC","IND_PAXS_BT","IND_PAXS_EP","IND_PAUP_REV"
                      ,"COD_ACTECONOMICA","COD_DEPARTAMENTO","COD_DISTRITO","COD_
                      PROFESION","RCC_TIPCLASIFRIESGO")
46 columnasEvaluar = c(columnasNum, columnasCat)
47 columnasNoEvaluar = c("CODCLAVECIC", "PI_SB", "PI_SM", "PI_SR", "PI_AJUSTADO_SB", "PI_
       AJUSTADO_SM", "PI_AJUSTADO_SR", "SB_FLGVTA", "SB_FLGVTA_PERSISTENCIA6M", "SM_
       FLGVTA", "SM_FLGVTA_PERSISTENCIA6M", "SR_FLGVTA", "SR_FLGVTA_PERSISTENCIA6M")
48
49 #Análisis de zero y near-zero variance
50 batchSize = 50
51 totalSize = length(columnasNum)
52
53 pt = proc.time()
54 seguros.nzv.1 = nearZeroVar(seguros.nba[, .SD, .SDcols=columnasNum[1:batchSize]],
```

```
    saveMetrics = TRUE)
proc.time() - pt
pt = proc.time()
seguros.nzv.2 = nearZeroVar(seguros.nba[, .SD, .SDcols=columnasNum[(batchSize+1):(
    batchSize*2)]], saveMetrics = TRUE)
proc.time() - pt
pt = proc.time()
seguros.nzv.3 = nearZeroVar(seguros.nba[, .SD, .SDcols=columnasNum[(batchSize*2+1):(
    batchSize*3)]], saveMetrics = TRUE)
proc.time() - pt
pt = proc.time()
seguros.nzv.4 = nearZeroVar(seguros.nba[, .SD, .SDcols=columnasNum[(batchSize*3+1):(
    batchSize*4)]], saveMetrics = TRUE)
proc.time() - pt
pt = proc.time()
seguros.nzv.5 = nearZeroVar(seguros.nba[, .SD, .SDcols=columnasNum[(batchSize*4+1):(
    batchSize*5)]], saveMetrics = TRUE)
proc.time() - pt
pt = proc.time()
seguros.nzv.6 = nearZeroVar(seguros.nba[, .SD, .SDcols=columnasNum[(batchSize*5+1):(
    batchSize*6)]], saveMetrics = TRUE)
proc.time() - pt
pt = proc.time()
seguros.nzv.7 = nearZeroVar(seguros.nba[, .SD, .SDcols=columnasNum[(batchSize*6+1):(
    batchSize*7)]], saveMetrics = TRUE)
proc.time() - pt
pt = proc.time()
seguros.nzv.8 = nearZeroVar(seguros.nba[, .SD, .SDcols=columnasNum[(batchSize*7+1):(
    batchSize*8)]], saveMetrics = TRUE)
proc.time() - pt
pt = proc.time()
seguros.nzv.9 = nearZeroVar(seguros.nba[, .SD, .SDcols=columnasNum[(batchSize*8+1):(
    batchSize*9)]], saveMetrics = TRUE)
proc.time() - pt
pt = proc.time()
seguros.nzv.10 = nearZeroVar(seguros.nba[, .SD, .SDcols=columnasNum[(batchSize*9+1):(
    totalSize)]], saveMetrics = TRUE)
proc.time() - pt

totalSizeCat = length(columnasCat)
pt = proc.time()
seguros.nzv.11 = nearZeroVar(seguros.nba[, .SD, .SDcols=columnasCat[1:batchSize]],
    saveMetrics = TRUE)
proc.time() - pt
pt = proc.time()
seguros.nzv.12 = nearZeroVar(seguros.nba[, .SD, .SDcols=columnasCat[(batchSize*1+1):(
    batchSize*2)]], saveMetrics = TRUE)
proc.time() - pt
pt = proc.time()
seguros.nzv.13 = nearZeroVar(seguros.nba[, .SD, .SDcols=columnasCat[(batchSize*2+1):(
    batchSize*3)]], saveMetrics = TRUE)
proc.time() - pt
pt = proc.time()
seguros.nzv.14 = nearZeroVar(seguros.nba[, .SD, .SDcols=columnasCat[(batchSize*3+1):(
    totalSizeCat)]], saveMetrics = TRUE)
proc.time() - pt

seguros.nzv = rbind(seguros.nzv.1, seguros.nzv.2, seguros.nzv.3, seguros.nzv.4,seguros
    .nzv.5, seguros.nzv.6, seguros.nzv.7, seguros.nzv.8, seguros.nzv.9, seguros.nzv
    .10, seguros.nzv.11, seguros.nzv.12, seguros.nzv.13, seguros.nzv.14)
```

```
nrow(seguros.nzv)
nrow(seguros.nzv[seguros.nzv$nzv == TRUE, ])
nrow(seguros.nzv[seguros.nzv$zeroVar == TRUE, ])

seguros.nzv[seguros.nzv$nzv == TRUE, ]
columnasEliminar = rownames(seguros.nzv[seguros.nzv$nzv == TRUE, ])
columnasMantener = c( "CODCLAVECIC", columnasEvaluar[ -match(columnasEliminar,
    columnasEvaluar) ], columnasNoEvaluar[-1])

seguros.nba.2 = seguros.nba[, .SD, .SDcols=columnasMantener]
rm(seguros.nba)
pt = proc.time()
write.table(seguros.nba.2, file="SegurosNBA_2.csv", row.names=FALSE, col.names=TRUE,
    sep=",")
proc.time() - pt
gc()
```

: ./scripts/03.1.1_AnalisisExploratorio_NZV1.R

```
#Evaluar correlaciones entre variables predictoras numéricas.
#Si una variable está correlacionada solo con otra variable, se descarta una de las
    variables
#Si una variable está correlacionada con otras dos o más variables, se aplica análisis
     factorial. Los factores resultantes reemplazan a las variables originales.
setwd("C://Users//Sergio//Documents//Cursos//PUCP - Maestría Estadística//2014.2 -
    Ciclo 4//1 - Investigación en Estadística//Aplicación//Datos//Versión 3")
library(data.table)
library(caret)
library(psych)

memory.limit(40000)
pt = proc.time()
seguros.nba.2 = fread("SegurosNBA_2.csv", header=T, sep=",", stringsAsFactors=T)
proc.time() - pt
setkey(seguros.nba.2, CODCLAVECIC)

columnasModelo = colnames(seguros.nba.2)
columnasModeloNoEvaluar = c("CODCLAVECIC","PI_SB","PI_SM","PI_SR","PI_AJUSTADO_SB","PI
    _AJUSTADO_SM","PI_AJUSTADO_SR","SB_FLGVTA","SB_FLGVTA_PERSISTENCIA6M","SM_FLGVTA",
    "SM_FLGVTA_PERSISTENCIA6M","SR_FLGVTA","SR_FLGVTA_PERSISTENCIA6M")

################################
#Correlaciones altas para variables numéricas

columnasModeloNum = columnasModelo[match(columnasNum, columnasModelo)]
columnasModeloNum = columnasModeloNum[!is.na(columnasModeloNum)]
seguros.cor.1 = cor(seguros.nba.2[, .SD, .SDcols=columnasModeloNum])
seguros.cor.1
seguros.high.cor.1 = findCorrelation(seguros.cor.1, cutoff = .75)
columnasModeloNum.cor = columnasModeloNum[seguros.high.cor.1]
colCorrelaciones1 = matrix(nrow=45, ncol=6)
for (i in 1:length(columnasModeloNum.cor)){
  k = 1
  for (j in 1:length(columnasModeloNum)){
    if (j == 1){
      colCorrelaciones1[i, k] = columnasModeloNum.cor[i]
      columnaReferencia = columnasModeloNum.cor[i]
    }
    cor1 = seguros.cor.1[match(columnasModeloNum.cor[i], columnasModeloNum), j]
    if (cor1 >= 0.75){
```

```
      if (columnasModeloNum[j] != columnaReferencia){
        k = k + 1
        colCorrelaciones1[i, k] = columnasModeloNum[j]
      }
    }
  }
}
colCorrelaciones1

seguros.prueba.fa.1 = factanal(seguros.nba.2[, .SD, .SDcols=c("IMP_OPE_TD_MED","IMP_
    OPE_TD","IMP_OPE_MP")], 1, rotation="varimax", scores="regression")
seguros.prueba.fa.2 = factanal(seguros.nba.2[, .SD, .SDcols=c("NUM_OPE_TD_MED","NUM_
    OPE_TD","NUM_OPE_MP")], 1, rotation="varimax", scores="regression")
seguros.prueba.fa.3 = factanal(seguros.nba.2[, .SD, .SDcols=c("IMP_INTANG_CTS_MED","
    IMP_INTANG_CTS1","IMP_CTS_MED","IMP_INTANG_SOL_CTS_MED","IMP_INTANG_SOL_CTS","IMP_
    SOL_CTS_MED","IMP_CTS","IMP_SOL_CTS")], 2, rotation="varimax", scores="regression"
    )
seguros.prueba.fa.4 = factanal(seguros.nba.2[, .SD, .SDcols=c("IMP_RCC_TC_COMPRA","IMP
    _DEUDA_TC_MED","IMP_RCC_TC")], 1, rotation="varimax", scores="regression")
seguros.prueba.fa.5 = factanal(seguros.nba.2[, .SD, .SDcols=c("IMP_MAX_12M_CREDITOS","
    IMP_RCC_VIGENTE","IMP_RCC_VIGENTE_MED")], 1, rotation="varimax", scores="
    regression")
seguros.prueba.fa.6 = factanal(seguros.nba.2[, .SD, .SDcols=c("IMP_VISTA_MED","IMP_MED
    _CAH_MED","IMP_MED_CVIS_MED")], 1, rotation="varimax", scores="regression")
seguros.prueba.fa.7 = factanal(seguros.nba.2[, .SD, .SDcols=c("NUM_CTATOT_PPE","NUM_
    PPE","NUM_CTACANC_PPE")], 1, rotation="varimax", scores="regression")
seguros.prueba.fa.8 = factanal(seguros.nba.2[, .SD, .SDcols=c("IMP_ACTIVO_MED","IMP_
    ACTIVO_TOTAL_ACTUAL","RAT_activo_total_Med","RAT_activo_total_Med2")], 1, rotation
    ="varimax", scores="regression")
seguros.prueba.fa.9 = factanal(seguros.nba.2[, .SD, .SDcols=c("IMP_OPEHAB_CVIS_MED","
    IMP_OPEDEB_CVIS_MED","IMP_INGREG_01","IMP_GASTREG_01")], 1, rotation="varimax",
    scores="regression")
seguros.prueba.fa.10 = factanal(seguros.nba.2[, .SD, .SDcols=c("PCT_IMP_INTANG_CTS_MED
    ","NUM_CTS","PCT_IMP_SOL_CTS_MED","PCT_IMP_INTANG_SOL_CTS_MED")], 1, rotation="
    varimax", scores="regression")
seguros.prueba.fa.11 = factanal(seguros.nba.2[, .SD, .SDcols=c("RAT_pasivo_total_Med2"
    ,"RAT_fmudap_Med2","RAT_fmudap_Med","RAT_pasivo_total_Med")], 1, rotation="varimax
    ", scores="regression")
seguros.prueba.fa.12 = factanal(seguros.nba.2[, .SD, .SDcols=c("RAT_pasivo_total_Med_
    Evol","IMP_PASIVO_MED_EVOL","RAT_fmudap_Med_Evol")], 1, rotation="varimax", scores
    ="regression")

columnasModeloNumEliminar = c("IMP_OPE_TD_MED","IMP_OPE_TD","IMP_OPE_MP","NUM_OPE_TD_
    MED","NUM_OPE_TD","NUM_OPE_MP","IMP_INTANG_CTS_MED","IMP_INTANG_CTS1","IMP_CTS_MED
    ","IMP_INTANG_SOL_CTS_MED","IMP_INTANG_SOL_CTS","IMP_SOL_CTS_MED","IMP_CTS","IMP_
    SOL_CTS",
                              "IMP_RCC_TC_COMPRA","IMP_DEUDA_TC_MED","IMP_RCC_TC","IMP
                                  _MAX_12M_CREDITOS","IMP_RCC_VIGENTE","IMP_RCC_
                                  VIGENTE_MED","IMP_VISTA_MED","IMP_MED_CAH_MED","IMP_
                                  MED_CVIS_MED","NUM_CTATOT_PPE","NUM_PPE","NUM_
                                  CTACANC_PPE",
                              "IMP_ACTIVO_MED","IMP_ACTIVO_TOTAL_ACTUAL","RAT_activo_
                                  total_Med","RAT_activo_total_Med2","IMP_MED_CVIS_MED
                                  ","IMP_MED_CAH_MED","IMP_VISTA_MED","IMP_OPEHAB_CVIS
                                  _MED","IMP_OPEDEB_CVIS_MED","IMP_INGREG_01","IMP_
                                  GASTREG_01",
                              "PCT_IMP_INTANG_CTS_MED","NUM_CTS","PCT_IMP_SOL_CTS_MED"
                                  ,"PCT_IMP_INTANG_SOL_CTS_MED","RAT_pasivo_total_Med2
                                  ","RAT_fmudap_Med2","RAT_fmudap_Med","RAT_pasivo_
                                  total_Med","RAT_pasivo_total_Med_Evol","IMP_PASIVO_
```

```
                                    MED_EVOL","RAT_fmudap_Med_Evol")
columnasModeloNumEliminar = c(columnasModeloNumEliminar, "PCT_RCC_BCP_MED", "PCT_
    ACTIVO_ACTUAL_MAX12", "NUM_MES_APEANT_CVIS_LN", "IMP_PASIVO_ACTUAL", "NUM_OPE_IN_
    TNS", "IMP_GASTREG_02", "NUM_MP", "NUM_CAMP_EXITO_INC_TASAS", "NUM_OPEDEB_CVIS_MED
    ",
                                "PCT_IMP_INTANG_DOL_CTS_MED", "IMP_INTANG_CTS_MED_EVOL",
                                     "IMP_INTANG_SOL_CTS_MED_EVOL", "IMP_PASIVO_ACTUAL",
                                     "RAT_ahorro_vista_Med2", "RAT_ahorro_vista_Med_Evol
                                    ", "NUM_OPE_IN_TNS", "IMP_GASTREG_02", "NUM_MP", "
                                    NUM_CAMP_EXITO_INC_TASAS", "PCT_ALUMBRADO")
columnasModeloNumMantener = columnasModeloNum[-match(columnasModeloNumEliminar,
    columnasModeloNum)]
columnasModeloNumNuevas = c("FR_OPE_TD_MP", "FR_NUM_OPE_TD_MP", "FR_IMP_CTS_1", "FR_
    IMP_CTS_2", "FR_IMP_RCC_TC", "FR_IMP_RCC", "FR_IMP_VISTA_CAH", "FR_NUM_PPE", "FR_
    ACTIVO", "FR_CVIS_INGGAST", "FR_PCT_CTS", "FR_RAT_PASIVO_FMU", "FR_RAT_PASIVO_FMU_
    EVOL")

###################################
columnasModeloCat = columnasModelo[match(columnasCat, columnasModelo)]
columnasModeloCat = columnasModeloCat[!is.na(columnasModeloCat)]


###################################
columnasModeloMantener = c("CODCLAVECIC", columnasModeloNumMantener, columnasModeloCat
    )
seguros.nba.3 = seguros.nba.2[, .SD, .SDcols=columnasModeloMantener]
seguros.nba.3[, c(columnasModeloNumNuevas) := list(seguros.prueba.fa.1$scores, seguros
    .prueba.fa.2$scores, seguros.prueba.fa.3$scores[,1], seguros.prueba.fa.3$scores
    [,2], seguros.prueba.fa.4$scores, seguros.prueba.fa.5$scores, seguros.prueba.fa.6$
    scores, seguros.prueba.fa.7$scores, seguros.prueba.fa.8$scores, seguros.prueba.fa
    .9$scores, seguros.prueba.fa.10$scores, seguros.prueba.fa.11$scores, seguros.
    prueba.fa.12$scores)]
seguros.nba.3[, c(columnasModeloNoEvaluar[-1]) := seguros.nba.2[, .SD, .SDcols=
    columnasModeloNoEvaluar[-1]] ]
write.table(seguros.nba.3, file="SegurosNBA_3.csv", row.names=FALSE, col.names=TRUE,
    sep=",")
rm(seguros.nba.2)
gc()
```

: ./scripts/03.2_1_AnalisisExploratorio_CorrelacionPredictores.R

```
#Encontrar variables predictoras numéricas que sean combinaciones lineales de otras
    variables numéricas.
setwd("C://Users//Sergio//Documents//Cursos//PUCP - Maestría Estadística//2014.2 -
    Ciclo 4//1 - Investigación en Estadística//Aplicación//Datos//Versión 3")
library(data.table)
library(caret)

memory.limit(40000)
pt = proc.time()
seguros.nba.3 = fread("SegurosNBA_3.csv", header=T, sep=",", stringsAsFactors=T)
proc.time() - pt
setkey(seguros.nba.3, CODCLAVECIC)

columnasModelo = colnames(seguros.nba.3)
columnasModeloCat = c("STR_TIPOBANCA","STR_MACROZONALIMA","STR_PROVINCIA","STR_
    ESTCIVIL","IND_SEXO","IND_NIVSOCIOECO","IND_SEGMENTOBEX","IND_SEGMENTOPDH","IND_
    CAH","IND_CSU","IND_CVIS","IND_CANCREC_CVIS","IND_APEREC_CAH","IND_APEREC_CSU","
    IND_APEREC_CVIS","IND_CTS","IND_REC_APER_CTS","IND_REC_CANC_CTS","IND_PEFE","IND_
    PPE",
```

```
14                      "IND_TC","IND_TC_PREMIUM","IND_TD","IND_REC_USO_IN_TRA","IND_REC
                            _USO_OUT_TRA","IND_MP","IND_SEG","IND_PROGRAMA_PDH","IND_
                            PROGRAMA_LANPASS","IND_INGREG","IND_GASTREG","IND_CAMP_EXITO
                            _CANAL_AGEN_REA","IND_CAMP_SOLI_CANAL_AGEN_REA","IND_
                            REVOLVING_6M","IND_EDAD_MED","IND_NSE_MED","GRP_TIPO_
                            VIVIENDA_MAX","GRP_VIVIENDA_MAX","GRP_PARED_MAX","GRP_PISO_
                            MAX",
15                      "GRP_TIPO_ABASTECIMIENTO_MAX","GRP_SERV_HIGIENICOS_MAX","GRP_
                            ENERGIA_MAX","GRP_LENGUA_MAX","IND_NIVEL_ESTUDIOS_MED","GRP_
                            ESTADO_CIVIL_MAX","IND_DENSI_FARM_DECIL","IND_DENSI_REST_
                            DECIL","SEGM","IND_PAFG_CAH","IND_PAXS_CSU","IND_PAFG_CSU","
                            IND_PAXS_CTS","IND_PAFG_CTS","IND_PAXS_HIP_AyB","IND_PAXS_
                            HIP_CyD","IND_PAXS_PPE","IND_PAUP_PPE","IND_PAFG_PPE","IND_
                            PAXS_TC",
16                      "IND_PAXS_BT","IND_PAXS_EP","IND_PAUP_REV","COD_ACTECONOMICA","
                            COD_DEPARTAMENTO","COD_DISTRITO","COD_PROFESION","RCC_
                            TIPCLASIFRIESGO")
17 columnasModeloNum = columnasModelo[ -match(columnasModeloCat, columnasModelo) ]
18 seguros.nba.3[, c("STR_TIPOBANCA","STR_MACROZONALIMA","STR_PROVINCIA","STR_ESTCIVIL","
       IND_SEXO","IND_NIVSOCIOECO","IND_SEGMENTOBEX","IND_SEGMENTOPDH","IND_CAH","IND_CSU
       ","IND_CVIS","IND_CANCREC_CVIS","IND_APEREC_CAH","IND_APEREC_CSU","IND_APEREC_CVIS
       ","IND_CTS","IND_REC_APER_CTS","IND_REC_CANC_CTS","IND_PEFE","IND_PPE",
19                    "IND_TC","IND_TC_PREMIUM","IND_TD","IND_REC_USO_IN_TRA","IND_REC_USO
                          _OUT_TRA","IND_MP","IND_SEG","IND_PROGRAMA_PDH","IND_PROGRAMA_
                          LANPASS","IND_INGREG","IND_GASTREG","IND_CAMP_EXITO_CANAL_AGEN_
                          REA","IND_CAMP_SOLI_CANAL_AGEN_REA","IND_REVOLVING_6M","IND_EDAD
                          _MED","IND_NSE_MED","GRP_TIPO_VIVIENDA_MAX","GRP_VIVIENDA_MAX","
                          GRP_PARED_MAX","GRP_PISO_MAX",
20                    "GRP_TIPO_ABASTECIMIENTO_MAX","GRP_SERV_HIGIENICOS_MAX","GRP_ENERGIA
                          _MAX","GRP_LENGUA_MAX","IND_NIVEL_ESTUDIOS_MED","GRP_ESTADO_
                          CIVIL_MAX","IND_DENSI_FARM_DECIL","IND_DENSI_REST_DECIL","SEGM",
                          "IND_PAFG_CAH","IND_PAXS_CSU","IND_PAFG_CSU","IND_PAXS_CTS","IND
                          _PAFG_CTS","IND_PAXS_HIP_AyB","IND_PAXS_HIP_CyD","IND_PAXS_PPE",
                          "IND_PAUP_PPE","IND_PAFG_PPE","IND_PAXS_TC",
21                    "IND_PAXS_BT","IND_PAXS_EP","IND_PAUP_REV","COD_ACTECONOMICA","COD_
                          DEPARTAMENTO","COD_DISTRITO","COD_PROFESION","RCC_
                          TIPCLASIFRIESGO") :=
22                list(as.factor(STR_TIPOBANCA),as.factor(STR_MACROZONALIMA),as.factor(STR
                      _PROVINCIA),as.factor(STR_ESTCIVIL),as.factor(IND_SEXO),as.factor(
                      IND_NIVSOCIOECO),as.factor(IND_SEGMENTOBEX),as.factor(IND_
                      SEGMENTOPDH),as.factor(IND_CAH),as.factor(IND_CSU),as.factor(IND_
                      CVIS),as.factor(IND_CANCREC_CVIS),as.factor(IND_APEREC_CAH),as.
                      factor(IND_APEREC_CSU),as.factor(IND_APEREC_CVIS),as.factor(IND_CTS)
                      ,as.factor(IND_REC_APER_CTS),as.factor(IND_REC_CANC_CTS),as.factor(
                      IND_PEFE),as.factor(IND_PPE),
23                     as.factor(IND_TC),as.factor(IND_TC_PREMIUM),as.factor(IND_TD),as.
                           factor(IND_REC_USO_IN_TRA),as.factor(IND_REC_USO_OUT_TRA),as.
                           factor(IND_MP),as.factor(IND_SEG),as.factor(IND_PROGRAMA_PDH),
                           as.factor(IND_PROGRAMA_LANPASS),as.factor(IND_INGREG),as.factor
                           (IND_GASTREG),as.factor(IND_CAMP_EXITO_CANAL_AGEN_REA),as.
                           factor(IND_CAMP_SOLI_CANAL_AGEN_REA),as.factor(IND_REVOLVING_6M
                           ),as.factor(IND_EDAD_MED),as.factor(IND_NSE_MED),as.factor(GRP_
                           TIPO_VIVIENDA_MAX),as.factor(GRP_VIVIENDA_MAX),as.factor(GRP_
                           PARED_MAX),as.factor(GRP_PISO_MAX),
24                     as.factor(GRP_TIPO_ABASTECIMIENTO_MAX),as.factor(GRP_SERV_
                           HIGIENICOS_MAX),as.factor(GRP_ENERGIA_MAX),as.factor(GRP_LENGUA
                           _MAX),as.factor(IND_NIVEL_ESTUDIOS_MED),as.factor(GRP_ESTADO_
                           CIVIL_MAX),as.factor(IND_DENSI_FARM_DECIL),as.factor(IND_DENSI_
                           REST_DECIL),as.factor(SEGM),as.factor(IND_PAFG_CAH),as.factor(
                           IND_PAXS_CSU),as.factor(IND_PAFG_CSU),as.factor(IND_PAXS_CTS),
```

```
                        as.factor(IND_PAFG_CTS),as.factor(IND_PAXS_HIP_AyB),as.factor(
                        IND_PAXS_HIP_CyD),as.factor(IND_PAXS_PPE),as.factor(IND_PAUP_
                        PPE),as.factor(IND_PAFG_PPE),as.factor(IND_PAXS_TC),
                    as.factor(IND_PAXS_BT),as.factor(IND_PAXS_EP),as.factor(IND_PAUP_
                        REV),as.factor(COD_ACTECONOMICA),as.factor(COD_DEPARTAMENTO),as
                        .factor(COD_DISTRITO),as.factor(COD_PROFESION),as.factor(RCC_
                        TIPCLASIFRIESGO)) ]

pt = proc.time()
seguros.comboInfo = findLinearCombos(seguros.nba.3[, .SD, .SDcols=columnasModeloNum])
proc.time() - pt

seguros.comboInfo

head(seguros.nba.3[, .SD, .SDcols=columnasModeloNum[c(95,92)]])
head(seguros.nba.3[, .SD, .SDcols=columnasModeloNum[c(96,93)]])
head(seguros.nba.3[, .SD, .SDcols=columnasModeloNum[c(97,94)]])
head(seguros.nba.3[, .SD, .SDcols=columnasModeloNum[c(103,102)]])
```

: ./scripts/03.3_1_AnalisisExploratorio_CombosLineales.R

```
setwd("C://Users//Sergio//Documents//Cursos//PUCP - Maestría Estadística//2014.2 -
    Ciclo 4//1 - Investigación en Estadística//Aplicación//Datos//Versión 3")
library(data.table)
library(caret)

memory.limit(40000)
pt = proc.time()
seg_tablon_15_total = fread("seg_tablon_15_total.csv", header=T, sep=",",
    stringsAsFactors=T)
proc.time() - pt
setkey(seg_tablon_15_total, CODCLAVECIC)

columnasModeloLogitTotal = colnames(seg_tablon_15_total)
columnasModeloLogitSB = c("SB_COD_DEPARTAMENTO_W", "SB_G_STR_ESTCIVIL_2", "SB_IMP_
    INTANG_CTS1", "SB_IMP_PASIVO_MED_EVOL", "SB_IND_CAH_W",
                          "SB_IND_INGREG_W", "SB_IND_REVOLVING_6M_W", "SB_NUM_CAMP_
                              EXITO", "SB_NUM_CTS", "SB_NUM_MES_ULTREC_CTS_LN", "SB_
                              NUM_OPE_GRUPO6_MED",
                          "SB_NUM_OPE_OUT_TNS_MED", "SB_NUM_OPEDEB_CVIS_MED", "SB_SEGM
                              _W", "SB_STR_MACROZONALIMA_W")
columnasModeloLogitSM = c("SM_COD_DEPARTAMENTO_W", "SM_g_profesion_W", "SM_IMP_CTS_MED
    _EVOL", "SM_IMP_INGREG_02", "SM_IMP_LIM_CRED_TOTAL",
                          "SM_IND_CAMP_SOLI_CANAL_AGEN_REA", "SM_IND_GASTREG_W", "SM_
                              IND_NSE_MED_W", "SM_NUM_CAMP_ENVIADAS", "SM_NUM_MES_
                              APEANT_CVIS_LN", "SM_NUM_MES_APEREC_CVIS_LN",
                          "SM_NUM_OPE_GRUPO1_MED", "SM_STR_MACROZONALIMA_W", "SM_woe_
                              NUM_OPE_GRUPO6_MED", "SM_woe_RAT_pasivo_total_Med2")
columnasModeloLogitSR = c("SR_COD_DEPARTAMENTO_W", "SR_G_COD_DEPARTAMENTO_1", "SR_g_
    profesion_W", "SR_IMP_LIMITE_TC", "SR_NUM_MES_APEREC_CSU_LN",
                          "SR_NUM_OPEDEB_CVIS_MED", "SR_NUM_TC", "SR_PCT_USO_CANAL_INT
                              ", "SR_RAT_activo_total_Med_Evol", "SR_SEGM_W", "SR_STR_
                              MACROZONALIMA_W", "SR_woe_IMP_OPE_GRUPO6_MED",
                          "SR_woe_IMP_OPE_TD", "SR_woe_IMP_PASIVO_MED_EVOL", "SR_woe_
                              NUM_CAMP_EXITO", "SR_woe_PCT_RCC_BCP_MED")
columnasModeloLogit = c("CODCLAVECIC", columnasModeloLogitSB, columnasModeloLogitSM,
    columnasModeloLogitSR)
columnasModeloLogitExcluir = columnasModeloLogitTotal[ -match(columnasModeloLogit,
    columnasModeloLogitTotal) ]

```

```
seg_tablon_15_total[, c(columnasModeloLogitExcluir) := NULL]
colnames(seg_tablon_15_total)

seguros.nba.4 = merge(seguros.nba.3, seg_tablon_15_total)
colnames(seguros.nba.4)

write.table(seguros.nba.4, file="SegurosNBA_4.csv", row.names=FALSE, col.names=TRUE,
    sep=",")
rm(seguros.nba.3)
gc()
```

: ./scripts/03.4_VariablesModeloLogit.R

```
#Seleccionar muestra para entrenamiento del modelo, validación y test
setwd("C://Users//Sergio//Documents//Cursos//PUCP - Maestría Estadística//2014.2 -
    Ciclo 4//1 - Investigación en Estadística//Aplicación//Datos//Versión 3")
library(data.table)

memory.limit(40000)
pt = proc.time()
seguros.nba.5 = fread("SegurosNBA_5.csv", header=T, sep=",", stringsAsFactors=T)
proc.time() - pt
setkey(seguros.nba.5, CODCLAVECIC)

#Selección de muestra de test y validación, con muestreo aleatorio simple
testvalIndex = sample(1:nrow(seguros.nba.5), 750000)
seguros.nba.test.val = seguros.nba.5[testvalIndex]
setkey(seguros.nba.test.val, CODCLAVECIC)
testIndex = sample(1:nrow(seguros.nba.test.val), 500000)
seguros.nba.test = seguros.nba.test.val[testIndex]
setkey(seguros.nba.test, CODCLAVECIC)
seguros.nba.val = seguros.nba.test.val[-testIndex]
setkey(seguros.nba.val, CODCLAVECIC)
write.table(seguros.nba.test, file="SegurosNBA_5_test.csv", row.names=FALSE, col.names
    =TRUE, sep=",")
write.table(seguros.nba.val, file="SegurosNBA_5_val.csv", row.names=FALSE, col.names=
    TRUE, sep=",")

#Selección de muestras de entrenamiento.
seguros.nba.train = seguros.nba.5[-testvalIndex]
seguros.nba.train_0 = seguros.nba.train[ SB_FLGVTA_PERSISTENCIA6M == 0 ]
seguros.nba.train_1 = seguros.nba.train[ SB_FLGVTA_PERSISTENCIA6M == 1 ]
n1
n0 = c(n1, n1*75/25, n1*97/5)
for (i in 1:length(n0)){
  trainIndex_0 = sample(1:nrow(seguros.nba.train_0), n0[i])
  trainIndex_1 = sample(1:nrow(seguros.nba.train_1), n1)
  seguros.nba.train.muestra_0 = seguros.nba.train_0[ trainIndex_0 ]
  seguros.nba.train.muestra_1 = seguros.nba.train_1[ trainIndex_1 ]
  seguros.nba.train.muestra = rbind(seguros.nba.train.muestra_0, seguros.nba.train.
      muestra_1)
  setkey(seguros.nba.train.muestra, CODCLAVECIC)
  write.table(seguros.nba.train.muestra, file=paste("SegurosNBA_5_",i,"_train.csv",
      sep=""), row.names=FALSE, col.names=TRUE, sep=",")
}
```

: ./scripts/04.1_SeleccionMuestrasModelo.R

```
#Lectura de la muestra de entrenamiento, de la muestra de validación y de la muestra
    de prueba
```

```
setwd("C://Users//Sergio//Documents//Cursos//PUCP - Maestría Estadística//2014.2 - Ciclo 4//1 - Investigación en Estadística//Aplicación//Datos//Versión 3")
library(data.table)
library(caret)
library(kernlab)
library(e1071)
library(doSNOW)
library(pROC)
library(plyr)
library(rpud)
library(caTools)
library(xtable)
library(tables)
memory.limit(40000)

#Leer muestra de entrenamiento M2: proporción de 1s a 0s de 50 a 50
pt = proc.time()
seguros.nba.train.1 = fread("SegurosNBA_5_1_train.csv", header=T, sep=",", stringsAsFactors=T)
proc.time() - pt
setkey(seguros.nba.modelo, CODCLAVECIC)
seguros.nba.train.1[, c("STR_TIPOBANCA","STR_MACROZONALIMA","STR_ESTCIVIL","IND_SEXO","IND_NIVSOCIOECO","IND_SEGMENTOBEX","IND_SEGMENTOPDH","IND_CAH","IND_CSU",
                        "IND_CVIS","IND_CANCREC_CVIS","IND_APEREC_CAH","IND_APEREC_CSU","IND_APEREC_CVIS","IND_CTS","IND_REC_APER_CTS","IND_REC_CANC_CTS","IND_PEFE","IND_PPE",
                        "IND_TC","IND_TC_PREMIUM","IND_TD","IND_REC_USO_IN_TRA","IND_REC_USO_OUT_TRA","IND_MP","IND_SEG","IND_PROGRAMA_PDH","IND_PROGRAMA_LANPASS","IND_INGREG",
                        "IND_GASTREG","IND_CAMP_EXITO_CANAL_AGEN_REA","IND_CAMP_SOLI_CANAL_AGEN_REA","IND_REVOLVING_6M","IND_EDAD_MED","IND_NSE_MED","GRP_TIPO_VIVIENDA_MAX","GRP_VIVIENDA_MAX","GRP_PARED_MAX","GRP_PISO_MAX",
                        "GRP_TIPO_ABASTECIMIENTO_MAX","GRP_SERV_HIGIENICOS_MAX","GRP_ENERGIA_MAX","GRP_LENGUA_MAX","IND_NIVEL_ESTUDIOS_MED","GRP_ESTADO_CIVIL_MAX","IND_DENSI_FARM_DECIL","IND_DENSI_REST_DECIL","SEGM","IND_PAFG_CAH",
                        "IND_PAXS_CSU","IND_PAFG_CSU","IND_PAXS_CTS","IND_PAFG_CTS","IND_PAXS_HIP_AyB","IND_PAXS_HIP_CyD","IND_PAXS_PPE","IND_PAUP_PPE","IND_PAFG_PPE","IND_PAXS_TC",
                        "IND_PAXS_BT","IND_PAXS_EP","IND_PAUP_REV","COD_DEPARTAMENTO","RCC_TIPCLASIFRIESGO",
                        "SB_FLGVTA","SB_FLGVTA_PERSISTENCIA6M","SM_FLGVTA","SM_FLGVTA_PERSISTENCIA6M","SR_FLGVTA","SR_FLGVTA_PERSISTENCIA6M") :=
                      list(as.factor(STR_TIPOBANCA),as.factor(STR_MACROZONALIMA),as.factor(STR_ESTCIVIL),as.factor(IND_SEXO),as.factor(IND_NIVSOCIOECO),as.factor(IND_SEGMENTOBEX),as.factor(IND_SEGMENTOPDH),as.factor(IND_CAH),as.factor(IND_CSU),
                           as.factor(IND_CVIS),as.factor(IND_CANCREC_CVIS),as.factor(IND_APEREC_CAH),as.factor(IND_APEREC_CSU),as.factor(IND_APEREC_CVIS),as.factor(IND_CTS),as.factor(IND_REC_APER_CTS),as.factor(IND_REC_CANC_CTS),as.factor(IND_PEFE),as.factor(IND_PPE),
                           as.factor(IND_TC),as.factor(IND_TC_PREMIUM),as.factor(IND_TD),as.factor(IND_REC_USO_IN_TRA),as.factor(IND_REC_USO_OUT_TRA),as.factor(IND_MP),as.factor(IND_SEG),as.factor(IND_PROGRAMA_PDH),as.factor(IND_PROGRAMA_LANPASS),as.factor(IND_INGREG),
                           as.factor(IND_GASTREG),as.factor(IND_CAMP_EXITO_CANAL_AGEN_REA),as.factor(IND_CAMP_SOLI_CANAL_AGEN_REA),as.factor(
```

```
                              IND_REVOLVING_6M),as.factor(IND_EDAD_MED),as.factor(IND_
                              NSE_MED),as.factor(GRP_TIPO_VIVIENDA_MAX),as.factor(GRP_
                              VIVIENDA_MAX),as.factor(GRP_PARED_MAX),as.factor(GRP_
                              PISO_MAX),
33                          as.factor(GRP_TIPO_ABASTECIMIENTO_MAX),as.factor(GRP_SERV_
                              HIGIENICOS_MAX),as.factor(GRP_ENERGIA_MAX),as.factor(GRP
                              _LENGUA_MAX),as.factor(IND_NIVEL_ESTUDIOS_MED),as.factor
                              (GRP_ESTADO_CIVIL_MAX),as.factor(IND_DENSI_FARM_DECIL),
                              as.factor(IND_DENSI_REST_DECIL),as.factor(SEGM),as.
                              factor(IND_PAFG_CAH),
34                          as.factor(IND_PAXS_CSU),as.factor(IND_PAFG_CSU),as.factor(
                              IND_PAXS_CTS),as.factor(IND_PAFG_CTS),as.factor(IND_PAXS
                              _HIP_AyB),as.factor(IND_PAXS_HIP_CyD),as.factor(IND_PAXS
                              _PPE),as.factor(IND_PAUP_PPE),as.factor(IND_PAFG_PPE),as
                              .factor(IND_PAXS_TC),
35                          as.factor(IND_PAXS_BT),as.factor(IND_PAXS_EP),as.factor(IND_
                              PAUP_REV),as.factor(COD_DEPARTAMENTO),as.factor(RCC_
                              TIPCLASIFRIESGO),
36                          as.factor(SB_FLGVTA),as.factor(SB_FLGVTA_PERSISTENCIA6M),as.
                              factor(SM_FLGVTA),as.factor(SM_FLGVTA_PERSISTENCIA6M),as
                              .factor(SR_FLGVTA),as.factor(SR_FLGVTA_PERSISTENCIA6M))]
37 seguros.nba.train.1[, c("SB_COD_DEPARTAMENTO_W", "SB_G_STR_ESTCIVIL_2", "SB_IND_CAH_W"
       , "SB_IND_INGREG_W", "SB_IND_REVOLVING_6M_W", "SB_SEGM_W", "SB_STR_MACROZONALIMA_W
       ") :=
38                  list(as.factor(SB_COD_DEPARTAMENTO_W), as.factor(SB_G_STR_ESTCIVIL_2),
                         as.factor(SB_IND_CAH_W), as.factor(SB_IND_INGREG_W), as.factor(SB
                        _IND_REVOLVING_6M_W), as.factor(SB_SEGM_W), as.factor(SB_STR_
                        MACROZONALIMA_W)) ]
39
40 #Leer muestra de entrenamiento M2: proporción de 1s a 0s de 25 a 75
41 pt = proc.time()
42 seguros.nba.train.2 = fread("SegurosNBA_5_2_train.csv", header=T, sep=",",
       stringsAsFactors=T)
43 proc.time() - pt
44 setkey(seguros.nba.modelo, CODCLAVECIC)
45 seguros.nba.train.2[, c("STR_TIPOBANCA","STR_MACROZONALIMA","STR_ESTCIVIL","IND_SEXO",
       "IND_NIVSOCIOECO","IND_SEGMENTOBEX","IND_SEGMENTOPDH","IND_CAH","IND_CSU",
46                        "IND_CVIS","IND_CANCREC_CVIS","IND_APEREC_CAH","IND_APEREC_CSU
                              ","IND_APEREC_CVIS","IND_CTS","IND_REC_APER_CTS","IND_REC_
                              CANC_CTS","IND_PEFE","IND_PPE",
47                        "IND_TC","IND_TC_PREMIUM","IND_TD","IND_REC_USO_IN_TRA","IND_
                              REC_USO_OUT_TRA","IND_MP","IND_SEG","IND_PROGRAMA_PDH","
                              IND_PROGRAMA_LANPASS","IND_INGREG",
48                        "IND_GASTREG","IND_CAMP_EXITO_CANAL_AGEN_REA","IND_CAMP_SOLI_
                              CANAL_AGEN_REA","IND_REVOLVING_6M","IND_EDAD_MED","IND_NSE
                              _MED","GRP_TIPO_VIVIENDA_MAX","GRP_VIVIENDA_MAX","GRP_
                              PARED_MAX","GRP_PISO_MAX",
49                        "GRP_TIPO_ABASTECIMIENTO_MAX","GRP_SERV_HIGIENICOS_MAX","GRP_
                              ENERGIA_MAX","GRP_LENGUA_MAX","IND_NIVEL_ESTUDIOS_MED","
                              GRP_ESTADO_CIVIL_MAX","IND_DENSI_FARM_DECIL","IND_DENSI_
                              REST_DECIL","SEGM","IND_PAFG_CAH",
50                        "IND_PAXS_CSU","IND_PAFG_CSU","IND_PAXS_CTS","IND_PAFG_CTS","
                              IND_PAXS_HIP_AyB","IND_PAXS_HIP_CyD","IND_PAXS_PPE","IND_
                              PAUP_PPE","IND_PAFG_PPE","IND_PAXS_TC",
51                        "IND_PAXS_BT","IND_PAXS_EP","IND_PAUP_REV","COD_DEPARTAMENTO",
                              "RCC_TIPCLASIFRIESGO",
52                        "SB_FLGVTA","SB_FLGVTA_PERSISTENCIA6M","SM_FLGVTA","SM_FLGVTA_
                              PERSISTENCIA6M","SR_FLGVTA","SR_FLGVTA_PERSISTENCIA6M") :=
53                      list(as.factor(STR_TIPOBANCA),as.factor(STR_MACROZONALIMA),as.
                              factor(STR_ESTCIVIL),as.factor(IND_SEXO),as.factor(IND_
```

```
                          NIVSOCIOECO),as.factor(IND_SEGMENTOBEX),as.factor(IND_
                          SEGMENTOPDH),as.factor(IND_CAH),as.factor(IND_CSU),
54                         as.factor(IND_CVIS),as.factor(IND_CANCREC_CVIS),as.factor(
                               IND_APEREC_CAH),as.factor(IND_APEREC_CSU),as.factor(IND
                               _APEREC_CVIS),as.factor(IND_CTS),as.factor(IND_REC_APER
                               _CTS),as.factor(IND_REC_CANC_CTS),as.factor(IND_PEFE),
                               as.factor(IND_PPE),
55                         as.factor(IND_TC),as.factor(IND_TC_PREMIUM),as.factor(IND_
                               TD),as.factor(IND_REC_USO_IN_TRA),as.factor(IND_REC_USO
                               _OUT_TRA),as.factor(IND_MP),as.factor(IND_SEG),as.
                               factor(IND_PROGRAMA_PDH),as.factor(IND_PROGRAMA_LANPASS
                               ),as.factor(IND_INGREG),
56                         as.factor(IND_GASTREG),as.factor(IND_CAMP_EXITO_CANAL_AGEN_
                               REA),as.factor(IND_CAMP_SOLI_CANAL_AGEN_REA),as.factor(
                               IND_REVOLVING_6M),as.factor(IND_EDAD_MED),as.factor(IND
                               _NSE_MED),as.factor(GRP_TIPO_VIVIENDA_MAX),as.factor(
                               GRP_VIVIENDA_MAX),as.factor(GRP_PARED_MAX),as.factor(
                               GRP_PISO_MAX),
57                         as.factor(GRP_TIPO_ABASTECIMIENTO_MAX),as.factor(GRP_SERV_
                               HIGIENICOS_MAX),as.factor(GRP_ENERGIA_MAX),as.factor(
                               GRP_LENGUA_MAX),as.factor(IND_NIVEL_ESTUDIOS_MED),as.
                               factor(GRP_ESTADO_CIVIL_MAX),as.factor(IND_DENSI_FARM_
                               DECIL),as.factor(IND_DENSI_REST_DECIL),as.factor(SEGM),
                               as.factor(IND_PAFG_CAH),
58                         as.factor(IND_PAXS_CSU),as.factor(IND_PAFG_CSU),as.factor(
                               IND_PAXS_CTS),as.factor(IND_PAFG_CTS),as.factor(IND_
                               PAXS_HIP_AyB),as.factor(IND_PAXS_HIP_CyD),as.factor(IND
                               _PAXS_PPE),as.factor(IND_PAUP_PPE),as.factor(IND_PAFG_
                               PPE),as.factor(IND_PAXS_TC),
59                         as.factor(IND_PAXS_BT),as.factor(IND_PAXS_EP),as.factor(IND
                               _PAUP_REV),as.factor(COD_DEPARTAMENTO),as.factor(RCC_
                               TIPCLASIFRIESGO),
60                         as.factor(SB_FLGVTA),as.factor(SB_FLGVTA_PERSISTENCIA6M),as
                               .factor(SM_FLGVTA),as.factor(SM_FLGVTA_PERSISTENCIA6M),
                               as.factor(SR_FLGVTA),as.factor(SR_FLGVTA_PERSISTENCIA6M
                               ))]
61 seguros.nba.train.2[, c("SB_COD_DEPARTAMENTO_W", "SB_G_STR_ESTCIVIL_2", "SB_IND_CAH_W"
       , "SB_IND_INGREG_W", "SB_IND_REVOLVING_6M_W", "SB_SEGM_W", "SB_STR_MACROZONALIMA_W
       ") :=
62                        list(as.factor(SB_COD_DEPARTAMENTO_W), as.factor(SB_G_STR_
                               ESTCIVIL_2), as.factor(SB_IND_CAH_W), as.factor(SB_IND_
                               INGREG_W), as.factor(SB_IND_REVOLVING_6M_W), as.factor(SB_
                               SEGM_W), as.factor(SB_STR_MACROZONALIMA_W)) ]
63
64 #Leer muestra de entrenamiento M3: proporción de 1s a 0s de 05 a 95
65 pt = proc.time()
66 seguros.nba.train.3 = fread("SegurosNBA_5_3_train.csv", header=T, sep=",",
       stringsAsFactors=T)
67 proc.time() - pt
68 setkey(seguros.nba.modelo, CODCLAVECIC)
69 seguros.nba.train.3[, c("STR_TIPOBANCA","STR_MACROZONALIMA","STR_ESTCIVIL","IND_SEXO",
       "IND_NIVSOCIOECO","IND_SEGMENTOBEX","IND_SEGMENTOPDH","IND_CAH","IND_CSU",
70                        "IND_CVIS","IND_CANCREC_CVIS","IND_APEREC_CAH","IND_APEREC_CSU
                               ","IND_APEREC_CVIS","IND_CTS","IND_REC_APER_CTS","IND_REC_
                               CANC_CTS","IND_PEFE","IND_PPE",
71                        "IND_TC","IND_TC_PREMIUM","IND_TD","IND_REC_USO_IN_TRA","IND_
                               REC_USO_OUT_TRA","IND_MP","IND_SEG","IND_PROGRAMA_PDH","
                               IND_PROGRAMA_LANPASS","IND_INGREG",
72                        "IND_GASTREG","IND_CAMP_EXITO_CANAL_AGEN_REA","IND_CAMP_SOLI_
                               CANAL_AGEN_REA","IND_REVOLVING_6M","IND_EDAD_MED","IND_NSE
```

```
                          _MED","GRP_TIPO_VIVIENDA_MAX","GRP_VIVIENDA_MAX","GRP_
                          PARED_MAX","GRP_PISO_MAX",
73                      "GRP_TIPO_ABASTECIMIENTO_MAX","GRP_SERV_HIGIENICOS_MAX","GRP_
                          ENERGIA_MAX","GRP_LENGUA_MAX","IND_NIVEL_ESTUDIOS_MED","
                          GRP_ESTADO_CIVIL_MAX","IND_DENSI_FARM_DECIL","IND_DENSI_
                          REST_DECIL","SEGM","IND_PAFG_CAH",
74                      "IND_PAXS_CSU","IND_PAFG_CSU","IND_PAXS_CTS","IND_PAFG_CTS","
                          IND_PAXS_HIP_AyB","IND_PAXS_HIP_CyD","IND_PAXS_PPE","IND_
                          PAUP_PPE","IND_PAFG_PPE","IND_PAXS_TC",
75                      "IND_PAXS_BT","IND_PAXS_EP","IND_PAUP_REV","COD_DEPARTAMENTO",
                          "RCC_TIPCLASIFRIESGO",
76                      "SB_FLGVTA","SB_FLGVTA_PERSISTENCIA6M","SM_FLGVTA","SM_FLGVTA_
                          PERSISTENCIA6M","SR_FLGVTA","SR_FLGVTA_PERSISTENCIA6M") :=
77                    list(as.factor(STR_TIPOBANCA),as.factor(STR_MACROZONALIMA),as.
                          factor(STR_ESTCIVIL),as.factor(IND_SEXO),as.factor(IND_
                          NIVSOCIOECO),as.factor(IND_SEGMENTOBEX),as.factor(IND_
                          SEGMENTOPDH),as.factor(IND_CAH),as.factor(IND_CSU),
78                         as.factor(IND_CVIS),as.factor(IND_CANCREC_CVIS),as.factor(
                              IND_APEREC_CAH),as.factor(IND_APEREC_CSU),as.factor(IND
                              _APEREC_CVIS),as.factor(IND_CTS),as.factor(IND_REC_APER
                              _CTS),as.factor(IND_REC_CANC_CTS),as.factor(IND_PEFE),
                              as.factor(IND_PPE),
79                         as.factor(IND_TC),as.factor(IND_TC_PREMIUM),as.factor(IND_
                              TD),as.factor(IND_REC_USO_IN_TRA),as.factor(IND_REC_USO
                              _OUT_TRA),as.factor(IND_MP),as.factor(IND_SEG),as.
                              factor(IND_PROGRAMA_PDH),as.factor(IND_PROGRAMA_LANPASS
                              ),as.factor(IND_INGREG),
80                         as.factor(IND_GASTREG),as.factor(IND_CAMP_EXITO_CANAL_AGEN_
                              REA),as.factor(IND_CAMP_SOLI_CANAL_AGEN_REA),as.factor(
                              IND_REVOLVING_6M),as.factor(IND_EDAD_MED),as.factor(IND
                              _NSE_MED),as.factor(GRP_TIPO_VIVIENDA_MAX),as.factor(
                              GRP_VIVIENDA_MAX),as.factor(GRP_PARED_MAX),as.factor(
                              GRP_PISO_MAX),
81                         as.factor(GRP_TIPO_ABASTECIMIENTO_MAX),as.factor(GRP_SERV_
                              HIGIENICOS_MAX),as.factor(GRP_ENERGIA_MAX),as.factor(
                              GRP_LENGUA_MAX),as.factor(IND_NIVEL_ESTUDIOS_MED),as.
                              factor(GRP_ESTADO_CIVIL_MAX),as.factor(IND_DENSI_FARM_
                              DECIL),as.factor(IND_DENSI_REST_DECIL),as.factor(SEGM),
                              as.factor(IND_PAFG_CAH),
82                         as.factor(IND_PAXS_CSU),as.factor(IND_PAFG_CSU),as.factor(
                              IND_PAXS_CTS),as.factor(IND_PAFG_CTS),as.factor(IND_
                              PAXS_HIP_AyB),as.factor(IND_PAXS_HIP_CyD),as.factor(IND
                              _PAXS_PPE),as.factor(IND_PAUP_PPE),as.factor(IND_PAFG_
                              PPE),as.factor(IND_PAXS_TC),
83                         as.factor(IND_PAXS_BT),as.factor(IND_PAXS_EP),as.factor(IND
                              _PAUP_REV),as.factor(COD_DEPARTAMENTO),as.factor(RCC_
                              TIPCLASIFRIESGO),
84                         as.factor(SB_FLGVTA),as.factor(SB_FLGVTA_PERSISTENCIA6M),as
                              .factor(SM_FLGVTA),as.factor(SM_FLGVTA_PERSISTENCIA6M),
                              as.factor(SR_FLGVTA),as.factor(SR_FLGVTA_PERSISTENCIA6M
                              ))]
85 seguros.nba.train.3[, c("SB_COD_DEPARTAMENTO_W", "SB_G_STR_ESTCIVIL_2", "SB_IND_CAH_W"
       , "SB_IND_INGREG_W", "SB_IND_REVOLVING_6M_W", "SB_SEGM_W", "SB_STR_MACROZONALIMA_W
       ") :=
86                        list(as.factor(SB_COD_DEPARTAMENTO_W), as.factor(SB_G_STR_
                              ESTCIVIL_2), as.factor(SB_IND_CAH_W), as.factor(SB_IND_
                              INGREG_W), as.factor(SB_IND_REVOLVING_6M_W), as.factor(SB_
                              SEGM_W), as.factor(SB_STR_MACROZONALIMA_W)) ]
87
88 #Leer muestra de validación
```

```
pt = proc.time()
seguros.nba.val = fread("SegurosNBA_5_val.csv", header=T, sep=",", stringsAsFactors=T)
proc.time() - pt
setkey(seguros.nba.val, CODCLAVECIC)

#Leer muestra de test
pt = proc.time()
seguros.nba.test = fread("SegurosNBA_5_test.csv", header=T, sep=",", stringsAsFactors=
    T)
proc.time() - pt
setkey(seguros.nba.test, CODCLAVECIC)

#Configurar hiperparámetros para SVM
seguros.svm.C = c(1, 2.5, 5, 10)
seguros.svm.wts = list()
pesos = c(0.50, 0.50)
names(pesos) = c("0", "1")
seguros.svm.wts[[1]] = pesos
pesos[1:2] = c(0.25, 0.75)
seguros.svm.wts[[2]] = pesos
pesos[1:2] = c(0.05, 0.95)
seguros.svm.wts[[3]] = pesos

#Armar muestras para entrenamiento
n = 22000
seguros.nba.train.muestra = list()
seguros.nba.train.muestra[[1]] = seguros.nba.train.1[sample(1:nrow(seguros.nba.train
    .1), n)]
seguros.nba.train.muestra[[2]] = seguros.nba.train.2[sample(1:nrow(seguros.nba.train
    .2), n)]
seguros.nba.train.muestra[[3]] = seguros.nba.train.3[sample(1:nrow(seguros.nba.train
    .3), n)]
```

: ./scripts/05.0_SVM_Lectura_TrainValTest.R

```
#Configurar variables para cada escenario, para los modelos SVM y Logit
varModeloLogitSB = c("SB_COD_DEPARTAMENTO_W", "SB_G_STR_ESTCIVIL_2", "SB_IMP_INTANG_
    CTS1", "SB_IMP_PASIVO_MED_EVOL", "SB_IND_CAH_W",
                         "SB_IND_INGREG_W", "SB_IND_REVOLVING_6M_W", "SB_NUM_CAMP_
                             EXITO", "SB_NUM_CTS", "SB_NUM_MES_ULTREC_CTS_LN", "SB_
                             NUM_OPE_GRUPO6_MED",
                         "SB_NUM_OPE_OUT_TNS_MED", "SB_NUM_OPEDEB_CVIS_MED", "SB_SEGM
                             _W", "SB_STR_MACROZONALIMA_W")
varModeloExcluir = c("CODCLAVECIC","PI_SB","PI_SM","PI_SR","PI_AJUSTADO_SB","PI_
    AJUSTADO_SM","PI_AJUSTADO_SR",
                         "SB_FLGVTA","SB_FLGVTA_PERSISTENCIA6M","SM_FLGVTA","SM_
                             FLGVTA_PERSISTENCIA6M","SR_FLGVTA","SR_FLGVTA_
                             PERSISTENCIA6M", varModeloLogitSB)
varModeloTotal = colnames(seguros.nba.train.1)
varModeloTotal = varModeloTotal[ -match(varModeloExcluir, varModeloTotal) ]
varModeloTotal_Y = c(varModeloTotal, "SB_FLGVTA_PERSISTENCIA6M")


#Variables escenario A (1): variables numéricas productos de análisis factoral
varModelo.E1.Num = c("FR_OPE_TD_MP","FR_NUM_OPE_TD_MP","FR_IMP_CTS_1","FR_IMP_CTS_2","
    FR_IMP_RCC_TC","FR_IMP_RCC","FR_IMP_VISTA_CAH","FR_NUM_PPE","FR_ACTIVO","FR_CVIS_
    INGGAST","FR_PCT_CTS","FR_RAT_PASIVO_FMU","FR_RAT_PASIVO_FMU_EVOL")
varModelo.E1.Num_Y = c(varModelo.E1.Num, "SB_FLGVTA_PERSISTENCIA6M")

#Variables escenario 1 (2): todas las variables numéricas
```

```
varModelo.E2.Num = character()
j = 0
for (i in 1:length(varModeloTotal)){
  if ( !is.factor(seguros.nba.train.1[, .SD, .SDcols=varModeloTotal[i]][[1]]) ){
    j = j + 1
    varModelo.E2.Num[j] = varModeloTotal[i]
  }
}
varModelo.E2.Num_Y = c(varModelo.E2.Num, "SB_FLGVTA_PERSISTENCIA6M")
varModelo.E2.Factor = varModeloTotal[-match(varModelo.E2.Num, varModeloTotal)]

#Variables escenario B (3): variables numéricas productos de análisis factoral más variables categóricas
varModelo.E3.Tot = c(varModelo.E1.Num, varModelo.E2.Factor)
varModelo.E3.Tot_Y = c(varModelo.E3.Tot, "SB_FLGVTA_PERSISTENCIA6M")

#Variables escenario 2 (4): todas las variables numéricas y categóricas
varModelo.E4.Tot = varModeloTotal
varModelo.E4.Tot_Y = varModeloTotal_Y

#Variables escenario C (5): variables seleccionadas para el modelo Logit "óptimo"
varModelo.E5 = c("COD_DEPARTAMENTO", "SB_G_STR_ESTCIVIL_2", "SB_IMP_INTANG_CTS1", "SB_IMP_PASIVO_MED_EVOL", "IND_CAH", "IND_INGREG", "IND_REVOLVING_6M", "NUM_CAMP_EXITO",
                 "SB_NUM_CTS", "NUM_MES_ULTREC_CTS_LN", "NUM_OPE_GRUPO6_MED", "SB_NUM_OPE_OUT_TNS_MED", "SB_NUM_OPEDEB_CVIS_MED", "SEGM", "STR_MACROZONALIMA")
varModelo.E5_Y = c(varModelo.E5, "SB_FLGVTA_PERSISTENCIA6M")
```

: ./scripts/05.1_SVM_VariablesPorEscenario.R

```
#Revisar tiempos de entrenamiento de las librerías SVM: e1071, kernlab, rpud
n = 10000
m = 10

seguros.svm.fit.e1071 = list()
seguros.svm.fit.kernlab = list()
seguros.svm.fit.rpud = list()
seguros.svm.fit.tiempos = matrix(nrow=10, ncol=3)
colnames(seguros.svm.fit.tiempos) = c("e1071", "kernlab", "rpud")
for (i in 1:m){
  seguros.nba.train.muestra.0 = seguros.nba.train.2[sample(1:nrow(seguros.nba.train.2), n)]
  pt = proc.time()
  seguros.svm.fit.e1071[[i]] = svm(SB_FLGVTA_PERSISTENCIA6M ~ ., data=seguros.nba.train.muestra.0[, .SD, .SDcols=varModelo.E4.Tot_Y], type="C-classification",
                                   scale=TRUE, kernel="radial", cost=1, class.weights=seguros.svm.wts[[3]])
  seguros.svm.fit.tiempos[i,1] = (proc.time() - pt)[3]

  pt = proc.time()
  seguros.svm.fit.kernlab[[i]] = ksvm(SB_FLGVTA_PERSISTENCIA6M ~ ., data=seguros.nba.train.muestra.0[, .SD, .SDcols=varModelo.E4.Tot_Y], type="C-svc",
                                      scaled=TRUE, kernel="rbfdot", kpar="automatic",
                                          C=1, class.weights=seguros.svm.wts[[3]],
                                          prob.model=FALSE)
  seguros.svm.fit.tiempos[i,2] = (proc.time() - pt)[3]

  pt = proc.time()
  seguros.svm.fit.rpud[[i]] = rpusvm(SB_FLGVTA_PERSISTENCIA6M ~ ., data=seguros.nba.
```

```
      train.muestra.0[, .SD, .SDcols=varModelo.E4.Tot_Y], type="C-classification",
24                                    scale=TRUE, kernel="radial", cost=1, class.
                                          weights=seguros.svm.wts[[3]])
25   seguros.svm.fit.tiempos[i,3] = (proc.time() - pt)[3]
26 }
27
28 seguros.nba.val.muestra.0 = seguros.nba.val[sample(1:nrow(seguros.nba.val), n)]
29 seguros.svm.pred.e1071 = list()
30 seguros.svm.pred.kernlab = list()
31 seguros.svm.pred.rpud = list()
32 seguros.svm.pred.tiempos = matrix(nrow=10, ncol=3)
33 colnames(seguros.svm.pred.tiempos) = c("e1071", "kernlab", "rpud")
34 for (i in 1:m){
35   pt = proc.time()
36   seguros.svm.pred.e1071[[i]] = predict(seguros.svm.fit.e1071[[i]], seguros.nba.val.
       muestra.0, decision.values=TRUE)
37   seguros.svm.pred.tiempos[i,1] = (proc.time() - pt)[3]
38
39   pt = proc.time()
40   seguros.svm.pred.kernlab[[i]] = predict(seguros.svm.fit.kernlab[[i]], seguros.nba.
       val.muestra.0, type="decision")
41   seguros.svm.pred.tiempos[i,2] = (proc.time() - pt)[3]
42
43   pt = proc.time()
44   seguros.svm.pred.rpud[[i]] = predict(seguros.svm.fit.rpud[[i]], seguros.nba.val.
       muestra.0, decision.values=TRUE)
45   seguros.svm.pred.tiempos[i,3] = (proc.time() - pt)[3]
46 }
47
48 #Revisar tiempos de entrenamiento
49 seguros.svm.fit.tiempos
50 seguros.svm.fit.tiempos.res = cbind(apply(seguros.svm.fit.tiempos, 2, sum), apply(
     seguros.svm.fit.tiempos, 2, mean), apply(seguros.svm.fit.tiempos, 2, sd))
51 colnames(seguros.svm.fit.tiempos.res) = c("Total", "Media", "Desv. Est.")
52 seguros.svm.fit.tiempos.res
53
54 #Revisar tiempos de predicción
55 seguros.svm.pred.tiempos
56 seguros.svm.pred.tiempos.res = cbind(apply(seguros.svm.pred.tiempos, 2, sum), apply(
     seguros.svm.pred.tiempos, 2, mean), apply(seguros.svm.pred.tiempos, 2, sd))
57 colnames(seguros.svm.pred.tiempos.res) = c("Total", "Media", "Desv. Est.")
58 seguros.svm.pred.tiempos.res
59
60 xtable(seguros.svm.fit.tiempos.res)
61 xtable(seguros.svm.pred.tiempos.res)
```

: ./scripts/05.2_SVM_EvaluacionTiemposLibreriasSVM.R

```
1  #Escenario 1 (2): SVM con todas las variables numéricas
2  seguros.svm.fit.E2.list = list()
3  tiempos.svm.fit.E2.list = list()
4  seguros.svm.pred.E2.list = list()
5  tiempos.svm.pred.E2.list = list()
6  seguros.logit.fit.E2 = list()
7  seguros.logit.pred.val.E2 = list()
8
9  for (k in 1:3){
10   seguros.svm.sigma = sigest(SB_FLGVTA_PERSISTENCIA6M ~ ., data=seguros.nba.train.
       muestra[[k]][, .SD, .SDcols=varModelo.E2.Num_Y])
11   seguros.Mi.svm.fit.E2.K1.list = list()
```

```
12  tiempos.Mi.svm.fit.E2.K1 = numeric()
13  for (i in 1:length(seguros.svm.C)){
14    pti = proc.time()
15    seguros.Mi.svm.fit.E2.K1.list[[i]] = rpusvm(SB_FLGVTA_PERSISTENCIA6M ~ ., data=
          seguros.nba.train.muestra[[k]][, .SD, .SDcols=varModelo.E2.Num_Y], type="C-
          classification",
16                                                scale=TRUE, kernel="radial", cost=
                                                      seguros.svm.C[i], gamma=seguros.
                                                      svm.sigma[3], class.weights=
                                                      seguros.svm.wts[[k]], decision.
                                                      values=TRUE)
17    tiempos.Mi.svm.fit.E2.K1[i] = (proc.time() - pti)[3]
18  }
19  seguros.Mi.svm.fit.E2.K2.list = list()
20  tiempos.Mi.svm.fit.E2.K2 = numeric()
21  for (i in 1:length(seguros.svm.C)){
22    pti = proc.time()
23    seguros.Mi.svm.fit.E2.K2.list[[i]] = rpusvm(SB_FLGVTA_PERSISTENCIA6M ~ ., data=
          seguros.nba.train.muestra[[k]][, .SD, .SDcols=varModelo.E2.Num_Y], type="C-
          classification",
24                                                scale=TRUE, kernel="polynomial", cost=
                                                      seguros.svm.C[i], class.weights=
                                                      seguros.svm.wts[[k]], decision.
                                                      values=TRUE)
25    tiempos.Mi.svm.fit.E2.K2[i] = (proc.time() - pti)[3]
26  }
27  seguros.Mi.svm.fit.E2.K3.list = list()
28  tiempos.Mi.svm.fit.E2.K3 = numeric()
29  for (i in 1:length(seguros.svm.C)){
30    pti = proc.time()
31    seguros.Mi.svm.fit.E2.K3.list[[i]] = rpusvm(SB_FLGVTA_PERSISTENCIA6M ~ ., data=
          seguros.nba.train.muestra[[k]][, .SD, .SDcols=varModelo.E2.Num_Y], type="C-
          classification",
32                                                scale=TRUE, kernel="linear", cost=
                                                      seguros.svm.C[i], class.weights=
                                                      seguros.svm.wts[[k]], decision.
                                                      values=TRUE)
33    tiempos.Mi.svm.fit.E2.K3[i] = (proc.time() - pti)[3]
34  }
35  seguros.Mi.svm.pred.E2.K1.list = list()
36  tiempos.Mi.svm.pred.E2.K1 = numeric()
37  for (i in 1:length(seguros.Mi.svm.fit.E2.K1.list)){
38    pti = proc.time()
39    seguros.Mi.svm.pred.E2.K1.list[[i]] = predict(seguros.Mi.svm.fit.E2.K1.list[[i]],
          seguros.nba.val, decision.values=TRUE)
40    tiempos.Mi.svm.pred.E2.K1[i] = (proc.time() - pti)[3]
41  }
42  seguros.Mi.svm.pred.E2.K2.list = list()
43  tiempos.Mi.svm.pred.E2.K2 = numeric()
44  for (i in 1:length(seguros.Mi.svm.fit.E2.K2.list)){
45    pti = proc.time()
46    seguros.Mi.svm.pred.E2.K2.list[[i]] = predict(seguros.Mi.svm.fit.E2.K2.list[[i]],
          seguros.nba.val, decision.values=TRUE)
47    tiempos.Mi.svm.pred.E2.K2[i] = (proc.time() - pti)[3]
48  }
49  seguros.Mi.svm.pred.E2.K3.list = list()
50  tiempos.Mi.svm.pred.E2.K3 = numeric()
51  for (i in 1:length(seguros.Mi.svm.fit.E2.K3.list)){
52    pti = proc.time()
53    seguros.Mi.svm.pred.E2.K3.list[[i]] = predict(seguros.Mi.svm.fit.E2.K3.list[[i]],
```

```
        seguros.nba.val, decision.values=TRUE)
    tiempos.Mi.svm.pred.E2.K3[i] = (proc.time() - pti)[3]
  }
  seguros.svm.fit.E2.list[[k]] = append(append(seguros.Mi.svm.fit.E2.K1.list, seguros.
      Mi.svm.fit.E2.K2.list), seguros.Mi.svm.fit.E2.K3.list)
  tiempos.svm.fit.E2.list[[k]] = cbind(tiempos.Mi.svm.fit.E2.K1, tiempos.Mi.svm.fit.E2
      .K2, tiempos.Mi.svm.fit.E2.K3)
  colnames(tiempos.svm.fit.E2.list[[k]]) = c("K1 Radial", "K2 Polynomial", "K3 Linear"
      )
  seguros.svm.pred.E2.list[[k]] = append(append(seguros.Mi.svm.pred.E2.K1.list,
      seguros.Mi.svm.pred.E2.K2.list), seguros.Mi.svm.pred.E2.K3.list)
  tiempos.svm.pred.E2.list[[k]] = cbind(tiempos.Mi.svm.pred.E2.K1, tiempos.Mi.svm.pred
      .E2.K2, tiempos.Mi.svm.pred.E2.K3)
  colnames(tiempos.svm.pred.E2.list[[k]]) = c("K1 Radial", "K2 Polynomial", "K3 Linear
      ")

  #Ajuste y predicción del modelo logit
  seguros.logit.fit.E2[[k]] = glm(SB_FLGVTA_PERSISTENCIA6M ~ ., family=binomial, data=
      seguros.nba.train.muestra[[k]][, .SD, .SDcols=varModelo.E2.Num_Y])
  seguros.logit.pred.val.E2[[k]] = predict(seguros.logit.fit.E2[[k]], seguros.nba.val
      [, .SD, .SDcols=varModelo.E2.Num], type="response")
}

saveRDS(seguros.svm.fit.E2.list, "Models//seguros.svm.fit.E2.list.RDS")
saveRDS(seguros.svm.pred.E2.list, "Models//seguros.svm.pred.E2.list.RDS")
saveRDS(seguros.logit.fit.E2, "Models//seguros.logit.fit.E2.RDS")
#seguros.svm.fit.E2.list = readRDS("Models//seguros.svm.fit.E2.list.RDS")
#seguros.svm.pred.E2.list = readRDS("Models//seguros.svm.pred.E2.list.RDS")
#seguros.logit.fit.E2 = readRDS("Models//seguros.logit.fit.E2.RDS")
#seguros.logit.pred.val.E2 = list()
#seguros.logit.pred.val.E2[[1]] = predict(seguros.logit.fit.E2[[1]], seguros.nba.val[,
    .SD, .SDcols=varModelo.E2.Num], type="response")
#seguros.logit.pred.val.E2[[2]] = predict(seguros.logit.fit.E2[[2]], seguros.nba.val[,
    .SD, .SDcols=varModelo.E2.Num], type="response")
#seguros.logit.pred.val.E2[[3]] = predict(seguros.logit.fit.E2[[3]], seguros.nba.val[,
    .SD, .SDcols=varModelo.E2.Num], type="response")

seguros.svm.roc.E2.list = list()
seguros.smv.result.E2 = matrix(nrow=1, ncol=7)
colnames(seguros.smv.result.E2) = c("Muestra", "Kernel", "C", "gamma", "d", "delta", "
    AUC")
seguros.smv.result.E2 = data.frame(seguros.smv.result.E2)
seguros.svm.kernel = c("Linear", "Polinomial", "Radial")
seguros.logit.result.E2 = matrix(nrow=1, ncol=4)
colnames(seguros.logit.result.E2) = c("Modelo", "Muestra", "AUC", "AIC")
seguros.logit.result.E2 = data.frame(seguros.logit.result.E2)
for (k in 1:length(seguros.svm.pred.E2.list)){
  muestra = paste("M",k, sep="")
  seguros.logit.result.E2[k, 1:2] = c("Logit", muestra)
  seguros.logit.result.E2[k, 3:4] = c(colAUC(seguros.logit.pred.val.E2[[k]], seguros.
      nba.val[, SB_FLGVTA_PERSISTENCIA6M], alg="ROC"), seguros.logit.fit.E2[[k]]$aic)
  for (i in 1:length(seguros.svm.pred.E2.list[[k]])){
    indice = i + (k-1)*length(seguros.svm.pred.E2.list[[k]])
    seguros.smv.result.E2[ indice, 1:2] = c(muestra, seguros.svm.kernel[ seguros.svm.
        fit.E2.list[[k]][[i]]$kernel+1 ])
    seguros.smv.result.E2[ indice, 3] = seguros.svm.fit.E2.list[[k]][[i]]$cost
    seguros.smv.result.E2[ indice, 4] = ifelse(seguros.svm.fit.E2.list[[k]][[i]]$
        kernel!=0, seguros.svm.fit.E2.list[[k]][[i]]$gamma, NA)
    seguros.smv.result.E2[ indice, 5] = ifelse(seguros.svm.fit.E2.list[[k]][[i]]$
        kernel==1, seguros.svm.fit.E2.list[[k]][[i]]$degree, NA)
```

```
97      seguros.smv.result.E2[ indice, 6] = ifelse(seguros.svm.fit.E2.list[[k]][[i]]$
            kernel==1, seguros.svm.fit.E2.list[[k]][[i]]$coef0, NA)
98      seguros.smv.result.E2[ indice, 7 ] = colAUC(attributes(seguros.svm.pred.E2.list[[k
            ]][[i]])$decision.values, seguros.nba.val[, SB_FLGVTA_PERSISTENCIA6M], alg="
            ROC")
99    }
100 }
101 seguros.smv.result.E2
102 seguros.logit.result.E2
103 seguros.smv.result.E2[ seguros.smv.result.E2$AUC >= max(seguros.smv.result.E2[seguros.
        smv.result.E2$Muestra=="M2", 7]) & seguros.smv.result.E2$Muestra=="M2", ]
104
105 seguros.smv.result.E2.res = data.frame( matrix(nrow=1, ncol=7) )
106 colnames(seguros.smv.result.E2.res) = colnames(seguros.smv.result.E2)
107 for (i in 1:3){
108   seguros.smv.result.E2.res[(i-1)*3 + 1, ] = seguros.smv.result.E2[ seguros.smv.result
          .E2$AUC >= max(seguros.smv.result.E2[seguros.smv.result.E2$Muestra==paste("M",i,
           sep="") & seguros.smv.result.E2$Kernel=="Radial", 7]) & seguros.smv.result.E2$
          Muestra==paste("M",i, sep="") & seguros.smv.result.E2$Kernel=="Radial", ]
109   seguros.smv.result.E2.res[(i-1)*3 + 2, ] = seguros.smv.result.E2[ seguros.smv.result
          .E2$AUC >= max(seguros.smv.result.E2[seguros.smv.result.E2$Muestra==paste("M",i,
           sep="") & seguros.smv.result.E2$Kernel=="Polinomial", 7]) & seguros.smv.result.
          E2$Muestra==paste("M",i, sep="") & seguros.smv.result.E2$Kernel=="Polinomial", ]
110   seguros.smv.result.E2.res[(i-1)*3 + 3, ] = seguros.smv.result.E2[ seguros.smv.result
          .E2$AUC >= max(seguros.smv.result.E2[seguros.smv.result.E2$Muestra==paste("M",i,
           sep="") & seguros.smv.result.E2$Kernel=="Linear", 7]) & seguros.smv.result.E2$
          Muestra==paste("M",i, sep="") & seguros.smv.result.E2$Kernel=="Linear", ]
111 }
112 seguros.smv.result.E2.res = seguros.smv.result.E2.res[ seguros.smv.result.E2.res$
        Muestra=="M2", 2:7]
113 seguros.logit.result.E2 = seguros.logit.result.E2[ seguros.logit.result.E2$Muestra==
        "M2", c(1,3)]
114 seguros.smv.result.E2.res
115 seguros.logit.result.E2
116
117 xtable(seguros.smv.result.E2.res, align=c("l","l","c","c","c","c","c"), digits=c
        (0,0,0,4,0,0,4), caption="Resultados del Modelo SVM en el Escenario 2")
118 xtable(seguros.logit.result.E2, align=c("l","l","c"), digits=c(0,0,4), caption="
        Resultados del Modelo Logit en el Escenario 2")
```

: ./scripts/05.3_SVM_E2_Entrenamiento_SVM_Logit.R

```
1  #Escenario 2 (4): todas las variables numéricas y categóricas
2  seguros.svm.fit.E4.list = list()
3  tiempos.svm.fit.E4.list = list()
4  seguros.svm.pred.E4.list = list()
5  tiempos.svm.pred.E4.list = list()
6  seguros.logit.fit.E4 = list()
7  seguros.logit.pred.val.E4 = list()
8
9  for (k in 1:3){
10   seguros.svm.sigma = sigest(SB_FLGVTA_PERSISTENCIA6M ~ ., data=seguros.nba.train.
         muestra[[k]][, .SD, .SDcols=varModelo.E4.Tot_Y])
11   seguros.Mi.svm.fit.E4.K1.list = list()
12   tiempos.Mi.svm.fit.E4.K1 = numeric()
13   for (i in 1:length(seguros.svm.C)){
14     pti = proc.time()
15     seguros.Mi.svm.fit.E4.K1.list[[i]] = rpusvm(SB_FLGVTA_PERSISTENCIA6M ~ ., data=
           seguros.nba.train.muestra[[k]][, .SD, .SDcols=varModelo.E4.Tot_Y], type="C-
           classification",
```

```
16                                                  scale=TRUE, kernel="radial", cost=
                                                        seguros.svm.C[i], gamma=seguros.
                                                        svm.sigma[3], class.weights=
                                                        seguros.svm.wts[[k]], decision.
                                                        values=TRUE)
17    tiempos.Mi.svm.fit.E4.K1[i] = (proc.time() - pti)[3]
18  }
19  seguros.Mi.svm.fit.E4.K2.list = list()
20  tiempos.Mi.svm.fit.E4.K2 = numeric()
21  for (i in 1:length(seguros.svm.C)){
22    pti = proc.time()
23    seguros.Mi.svm.fit.E4.K2.list[[i]] = rpusvm(SB_FLGVTA_PERSISTENCIA6M ~ ., data=
          seguros.nba.train.muestra[[k]][, .SD, .SDcols=varModelo.E4.Tot_Y], type="C-
          classification",
24                                                  scale=TRUE, kernel="polynomial", cost=
                                                        seguros.svm.C[i], class.weights=
                                                        seguros.svm.wts[[k]], decision.
                                                        values=TRUE)
25    tiempos.Mi.svm.fit.E4.K2[i] = (proc.time() - pti)[3]
26  }
27  seguros.Mi.svm.fit.E4.K3.list = list()
28  tiempos.Mi.svm.fit.E4.K3 = numeric()
29  for (i in 1:length(seguros.svm.C)){
30    pti = proc.time()
31    seguros.Mi.svm.fit.E4.K3.list[[i]] = rpusvm(SB_FLGVTA_PERSISTENCIA6M ~ ., data=
          seguros.nba.train.muestra[[k]][, .SD, .SDcols=varModelo.E4.Tot_Y], type="C-
          classification",
32                                                  scale=TRUE, kernel="linear", cost=
                                                        seguros.svm.C[i], class.weights=
                                                        seguros.svm.wts[[k]], decision.
                                                        values=TRUE)
33    tiempos.Mi.svm.fit.E4.K3[i] = (proc.time() - pti)[3]
34  }
35  seguros.Mi.svm.pred.E4.K1.list = list()
36  tiempos.Mi.svm.pred.E4.K1 = numeric()
37  for (i in 1:length(seguros.Mi.svm.fit.E4.K1.list)){
38    pti = proc.time()
39    seguros.Mi.svm.pred.E4.K1.list[[i]] = predict(seguros.Mi.svm.fit.E4.K1.list[[i]],
          seguros.nba.val, decision.values=TRUE)
40    tiempos.Mi.svm.pred.E4.K1[i] = (proc.time() - pti)[3]
41  }
42  seguros.Mi.svm.pred.E4.K2.list = list()
43  tiempos.Mi.svm.pred.E4.K2 = numeric()
44  for (i in 1:length(seguros.Mi.svm.fit.E4.K2.list)){
45    pti = proc.time()
46    seguros.Mi.svm.pred.E4.K2.list[[i]] = predict(seguros.Mi.svm.fit.E4.K2.list[[i]],
          seguros.nba.val, decision.values=TRUE)
47    tiempos.Mi.svm.pred.E4.K2[i] = (proc.time() - pti)[3]
48  }
49  seguros.Mi.svm.pred.E4.K3.list = list()
50  tiempos.Mi.svm.pred.E4.K3 = numeric()
51  for (i in 1:length(seguros.Mi.svm.fit.E4.K3.list)){
52    pti = proc.time()
53    seguros.Mi.svm.pred.E4.K3.list[[i]] = predict(seguros.Mi.svm.fit.E4.K3.list[[i]],
          seguros.nba.val, decision.values=TRUE)
54    tiempos.Mi.svm.pred.E4.K3[i] = (proc.time() - pti)[3]
55  }
56  seguros.svm.fit.E4.list[[k]] = append(append(seguros.Mi.svm.fit.E4.K1.list, seguros.
        Mi.svm.fit.E4.K2.list), seguros.Mi.svm.fit.E4.K3.list)
57  tiempos.svm.fit.E4.list[[k]] = cbind(tiempos.Mi.svm.fit.E4.K1, tiempos.Mi.svm.fit.E4
```

```
      .K2, tiempos.Mi.svm.fit.E4.K3)
58  colnames(tiempos.svm.fit.E4.list[[k]]) = c("K1 Radial", "K2 Polynomial", "K3 Linear"
      )
59  seguros.svm.pred.E4.list[[k]] = append(append(seguros.Mi.svm.pred.E4.K1.list,
      seguros.Mi.svm.pred.E4.K2.list), seguros.Mi.svm.pred.E4.K3.list)
60  tiempos.svm.pred.E4.list[[k]] = cbind(tiempos.Mi.svm.pred.E4.K1, tiempos.Mi.svm.pred
      .E4.K2, tiempos.Mi.svm.pred.E4.K3)
61  colnames(tiempos.svm.pred.E4.list[[k]]) = c("K1 Radial", "K2 Polynomial", "K3 Linear
      ")
62
63  #Ajuste y predicción del modelo logit
64  seguros.logit.fit.E4[[k]] = glm(SB_FLGVTA_PERSISTENCIA6M ~ ., family=binomial, data=
      seguros.nba.train.muestra[[k]][, .SD, .SDcols=varModelo.E4.Tot_Y])
65  seguros.logit.pred.val.E4[[k]] = predict(seguros.logit.fit.E4[[k]], seguros.nba.val
      [, .SD, .SDcols=varModelo.E4.Tot], type="response")
66 }
67
68 saveRDS(seguros.svm.fit.E4.list, "Models//seguros.svm.fit.E4.list.RDS")
69 saveRDS(seguros.svm.pred.E4.list, "Models//seguros.svm.pred.E4.list.RDS")
70 saveRDS(seguros.logit.fit.E4, "Models//seguros.logit.fit.E4.RDS")
71 #seguros.svm.fit.E4.list = readRDS("Models//seguros.svm.fit.E4.list.RDS")
72 #seguros.svm.pred.E4.list = readRDS("Models//seguros.svm.pred.E4.list.RDS")
73 #seguros.logit.fit.E4 = readRDS("Models//seguros.logit.fit.E4.RDS")
74 #seguros.logit.pred.val.E4 = list()
75 #seguros.logit.pred.val.E4[[1]] = predict(seguros.logit.fit.E4[[1]], seguros.nba.val[,
     .SD, .SDcols=varModelo.E4.Tot], type="response")
76 #seguros.logit.pred.val.E4[[2]] = predict(seguros.logit.fit.E4[[2]], seguros.nba.val[,
     .SD, .SDcols=varModelo.E4.Tot], type="response")
77 #seguros.logit.pred.val.E4[[3]] = predict(seguros.logit.fit.E4[[3]], seguros.nba.val[,
     .SD, .SDcols=varModelo.E4.Tot], type="response")
78
79 #WARNING: algoritmo logit NO converge
80 seguros.svm.roc.E4.list = list()
81 seguros.smv.result.E4 = matrix(nrow=1, ncol=7)
82 colnames(seguros.smv.result.E4) = c("Muestra", "Kernel", "C", "gamma", "d", "delta", "
     AUC")
83 seguros.smv.result.E4 = data.frame(seguros.smv.result.E4)
84 seguros.svm.kernel = c("Linear", "Polinomial", "Radial")
85 seguros.logit.result.E4 = matrix(nrow=1, ncol=4)
86 colnames(seguros.logit.result.E4) = c("Modelo", "Muestra", "AUC", "AIC")
87 seguros.logit.result.E4 = data.frame(seguros.logit.result.E4)
88 for (k in 1:length(seguros.svm.pred.E4.list)){
89   muestra = paste("M",k, sep="")
90   seguros.logit.result.E4[k, 1:2] = c("Logit", muestra)
91   seguros.logit.result.E4[k, 3:4] = c(colAUC(seguros.logit.pred.val.E4[[k]], seguros.
       nba.val[, SB_FLGVTA_PERSISTENCIA6M], alg="ROC"), seguros.logit.fit.E4[[k]]$aic)
92   for (i in 1:length(seguros.svm.pred.E4.list[[k]])){
93     indice = i + (k-1)*length(seguros.svm.pred.E4.list[[k]])
94     seguros.smv.result.E4[ indice, 1:2] = c(muestra, seguros.svm.kernel[ seguros.svm.
         fit.E4.list[[k]][[i]]$kernel+1 ])
95     seguros.smv.result.E4[ indice, 3] = seguros.svm.fit.E4.list[[k]][[i]]$cost
96     seguros.smv.result.E4[ indice, 4] = ifelse(seguros.svm.fit.E4.list[[k]][[i]]$
         kernel!=0, seguros.svm.fit.E4.list[[k]][[i]]$gamma, NA)
97     seguros.smv.result.E4[ indice, 5] = ifelse(seguros.svm.fit.E4.list[[k]][[i]]$
         kernel==1, seguros.svm.fit.E4.list[[k]][[i]]$degree, NA)
98     seguros.smv.result.E4[ indice, 6] = ifelse(seguros.svm.fit.E4.list[[k]][[i]]$
         kernel==1, seguros.svm.fit.E4.list[[k]][[i]]$coef0, NA)
99     seguros.smv.result.E4[ indice, 7 ] = colAUC(attributes(seguros.svm.pred.E4.list[[k
         ]][[i]])$decision.values, seguros.nba.val[, SB_FLGVTA_PERSISTENCIA6M], alg="
         ROC")
```

```
  }
}
seguros.smv.result.E4
seguros.logit.result.E4
seguros.smv.result.E4[ seguros.smv.result.E4$AUC >= max(seguros.smv.result.E4[seguros.smv.result.E4$Muestra=="M2", 7]) & seguros.smv.result.E4$Muestra=="M2", ]

seguros.smv.result.E4.res = data.frame( matrix(nrow=1, ncol=7) )
colnames(seguros.smv.result.E4.res) = colnames(seguros.smv.result.E4)
for (i in 1:3){
  seguros.smv.result.E4.res[(i-1)*3 + 1, ] = seguros.smv.result.E4[ seguros.smv.result.E4$AUC >= max(seguros.smv.result.E4[seguros.smv.result.E4$Muestra==paste("M",i, sep="") & seguros.smv.result.E4$Kernel=="Radial", 7]) & seguros.smv.result.E4$Muestra==paste("M",i, sep="") & seguros.smv.result.E4$Kernel=="Radial", ]
  seguros.smv.result.E4.res[(i-1)*3 + 2, ] = seguros.smv.result.E4[ seguros.smv.result.E4$AUC >= max(seguros.smv.result.E4[seguros.smv.result.E4$Muestra==paste("M",i, sep="") & seguros.smv.result.E4$Kernel=="Polinomial", 7]) & seguros.smv.result.E4$Muestra==paste("M",i, sep="") & seguros.smv.result.E4$Kernel=="Polinomial", ]
  seguros.smv.result.E4.res[(i-1)*3 + 3, ] = seguros.smv.result.E4[ seguros.smv.result.E4$AUC >= max(seguros.smv.result.E4[seguros.smv.result.E4$Muestra==paste("M",i, sep="") & seguros.smv.result.E4$Kernel=="Linear", 7]) & seguros.smv.result.E4$Muestra==paste("M",i, sep="") & seguros.smv.result.E4$Kernel=="Linear", ]
}
seguros.smv.result.E4.res = seguros.smv.result.E4.res[ seguros.smv.result.E4.res$Muestra=="M2", 2:7]
seguros.logit.result.E4 = seguros.logit.result.E4[ seguros.logit.result.E4$Muestra=="M2", c(1,3)]
seguros.smv.result.E4.res
seguros.logit.result.E4

xtable(seguros.smv.result.E4.res, align=c("l","l","l","c","c","c","c","c"), digits=c(0,0,0,0,4,0,0,4), caption="Resultados del Modelo SVM en el Escenario 3")
xtable(seguros.logit.result.E4, align=c("l","l","l","c","c"), digits=c(0,0,0,4,2), caption="Resultados del Modelo Logit en el Escenario 3")
```

: ./scripts/05.3_SVM_E4_Entrenamiento_SVM_Logit.R

```
#Revisión de AUC para modelos SVM y Logit
seguros.smv.result.E1.res
seguros.logit.result.E1

seguros.smv.result.E2.res
seguros.logit.result.E2

seguros.smv.result.E3.res
seguros.logit.result.E3

seguros.smv.result.E4.res
seguros.logit.result.E4

seguros.smv.result.E1.res[ seguros.smv.result.E1.res$AUC >= max(seguros.smv.result.E1.res$AUC), ]
seguros.smv.result.E2.res[ seguros.smv.result.E2.res$AUC >= max(seguros.smv.result.E2.res$AUC), ]
seguros.smv.result.E3.res[ seguros.smv.result.E3.res$AUC >= max(seguros.smv.result.E3.res$AUC), ]
seguros.smv.result.E4.res[ seguros.smv.result.E4.res$AUC >= max(seguros.smv.result.E4.res$AUC), ]

#Punto de corte para modelo SVM. V1
```

```
seguros.svm.fit.E4 = seguros.svm.fit.E4.list[[2]][[19-12]]
seguros.svm.pred.val.E4 = seguros.svm.pred.E4.list[[2]][[19-12]]
seguros.svm.pred.val.E4.dv = attributes(seguros.svm.pred.val.E4)$decision.values
seguros.svm.cutoff_1 = quantile(seguros.svm.pred.val.E4.dv, p=seq(0,1, 0.01))
seguros.svm.cutoff = data.frame( t(rep(NA, 5)) )
colnames(seguros.svm.cutoff) = c("i", "Cutoff", "Sensitividad", "Especificidad", "Bal_
    Accuracy")
for (i in 1:length(seguros.svm.cutoff_1)){
  seguros.svm.matrizconf = confusionMatrix( ifelse(seguros.svm.pred.val.E4.dv <=
      seguros.svm.cutoff_1[i], 1, 0), seguros.nba.val[, SB_FLGVTA_PERSISTENCIA6M],
      positive='1' )
  seguros.svm.cutoff[i,] = c(i, seguros.svm.cutoff_1[i], seguros.svm.matrizconf$
      byClass[1], seguros.svm.matrizconf$byClass[2], seguros.svm.matrizconf$byClass
      [8])
}
seguros.svm.cutoff
seguros.svm.cutoff[ seguros.svm.cutoff$Bal_Accuracy >= max(seguros.svm.cutoff$Bal_
    Accuracy), ]
seguros.svm.cutoff.final = seguros.svm.cutoff_1[34]
seguros.svm.cutoff[34:37, ]
seguros.svm.matrizconf.val = confusionMatrix( ifelse(seguros.svm.pred.val.E4.dv <=
    seguros.svm.cutoff.final, 1, 0), seguros.nba.val[, SB_FLGVTA_PERSISTENCIA6M],
    positive='1' )
seguros.svm.matrizconf.val

seguros.svm.pred.test.E4 = predict(seguros.svm.fit.E4, seguros.nba.test, decision.
    values=TRUE)
seguros.svm.pred.test.E4.dv = attributes(seguros.svm.pred.test.E4)$decision.values
seguros.svm.roc.test.E4 = colAUC(seguros.svm.pred.test.E4.dv, seguros.nba.test[, SB_
    FLGVTA_PERSISTENCIA6M], alg="ROC")
seguros.svm.matrizconf.test = confusionMatrix( ifelse(seguros.svm.pred.test.E4.dv <=
    seguros.svm.cutoff.final, 1, 0), seguros.nba.test[, SB_FLGVTA_PERSISTENCIA6M],
    positive='1' )
seguros.svm.matrizconf.test
seguros.svm.roc.test.E4

#Punto de corte para modelo SVM. V2. Usando regla estándard de SVM: de acuerdo al
    signo de la función f(x)
seguros.svm.matrizconf.val.2 = confusionMatrix( ifelse(seguros.svm.pred.val.E4.dv <=
    0, 1, 0), seguros.nba.val[, SB_FLGVTA_PERSISTENCIA6M], positive='1' )
seguros.svm.matrizconf.val.2
seguros.svm.matrizconf.test.2 = confusionMatrix( ifelse(seguros.svm.pred.test.E4.dv <=
     0, 1, 0), seguros.nba.test[, SB_FLGVTA_PERSISTENCIA6M], positive='1' )
seguros.svm.matrizconf.test.2


#Punto de corte para modelo Logit
seguros.logit.cutoff_1 = quantile(seguros.logit.pred.val.E2[[2]], p=seq(0,1, 0.01))
seguros.logit.cutoff = data.frame( t(rep(NA, 5)) )
colnames(seguros.logit.cutoff) = c("i", "Cutoff", "Sensitividad", "Especificidad", "
    Bal_Accuracy")
for (i in 1:length(seguros.logit.cutoff_1)){
  seguros.logit.matrizconf = confusionMatrix( ifelse(seguros.logit.pred.val.E2[[2]] >
      seguros.logit.cutoff_1[i], 1, 0), seguros.nba.val[, SB_FLGVTA_PERSISTENCIA6M],
      positive='1' )
  seguros.logit.cutoff[i,] = c(i, seguros.logit.cutoff_1[i], seguros.logit.matrizconf$
      byClass[1], seguros.logit.matrizconf$byClass[2], seguros.logit.matrizconf$
      byClass[8])
}
seguros.logit.cutoff
```

```
seguros.logit.cutoff[ seguros.logit.cutoff$Bal_Accuracy >= max(seguros.logit.cutoff$
    Bal_Accuracy), ]
seguros.logit.cutoff.final = seguros.logit.cutoff_1[65]
seguros.logit.matrizconf.val = confusionMatrix( ifelse(seguros.logit.pred.val.E2[[2]]
    > seguros.logit.cutoff.final, 1, 0), seguros.nba.val[, SB_FLGVTA_PERSISTENCIA6M],
    positive='1' )
seguros.logit.matrizconf.val

seguros.logit.pred.test = predict(seguros.logit.fit.E2[[2]], seguros.nba.test, type="
    response")
seguros.logit.roc.test = colAUC(seguros.logit.pred.test, seguros.nba.test[, SB_FLGVTA_
    PERSISTENCIA6M], alg="ROC")
seguros.logit.matrizconf.test = confusionMatrix( ifelse(seguros.logit.pred.test >
    seguros.logit.cutoff.final, 1, 0), seguros.nba.test[, SB_FLGVTA_PERSISTENCIA6M],
    positive='1' )
seguros.logit.matrizconf.test
seguros.logit.roc.test


#Resumen
seguros.resumen = data.frame( t(c(NA, NA, NA)) )
colnames(seguros.resumen) = c("Modelo", "AUC Validación", "AUC Test")
seguros.resumen[1, 1] = "SVM Polinomial"
seguros.resumen[1, 2:3] = c(seguros.smv.result.E4.res[2,]$AUC, seguros.svm.roc.test.E4
    [1])
seguros.resumen[2, 1] = "Logit"
seguros.resumen[2, 2:3] = c(seguros.logit.result.E2$AUC, seguros.logit.roc.test[1])
seguros.resumen

xtable(seguros.resumen, align=c("l","l","c","c"), digits=c(0,0,5,5), caption="Medición
     de desempeño de predicción de los modelos SVM y Logit")
xtable(seguros.svm.matrizconf.test.2$table, align=c("l", "c", "c"), caption="Matriz de
     confusión en los datos de prueba para el modelo SVM")
xtable(seguros.logit.matrizconf.test$table, align=c("l", "c", "c"), caption="Matriz de
     confusión en los datos de prueba para el modelo Logit")

seguros.resumen_2 = cbind(seguros.svm.matrizconf.test$byClass[c(1:4,8)], seguros.logit
    .matrizconf.test$byClass[c(1:4,8)],
                          seguros.svm.matrizconf.test$byClass[c(1:4,8)] - seguros.
                              logit.matrizconf.test$byClass[c(1:4,8)],
                          (seguros.svm.matrizconf.test$byClass[c(1:4,8)] - seguros.
                              logit.matrizconf.test$byClass[c(1:4,8)])/seguros.svm.
                              matrizconf.test$byClass[c(1:4,8)]*100 )
colnames(seguros.resumen_2) = c("SVM Polinomial", "Logit", "Diferencia", "Porcentaje")
xtable(seguros.resumen_2, align=c("l","c","c","c","c"), digits=c(0,4,4,5,4), caption="
    Indicadores de Desempeño de los modelos SVM y Logit")

#Curva ROC
seguros.svm.roc.test.E4.graph = roc(seguros.nba.test[, SB_FLGVTA_PERSISTENCIA6M],
    seguros.svm.pred.test.E4.dv)
seguros.logit.roc.test.graph = roc(seguros.nba.test[, SB_FLGVTA_PERSISTENCIA6M],
    seguros.logit.pred.test)
plot(seguros.svm.roc.test.E4.graph, col="blue", main="Curva ROC para modelos SVM y
    Logit", xlab="Especificidad", ylab="Sensitividad")
plot(seguros.logit.roc.test.graph, col="red", add=T)
legend(0.2, 0.2, lty=c(1,1), lwd=c(2.5,2.5), col=c("blue","red"), legend=c("SVM", "
    Logit"))
```

: ./scripts/05.4_SVM_ResumenMedidasPerformance.R

```
seguros.smv.result.E4[seguros.smv.result.E4$Muestra=="M2",c(2,3,7)]
seguros.smv.result.E2[seguros.smv.result.E4$Muestra=="M2",c(2,3,7)]


seguros.smv.result.E4.det = cbind( rep("Escenario 2", 4),
      seguros.smv.result.E4[seguros.smv.result.E4$Muestra=="M2" & seguros.smv.result.
          E4$Kernel=="Radial",c(3,7)],
      seguros.smv.result.E4[seguros.smv.result.E4$Muestra=="M2" & seguros.smv.result.
          E4$Kernel=="Polinomial",c(7)],
      seguros.smv.result.E4[seguros.smv.result.E4$Muestra=="M2" & seguros.smv.result.
          E4$Kernel=="Linear",c(7)])
colnames(seguros.smv.result.E4.det) = c("Escenario", "C", "Kernel Radial", "Kernel
    Polinomial", "Kernel Lineal")

seguros.smv.result.E2.det = cbind( rep("Escenario 1", 4),
                                   seguros.smv.result.E2[seguros.smv.result.E2$Muestra
                                       =="M2" & seguros.smv.result.E2$Kernel=="Radial"
                                       ,c(3,7)],
                                   seguros.smv.result.E2[seguros.smv.result.E2$Muestra
                                       =="M2" & seguros.smv.result.E2$Kernel=="
                                       Polinomial",c(7)],
                                   seguros.smv.result.E2[seguros.smv.result.E2$Muestra
                                       =="M2" & seguros.smv.result.E2$Kernel=="Linear"
                                       ,c(7)])
colnames(seguros.smv.result.E2.det) = c("Escenario", "C", "Kernel Radial", "Kernel
    Polinomial", "Kernel Lineal")

seguros.smv.result.E2.det
seguros.smv.result.E4.det

rbind(seguros.smv.result.E2.det, seguros.smv.result.E4.det)
xtable(rbind(seguros.smv.result.E2.det, seguros.smv.result.E4.det), align=c("l","l","l
    ","c","c","c"), digits=c(0,0,1,4,4,4), caption="Medición de desempeño de
    predicción de los modelos SVM y Logit")
```

: ./scripts/05.6_SVM_CuadroDetalleAUC_SVM.R

```
#Idea: un cuadro con la performance de "valor predicción positivo" para modelos SVM,
    probando varios valores de la constante de costo C

#Medidas de performance para Escenario 2 (usando variables numéricas y categóricas).
    Sólo para muestra M2 (desbalance de 1 a 0: 75 a 25)
#AUC
seguros.smv.result.E4.new = seguros.smv.result.E4[seguros.smv.result.E4$Muestra == "M2
    ", c(2,3,7)]
seguros.smv.result.E4.res.new = seguros.smv.result.E4.res[seguros.smv.result.E4.res$
    Muestra == "M2", c(2,3,7)]

#Muestra de validación: Predicción incorrecta de 1s, Predicción correcta de 1s, Total
    predicción de 1s, Valor Predicción Positivo
seguros.svm.matrizconf.val.E4.list = list()
for (i in 1:nrow(seguros.smv.result.E4.new)){
  seguros.svm.pred.val.E4.dv.tp = attributes(seguros.svm.pred.E4.list[[2]][[i]])$
      decision.values
  seguros.svm.matrizconf.val.E4.list[[i]] = confusionMatrix( ifelse(seguros.svm.pred.
      val.E4.dv.tp <= 0, 1, 0), seguros.nba.val[, SB_FLGVTA_PERSISTENCIA6M], positive=
      '1' )
}
seguros.smv.result.E4.new$Pred1Falso = 0
seguros.smv.result.E4.new$Pred1Verdadero = 0
```

```
seguros.smv.result.E4.new$Pred1 = 0
seguros.smv.result.E4.new$ValPredPos = 0
for (i in 1:length(seguros.svm.matrizconf.val.E4.list)){
  seguros.smv.result.E4.new[i,] = cbind( seguros.smv.result.E4.new[i, c(1:3)],
                                         cbind( rbind(seguros.svm.matrizconf.val.E4.
                                             list[[i]]$table[2,]),
                                                sum(seguros.svm.matrizconf.val.E4.list
                                                    [[i]]$table[2,]),
                                                seguros.svm.matrizconf.val.E4.list[[i
                                                    ]]$byClass[3] )
  )
}
seguros.smv.result.E4.new
seguros.svm.matrizconf.val.E4.list[19-12]


#Muestra de prueba: Predicción incorrecta de 1s, Predicción correcta de 1s, Total
    predicción de 1s, Valor Predicción Positivo
#Para los distintos tipos de Kernel, probando distintos valores de la constante de
    costo C
seguros.svm.matrizconf.test.E4.list = list()
seguros.svm.pred.test.E4.list = list()
seguros.smv.result.E4.test = seguros.smv.result.E4.new
for (i in 1:nrow(seguros.smv.result.E4.new)){
  seguros.svm.pred.test.E4.list[[i]] = predict(seguros.svm.fit.E4.list[[2]][[i]],
      seguros.nba.test, decision.values=TRUE)
  seguros.svm.pred.test.E4.dv.tp = attributes(seguros.svm.pred.test.E4.list[[i]])$
      decision.values
  seguros.smv.result.E4.test[i, 3] = colAUC(seguros.svm.pred.test.E4.dv.tp, seguros.
      nba.test[, SB_FLGVTA_PERSISTENCIA6M], alg="ROC")
  seguros.svm.matrizconf.test.E4.list[[i]] = confusionMatrix( ifelse(seguros.svm.pred.
      test.E4.dv.tp <= 0, 1, 0), seguros.nba.test[, SB_FLGVTA_PERSISTENCIA6M],
      positive='1' )
}
for (i in 1:length(seguros.svm.matrizconf.test.E4.list)){
  seguros.smv.result.E4.test[i,] = cbind( seguros.smv.result.E4.test[i, c(1:3)],
                                         cbind( rbind(seguros.svm.matrizconf.test.E4.
                                             list[[i]]$table[2,]),
                                                sum(seguros.svm.matrizconf.test.E4.
                                                    list[[i]]$table[2,]),
                                                seguros.svm.matrizconf.test.E4.list[[i
                                                    ]]$byClass[3] )
  )
}
seguros.smv.result.E4.test
xtable(seguros.smv.result.E4.test, align=c("l","l","c","c","c","c","c","c"), digits=c
    (0,0,0,4,0,0,0,4), caption="Resultados de predicción de 1s verdaderos, 1s falsos,
    y valor de predicción positiva")
seguros.svm.matrizconf.test.E4.list[[19-12]]


#Muestra de prueba: Predicción incorrecta de 1s, Predicción correcta de 1s, Total
    predicción de 1s, Valor Predicción Positivo
#Para el mejor resultado: Kernel Polinomial, Constante de Costo C=5. Cambiando los
    valores del parámetro de pesos: parámetro usual (0-1:25-75), sin parámetro, más
    desbalance(0-1:90-10); menos desbalance(0-1:40-60); balanceo inverso (0-1:75-25)
seguros.svm.wts.test = list()
pesos = c(0.25, 0.75)
names(pesos) = c("0", "1")
seguros.svm.wts.test[[1]] = pesos
```

```
58 seguros.svm.wts.test[[2]] = NULL
59 pesos[1:2] = c(0.10, 0.90)
60 seguros.svm.wts.test[[3]] = pesos
61 pesos[1:2] = c(0.40, 0.60)
62 seguros.svm.wts.test[[4]] = pesos
63 pesos[1:2] = c(0.75, 0.25)
64 seguros.svm.wts.test[[5]] = pesos
65
66 seguros.svm.fit.E4.wts.list = list()
67 seguros.svm.fit.E4.wts.list[[1]] = seguros.svm.fit.E4.list[[2]][[19-12]]
68 seguros.svm.fit.E4.wts.list[[2]] = rpusvm(SB_FLGVTA_PERSISTENCIA6M ~ ., data=seguros.
       nba.train.muestra[[2]][, .SD, .SDcols=varModelo.E4.Tot_Y], type="C-classification"
       ,
69                                          scale=TRUE, kernel="polynomial", cost=5,
                                                decision.values=TRUE)
70 seguros.svm.fit.E4.wts.list[[3]] = rpusvm(SB_FLGVTA_PERSISTENCIA6M ~ ., data=seguros.
       nba.train.muestra[[2]][, .SD, .SDcols=varModelo.E4.Tot_Y], type="C-classification"
       ,
71                                          scale=TRUE, kernel="polynomial", cost=5,
                                                class.weights=seguros.svm.wts.test[[3]],
                                                 decision.values=TRUE)
72 seguros.svm.fit.E4.wts.list[[4]] = rpusvm(SB_FLGVTA_PERSISTENCIA6M ~ ., data=seguros.
       nba.train.muestra[[2]][, .SD, .SDcols=varModelo.E4.Tot_Y], type="C-classification"
       ,
73                                          scale=TRUE, kernel="polynomial", cost=5,
                                                class.weights=seguros.svm.wts.test[[4]],
                                                 decision.values=TRUE)
74 seguros.svm.fit.E4.wts.list[[5]] = rpusvm(SB_FLGVTA_PERSISTENCIA6M ~ ., data=seguros.
       nba.train.muestra[[2]][, .SD, .SDcols=varModelo.E4.Tot_Y], type="C-classification"
       ,
75                                          scale=TRUE, kernel="polynomial", cost=5,
                                                class.weights=seguros.svm.wts.test[[5]],
                                                 decision.values=TRUE)
76
77 seguros.svm.pred.test.E4.wts.list = list()
78 seguros.svm.matrizconf.test.E4.wts.list = list()
79 seguros.smv.result.E4.test.wts = data.frame( matrix(nrow=4, ncol=8) )
80 colnames(seguros.smv.result.E4.test.wts) =  c("Kernel","C","Pesos 0-1","AUC","
       Pred1Falso","Pred1Verdadero","Pred1","ValPredPos")
81 seguros.smv.result.E4.test.wts$Kernel = rep("Polinomial",4)
82 seguros.smv.result.E4.test.wts$C = rep(5,4)
83 seguros.smv.result.E4.test.wts[,3] = c("25-75","-","10-90","40-60")
84 for (i in 1:length(seguros.svm.fit.E4.wts.list)){
85   seguros.svm.pred.test.E4.wts.list[[i]] = predict(seguros.svm.fit.E4.wts.list[[i]],
         seguros.nba.test, decision.values=TRUE)
86   seguros.svm.pred.test.E4.dv.tp = attributes(seguros.svm.pred.test.E4.wts.list[[i]])$
         decision.values
87   seguros.smv.result.E4.test.wts[i, 4] = colAUC(seguros.svm.pred.test.E4.dv.tp,
         seguros.nba.test[, SB_FLGVTA_PERSISTENCIA6M], alg="ROC")
88   seguros.svm.matrizconf.test.E4.wts.list[[i]] = confusionMatrix( ifelse(seguros.svm.
         pred.test.E4.dv.tp <= 0, 1, 0), seguros.nba.test[, SB_FLGVTA_PERSISTENCIA6M],
         positive='1' )
89 }
90 for (i in 1:length(seguros.svm.fit.E4.wts.list)){
91   seguros.smv.result.E4.test.wts[i,] = cbind( seguros.smv.result.E4.test.wts[i, c(1:4)
         ],
92                                          cbind( rbind(seguros.svm.matrizconf.test.E4.
                                                wts.list[[i]]$table[2,]),
93                                                   sum(seguros.svm.matrizconf.test.E4.
                                                       wts.list[[i]]$table[2,]),
```

```
                                                    seguros.svm.matrizconf.test.E4.wts.
                                                        list[[i]]$byClass[3] )
  )
}
seguros.smv.result.E4.test.wts
xtable(seguros.smv.result.E4.test.wts, align=c("l","l","c","c","c","c","c","c","c"),
    digits=c(0,0,0,0,4,0,0,0,4), caption="Resultados de predicción de 1s verdaderos, 1
    s falsos, y valor de predicción positiva")


###########
#Guardar predicciones de modelos SVM en muestra de test
saveRDS(seguros.svm.pred.test.E4.list, "Models//seguros.svm.pred.test.E4.list.RDS")

#Guardar modelos SVM entrenedos con diferentes valores de pesos de clase, y las
    predicciones realizadas con estos modelos en la muestra de test
saveRDS(seguros.svm.fit.E4.wts.list, "Models//seguros.svm.fit.E4.wts.list.RDS")
saveRDS(seguros.svm.pred.test.E4.wts.list, "Models//seguros.svm.pred.test.E4.wts.list.
    RDS")
```

: ./scripts/05.7_SVM_Pruebas_C_Pesos.R

Bibliografía

Adler, D., Glaser, C., Nenadic, O., Oehlschlagel, J. y Zucchini, W. (2014). *ff: memory-efficient storage of large data on disk and fast access functions*. R package version 2.2-13.
URL: *http://CRAN.R-project.org/package=ff*

Basim Alwan, H. y Ruhana Ku-Mahamud, K. (2013). Solving support vector machine model selection problem using continuous ant colony optimization, *International Journal of Information Processing and Management (IJIPM)* **4**(2): 86–97.
URL: *http://www.aicit.org/IJIPM/ppl/IJIPM162PPL.pdf*

Batuwita, R. y Palade, V. (2013). *Class Imbalance Learning Methods for Support Vector Machines*.
URL: *http://www.cs.ox.ac.uk/people/vasile.palade/papers/Class-Imbalance-SVM.pdf*

Breiman, L. (2001). Statistical modeling: The two cultures, *Statistical Science* **16(3)**: 199–231.

Chang, C.-C. y Lin, C.-J. (2011). Libsvm: A library for support vector machines, *ACM Trans. Intell. Syst. Technol.* **2**(3): 27:1–27:27.
URL: *http://doi.acm.org/10.1145/1961189.1961199*

Dowle, M., Short, T., Lianoglou, S., Srinivasan, A., Saporta, R. y Antonyan, E. (2014). *data.table: Extension of data.frame*. R package version 1.9.4.
URL: *http://CRAN.R-project.org/package=data.table*

Fehr, J., Zapien Arreola, K. y Burkhardt, H. (2007). Fast support vector machine classification of very large datasets, *Data Analysis, Machine Learning and Applications - Proceedings of the 31st Annual Conference of the Gesellschaft fur Klassifikation e.V., Albert-Ludwigs-Universitat Freiburg, March 7-9, 2007*, pp. 11–18.

Grossman, S. y Flores, J. (2012). *Algebra Lineal*, McGraw-Hill.

Hamel, L. (2009). *Knowledge Discovery with Support Vector Machines*, Wiley.

Hastie, T., Tibshirani, R. y Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer.

Hsu, C.-W., Chang, C.-C. y Lin, C.-J. (2000). A Practical Guide to Support Vector Classification.
URL: *http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.6.3096*

James, G., Witten, D., Hastie, T. y Tibshirani, R. (2013). *An Introduction to Statistical Learning with Applications in R*, Springer.

Karatzoglou, A., Smola, A., Hornik, K. y Zeileis, A. (2004). kernlab – an S4 package for kernel methods in R, *Journal of Statistical Software* **11**(9): 1–20.
URL: *http://www.jstatsoft.org/v11/i09/*

Kuhn, M., Wing, J., Weston, S., Williams, A., Keefer, C., Engelhardt, A., Cooper, T., Mayer, Z., Kenkel, B., the R Core Team, Benesty, M., Lescarbeau, R., Ziem, A. y Scrucca, L. (2015). *caret: Classification and Regression Training*. R package version 6.0-41.
URL: *http://CRAN.R-project.org/package=caret*

Meyer, D., Dimitriadou, E., Hornik, K., Weingessel, A. y Leisch, F. (2014). *e1071: Misc Functions of the Department of Statistics (e1071), TU Wien*. R package version 1.6-3.
URL: *http://CRAN.R-project.org/package=e1071*

Scholkopf, B. y Smola, A. (2002). *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*, The MIT Press.

Tsang, I. W., Kwok, J. T., Cheung, P.-m. y Cristianini, N. (2005). Core vector machines: Fast SVM training on very large data sets, *Journal of Machine Learning Research* **6**: 363–392.
URL: *http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.66.9158*

Vapnik, V. (1998). *Statistical Learning Theory*, Wiley-Interscience.

Vapnik, V. (2000). *The Nature of Statistical Learning Theory*, Springer.

Yuh-jye, L. y L. Mangasarian, O. (2001). Rsvm: Reduced support vector machines, *Data Mining Institute, Computer Sciences Department, University of Wisconsin*, pp. 00–07.