PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

ESCUELA DE POSGRADO

Identificación de Obras Urbanas para la Ciudad de Lima a través del uso de Herramientas basadas en *Machine Learning*

TESIS PARA OBTENER EL GRADO DE MAGÍSTER EN GERENCIA DE TECNOLOGÍAS DE INFORMACIÓN OTORGADO POR LA PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

PRESENTADA POR

Juan Francisco Mendoza Bernedo, DNI: 43977334

Fernando Jesús Saldaña Bustamante, DNI: 41275676

Rocío Susana Vivanco Yovera, DNI: 03898287

ASESOR

Dr. Luis Negrón Naldos, DNI: 10788917

ORCID 0000-0003-1328-0323

JURADO

Dr. Percy Maquina Feldman

Dr. Yván García López

Dr. Luis Negrón Naldos

Surco, octubre de 2021

Agradecimientos

Expresamos nuestra mayor gratitud y aprecio a:

El profesor Yván García, por los conocimientos transmitidos, los mismos que hicieron posible el desarrollo y entrega de esta tesis, con una base sólida de liderazgo y planeamiento estratégico.

Un especial agradecimiento al profesor Luis Negrón, nuestro asesor, por el apoyo y asesoría brindada para el desarrollo de esta tesis.

Todos los profesores de CENTRUM PUCP y a la Universidad de Miami, por los valiosos conocimientos transmitidos.

Dedicatorias

A mis padres, Juan y Gladys, quienes con su esfuerzo me brindaron todas las herramientas necesarias para lograr mis metas; gracias por inculcarme los valores necesarios para afrontar los desafíos y adversidades de la vida. A mi hermana Angélica, por su constante ejemplo de superación y compañía en esta etapa profesional.

Juan Francisco Mendoza Bernedo

A Flor, mi esposa, y a mis hijos Esteban y Vicente, quienes me apoyaron y acompañaron durante el desarrollo de este gran paso en mi vida profesional; gracias por su amor y comprensión. A mis padres, por su ejemplo y por el gran esfuerzo que dedicaron en mi educación.

Fernando Saldaña Bustamante

A mi esposo y amigo Eduardo Roncal, por apoyarme en esta travesía de poder continuar con los estudios de la maestría y ser mi soporte en todo momento. A mis hijos y mis padres, que son las personas que siempre están presentes en cada acción de mi vida.

Rocío Vivanco Yovera

Resumen Ejecutivo

La presente investigación tiene como objetivo utilizar la tecnología basada en *machine learning*, para la identificación de obras urbanas en la ciudad de Lima. La posibilidad de extraer y analizar información de medios sociales mediante el análisis de sentimientos, también conocido como minería de opinión (*opinion mining*), es, para Liu (2015), un campo de estudio que se centra en analizar las opiniones que expresan o implican sentimientos positivos o negativos. Para abordar esta problemática se propone un modelo para la clasificación de mensajes de Twitter de forma automática, a fin de comprender cuál es la intención que tiene el usuario cuando publica un mensaje sobre las obras urbanas en la ciudad de Lima, en especial pistas, parques y veredas, además de identificar la ubicación de estas obras en sus diferentes distritos. La investigación permitió reconocer patrones de comportamiento que son de gran importancia para la Municipalidad de Lima, debido a que, al tener identificada la problemática de las obras urbanas por distritos, podrá plantear estrategias para priorizar obras de manera anticipada y así poder planificarlas para su ejecución en el periodo municipal. Los resultados obtenidos utilizando el algoritmo de clasificación supervisada *support vector machine* (SVM) muestran valores de aciertos del modelo de alrededor del 78% en análisis de sentimientos. Se realizó una primera clasificación de distritos que necesitan urgentemente de obras urbanas, disgregada en tres tipos: parques, pistas y veredas. Los resultados generales del modelo son buenos en comparación con las investigaciones de otros autores como Aiala et al. (2017).

Abstract

This research is called "Identification of urban works for the city of Lima through the use of tools based on Machine Learning", it has as goal the use of technology based on Machine Learning for the identification of urban works in the city of Lima. The possibility of extracting and analyzing information from different social media through sentiment analysis, also known as opinion mining; that for Liu (2015), is a field of study that focuses mainly on analyzing the opinions that express or imply positive or negative feelings. To address this topic, a model is proposed for the automatic classification of Twitter messages to try to understand the intention of the user when he publishes a message about urban works in the city of Lima, especially roads, parks, and sidewalks, additionally it is necessary to identify the location of urban works in the districts of Lima. This research allowed to identify patterns of behavior that are of great importance for Lima Municipality, because by having identified the problem of urban works by districts, it will allow them to propose strategies that allow prioritizing the districts with demand for urban works in advance and be able to plan them for execution in the governance period. The results obtained using the Support Vector Machine (SVM) supervised classification algorithm, show us values of correctness of the model around 78% in sentiment analysis. A first classification of districts was made with the urgently needs urban works and has been classified into the three types of urban works: parks, tracks, sidewalks. The general results of the model are good when comparing the research of other authors such as Aiala et al. (2017).

Tabla de Contenidos

Lista de Tablas

Lista de Figuras

Capítulo I: Introducción

1.1 Antecedentes

Actualmente, existe una convergencia de dos fenómenos importantes: la aceleración de la urbanización a nivel mundial y la revolución digital. La Organización de las Naciones Unidas (ONU, 2018) señaló que un 54.6% de personas vive en las ciudades, y se proyecta que para el 2050 el 64.1% haga lo propio.

Este crecimiento imparable convierte a las ciudades en un centro de atención de los poderes públicos y privados en la medida en que las urbes deben establecer estrategias de adaptación continua a estos procesos de cambio (Agencia Europea de Medio Ambiente, 2015).

Los grandes retos que los países enfrentarán por la aglomeración urbana son la planificación urbana, la administración y la gobernanza de las ciudades para hacerlas sostenibles, maximizando la economía y minimizando los daños medioambientales (ONU, 2018).

Las ciudades desarrollan estrategias para poder enfrentar los grandes retos derivados del crecimiento de la población y del agotamiento de los recursos naturales. Todo ello les impone la necesidad de adoptar mecanismos de adaptación para garantizar su sostenibilidad (Muñoz et al., 2018).

El desarrollo de las ciudades en el mundo apunta a un enfoque en donde se coloca como centro al ciudadano, buscando que sean sostenibles, inclusivas y pensada para sus habitantes, con el objetivo de satisfacer sus necesidades y resolver sus problemas de forma inteligente. Una ciudad inteligente es aquella que coloca a las personas en el centro del desarrollo, incorpora tecnologías de la información y la comunicación en la gestión urbana, y usa estos elementos como herramientas para estimular la información de un gobierno eficiente, que incluya los procesos de planificación colaborativa y participación ciudadana. Al promover un desarrollo integrado y sostenible, las ciudades inteligentes se tornan más

innovadoras y competitivas, atractivas y resilientes, mejorando así las vidas de sus ciudadanos y empresarios (Bouskela et al., 2016).

Para tener una referencia sobre el desarrollo de las ciudades, se puede observar el índice IESE *Cities in Motion* (ICIM), estudio anual realizado por la escuela de negocios de la Universidad de Navarra, donde construyeron un indicador que permite medir la sostenibilidad de las principales ciudades del mundo en búsqueda de que sean más inteligentes. De acuerdo con el ICIM publicado el año 2020, Lima se encuentra en el puesto 155 de un total de 174 ciudades estudiadas a nivel global, con un ICIM de 34.23, que la ubica en la categoría (B) por desempeño "bajo". La Tabla 1 muestra el *top five* de posiciones en Latinoamérica en comparación con Lima, utilizando datos del ICIM de los últimos tres años (IESE Business School, 2020).

Tabla 1

Top Five de Posiciones ICIM Región Latinoamérica vs. Lima

Ciudad	Posición regional	Posición global 2017	Posición global 2018	Posición global 2019
Santiago, Chile	1	85	75	68
Buenos Aires, Argentina	2	87	94	90
Montevideo, Uruguay	3	109	108	110
Panamá, Panamá	4	116	116	113
San José, Costa Rica	5	113	115	114
Lima, Perú	21	131	138	155

Nota. Adaptado de "Índice IESE Cities in Motion 2020," por IESE Business School, 2020 (https://media.iese.edu/research/pdfs/ST-0542.pdf).

Dentro de la arquitectura de las ciudades inteligentes es necesario el uso de datos abiertos, *Big Data* y *Analytics*, que permitan desarrollar análisis predictivos y sean soporte en la toma de decisiones y el planeamiento orientado al ciudadano (Bouskela et al., 2016). Asimismo, según estos mismos autores, para administrar y mejorar las ciudades, es necesario conocer lo que sucede en ellas y en sus diferentes regiones. Esto es posible modificando las estructuras de gobierno y los procesos de comunicación y participación de diferentes autores que intervienen en su gestión, incluido el ciudadano.

En Perú, como parte de la respuesta del Gobierno en busca del desarrollo del país, la Presidencia del Consejo de Ministros (PCM), a través de la Secretaría de Gobierno Digital, viene desarrollando la Estrategia Nacional de Inteligencia Artificial, conocida por sus siglas IA, en donde se toma como definición actual de IA la brindada por la Organización para la Cooperación y el Desarrollo Económicos (OECD): "Un sistema de IA es un sistema electrónico-mecánico que puede, para una serie de objetivos definidos por humanos, hacer predicciones, recomendaciones, o tomar decisiones, influenciando ambientes reales o virtuales" (citado en Secretaría de Gobierno Digital de la PCM, 2021, p. 11). Además, en la misma Estrategia Nacional de IA se considera al *machine learning* (aprendizaje automático) como "Conjunto de modelos de inteligencia artificial que aprenden en base a datos (de entrenamiento) para poder predecir resultados o tomar decisiones sin ser explícitamente programados para ello" (Secretaría de Gobierno y Transformación Digital de la PCM, 2021, p. 12). En la Tabla 2 se muestran los resultados de la encuesta del estado de IA en el sector público peruano de enero 2021, donde solo el 7% de las instituciones que contestaron indica que utiliza IA en su institución (Secretaría de Gobierno y Transformación Digital de la PCM, 2021).

Tabla 2

Resultado de Encuestas IA en el Sector Público Peruano, Enero 2021

Tipo de institución	No	Sí	Total general
Empresa del Estado	12	0	12
Gobierno local	375	29	404
Gobierno regional	28	2	30
Organismos autónomos	7	1	8
Poder Ejecutivo	17	6	23
Poder Judicial	1	0	1
Programa social	3	0	3
Universidad	6	1	7
Total general	449	39	488

Nota. Adaptado de "Estrategia nacional de inteligencia artificial. Documento de trabajo para la participación de la ciudadanía 2021-2026," por la Secretaría de Gobierno y Transformación Digital de la PCM, 2021 (https://cdn.www.gob.pe/uploads/document/file/1899077/Estrategia%20Nacional%20de%20Inteligencia%20Artificial.pdf).

La Estrategia Nacional de IA peruana propone los siguientes seis ejes estratégicos: (a) formación y atracción de talento, (b) modelo económico, (c) infraestructura tecnológica, (d) datos, (e) ética, y (f) colaboración. Cada uno de los ejes cuenta con objetivos estratégicos. El segundo objetivo estratégico dentro del segundo eje, modelo económico, es "impulsar en los organismos públicos, la incorporación de la inteligencia artificial en su operación y servicios a los ciudadanos" (Secretaría de Gobierno y Transformación Digital de la PCM, 2021, p. 65.). Este objetivo permite apreciar el esfuerzo del Gobierno del Perú, alineado con la guía del BID, como parte del desarrollo de las ciudades inteligentes.

La Municipalidad Metropolitana de Lima es la institución pública responsable de velar por el desarrollo de la ciudad de Lima; posee un órgano descentralizado encargado de la supervisión y ejecución de obras urbanas: Fondo Metropolitano de Inversiones (INVERMET). Lo que el presente trabajo de investigación pretende es determinar si con el apoyo de herramientas tecnológicas basadas en el análisis de información que brindan los ciudadanos de Lima Metropolitana, aprovechando fuentes de redes sociales que son de acceso público y utilizando herramientas específicas de *machine learning*, se pueden determinar las necesidades de la población para la identificación de obras urbanas como parques, pistas y veredas, las cuales son parte de la infraestructura que propone y ejecuta el INVERMET.

1.2 Problema de Investigación

Existe una falta de sintonía entre las necesidades actuales de la ciudadanía y las obras públicas planteadas para la ciudad de Lima, donde no se consideran las urgencias y reclamos de los ciudadanos como apoyo para la definición de obras urbanas en Lima Metropolitana.

Según un estudio realizado por la Asociación de Víctimas de Accidentes de Tránsito (AVIACTRAN), por cada kilómetro recorrido se puede encontrar ocho baches en zonas urbanas de Lima Metropolitana. En las vías denominadas rápidas hay al menos dos huecos por cada kilómetro ("Lima: Pistas con huecos," 2020).

En la encuesta realizada por el Instituto de Opinión Pública de la Pontificia Universidad Católica del Perú (IOP- PUCP), por encargo del observatorio ciudadano Lima Cómo Vamos, el 40.8% de la población de Lima Metropolitana considera a la falta de árboles y el mantenimiento de las áreas verdes como el segundo mayor problema de la gestión ambiental en la ciudad, superado solo por la contaminación ocasionada por los vehículos automotores (Lima Cómo Vamos, 2019). La misma encuesta reveló, asimismo, que solo un 18.1% está conforme con la gestión de la Municipalidad de Lima (ver Tabla 3). La satisfacción con los parques y áreas verdes de uso público en Lima Metropolitana indica un 17%.

Tabla 3

Resultado de Encuesta Gestión de la Municipalidad Distrital de Lima Metropolitana, 2019

Evaluación de la gestión municipal	% Lima
Muy mala gestión / Mala gestión	40.4
Ni buena ni mala gestión	39.4
Buena gestión / Muy buena gestión	18.1

Nota. Adaptado de "Lima y Callao según sus ciudadanos: Décimo informe urbano de percepción sobre calidad de vida en la ciudad," por Lima Cómo Vamos, 2019 (http://www.limacomovamos.org/wp-content/uploads/2019/11/Encuesta-2019_web.pdf).

A la pregunta si están de acuerdo o no con la afirmación "Existen mecanismos de consulta y participación ciudadana en la Municipalidad de Lima antes de aprobar proyectos importantes para la ciudad", solo un 22.1% confirmó dicha afirmación, en tanto que a la pregunta "Las acciones de la Municipalidad de Lima realmente contribuyen al desarrollo de la ciudad", el 24.3% estuvo de acuerdo (Lima Cómo Vamos, 2019), tal como se puede apreciar a continuación en la Tabla 4.

Tabla 4

Resultados Gestión de la Municipalidad Metropolitana de Lima, 2019

Gestión de la Municipalidad Metropolitana		% Lima
Las decisiones de la Municipalidad favorecen a unas pocas personas o grupos	En desacuerdo	22.3
	De acuerdo	45.1
Las acciones de la Municipalidad de Lima Metropolitana realmente contribuyen al desarrollo de la ciudad	En desacuerdo	34.8
	De acuerdo	24.3
Existen mecanismos de consulta y participación ciudadana en la Municipalidad de Lima antes de aprobar proyectos importantes para la ciudad	En desacuerdo	39.4
	De acuerdo	22.1
Existe corrupción en la gestión de los recursos públicos	En desacuerdo	9.8
	De acuerdo	72.6

Nota. Adaptado de "Lima y Callao según sus ciudadanos: Décimo informe urbano de percepción sobre calidad de vida en la ciudad," por Lima Cómo Vamos, 2019 (http://www.limacomovamos.org/wp-content/uploads/2019/11/Encuesta-2019_web.pdf).

1.3 Propósito de Investigación

El desarrollo de la presente tesis ayudará al sector público, a nivel de municipalidades, a la adecuada identificación de obras urbanas, específicamente para el mejoramiento de parques, pistas y veredas, con el uso de los datos a través de un modelo analítico basado en *machine learning*, con el objetivo de brindar estas obras públicas enfocadas en las necesidades de los ciudadanos y así aportar al desarrollo de las ciudades de forma inteligente.

1.4 Objetivos de la Investigación

1.4.1 Objetivo general

Utilizar la tecnología basada en *machine learning* para la identificación de obras urbanas en la ciudad de Lima.

1.4.2 Objetivos específicos

- Desarrollar un algoritmo basado en *machine learning* que permita identificar los problemas que presenta la ciudadanía limeña sobre obras públicas referentes a parques, veredas y pistas.
- Utilizar la información registrada en las redes sociales de Twitter para identificar las zonas en la ciudad de Lima donde se presentan los problemas referidos a obras públicas, específicamente parques, veredas y pistas.
- Determinar las obras públicas de parques, pistas y veredas que presentan el mayor número de quejas y reclamos utilizando la información obtenida de la red social de Twitter con herramientas analíticas basadas en *machine learning.*

1.5 Preguntas de Investigación

- ¿El uso de herramientas analíticas de *machine learning* permite la identificación de problemas que presenta el ciudadano limeño con referencia a obras públicas como son parques, veredas y pistas?
- ¿En qué medida el uso de herramientas analíticas permite identificar las zonas de la ciudad de Lima donde presentan problemas referidos a obras públicas?
- ¿El uso de la información de la red social de Twitter obtenida con herramientas analíticas de *machine learning* permite determinar las obras públicas de parques, pistas y veredas con mayor número de quejas?

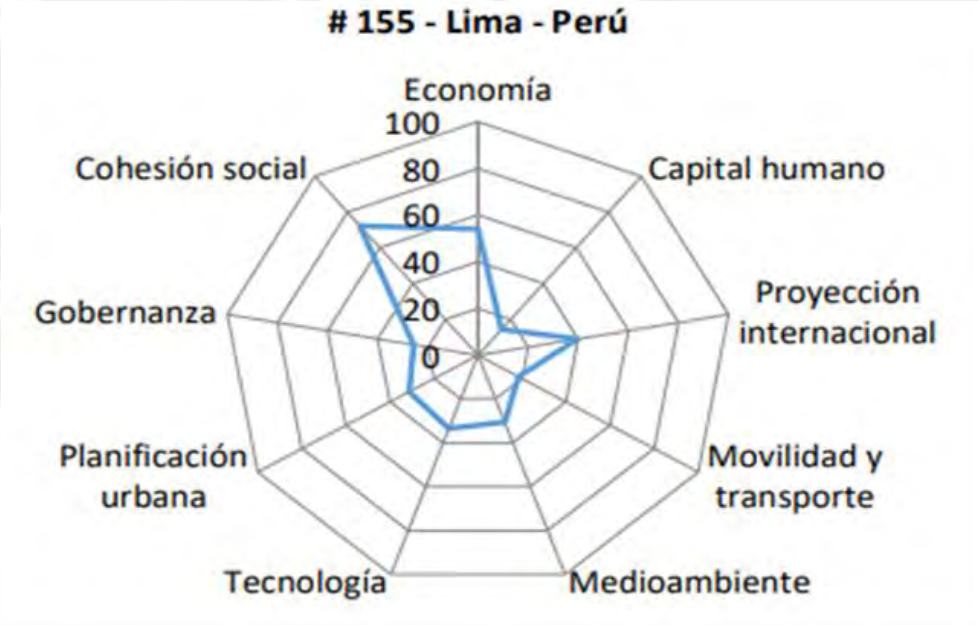
1.6 Justificación del Tema

El tema desarrollado es de gran interés por su aporte en la aplicación de herramientas basadas en *machine learning* para el sector público en Perú, ya que solo el 7% de las empresas públicas peruanas afirma utilizar inteligencia artificial en sus instituciones (Secretaría de Gobierno y Transformación Digital de la PCM, 2021). Esto forma parte de uno de los objetivos de la Estrategia de Inteligencia Artificial planteada por la PCM, además de

aportar como una herramienta inteligente en la identificación de problemas con obras urbanas de mejoramiento de parques, veredas y pistas de la población de Lima, cuya posición en el ranking global de desarrollo de ciudades la ubica en el puesto 155 de 174 ciudades (IESE Business School, 2020), lo que demuestra la existencia de una gran brecha por cubrir en busca de mejorar la calidad de vida de sus habitantes y del desarrollo a una ciudad inteligente. La Figura 1 presenta el resultado obtenido por Lima en cada una de las dimensiones evaluadas en el Índice IESE Cities in Motion (ICIM) publicado el año 2020.

Figura 1

Perfil de la Ciudad de Lima en las Dimensiones del Índice ICIM



Nota. Tomado de "Índice IESE Cities in Motion 2020," por IESE Business School, 2020 (https://media.iese.edu/research/pdfs/ST-0542.pdf).

1.6.1 Justificación tecnológica

Desde el punto de vista tecnológico, el tema de investigación aporta con un caso de uso de *machine learning* para entidades públicas peruanas. El empleo de la analítica permitirá incluir al ciudadano dentro de las decisiones de la Municipalidad de Lima, y esto es un aporte a la dimensión de tecnología, donde en el último resultado la ciudad de Lima se ubicó en el puesto 163 de las 174 estudiadas para esta dimensión (IESE Business School, 2020). La aplicación

de *big data* y *analitics* a través de *machine learning* para la identificación de problemas de obras públicas de mejoramiento de parques, veredas y pistas de Lima Metropolitana tendrá un aporte como un caso de aplicación de esta tecnología en beneficio del sector público.

1.6.2 Justificación social

En el aspecto social, de acuerdo con lo indicado por el Banco Interamericano de Desarrollo (BID) en su camino hacia las ciudades inteligentes, uno de los pasos por seguir es la participación ciudadana, que trata de crear mecanismos para oír a la población en cada etapa a partir de la identificación de los problemas. Es en este tema de investigación en donde se involucra a la ciudadanía considerando sus aportes a través de las redes sociales, justamente para identificar problemas sobre mejoramiento de parques, pistas y veredas, los cuales, de acuerdo con la dimensión de movilidad y transporte, ubican a Lima en el puesto 167 de 174 ciudades estudiadas (IESE Business School, 2020). Algunos de los indicadores que se evalúan en esta dimensión son: índice de ineficiencia en el tráfico, tráfico exponencial, índice de tráfico. Claramente, existen muchos problemas por identificar en este aspecto, y con esta investigación se aporta al abrir un canal más de análisis, como son las redes sociales, para poder brindar soluciones inteligentes poniendo en el centro al ciudadano.

1.6.3 Justificación política

En lo político, el tema aporta directamente a uno de los objetivos de la Estrategia Nacional de IA en el Perú que viene planteando la Secretaría de Gobierno y Transformación Digital de la PCM (2021): “Impulsar en los organismos públicos, la incorporación de la inteligencia artificial en su operación y servicios a los ciudadanos” (p. 65). En la encuesta aplicada para la elaboración de dicha estrategia, solo el 7% de empresas públicas peruanas señaló utilizar inteligencia artificial (Secretaría de Gobierno y Transformación Digital de la PCM, 2021). Con este tema, se impulsa el uso de la IA en una de las empresas públicas que actualmente no la viene aplicando. Además, está alineada con la política del Estado de

ofrecer los servicios públicos teniendo como centro al ciudadano. El uso de *machine learning* en la presente investigación y tomando en consideración las opiniones de los ciudadanos en este análisis, responde a la política que se viene desarrollando en el país.

1.7 Limitaciones

Según la encuesta nacional de hogares ENAHO, al tercer trimestre del 2020, el 70.3% de la población en Perú tiene acceso a Internet (ComexPerú, 2021). Si bien en los últimos años este acceso ha registrado avances significativos, ha sido bastante heterogéneo, lo que ha limitado el uso óptimo de los servicios digitales y el uso de las redes sociales para realizar quejas de toda índole, incluidas las de obras urbanas.

Por la presencia de la covid-19 desde inicio del 2020, el Perú ingresa a varias cuarentenas, que obligan a la población a no salir de sus domicilios. Los ciudadanos prestan atención a la covid-19 y dejan de lados otros temas pendientes de la ciudad, como son las obras urbanas.

Para este trabajo de investigación se limitó el uso de los comentarios de la red social Twitter al utilizar los módulos de desarrollador que puede compartir. Otros datos son restringidos por la misma red social.

El tiempo utilizado para la recolección de datos de la red social Twitter fue de dos meses, aproximadamente, que aunado a las tendencias de la pandemia y factores políticos, limitaron los datos encontrados sobre el tema de investigación, enfocados en las obras urbanas de la ciudad de Lima y en idioma español.

1.8 Delimitación

1.8.1 Delimitación poblacional

Para la investigación se delimitará la población a los comentarios en español de la red social Twitter que contengan referencias al estado de obras públicas en Lima Metropolitana, específicamente a parques, pistas y veredas. En estos comentarios se considerarán aquellos

que reporten problemas o quejas de las obras en mención y a los que hagan referencia a la zona del problema.

1.8.2 Delimitación espacial

La investigación se realizará para las obras públicas que son gestionadas por la Municipalidad Metropolitana de Lima. Las obras relacionadas con parques se ejecutan en la ciudad de Lima y las obras vinculadas a pistas y veredas abarcan toda Lima Metropolitana. Con esta delimitación se hace factible el desarrollo de la investigación, pues permite enfocarse únicamente con la información referente a la ciudad de Lima.

1.8.3 Delimitación temporal

Para propósitos de la investigación, se enmarcará el uso de información de redes sociales con las herramientas basadas en *machine learning* en un espacio temporal de cuatro a seis meses, considerando que la información obtenida de estas fuentes de datos no estructurados es muy amplia y deberá ser delimitada a este espacio de tiempo para un mejor análisis.

1.9 Resumen del Capítulo

El presente trabajo plantea diseñar una solución que apoye a la toma de decisiones para la determinación y priorización de obras públicas, para lo cual se plantean dos componentes:

1. Tecnologías de información: El componente de tecnologías de información hace referencia a las herramientas que se utilizarán, el modelo de gestión de toma de decisiones, donde se incluyen una herramienta de *big data* para la obtención y tratamiento de datos; una herramienta que permita la analítica de la información, con la se podrá determinar necesidades de la población que puedan ser traducidas en proyectos para que se propongan como alternativas de ejecución y, por último, una herramienta que permita la aplicación de inteligencia de negocios que muestre

los resultados de la información analizada y brinde un soporte para la toma de decisiones a través de representaciones gráficas que faciliten el análisis y entendimiento de los resultados obtenidos a través del análisis.

2. La opinión pública en redes sociales: Para que toda esta propuesta tenga los resultados esperados, debe considerarse al valor humano en relación con los comentarios de las redes sociales de los ciudadanos respecto de sus necesidades.

Capítulo II: Revisión de la Literatura

2.1 Introducción

Una ciudad inteligente es aquella que "utiliza la tecnología para prestar de forma más eficiente los servicios urbanos, mejorar la calidad de vida de los ciudadanos y transformar la relación entre entidades locales, empresas y ciudadanos, facilitando una nueva forma de vivir la ciudad" (Grupo Tecma Red, 2015, párr. 5).

La utilización de la tecnología que se acomode a esta necesidad de la ciudad inteligente se refleja en dos términos. El primer término es *Artificial Intelligence* (AI), definida como "la ciencia e ingenio de hacer máquinas inteligentes, especialmente programas de cálculo inteligente" (McCarthy et al., 1956, p. 12). Con este concepto crearon la base y la simulación de procesos de inteligencia humana de manera precisa por máquinas y sistemas informáticos, que permiten replicar por medio de algoritmos ciertas capacidades y actividades humanas.

La evolución de la AI está estrechamente relacionada con el desarrollo del segundo término, tecnologías de *big data,* al complementarse entre sí (Allam & Dhunny, 2019). Según Thakuriah et al. (2017), el término *big data* son los "datos estructurados y no estructurados proporcionados naturalmente a través de actividades transaccionales, operativas, de planificación y sociales que no están diseñadas para la investigación" (p. 5).

Una vez recopilados, los algoritmos de *big data* se emplean para procesar y analizar los datos con AI y *softwares* avanzados con el fin de extraer información útil para descifrar y predecir todo tipo de patrones en tiempo real en entornos urbanos.

El objetivo del desarrollo del presente acápite es introducir y esclarecer algunos conceptos importantes, con la finalidad de evaluar el origen de este tipo de metodologías, sus objetivos, las herramientas que necesitan para su aplicación, y el escenario previo a ello, lo que conllevará a contar con un panorama claro y preciso acerca de lo que se requiere para su exitosa aplicación en el destino propuesto.

2.2 Investigaciones Realizadas

La presente investigación propone la identificación de obras urbanas para la ciudad de Lima. Los antecedentes están divididos en tres pilares que logran el objetivo de esta tesis.

El primer pilar viene en conjunto con el término *smart city*. Dicha evolución del *smart city* implica comprender la necesidad de la comunidad. Fernández (2015) empleó la visión predominante de las *smart cities* como imaginario tecnológico generalizado en la agenda de las políticas urbanas, que ofrezca un marco de análisis para comprender los conceptos que están detrás de una narrativa de las ciudades inteligentes y de la experiencia cotidiana de la vida en la ciudad, y así satisfacer la necesidad del ciudadano. Por su parte, Sota (2018) mencionó que las necesidades actuales de las urbes que se constituyen hoy están haciendo que se generen nuevos retos en la construcción de plataformas de servicios para los ciudadanos.

De otro lado, el crecimiento poblacional acelerado y el desarrollo urbano no planificado han provocado problemas medioambientales, sociales, económicos y políticos que afectan el progreso de las ciudades y la calidad de vida de las personas. El modelo de ciudad inteligente, mencionado en Maldonado et al. (2020), pretendió impactar significativamente en la vida de los ciudadanos y solucionar los retos urbanos más urgentes de la ciudad a través de herramientas tecnológicas y servicios que mejoren la eficiencia de los recursos naturales y económicos.

En esta línea, Maldonado et al. (2020) expusieron la situación en Latinoamérica, en particular de Lima. Señalaron que es necesario definir los factores determinantes para la transformación de un distrito de la ciudad de Lima Metropolitana en una ciudad inteligente que logren la continuidad y sostenibilidad de los servicios inteligentes concedidos, sin importar el periodo de gobernanza de la ciudad, por lo cual se plantea una estrategia de comunicación y empoderamiento de la población mediante campañas de sensibilización.

El segundo pilar considera que el uso de la tecnología para poder obtener la abundante información estructurada de las entidades públicas y la información no estructurada de las redes sociales llevan a utilizar una arquitectura plana para el almacenamiento. Jerí y Sosa (2019) relacionaron el número de arribos en los aeropuertos con la búsqueda de información en Google y YouTube vinculada a viajar. Esta búsqueda fue cuantificada con herramientas como Google Trends. Los índices que se pueden obtener con esta herramienta forman parte del *big data*.

Carrasco (2019) realizó un estudio para la extracción de datos en las redes sociales Facebook y Twitter, enviándolo a cierta base de datos para su posterior procesamiento y limpieza de datos. Utilizó también un disparador que reúne en determinado tiempo los mismos datos, pero seleccionando los campos interesados.

Grández (2017) hizo una investigación de aplicación de minería de datos a través de un *software* que puede encontrar patrones de consumo en una distribuidora de suplementos nutricionales y luego implementar políticas que ayuden a subir el nivel de ventas. La minería de datos puede ser aplicada a cualquier modelado de datos sobre clientes; todo el proceso debe estar encaminado a soportar una futura toma de decisiones estratégicas, que es el principal objetivo. Se puede analizar compra de productos, patrones secuenciales de compra, efectividad de promociones, segmentación de clientes (Fandiño, 2005).

Finalmente, el tercer pilar son las ciencias de la computación, que han tomado un papel fundamental en el desarrollo de las actividades cotidianas del ser humano, presentando herramientas que dan solución a problemas en distintos ámbitos. Marín y Díaz (2015) desarrollaron un método que extrae la opinión de usuarios en Twitter, donde los *tweets* son clasificados en negativos o positivos acerca de los candidatos de dos partidos políticos, para al final predecir cuál sería el resultado de las elecciones. Este tipo de análisis de sentimientos ayuda a aplicar a los datos recogidos de las redes sociales para luego hacer una clasificación y predicción sobre los usuarios. Medhat et al. (2014) analizaron un enfoque en el que un flujo

publicitado de *tweets* desde el sitio de *microblogging* de Twitter se procesa y clasifica según su contenido emocional como positivo, negativo e irrelevante.

El reto de las ciencias computacionales es extraer información útil del medio en el que el ser humano interactúa, con el fin de crear modelos matemáticos, estadísticos o cuantitativos que puedan representar estos procesos naturales del hombre (Gamarra & Ríos, 2018). Sobrino (2018) explicó los fundamentos teóricos, aplicaciones y la relación entre procesamiento de lenguaje natural (NLP, por sus siglas en inglés) y el análisis de sentimiento, así como los métodos utilizados en estas tareas. En su trabajo se implementaron clasificadores de polaridad basados en algoritmos de aprendizaje supervisado (excluyendo a modelos de redes neuronales), para posteriormente comparar los resultados de los métodos más utilizados.

Del mismo modo, Olarte y Casaverde (2020) implementaron una arquitectura para identificar masivamente opiniones de las publicaciones en Twitter mediante máquinas de soporte vectorial, para obtener como resultado una calificación positiva o negativa. Para el desarrollo del trabajo se utilizó el DataSet TASS de la Sociedad Española para el Procesamiento del Lenguaje Natural (SEPLN). Se implementaron los módulos respectivos de preprocesamiento de texto, extracción de características, aprendizaje automático y la manipulación de datos con librerías de Python. Se aplicó el algoritmo SVM (*Support Vector Machines*: máquinas de soporte vectorial) para la identificación de opiniones, debido a que presenta mejor resultado para esta tarea de identificación de opinión.

Gordon (2018) seleccionó otros algoritmos de clasificación*: Linear Support Vector Machine* (LSVM)*, K-Nearest Neighbor* (K-NN)*, Multinomial Naïve Bayes* (NB) *y Logistic Rregression* (LR). Los algoritmos fueron entrenados utilizando textos en idioma español, y superaron el 70% de precisión en su etapa de entrenamiento. El modelo de clasificación de mejor rendimiento fue *Multinomial NB*, debido a sus características de entrenamiento con textos cortos, seguido por SVM como el segundo mejor.

2.3 *Smart City*

Las políticas públicas o estrategias de intervención están dirigidas a cubrir las demandas de un determinado sector de la población en un territorio determinado. Según la Organización de Naciones Unidas (ONU, 2018), por primera vez más de la mitad de la población mundial vive en ciudades (54.6%) y el proceso de migración rural hacia lo urbano iniciado en la primera revolución industrial está en constante crecimiento; se proyecta que para el año 2050, más del 70% de personas vivirán en ciudades. De acuerdo con el censo de población realizado el año 2017, el 79.3% de los habitantes del Perú vive en centros poblados urbanos y solo el 20.7% lo hace en centros poblados rurales (ComexPerú, 2018).

La definición de *smart city* varía según su origen académico, gobierno, instituciones, etc.; destaca la necesidad de crear mejores condiciones de habitabilidad para los ciudadanos a través de decisiones de gobierno que se apoyen en el uso de tecnología. Como parte de las actividades del World Smart City Forum, realizado en julio del 2016, los representantes de la International Organization for Standarization (ISO), la Comisión Electrotécnica Internacional (IEC), la Unión Internacional de Telecomunicaciones (ITU), el Instituto de Ingeniería Eléctrica y Electrónica (IEEE) y el Instituto Europeo de Normas de Telecomunicaciones (ETSI, por sus siglas en inglés) se reunieron con el objetivo de acelerar y alinear el trabajo de estandarización de las ciudades inteligentes para su desarrollo (Piva, 2016).

Dentro de la variedad de definiciones, se puede indicar que el objetivo es buscar un beneficio para el ciudadano y el crecimiento ordenado de las ciudades. Según Caragliu y Del Bo (2019):

> Una ciudad puede definirse como ‘Smart’ cuando invierte en capital humano y social; infraestructura de comunicación tradicional (transporte) y moderna (TIC) fomentando el desarrollo económico sostenible y una alta calidad de vida; con una gestión inteligente de los recursos naturales, a través de la acción y el compromiso participativo. (p. 5)

Por otro lado, la compresión de la inteligencia o *smartness* de las ciudades depende del campo en la que se estudie (Nam & Prado, 2011) y está condicionada por factores tecnológicos, humanos e institucionales. La Agenda 2030 sobre Desarrollo Sostenible, aprobada en 2015 por la Organización de las Naciones Unidas (ONU), mencionó que los países y sociedades emprenden un nuevo camino para mejorar la vida de todos. Esta agenda está basada en 17 objetivos de desarrollo sostenible (ODS), dentro de los cuales el número 11, Ciudades y Comunidades Sostenibles, busca crear espacios inclusivos, seguros, resilientes y sostenibles (ONU, s.f.).

Un proyecto integral de *smart city* debe de incorporar aspectos relativos a la gobernanza, la infraestructura y el capital humano y social, para lograr fomentar un desarrollo sostenible e integrado (Bouskela et al, 2016).

La utilización de las *smart cities* viene siendo impulsada por los proveedores de tecnología, con la implementación de sistemas basados en Internet de las cosas (IoT: *Internet of the Things*) y la implementación de sistemas que tienen como base las tecnologías de información. Estos dos caminos se unen a otro más amplio que consiste en la mejora de la gestión de los centros urbanos (Cohen, 2015). Finalmente, todo se engloba en un cambio de procesos internos para poder lograrlos. Así, al realizar esta implementación se inicia el *smart city.*

Es vital el rol de los municipios y sus líderes para determinar el futuro deseado para sus comunidades y el ingreso al uso de tecnologías de sus habitantes. La evolución de las *smart cities* se complementa con la mejora de los servicios estatales, la calidad y el énfasis en los problemas de transporte, salud pública y ecología (Vishnivetskaya & Alexandrova, 2019).

La evolución de la *smart city* exige una mayor participación de los ciudadanos y su involucramiento en las soluciones generales. Se debe tener especial cuidado en incluir las demandas de cada grupo de *stakeholders*. La abundante información y datos en esta evolución de las ciudades establece nuevos retos para las comunidades, como el reforzamiento de los sistemas de seguridad de manejo de datos.

2.3.1 *Smart city* en Perú

El Perú es un territorio poco explorado en materia de *smart city*, las instituciones públicas han tomado algunos conceptos desarrollados por organizaciones extranjeras como el Banco Interamericano de Desarrollo (BID) y la Unión Internacional de Telecomunicaciones (UIT).

En el Perú, el Proyecto de Ley N°1630/2016-CR, que promueve y garantiza la ejecución del Plan Nacional de Ciudades Inteligentes, presentado en el Congreso de la República en el periodo parlamentario 2016-2019 y dictaminado por la Comisión de Transporte y Comunicaciones, recogió la siguiente definición de Bouskela et al. (2016):

> Una ciudad inteligente es aquella que coloca a las personas en el centro del desarrollo, incorpora tecnologías de la información y comunicación en la gestión urbana y usa estos elementos como herramientas para estimular la formación de un gobierno eficiente que incluya procesos de planificación colaborativa y participación ciudadana. Al promover un desarrollo integrado y sostenible, las *smart cities* se tornan más innovadoras, competitivas, atractivas y resilientes, mejorando así las vidas. (p. 33)

El Ministerio de Transportes y Comunicaciones (MTC), en el Seminario Peruano-Alemán Smart City realizado en noviembre del año 2015, definió a las ciudades inteligentes como "lugares donde las tecnologías de información son usadas para mejorar la calidad y desempeño de los servicios urbanos, reducir costos y consumo de recursos, e involucrar más efectiva y activamente a sus ciudadanos" (MTC, 2015, p. 6).

2.3.2 Indicadores de *smart city*

Según Jorge Guerra, docente de la Universidad Nacional Mayor de San Marcos (UNMSM) ante la variedad de definiciones se debería considerar la establecida por la norma ISO 37120, actualizada a julio del 2018, que señala y estandariza los indicadores para los servicios y calidad de vida en las comunidades y las ciudades (Apolitano, 2019). La ISO 37122 define seis indicadores para una *smart city*, los cuales buscan estandarizar sus procesos

de desarrollo. Tales indicadores son los siguientes: (a) *smart economy*, (b) *smart people*, (c) *smart governance*, (d) *smart mobility*, (e) *smart environment*, y (f) *smart living* (International Standardization Organization [ISO], 2019).

2.4 Analítica de la Ciudad

La analítica predictiva surge como una idea de negocio, donde es utilizada para personalizar propuestas de valor en tiempo real, las cuales son pensadas para los consumidores y, en algunas ocasiones, su alcance puede ser individual y grupal. Fortalece y tiende a prolongar la participación de mercado de las empresas, sobre todo cuando los nativos digitales se volvieron un arma importante para su desarrollo.

La analítica es un campo incluyente y multidimensional que utiliza matemáticas, estadística, modelos predictivos y técnicas de aprendizaje basado en *machine learning,* para hallar patrones y conocimientos significativos en datos grabados (SAS Institute, 2018).

Esta herramienta, al ser trasladada a un entorno urbano, analiza a diferentes escalas el comportamiento de los ciudadanos, a través de datos de geolocalización, sensores y cámaras instaladas en la ciudad, redes sociales, entre otros. Así, este cúmulo de información permite desarrollar modelos de gestión y servicios.

Al obtener una gran cantidad de información tanto estructurada como no estructurada, se puede predecir el comportamiento de los agentes y presentar alternativas de servicios que mejoren su experiencia en la ciudad en distintos ámbitos.

Los tres tipos de analítica que existen son los siguientes:

- Analítica descriptiva: Desarrolla modelos para entender los hechos y su razón.
- Analítica predictiva: Identifica probables escenarios futuros basados en uso de datos, estadística y *machine learning*.
- Analítica prescriptiva: Presenta posibles soluciones a realidades futuras identificadas permitiendo actuar en tiempo real.

2.5 Inteligencia Artificial

Se entiende por inteligencia artificial a aquellos sistemas informáticos que pueden percibir su entorno, pensar, aprender y actuar en respuesta a lo que detectan y a sus objetivos. Se trata de la simulación por parte de una máquina de un proceso mental que le permite tomar decisiones y efectuar tareas propias de los seres humanos por medio de algoritmos, los cuales son capaces de ingerir y analizar datos para convertirlos en información relevante.

La inteligencia artificial representa un conjunto de disciplinas de *software*, lógica, informática y filosofía que están destinadas a hacer que los computadores realicen funciones que se pensaba que eran exclusivamente humanas, como percibir el significado en el lenguaje escrito o hablado, aprender, reconocer expresiones faciales, etc.

El campo de la inteligencia artificial tiene una larga historia tras de sí, con muchos avances anteriores, como el reconocimiento de caracteres ópticos, que en la actualidad se consideran como algo cotidiano (Hewlett Packard Enterprise, 2018).

Según López (2007), la IA es una rama de las ciencias computacionales encargada de estudiar modelos de cómputo capaces de realizar actividades propias de los seres humanos con base en dos de sus características primordiales: el razonamiento y la conducta.

Por su parte, autores como McCarthy et al. (1955) y Torra (2011) definieron la inteligencia artificial como el de construir una máquina que se comporte de tal manera que, si el mismo comportamiento lo realizara un ser humano, este sería llamado inteligente. De acuerdo con Rouse (2019), la Inteligencia Artificial (IA), en función de sus objetivos finales de investigación, se puede clasificar en:

- Inteligencia artificial débil: Se considera que los ordenadores únicamente pueden simular que razonan, y únicamente pueden actuar de forma inteligente.

- Inteligencia artificial fuerte: Se considera que un ordenador puede tener una mente y unos estados mentales, y que, por lo tanto, un día será posible construir uno con todas las capacidades de la mente humana. Este ordenador será capaz de razonar, imaginar, etc. En la actualidad, en el ámbito de la extracción y análisis de grandes volúmenes de datos se utiliza la IA, cuyo objetivo en los ámbitos descritos anteriormente es el tratamiento de datos de forma masiva y automática, que pudieran contener un alto grado de complejidad.

Los datos generados por los usuarios en Internet mediante sus interacciones y comportamiento pueden convertirse en una fuente de información si se analizan sus quejas, sentimientos, preferencias, con la finalidad de crear servicios personalizados o tomar decisiones empresariales, las mismas que están basadas en datos obtenidos mediante el uso de IA (Costa, 2015). Con las herramientas y recursos tecnológicos disponibles, y cada vez más asequibles, para aplicar la inteligencia artificial y la ciencia de datos a las decisiones empresariales, se pretende ayudar en la gran tarea de identificar qué sectores e industrias son los que poseen mayor interactividad y gran connotación positiva en los comentarios, que demuestren un índice de estabilidad, de acuerdo con el análisis de *tweets* pertenecientes a un sector e industria. Todo ello será posible gracias a diferentes algoritmos de aprendizaje automático (*machine learning*).

Según Russell y Norvig (2004), existen varios tipos de inteligencia artificial, que son los siguientes:

- Sistemas que piensan como humanos*:* Estos sistemas tratan de emular el pensamiento humano; por ejemplo, las redes neuronales artificiales, o la automatización de actividades vinculadas a procesos de pensamiento humano, como la toma de decisiones, resolución de problemas y aprendizaje.

- Sistemas que actúan como humanos: Estos sistemas tratan de actuar como humanos, es decir, imitan el comportamiento humano; por ejemplo, la robótica (el estudio de cómo lograr que los computadores realicen tareas que, por el momento, los humanos hacen mejor).
- Sistemas que piensan racionalmente: Mediante la lógica (idealmente), estos sistemas tratan de imitar el pensamiento racional del ser humano; por ejemplo, los sistemas expertos (el estudio de los cálculos que hacen posible percibir, razonar y actuar).
- Sistemas que actúan racionalmente: Tratan de emular de forma racional el comportamiento humano; por ejemplo, los agentes inteligentes, que están relacionados con conductas inteligentes en artefactos.

La IA se divide en dos escuelas de pensamiento: la inteligencia artificial convencional y la inteligencia basada en comportamiento. Tales escuelas serán reseñadas a continuación.

- Inteligencia artificial convencional: Se conoce también como IA simbólico-deductiva. Está basada en el análisis formal y estadístico del comportamiento humano ante diferentes problemas:
  - Razonamiento basado en casos: Ayuda a tomar decisiones mientras se resuelven ciertos problemas concretos y, aparte de que son muy importantes, requieren de un buen funcionamiento.
  - Sistemas expertos: Infieren una solución a través del conocimiento previo del contexto en que se aplica y se ocupan de ciertas reglas o relaciones.
  - Redes bayesianas: Propone soluciones mediante inferencia probabilística.
- Inteligencia artificial basada en comportamientos: Esta inteligencia contiene autonomía y puede autorregularse y controlarse para mejorar.

2.6 *Machine Learning*

El *machine learning* es un tipo de inteligencia artificial que permite a las máquinas aprender por sí mismas, sin necesidad de ser programadas por una persona previamente. Con el aprendizaje automático, las máquinas son capaces de aprender de forma constante y no solo para realizar las tareas para las que han sido creadas en un primer momento. También es definido como un algoritmo que tiene la finalidad de predecir sucesos futuros que son desconocidos para el sistema (Aguilar & Vásquez, 2016).

Por su parte, Mitchell (1997) señaló que el *machine learning* es el estudio de algoritmos de computación que mejoran automáticamente su rendimiento gracias a la experiencia. Es decir, un programa informático aprende sobre un conjunto de tareas, y usa una medida de rendimiento si su desempeño mejora.

Así mismo, Candia (2019) definió al *machine learning* como una manera en que las computadoras pueden aprender por sí solas, pero con ayuda de los humanos. Y, a su vez, estos pueden obtener patrones para construir modelos y predecir comportamientos anticipadamente.

En conclusión, el *machine learning* ofrece formas de acciones de predicción y puede ser utilizado en el presente trabajo de investigación, ya que por medio de algoritmos se podrán detectar patrones que ayudarán a determinar su objetivo.

El *machine learning* o aprendizaje automático tiene tres categorías:

- Aprendizaje supervisado: Se otorgan preguntas, características y respuestas a los algoritmos, denominados etiquetas, con la finalidad de que puedan combinarlos y hacer predicciones (Candia, 2019). Su objetivo es generar modelos predictivos mediante el modelo de análisis de datos como la etiquetación y clasificación (Mamani, 2019). Según García (2019), los tipos de aprendizaje supervisado son de regresión (indica la tendencia de un grupo de datos continuos) y de clasificación (patrones).

- Aprendizaje no supervisado: Requiere de los datos de entrada para que pueda almacenar y clasificar los patrones de comportamiento (Candia, 2019).
- Aprendizaje reforzado: emite un entorno de control para identificar la conducta dentro de una situación única (García, 2019). El objetivo del aprendizaje es verificar si tiene la capacidad de adaptarse al entorno.

2.6.1 Clasificadores supervisados

Los métodos supervisados son los más utilizados, porque constan de un proceso de entrenamiento con base en ejemplos entregados por humanos, donde se indica explícitamente al sistema, a qué clase pertenece cada ejemplo (Manupati et al., 2013).

Cada comentario que define una expresión hecha por las personas es explicado sobre la base de características que pueden ser de una palabra (unigramas), conjunto de dos palabras (bigramas), etc. Ciertos sentimientos son expresados con dos o más palabras que abarcan características importantes, como la negación de una frase (Manupati et al., 2013).

Los métodos supervisados constan de dos etapas: el entrenamiento del sistema y la clasificación de los nuevos datos entregados. Para pasar de etapas son necesarias la selección y extracción de las *features* a cada dato. Uno de los métodos más utilizados son las *support vector machine* (Manupati et al., 2013).

2.6.2 *Support vector machine* (SVM)

Las máquinas de vectores de soporte o *support vector machines* (SVM) son altamente utilizadas en la clasificación y detección de sentimientos (Montesinos, 2014). Se basan en métodos de *kernel*, los cuales toman los datos y lo ponen dentro de un espacio de características apropiado (ver Figura 2). Se utilizan algoritmos lineales para determinar patrones no lineales (González et al., 2017).

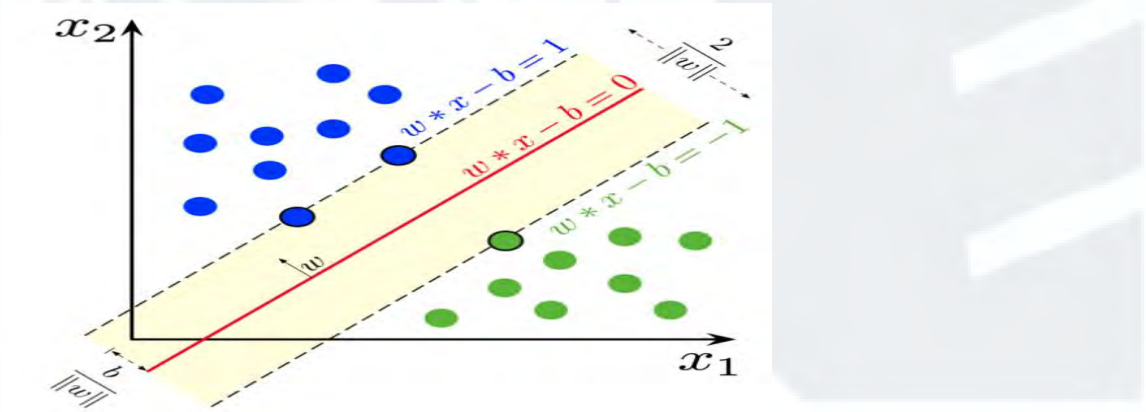
Estos algoritmos aprenden de un conjunto de datos de entrenamiento, que se aplican a una serie de datos nuevos. Los datos son clasificados mediante la generación de hiperplanos

que tienen como objetivo minimizar el porcentaje de error del conjunto de entrenamiento y maximizar el margen de separación (González et al., 2017).

Las SVM fijan el criterio de separación entre clases que estén lo más lejos posible de cualquier dato. La distancia del punto de decisión al punto más cercano es el margen del clasificador. El método queda definido por una función de decisión que involucra un subconjunto de características o datos (*support vectors*) que definirán la posición del separador. La decisión del límite o margen es bastante importante, ya que los datos que queden en torno a este tendrán una menor probabilidad de ser catalogados correctamente.

Figura 2

Support Vector Machine (SVM)



Nota. Adaptado de "Tutorial sobre máquinas de vectores soporte - SVM" (p. 6), por E. Carmona, 2016 (https://www.researchgate.net/publication/263817587_Tutorial_sobre_Maquinas_de_Vectores_Soporte_SVM).

Se define el hiperplano *H* dado que:

H: w.x+b = 0

Posteriormente, se define H1 y H2 para los planos:

H1: w.x + b = 1

H2: w.x + b = -1

Donde:

W = Vector normal al hiperplano

B = Constante de posición del plano respecto al origen de coordenadas

Los puntos en los planos H1 y H2 son los vectores de soporte.

El objetivo del algoritmo SVM es encontrar en el *hiperplano* el máximo margen que separa el conjunto de puntos de datos que tienen el rótulo de clase de X2=1 del conjunto de puntos de datos que tienen el rótulo de clase X2=−1, de tal manera que la distancia entre el *hiperplano* y la muestra de los puntos de datos de cualquier clase más cercana se maximiza. Estos puntos de datos de muestra se conocen como vectores de soporte.

2.6.3 Matriz de confusión

Una matriz de confusión es una técnica para resumir el rendimiento de un algoritmo de clasificación. Consta de una tabla de dos dimensiones: real y predicción. Las columnas de la matriz indican la clase observada o real, y las filas indican la clase predicha. La dimensión real se refiere a los datos reales, los cuales van a ser comparados con las predicciones. Estas predicciones se hacen con los modelos de clasificación como el SVM (ver Tabla 5).

Tabla 5

Matriz de Confusión

		Clase	
		+	-
Predicción de clases	+	PT	PF
	-	NF	NT

Nota. PT= N° de positivos definidos como positivos correctos; NT= N° de negativos definidos como negativos correctos; NF= N° de positivos definidos como negativos incorrectos; PF: N° de negativos definidos como positivos incorrectos. Adaptado de "Diseño de un proceso de alertas tempranas para disminuir las deserciones de los estudiantes de primer año en una institución de educación superior," (Tesis de maestría), por F. P. Miranda, 2019 (http://repositorio.uchile.cl/bitstream/handle/2250/172649/Dise%C3%B1o-de-un-proceso-de-alertas-tempranas-para-disminuir-las-deserciones.pdf?sequence=1).

A partir de la información de la matriz de confusión se determina la eficiencia de los algoritmos, que se reseñan a continuación (Miranda, 2019):

- *Accuracy*: Es el número de predicciones correctas realizadas por el número total de registros. Se refiere a lo cerca que está el resultado de una medición al valor verdadero.

- *Presicion*: Se define como la proposición de predicciones positivas con respecto al número de observaciones que son realmente positivas.
- *Recall*: Es la proposición de observaciones positivas predichas correctamente a todas las observaciones en la clase real.
- *Sentitivity*: Son valores que pueden identificar la capacidad de limitar los casos positivos de los casos negativos. Se entienden como la habilidad de tener sensibilidad con los sucesos positivos.
- *F1-score*: En este cálculo se busca obtener una puntuación única que represente las variables de "precisión" y "sensibilidad", a fin de obtener un ponderado. Este valor será la base de optimismo del modelo que se realizará y por referencias de estudios previos.

2.6.4 Corpus

Para el caso de los métodos supervisados es necesario contar con un *dataset* o corpus previamente categorizado. La técnica de creación de corpus incluye una cantidad determinada de comentarios en las redes sociales sobre el tópico por investigar (Sobrino, 2018).

Los mensajes de corpus antes de ser valorarizados pasan por un proceso de limpieza de datos, se quitan los comentarios que no aportan valor al *dataset*. Los corpus pueden ser clasificados en dos o más categorías según el interés. En ocasiones se requiere la opinión de un profesional para determinar el valor de la clasificación de los comentarios para agregarles valor al *dataset* (Sobrino, 2018).

2.7 *Big Data*

La definición de *big data* se ha venido ampliando en el tiempo, de acuerdo con su aplicación y mayor extensión en distintos casos de uso, pero se puede partir de una definición original que la mayoría de los autores toma como base para iniciar en el concepto de *big data.* De acuerdo con Laney (2001), *big data* es un término utilizado para referirse a la gran

cantidad de datos con tres características denominadas como las 3 V (volumen, velocidad y variedad), las cuales superan los parámetros de cualquier *software* tradicional en la gestión de datos. En otras palabras, lo que se busca transmitir con este concepto es que existe una gran cantidad de datos que ingresan rápidamente al sistema para su procesamiento provenientes de distintas fuentes de origen y que, además, vienen en muy variados tipos de formatos (Kalbandi & Anuradha, 2015).

Con el tiempo, al primer concepto planteado por Laney (2001) se le agregaron otras V, que representan características que deben ser consideradas para que el concepto de *big data* abarque con amplitud lo que realmente se viene desarrollando en la práctica (Kalbandi & Anuradha, 2015). La siguiente característica agregada a las tres iniciales es la de *valor*, haciendo referencia a que toda esta gran cantidad de datos debe aportar algún tipo de valor para realizar su procesamiento. Como una quinta característica adicional a este concepto se incorpora la *veracidad*, que es la validez de los datos que ingresan al sistema. Teniendo estas dos V adicionales, el concepto de *big data* se amplía a cinco características. No obstante, una mayor revisión sobre la definición mostrará que distintos autores van añadiendo más características (Kalbandi & Anuradha, 2015).

En cuanto al volumen, es la cantidad de datos por analizar, los cuales varían sin que se tenga definida una cantidad en específico, pero hoy en día existen empresas que manejan información en terabytes hasta petabytes. Es importante mencionar que el volumen de datos debe superar la capacidad de administración de aquellos motores de bases de datos relacionales (Kalbandi & Anuradha, 2015).

La velocidad hace referencia a cuán rápido se generan los datos y a su velocidad de distribución. Hoy en día, la generación de datos a través de redes sociales, *smartphones*, sensores y el auge del IoT (Internet de las cosas) lleva consigo que muchos dispositivos se

encuentren conectados a la red, y la cantidad de información vaya en aumento exponencialmente (Kalbandi & Anuradha, 2015).

Por su parte, la variedad tiene tres enfoques. El primero alude a la variedad en la fuente de los datos, pues en la actualidad existen muchas fuentes de datos, distintas redes sociales, información pública del Gobierno, además de las fuentes internas de una empresa. El segundo enfoque de variedad hace referencia a las distintas estructuras en que se presentan los datos. En el pasado, la información venía principalmente en data estructurada almacenada en bases de datos relacionales; hoy se cuenta además con data semiestructurada y data no estructurada, lo cual lleva al tercer enfoque de variedad, referido a los distintos tipos de formatos en los que se presentan los datos, como imágenes, videos, audios. El *streaming*, por ejemplo, es en la actualidad un servicio que genera mucha información (Kalbandi & Anuradha, 2015).

A estas tres características mencionadas se le agrega el valor, ya que se requiere que los datos muestren utilidad para el negocio; si los datos no logran aportar a la toma de decisiones, no servirá la inversión realizada en su procesamiento y análisis (Kalbandi & Anuradha, 2015).

Por último, la veracidad se refiere a la certeza que tienen los datos, disminuyendo la incertidumbre y tomando en consideración la calidad de estos. De nada sirve trabajar con datos falsos que llevarán al error en el análisis de información y, por tanto, al error en la toma de decisiones para el negocio (Kalbandi & Anuradha, 2015).

2.8 Redes Sociales

Las redes sociales son estructuras formadas en Internet por personas u organizaciones que se conectan a partir de intereses o valores comunes. A través de ellas, se crean relaciones entre individuos o empresas de forma rápida, sin jerarquía o límites físicos.

El concepto de red social, desde un enfoque analítico, está más cerca de la idea de sociograma de Moreno (1940), cuando definió a la estructura social como "la red de las relaciones existentes entre las personas implicadas en una sociedad" (Requena, 1989, p. 138).

Genéricamente, una red social puede concebirse como una estructura social formada por individuos que están vinculados por algún motivo, ya sea amistad, parentesco, ideas, aficiones, relaciones de trabajo, docentes, etc. Para la caracterización de las redes sociales se establecen dos conceptos fundamentales: los nodos, que caracterizan a los individuos en la red, y los enlaces o aristas, que vinculan a los individuos (Vidal & Vialart, 2013).

2.9 Resumen del Capítulo

Las redes sociales recogen mucha información relevante sobre nuestra sociedad, y los *microblogging* son una parte importante de la comunicación actual. Esto permite que los usuarios puedan publicar una opinión acerca de un determinado tema haciendo uso de Internet y los sitios web en un repositorio grande de información.

Las redes sociales como Twitter, Google+, Facebook y WhatsApp contienen gran cantidad de publicaciones en sus sitios web. Esto hace de estas plataformas una fuente para exploración de información haciendo uso de métodos de inteligencia artificial, debido a la masiva cantidad de información, que en casos como Twitter llegan a ser millones de comentarios por día alrededor del mundo. Esta vasta información debe ser aprovechada para interpretar los datos con la finalidad de saber cuántas opiniones son realizadas positivamente y cuántas son negativas.

Sin embargo, se carece de una herramienta capaz de hacer un análisis de todos estos comentarios y determinar una polaridad (si una opinión es positiva o negativa). Por ello, se propone desarrollar una herramienta de identificación de texto en Twitter usando técnicas de aprendizaje automático y procesamiento del lenguaje natural o *natural language processing* (NLP); procesamiento basado en texto con análisis de sentimientos o *sentiment analysis*

(SA); y técnica de *machine learning* de máquinas de soporte vectorial o *support vector machine* (SVM).

Los países latinoamericanos tienen la capacidad para aprovechar todo el potencial de la IA; sin embargo, debido a las limitaciones sociales y económicas se ha realizado poca inversión en el Gobierno, la industria y la investigación para avanzar en IA. Esto es una desventaja, ya que la IA es una tecnología importante y fundamental en la cuarta revolución industrial, y dada su naturaleza multipropósito, poder exponencial y capacidad predictiva podría ser una herramienta importante para abordar diversos desafíos que afectan el desarrollo de un país.

Dependencias de Gobierno como seguridad pública y los servicios públicos tienen una necesidad particular del *machine learning,* porque requieren múltiples fuentes de datos de las que se pueden extraer *insights*[1]. Esto permite incrementar la eficiencia y mejorar los servicios que son requeridos y necesarios para el ciudadano. Asimismo, el aprendizaje automático puede ayudar a detectar problemas de fraude y minimizar los riesgos de realizar una obra en un lugar donde no se requiera.

[1] Un *insight* es una comprensión de las necesidades reales expresadas y no expresadas por los clientes.

Capítulo III: Metodología

La metodología utilizada para llevar a cabo el presente trabajo consideró dos aspectos relacionados entre sí: el primero es la metodología de la investigación y el segundo es el método de trabajo que se utilizó, dentro del cual se aplicó un modelo de investigación basado en *machine learning*. Este enfoque se explica por la naturaleza de la investigación, que utiliza datos no estructurados obtenidos de una red social para el entrenamiento del modelo de *support vector machine*. Este modelo permitió realizar una clasificación a los datos no estructurados y previamente preparados dentro de una de las etapas del método de trabajo aplicado, para luego ejecutar un entrenamiento al modelo y de esta forma lograr el aprendizaje del modelo.

El modelo de *support vector machine* utiliza una serie de parámetros para obtener su resultado; por este motivo, se eligen los mejores parámetros para buscar el mejor resultado que pueda proporcionar el modelo. Esto se efectuó con distintas iteraciones en busca de los mejores valores para asignar a cada uno de los parámetros del modelo. Una vez definidos estos parámetros, se procedió con el entrenamiento del modelo y luego con una validación, para conocer qué tan precisas son las predicciones que realiza.

Los resultados obtenidos a partir de las predicciones del modelo mostraron con un porcentaje de precisión la positividad y negatividad de los comentarios referentes a las obras del alcance de la investigación, que son parques, veredas y pistas. Con base en este resultado se identificaron las zonas a las que hacen referencia los comentarios negativos a estas obras en la ciudad de Lima, utilizando para este fin la data de localización obtenida de la red social.

Luego de obtener los resultados del modelo de *support vector machine* y la identificación de las zonas, se efectuó un análisis para determinar el grado de precisión que brindó el modelo en sus resultados y así responder a las preguntas de investigación. Todos

estos procedimientos no son tradicionales para una investigación, debido al uso de *machine learning*, por lo que es importante resaltar esta estructura dentro de la metodología.

3.1 Metodología de Investigación

El trabajo de investigación tiene un alcance exploratorio, debido a que este tipo de problema ha sido poco estudiado. Las referencias encontradas sobre la aplicación de *machine learning* en el sector público peruano para la identificación de obras públicas en Lima corresponden a distintos casos de uso en otros sectores privados (Hernández et al., 2014).

Como el trabajo de investigación comprende el análisis de una gran cantidad de información no estructurada proveniente de redes sociales, la aplicación de métodos específicos tradicionales para el análisis no fue factible, y como se utilizaron técnicas de entrenamiento de datos para la definición de un modelo, se empleó un método que permite este tipo de trabajo, por lo que se recurrió a la metodología fundamental de IBM para la ciencia de datos (Rollins, 2015).

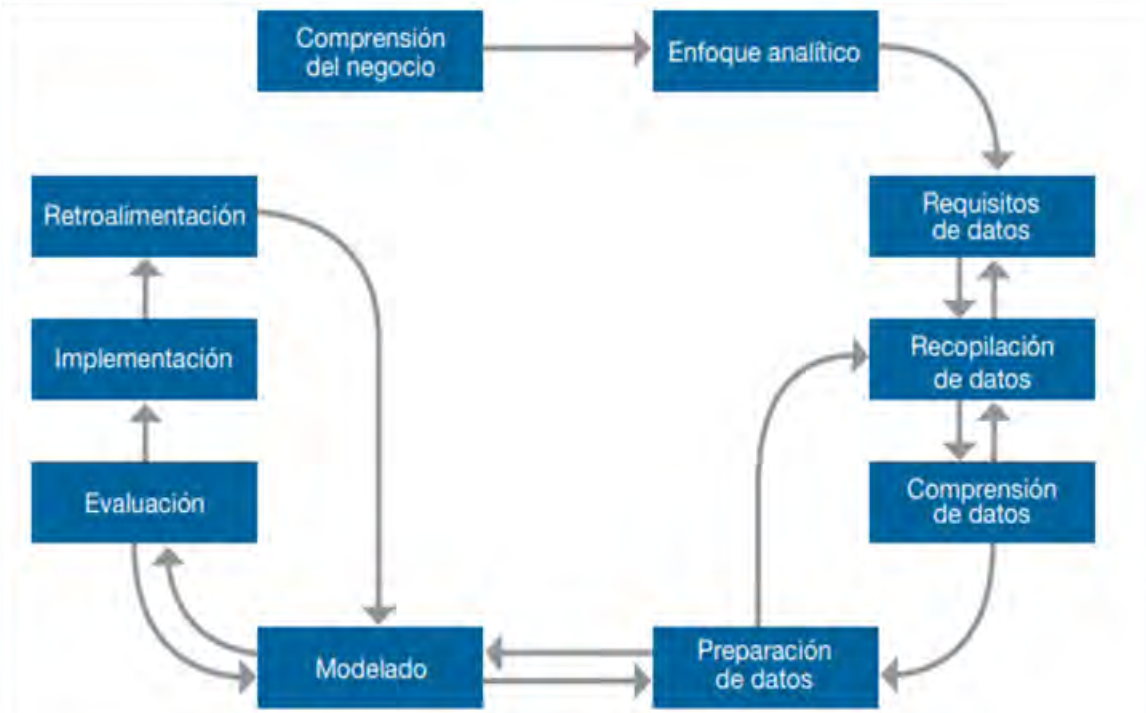
3.2 Método de Trabajo

Existen distintas formas de trabajo o metodologías que se aplican para el análisis de datos, que incluyen modelos analíticos de *machine learning*, en su mayoría muy similares. Para este trabajo se tomó como referencia la metodología propuesta de IBM, de John B. Rollins, Metodología Fundamental para la Ciencia de Datos, al tratarse de un referente en la industria del análisis de datos. En la Figura 3 se muestra la secuencia de etapas de la metodología por utilizar (Rollins, 2015).

La metodología propuesta por IBM consta de 10 etapas. Inicia con la comprensión del negocio; esta primera etapa lo que busca es definir el problema, objetivos y requisitos de solución. La segunda etapa, el enfoque analítico, busca expresar el problema bajo el contexto de técnicas estadísticas y de aprendizaje automático.

Figura 3

Metodología Fundamental para la Ciencia de Datos



Nota. Adaptado de *Metodología fundamental para la ciencia de datos* (p. 3), por J. B. Rollins, 2015, IBM Corporation (https://www.ibm.com/downloads/cas/WKK9DX51).

Las siguientes cuatro etapas están relacionadas con los datos en sí mismos: (a) requisitos de datos, (b) recopilación de datos, (c) preparación de datos, y (d) comprensión de datos. Tales etapas hacen referencia a los requerimientos de los datos, como son el formato y representaciones que deben tener los datos para realizar el análisis. La recolección de los datos son las fuentes que se utilizaron como datos estructurados, no estructurados o semiestructurados, porque la investigación debe enfocarse en obtener aquellos datos que sean relevantes para resolver la necesidad del negocio. En la compresión de los datos, se utiliza la estadística descriptiva y herramientas gráficas para poder obtener los primeros *insights*; se evalúa la calidad de los datos y descripción de la metadata. Por último, la preparación de los datos consiste en todas aquellas actividades realizadas para construir el conjunto de datos que fue utilizado en el modelamiento analítico. Estas actividades pueden ser la limpieza de datos,

cambio de formatos, eliminación de duplicados, combinación de datos de distintas fuentes, y creación de nuevas variables a partir de los datos obtenidos.

La séptima etapa, el modelado, consiste en el desarrollo del modelo analítico aplicando modelos descriptivos o predictivos de acuerdo con lo definido en la segunda etapa del enfoque analítico. En esta etapa, si el modelo es predictivo se deberá entrenar al modelo con data histórica para obtener resultados del modelo. La octava etapa, evaluación, consiste en la evaluación del modelo de la etapa anterior, buscando calidad y la seguridad de que aborda el problema de manera adecuada. Para los modelos predictivos se realiza un conjunto de pruebas independientes al entrenamiento para validar sus resultados.

Las dos últimas etapas de la metodología, implementación y retroalimentación, consisten en llevar a producción el modelo debidamente probado y brindar un continuo *feedback* a los resultados del modelo para ir mejorándolo de acuerdo con los cambios que se presenten más adelante, con el objetivo de que el modelo continúe otorgando valor. Estas dos últimas etapas no fueron consideradas para el presente trabajo porque exceden el alcance del trabajo de investigación, por lo que la metodología finaliza en los resultados obtenidos en la etapa de evaluación.

3.2.1 Comprensión del negocio

Para este trabajo de investigación, se ha elaborado un formulario en donde se estructura la información que fue levantada para esta etapa de *comprensión del negocio*, que busca definir el problema, los objetivos, los requisitos de la solución y los campos adicionales que ayudaron a comprender y elaborar la propuesta (ver Apéndice A).

Para el presente caso, al tratarse de un tema público y no de una empresa en particular, debe tenerse en claro el contexto, por lo que se consideró este campo adicional en el levantamiento de información, donde se detalló el bien público que se busca lograr con el desarrollo del trabajo de investigación.

Otros campos que se consideraron y se pidieron especificar son el alcance de las obras públicas, que para el presente estudio son parques, pistas y veredas. También se pidió delimitar el alcance geográfico, que podía llegar hasta distritos, y, por último, se solicitó un listado de palabras críticas, que son las que ayudaron a definir el modelo. El objetivo de esta primera etapa fue obtener toda aquella información que ayudó en las siguientes etapas para el planteamiento correcto del modelo.

3.2.2 Enfoque analítico

En esta etapa de la metodología se alinearon los objetivos definidos en la etapa anterior con objetivos analíticos, y sobre la base de ellos se definieron enfoques analíticos. Por ejemplo, si el objetivo analítico fue predecir si un comentario de Twitter hace o no referencia a un problema de una de las obras públicas definidas en el paso anterior, el enfoque analítico será la construcción, pruebas e implementación de un modelo de clasificación. Como soporte en esta etapa se elaboró una matriz que permitió hacer el mapeo entre los objetivos y los enfoques analíticos por cada objetivo. A modo de ejemplo, la Tabla 6 muestra el formato de la matriz por implementar y en el Apéndice B se puede visualizar la matriz que se completó para el enfoque analítico.

Tabla 6

Estructura de Matriz de Enfoque Analítico

Problema	Objetivos	Objetivos analíticos	Enfoque analítico
Descripción del problema	Objetivo 1	Predecir si un comentario de Twitter relacionado a un problema de obra pública "Si" o "No".	Construcción, pruebas e implementación de un modelo de clasificación.
	Objetivo 2	Objetivo analítico 2	Enfoque analítico 2
	Objetivo 3	Objetivo analítico 3	Enfoque analítico 3

Nota. Adaptado de *Metodología fundamental para la ciencia de datos* (p. 3), por J. B. Rollins, 2015, IBM Corporation (https://www.ibm.com/downloads/cas/WKK9DX51).

3.2.3 Requisitos de datos

En esta etapa de la metodología se determinan los datos que se utilizaron para la elaboración del modelo, que, a su vez, fue establecido por el enfoque analítico definido en la etapa anterior. En el presente caso se utilizaron datos procedentes de la red social Twitter, que se encontraron en formato de texto, ya que fueron los comentarios que se obtuvieron de las distintas publicaciones con las que se trabajó. Además, se tuvo a disposición datos correspondientes a ubicaciones que también se encuentran en formato de texto.

Para un correcto mapeo de los requisitos de datos con los enfoques analíticos identificados en la etapa anterior, se elaboró una matriz con el detalle correspondiente por enfoque. La Tabla 7 muestra un ejemplo de la matriz por completar en esta etapa, y en el Apéndice C se presenta la matriz que realizada para el trabajo de investigación.

Tabla 7

Matriz de Requisitos de Datos

Enfoque Analítico	Fuente	Dato	Tipo
Enfoque analítico 1	Red social Twitter	Comentarios de Twitter	Texto
	Red social Twitter	Ubicación de Twitter	Texto
Enfoque analítico 2			
Enfoque analítico 3			

Nota. Adaptado de *Metodología fundamental para la ciencia de datos* (p. 3), por J. B. Rollins, 2015, IBM Corporation (https://www.ibm.com/downloads/cas/WKK9DX51).

3.2.4 Recopilación de datos

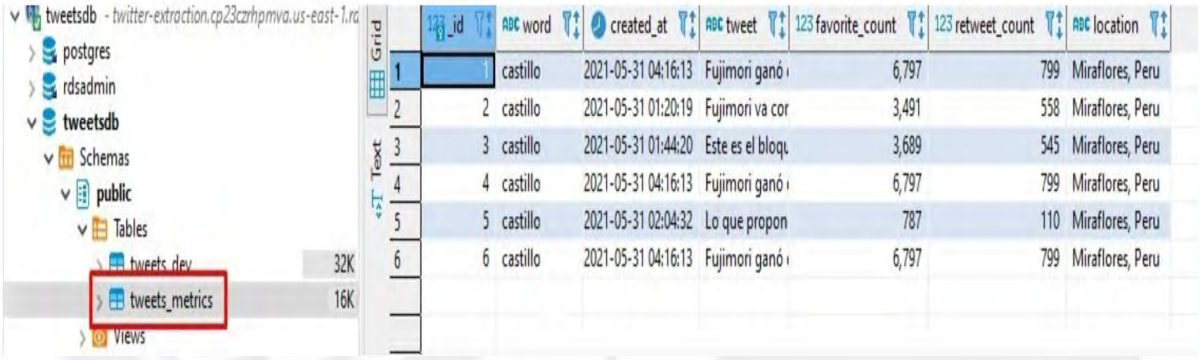
La recopilación de datos se ejecutó con el apoyo de una API[2] de Twitter, la cual permitió extraer de esta red social los *tweets* relacionados con las palabras críticas que se indicaron. Esta API también permitió definir un alcance geográfico, para poder considerar aquellos *tweets* que sean de Perú y referidos a la ciudad de Lima.

[2] *Application Programming Interface* (API): Son un conjunto de subrutinas, funciones y procedimientos que son utilizados como una capa que permite la comunicación entre componentes de *software*.

Toda esta información fue trasladada temporalmente a una base de datos y a un ambiente elaborado para este fin con apoyo del *cloud* de Amazon, AWS. En esta base de datos se podrán hacer los filtros que se consideren importantes sobre la base de los campos extraídos de Twitter, ya que no solo se obtienen comentarios, sino también campos de fechas, contadores de favoritos y *retweets*. La Figura 4 muestra un ejemplo de los datos obtenidos con el uso de la API de Twitter, y en el Apéndice F se presentan las actividades ejecutadas en esta etapa.

Figura 4

Datos Obtenidos con la API de Twitter



Para poder procesar y trabajar con la información obtenida, la data fue exportada a un archivo CSV (*Comma Separated Values*), que permite un gran almacenamiento de información. Todos los *tweets* obtenidos hasta ese momento vienen con caracteres que causan ruido cuando se procede a analizar los datos. Se debe tener en cuenta esto cuando se realice la etapa de preparación de los datos. Para más detalles de la estructura y dimensionamiento relacionado con la base de datos, ver el Apéndice F.

3.2.5 Comprensión de datos

En la etapa de comprensión de datos, se analizó la data disponible en el archivo CSV generado en la etapa anterior, para comprender cada uno de los atributos presentes en el CSV. Para ello se generó la metadata con la descripción de cada uno de ellos. La Tabla 8 muestra una matriz como ejemplo de la metadata.

Tabla 8

Matriz Metadata del Archivo CSV

Atributo	Descripción
id	Indicador de *tweet*
word	Palabra crítica identificada
created_at	Fecha de creación del *tweet*
tweet	Texto del comentario del *tweet*
favorite_count	Cantidad de favoritos que recibió el *tweet*
retweet_count	Cantidad de *retweet*
location	Ubicación del usuario que generó el *tweet*

En esta etapa también se puede realizar un análisis descriptivo de forma general, para tener un mejor entendimiento de los datos disponibles, como ver los *tweets* que tienen la mayor cantidad de *retweets*, o determinar de qué ubicación proviene la mayor parte de comentarios. Todo esto puede ser llevado a gráficos simples solo para una mejor comprensión de los *tweets* obtenidos.

3.2.6 Preparación de datos

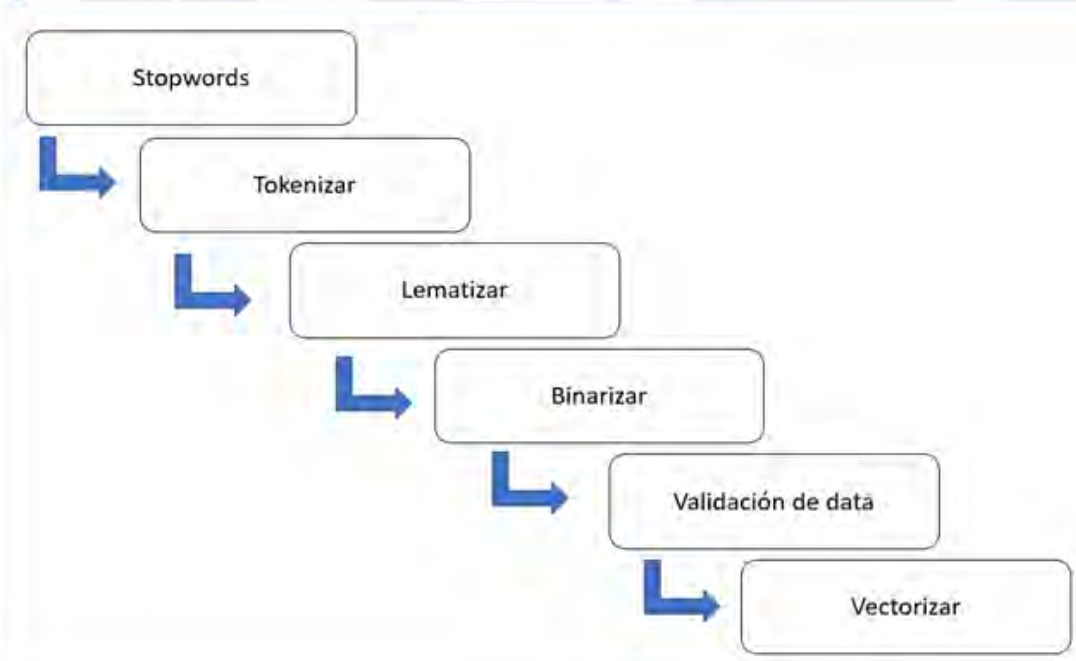
La preparación de los datos es la etapa que demandó mayor tiempo de toda la metodología, por la cantidad de ajustes que fueron realizados a la data para que sea mejor analizada. Para el presente caso, la principal fuente de información fueron los comentarios de cada *tweet*; esta data se encuentra en formato texto y en muchas ocasiones vino con errores gramaticales, caracteres duplicados, emojis, *hashtags*, entre otros. Por este motivo, es muy importante limpiar de la mejor forma estos comentarios en esta etapa, para que el modelo aplicado con el analizador de texto pueda interpretarlo correctamente. Las herramientas de apoyo empleadas son dos: Google Colab y la biblioteca Scikit-learn. Google Colab es una plataforma proporcionada por Google en *cloud*, donde están disponibles recursos de procesamiento, memoria y librerías de lenguaje de programación como Python, para el tratamiento que se le darán a los datos. Scikit-learn es una biblioteca *open source* para el

aprendizaje automático; en esta biblioteca se pueden encontrar algoritmos de clasificación, regresión, máquinas de soporte vectorial, *K-means,* entre otros, además de funciones que permiten realizar distintos tratamientos a los datos.

En esta etapa de preparación de datos, se llevaron a cabo varias actividades para lograr el objetivo de limpiar los datos y que se encuentren de una forma tal que, en el modelado, se puedan utilizar para la definición, entrenamiento y ejecución del modelo. Las actividades ejecutadas son seis: (a) no considerar las *stopwords,* (b) *tokenizar*, (c) lematizar, (d) binarizar, (e) validación de data, y (f) vectorizar. La Figura 5 muestra de forma gráfica las actividades dentro de esta etapa.

Figura 5

Actividades en la Preparación de Datos



Nota. Adaptado de "Análisis masivo de datos en Twitter para la identificación de opinión" (pp. 20-25), por A. Olarte y A. Casaverde, 2020 (http://repositorio.unsaac.edu.pe/bitstream/handle/20.500.12918/5252/253T20200108_TC.pdf?sequence=1&isAllowed=y).

Stopwords. En esta actividad, lo que se buscó es retirar de los comentarios de Twitter aquellas palabras que dentro del procesamiento del lenguaje natural no brinden un valor semántico para el fin propuesto. Estas palabras fueron artículos, conectores, pronombres, etc.;

por ejemplo: 'de', 'la', 'que', 'el', 'en', 'y', 'a', 'los', 'del', 'se', 'las', 'por', 'un', 'para', 'con', 'no', 'una', 'su', 'al', 'lo'. La Tabla 9 muestra el resultado de esta actividad en un comentario.

Tabla 9

Resultado de Comentario sin Stopwords

Tipo de comentario	Comentario
Comentario de Twitter original	Parque costado de la iglesia de la cúpula por la avenida sucre está descuidado y muy sucio
Comentario sin *stopwords*	Parque costado iglesia cúpula avenida sucre está descuidado sucio

Nota. Datos recolectados de comentarios de Twitter sobre parques de Lima.

Tokenizar. El objetivo de esta actividad fue separar el texto de los comentarios en un listado de palabras de forma independiente, que permitió dar un tratamiento de los comentarios palabra por palabra. La Tabla 10 muestra el resultado de esta actividad con un comentario.

Tabla 10

Resultado de Comentario Tokenizado.

Tipo de comentario	Comentario
Comentario sin *stopwords*	Parque costado iglesia cúpula avenida sucre está descuidado sucio
Comentario tokenizado	[Parque,costado,iglesia,cúpula,avenida,sucre,está,descuidado,sucio]

Nota. Datos recolectados de comentarios de Twitter sobre parques de Lima.

Lematizar. Esta actividad consistió en llevar cada una de las palabras a su lema correspondiente, es decir, a su forma base; por ejemplo, una conjugación de un verbo en

tercera persona, ella baila, el verbo en su forma base es bailar. La Tabla 11 muestra el resultado de una lematización de palabras de un comentario.

Tabla 11

Resultado de Comentario con Lematización

Tipo de comentario	Comentario
Comentario tokenizado	[Parque,costado,iglesia,cúpula,avenida,sucre,está,descuidado,sucio]
Comentario con lematización	[Parque,costado,iglesia,cúpula,avenida,sucre,estar,descuidar,sucio]

Nota. Datos recolectados de comentarios de Twitter sobre parques de Lima.

Binarizar. Dado que el objetivo es clasificar los comentarios, es necesario que estos tengan una etiqueta para que el modelo pueda aprender y luego predecir. En este caso, se clasificaron los comentarios referentes a las obras en positivos y negativos, los cuales fueron representados por el número 1 para aquellos que sean positivos y 0 para los comentarios que sean negativos.

Validación de data. Es una actividad que se agregó para verificar la data disponible al momento; se valida la cantidad de comentarios de Twitter reunidos para el entrenamiento del modelo. Para el presente caso, el corpus posee 12,546 comentarios. Otro criterio que se debe tener en cuenta es conocer si la data se encuentra o no equilibrada, ya que esto afectará en los resultados de la predicción del modelo.

Vectorizar. Es la última actividad considerada en esta etapa. Las palabras de los comentarios recopilados hasta el momento se llevaron a una representación numérica en una matriz de unos y ceros con la que el modelo pueda trabajar para poder determinar su clasificación. En la Tabla 12 se muestra cómo se realiza esta conversión de palabras a números.

Tabla 12

Resultado de Comentario con Vectorización

Palabra	parque	costado	iglesia	cúpula	avenida	estar	descuidar	sucio
parque	1	0	0	0	0	0	0	0
costado	0	1	0	0	0	0	0	0
iglesia	0	0	1	0	0	0	0	0
cúpula	0	0	0	1	0	0	0	0
avenida	0	0	0	0	1	0	0	0
estar	0	0	0	0	0	1	0	0
descuidar	0	0	0	0	0	0	1	0
sucio	0	0	0	0	0	0	0	1

Nota. Datos recolectados de comentarios de Twitter sobre parques de Lima.

Al finalizar con estas seis actividades, los datos quedan preparados y listos para iniciar la siguiente etapa del modelado. Es importante mencionar que en el caso de identificar algún problema relacionado con la preparación de los datos, se deberá regresar nuevamente a esta etapa para ejecutar alguna de las actividades o, si es preciso, agregar otra adicional.

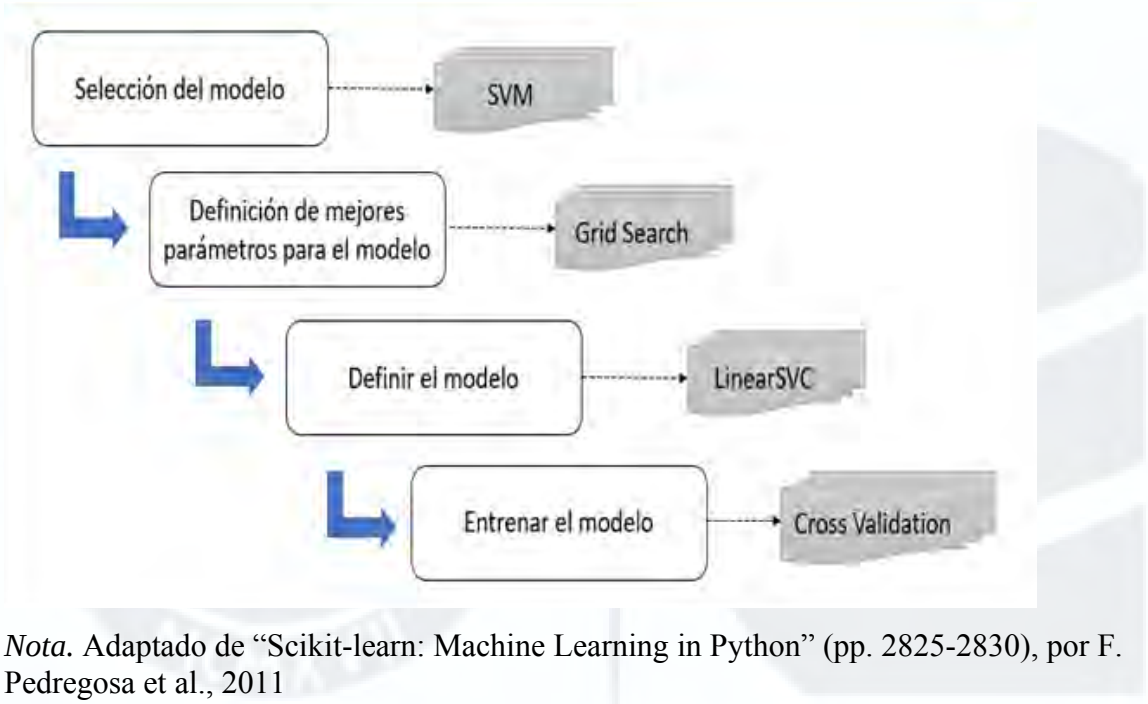
3.2.7 Modelado

En esta etapa se define el modelo que debe ser utilizado. En el presente caso, se buscó un modelo de clasificación que permita clasificar los comentarios que obtenidos de Twitter en positivos y negativos con referencia a las obras de parques, veredas y pistas, para luego poder identificar la ubicación. Existen distintos modelos de clasificación de acuerdo con la literatura revisada y que brindan buenos resultados para la clasificación; en especial, para el análisis de atributos bivariados es *support vector machine* (SVM). Este modelo lo que busca es encontrar una línea que separe a los datos de acuerdo con la clasificación que se haya definido, utilizando parámetros que permitan definir de una mejor manera los vectores de soporte para encontrar la línea de separación en el hiperplano, ya que al utilizar distintos atributos presentes en los comentarios, no se puede hablar de un plano simple.

Para el modelado, se efectuaron las siguientes cuatro actividades: (a) selección del modelo, (b) definición de mejores parámetros para el modelo, (c) definición del modelo, y (d) entrenar el modelo. La Figura 6 muestra de forma gráfica las actividades realizadas en esta etapa del modelado.

Figura 6

Actividades del Modelado



Nota. Adaptado de "Scikit-learn: Machine Learning in Python" (pp. 2825-2830), por F. Pedregosa et al., 2011 (http://jmlr.csail.mit.edu/papers/v12/pedregosa11a.html).

Selección del modelo. En esta actividad se elige el modelo, por lo que se deberá tener en cuenta la tarea prevista. En este caso se trata de una tarea de clasificación. Además, debe considerarse la variable objetivo en la que se busca realizar la clasificación. Para efectos del presente trabajo de investigación y al desarrollar un piloto del modelo, se tendrán dos clases: clasificación de comentarios de Twitter positivos, y comentarios de Twitter negativos.

Los modelos de *Support Vector Machine* (SVM) son buenos para la tarea de clasificación que se busca realizar (Aiala et al., 2017; Segura, 2019), pero dentro de estos modelos existen diferentes estimadores que se alinean a distintas tareas de clasificación, regresión, detección. La Tabla 13 muestra estos estimadores de SVM. Para el presente trabajo

de investigación se utilizará el *Linear Support Vector Classification*, al tratarse de una tarea de clasificación.

Tabla 13

Estimadores Support Vector Machines

Estimadores SVM
Linear Support Vector Classification
Linear Support Vector Regression
Nu-Support Vector Classification
Nu-Support Vector Regression
Unsupervised Outlier Detection
C-Support Vector Classification
Epsilon-Support Vector Regression

Nota. Adaptado de "Scikit-learn: Machine Learning in Python," por F. Pedregosa et al., 2011 (https://scikit-learn.org/stable/).

Por otra parte, debe tenerse un indicador de medición para ver el rendimiento del modelo, para lo cual se utilizará el *F1-score*, que es una métrica empleada en los modelos de clasificación. Las pruebas realizadas para el piloto de este estudio muestran un buen indicador para este modelo. La Tabla 14 presenta un resumen del modelo elegido y el resultado de la métrica.

Tabla 14

Valores de la Selección del Modelo

Parámetro de selección	Valor
Modelo	SVM
SVM Estimator	Linear Support Vector Classification
Métrica	F1-score
Resultado de métrica AUC ROC	0.76747067

Definición de mejores parámetros para el modelo. El modelo SVM requiere de ciertos parámetros para su ejecución, los valores que se asignen a estos parámetros

determinarán una mejor efectividad del modelo. Los tres parámetros que se utilizaron son el coste, la pérdida y el número máximo de iteraciones.

El coste hace referencia al margen de flexibilidad del modelo, mientras mayor sea el costo se producirá *overfitting*, es decir, el modelo quedará demasiado ajustado a los datos con los que se está trabajando; por el contrario, si el coste es menor, el modelo se flexibilizará y podrá ser utilizado con otros datos. La pérdida es la métrica de error para el ajuste *del Linear SVM*. Se consideran dos posibles valores para este parámetro: *hinge* y *squared hinge* o llamado hinge cuadrático. El último parámetro para considerar es la cantidad máxima de iteraciones, que es el valor máximo de repeticiones que ejecutará el modelo para su ajuste.

Estos parámetros también son conocidos como hiperparámetros, por la importancia que tienen en el aprendizaje del modelo. Para obtener los mejores valores en los hiperparámetros se ha recurrido a un método llamado *grid search* o también conocido como la búsqueda en cuadrícula. Este método prueba entre las alternativas brindadas a cada parámetro para obtener los valores óptimos de los hiperparámetros. La Tabla 15 muestra los valores óptimos obtenidos para el modelo SVM empleado.

Tabla 15

Valores Óptimos de los Hiperparámetros

Hiperparámetro	Valor óptimo
Coste	0.2
Pérdida	*squared_hinge*
Iteraciones máximas	1000

Definir el modelo. Para definir el modelo se necesitan dos elementos: el modelo elegido, en este caso el Linear SVM, y los valores de los hiperparámetros óptimos, que son los obtenidos previamente. Para efectos del trabajo de investigación, el modelo se definió dentro del ambiente de Google Colab con Python, utilizando la biblioteca SKLearn, donde se

encuentra disponible este modelo; es de esta forma en que solo debe importarse y utilizarse. Al momento de definirlo, se especifican los valores de los hiperparámetros obtenidos. Luego de tener el modelo definido, se utiliza para el entrenamiento. La Figura 7 muestra cómo fue definido el modelo con código Python. La totalidad del código, tanto para la selección de los parámetros como para los demás tratamientos de los datos, se encuentra en el Apéndice D.

Figura 7

Actividades del Modelado

```
# definimos el modelo de acuerdo a los hiper-paremetros encontrados
model = LinearSVC(C=.2, loss='squared_hinge',max_iter=1000,multi_class='ovr',
                  random_state=None,
                  penalty='l2',
                  tol=0.0001
)
```

Nota. Programación Python. Adaptado de "Scikit-learn: Machine Learning in Python" (pp. 2825-2830), por F. Pedregosa et al., 2011 (http://jmlr.csail.mit.edu/papers/v12/pedregosa11a.html).

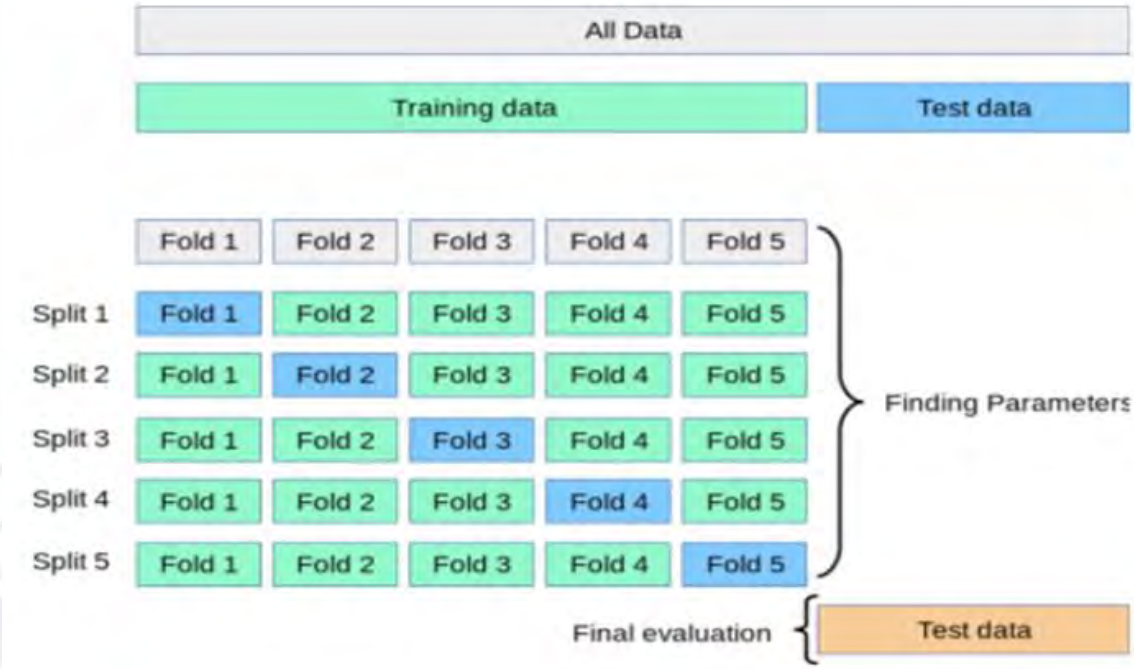
Como se puede ver en la Figura 7, se definió el modelo en "model", y se utiliza el modelo importado de *SKLearn, LinearSVC*; dentro de los paréntesis se ingresan los hiperparámetros con sus valores óptimos, donde C es el costo con valor de 2, *loss* es la pérdida con valor de *squad_hinge; max_iter* es el valor máximo de iteraciones, en este caso 1000. El resto de los valores son necesarios establecerlos para la definición del modelo, el estado aleatorio queda en ninguno, la penalidad establecida en l2 y el tol en 0.0001. Estos últimos valores son tomados como *default* de acuerdo con lo definido en la documentación de la librería de *LinearSVC*.

Entrenamiento del modelo. Para esta actividad debe tenerse listo el *dataset* de comentarios de Twitter debidamente preparado y etiquetado. También, el modelo definido, y buscar la mejor forma de entrenamiento. Lo ideal es dividir la data de entrenamiento en porciones pequeñas, para comprobar los resultados obtenidos por cada evaluación con las

diferentes porciones de datos utilizados. La Figura 8 muestra una representación gráfica de esta forma de entrenamiento.

Figura 8

Representación del Entrenamiento por Pliegues



Nota. Adaptado de "Scikit-learn: Machine Learning in Python," por F. Pedregosa et al., 2011 (https://scikit-learn.org/stable/).

Lo que se buscó fue dividir la data en grupos pequeños de datos para el entrenamiento y otro grupo pequeño para la prueba. La data de entrenamiento se divide en pequeños subconjuntos de datos menores, donde se toma uno de estos pequeños subconjuntos como data de prueba, representado de color celeste en la Figura 8, y se realiza una segunda iteración alternando la data de prueba en otro subconjunto de datos, y así sucesivamente hasta que todos los subconjuntos hayan sido data de prueba en alguna de las iteraciones. Con esta forma de trabajo se busca mejorar el entrenamiento y tener una mejor efectividad, la cual será comprobada en la etapa de evaluación. Para el presente caso se utilizó *grid search* en esta actividad, se envió en el ambiente tanto los comentarios debidamente preprocesados como la vectorización y el etiquetado, para que *grid search* realice el entrenamiento con esta división

de la información. El resultado del entrenamiento debe ser evaluado para que luego el modelo pueda ser ejecutado con información real.

3.2.8 Evaluación

En la etapa de evaluación, se validó el rendimiento del modelo entrenado y se obtuvieron las métricas que permitieron medir la precisión del modelo. Para efectos del trabajo de investigación, en esta etapa se utilizaron dos métricas: el área bajo la curva, conocida como AUC, y la métrica *F1-score*. Se tomará el valor obtenido del *F1-score* como el indicador principal del rendimiento, por ser el que representa la precisión y sensibilidad en una sola métrica. La forma de evaluación utilizada fue la validación cruzada, ya que al tener un entrenamiento del modelo por iteraciones y subconjuntos de datos con cambio del subconjunto de datos de prueba por cada iteración, lleva a resultados de las dos métricas en cada ejecución. La Tabla 16 muestra los resultados obtenidos en el proceso iterativo.

Tabla 16

Resultados del Proceso Iterativo de la Validación Cruzada

Iteración	*Score* AUC	F1-*score*
1ra	0.85775386	0.80106707
2da	0.85186993	0.7940833
3ra	0.6962776	0.68423092
4ta	0.85932109	0.80594739
5ta	0.79262029	0.75202468
Promedio	0.811568555	0.767470673

De acuerdo con los resultados obtenidos en la evaluación cruzada, se realizaron cinco iteraciones, con un resultado promedio de 0.767470673 para la métrica del *F1-score*, lo que muestra una precisión y sensibilidad del modelo de 76.75%.

Para validar este resultado se buscaron referencias de otras investigaciones en donde utilizaron el mismo modelo, *support vector machine*, con la misma tarea de clasificación de comentarios, de tal manera que se valida si el resultado obtenido es comparable con el de las otras investigaciones (Condori & Valeriano, 2020; Herrera, 2020; Segura, 2019). La Tabla 17 muestra los resultados de tres investigaciones en comparación con el resultado del presente modelo.

Tabla 17

Resultados del Modelo SVM en Investigaciones Referenciales

Investigación	Modelo	F1-*score*
Investigación referencial 1	SVM	0.78
Investigación referencial 2	SVM	0.72
Investigación referencial 3	SVM	0.7
Investigación propia	SVM	0.76

La investigación referencial 1 fue un estudio para clasificar comentarios de lugares turísticos utilizando el modelo SVM, con un resultado de *F1-score* de 78% de precisión y sensibilidad en la aplicación del modelo en esta investigación (Herrera, 2020).

La investigación referencial 2 fue estudio realizado para detectar intenciones de suicidios en el lenguaje español a través del análisis de texto en redes sociales, utilizando en uno de sus modelos SVM, con un resultado de *F1-score* de 72% (Condori & Valeriano, 2020).

En la investigación referencial 3 se evaluaron algoritmos de clasificación de opinión en Twitter, donde se utilizó el modelo SVM, con un *F1-score* de 0.70% de precisión y sensibilidad en la aplicación de este modelo (Segura, 2019).

De acuerdo con los resultados de la Tabla 13, el *F1-score* de 76% de precisión y sensibilidad obtenido con el modelo de la presente investigación es comparable con los

modelos de las investigaciones tomadas como referencia. Con este resultado y ya teniendo el modelo definido y entrenado es que se decide ejecutarlo para realizar la predicción con los comentarios de Twitter referentes a las obras que son motivo de este trabajo de investigación —parques, veredas y pistas— y así realizar el análisis de los resultados en el siguiente capítulo.

Capítulo IV: Resultados

En este capítulo se revisarán los resultados obtenidos luego de aplicar toda la metodología planteada en el capítulo anterior. Con el modelo debidamente entrenado, se ejecutará con los comentarios de las obras definidas para esta investigación —parques, veredas y pistas—, se evaluarán los resultados obtenidos con la ayuda de la matriz de consistencia y las métricas *F1-score*, *precisión*, *recall* y *accuracy*.

Para el trabajo de investigación, el indicador de éxito utilizado es el *F1-score*, por representar la sensibilidad y la precisión en un solo valor; además, la distribución de clases empleada fue desigual o no equilibrada. Los resultados de la predicción serán revisados en conjunto, es decir, todos los comentarios referentes a los tres tipos de obras públicas.

Posteriormente, se revisarán los resultados de forma independiente por cada uno de los tipos de obras. Al final del capítulo, se analizan los resultados para poder comprender los valores obtenidos y si el modelo responde a la investigación efectuada.

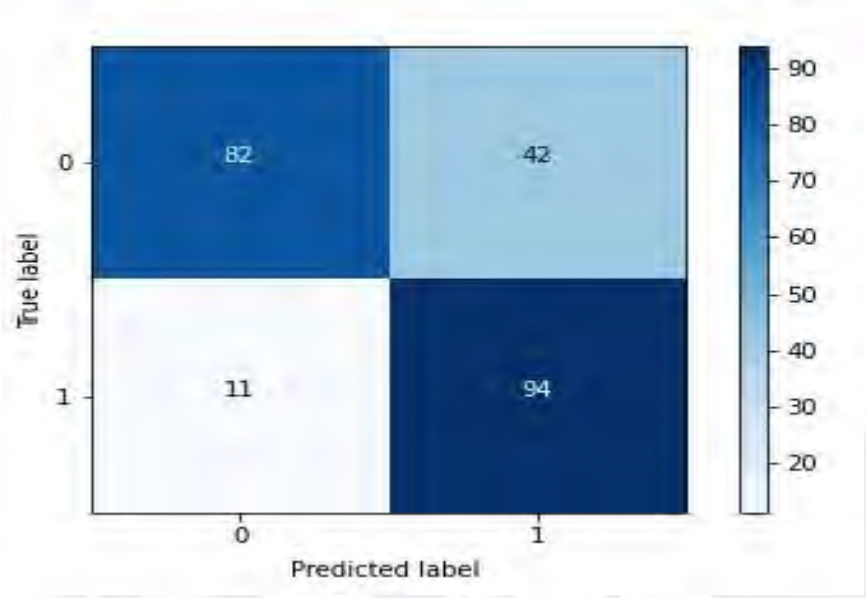
4.1 Resultados de Predicción del Modelo

A fin de tener una visión global de los resultados obtenidos para la predicción del modelo, se tomó el total de los comentarios referidos a las tres obras —parques, veredas y pistas—. Como primer resultado, se muestra la matriz de confusión (ver Figura 9), donde puede identificarse la cantidad de falsos positivos, falsos negativos, verdaderos positivos y verdaderos negativos.

De acuerdo con lo obtenido en la matriz de confusión, el número de verdaderos positivos y verdaderos negativos es elevado y mayor que los falsos positivos y falsos negativos, por lo que se puede apreciar que los aciertos del modelo son de mayor cantidad en la diagonal principal. La Tabla 18 muestra una descripción de los resultados de la matriz de confusión.

Figura 9

Matriz de Confusión Predicción del Modelo



Nota. Adaptado de "La matriz de confusión y sus métricas," por J. Barrios, 2019 (https://www.juanbarrios.com/la-matriz-de-confusion-y-sus-metricas/).

Tabla 18

Descripción de Resultados de Predicción del Modelo

Cuadrante	Total	Descripción	Estado de predicción
TN: Verdaderos negativos	82	Número de comentarios negativos de Twitter referentes a obras, que el modelo ha predicho como negativos y que realmente son negativos.	Correcta
FP: Falsos positivos	42	Número de comentarios positivos de Twitter referente a obras, que el modelo ha predicho como positivos pero que realmente son negativos.	Errónea
FN: Falsos negativos	11	Número de comentarios negativos de Twitter referente a obras, que el modelo ha predicho como negativos, pero realmente son positivos.	Errónea
TP: Verdaderos positivos	94	Número de comentarios positivos de Twitter referente a obras, que el modelo ha predicho como positivos y que realmente son positivos.	Correcta

Estos resultados serán revisados con el apoyo de cuatro métricas a partir de la información obtenida en la matriz de confusión, las cuales son la precisión, el *recall*, el *F1-score* y el *accuracy*. El cálculo de la precisión arrojó un resultado de 0.69, que indica que el 69% de predicciones positivas son correctas. Para ello se utilizó la siguiente fórmula:

$$Precisión = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

$$Precisión = \frac{94}{94 + 42}$$

$$Precisión = 0.69$$

Para el cálculo del *recall* se utilizó la siguiente fórmula, reemplazando los valores obtenidos en la matriz de confusión, con lo que se logró el resultado de 0.89, que indica que el 89% de los comentarios positivos fueron identificados de forma correcta por el modelo.

$$Recall = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

$$Recall = \frac{94}{94 + 11}$$

$$Recall = 0.89$$

Para el *F1-score*, el cálculo se realizó con la siguiente fórmula, que luego de reemplazar los valores obtenidos previamente en la precisión y el *recall* se logró el resultado de 0.78, que indica que el 78% del modelo propuesto cumple con el propósito del caso de estudio.

$$F1\ score = \frac{2 * \text{Precisión} * \text{Recall}}{\text{Precisión} + \text{Recall}}$$

$$F1\ score = \frac{2 * 0.69 * 0.89}{0.69 + 0.89}$$

$$F1\ score = 0.78$$

Por último, para el *accuracy* el cálculo se realizó con la fórmula abajo mencionada. Al reemplazar los valores obtenidos previamente en la matriz de confusión, el resultado fue de 0.77, es decir, el 77% de las predicciones positivas son correctas.

$$Accuracy = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}}$$

$$Accuracy = \frac{94 + 82}{94 + 82 + 42 + 11}$$

$$Accuracy = 0.77$$

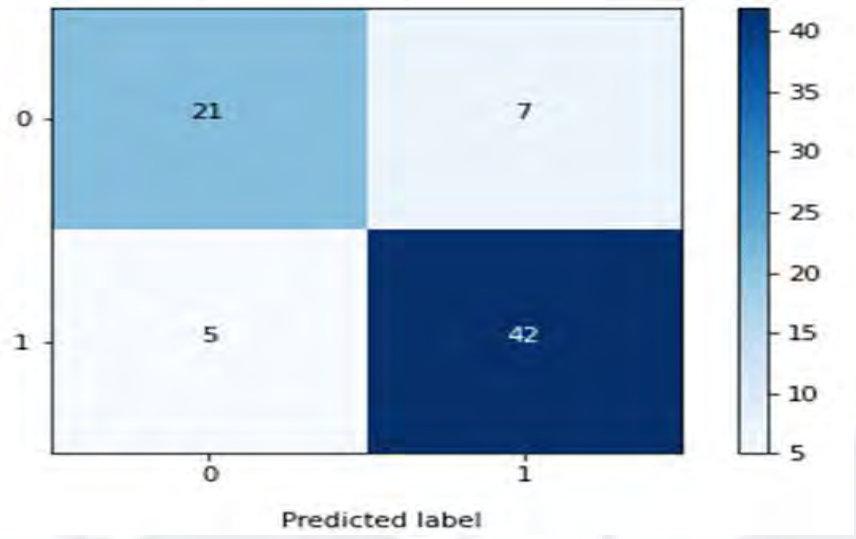
A partir de los cuatro resultados obtenidos, se prestó atención al ponderado de la precisión y de la sensibilidad. Esta variable es el *F1-score*. El valor de esta medida fue de 0.78. El resultado demuestra que el modelo predice el trabajo de investigación con una precisión y sensibilidad a un 78%. Tal resultado es bueno tomando como referencia la evaluación cruzada de cinco interacciones, donde el valor obtenido del promedio del *F1-score* fue 76%. A esto se adicionaron las referencias de investigaciones anteriores utilizando el modelo SVM con resultados equivalentes al obtenido (Herrera, 2020; Condori y Valeriano, 2020; Segura, 2019).

4.2 Resultados de Predicción del Modelo en Parques

Para analizar los resultados de las obras públicas relacionadas con parques, solo se han considerado aquellos comentarios de Twitter referidos a estas obras. De la misma forma en que se trabajó con el resultado global, se utilizó una matriz de confusión y las cuatro métricas de precisión, *recall*, *F1-score* y *Accuracy*. La Figura 10 muestra el resultado de la matriz de confusión con relación a parques. Asimismo, en la Tabla 19 se aprecian los resultados de la matriz de confusión con referencia a los comentarios y cada uno de sus cuadrantes con sus respectivos comentarios.

Figura 10

Matriz de Confusión Predicción del Modelo en Parques



Nota. Adaptado de "La matriz de confusión y sus métricas," por J. Barrios, 2019 (https://www.juanbarrios.com/la-matriz-de-confusion-y-sus-metricas/).

Tabla 19

Descripción de Resultados Predicción del Modelo en Parques

Cuadrante	Total	Descripción	Estado de predicción
TN: Verdaderos negativos	21	Número de comentarios negativos de Twitter referente a parques, que el modelo ha predicho como negativos y que realmente son negativos.	Correcta
FP: Falsos positivos	7	Número de comentarios positivos de Twitter referente a parques, que el modelo ha predicho como positivos pero que realmente son negativos.	Errónea
FN: Falsos negativos	5	Número de comentarios negativos de Twitter referente a parques, que el modelo ha predicho como negativos pero que realmente son positivos.	Errónea
TP: Verdaderos positivos	42	Número de comentarios positivos de Twitter referente a parques, que el modelo ha predicho como positivos y que realmente son positivos.	Correcta

Ya con los datos obtenidos y las fórmulas conocidas, se realizó el cálculo de las cuatro métricas: Precisión, *Recall*, *F1-score* y el *Accuracy*. Los resultados de estos cuatro indicadores se aprecian en la Tabla 20.

Tabla 20

Resultado de Métricas del Modelo de Predicción en Parques

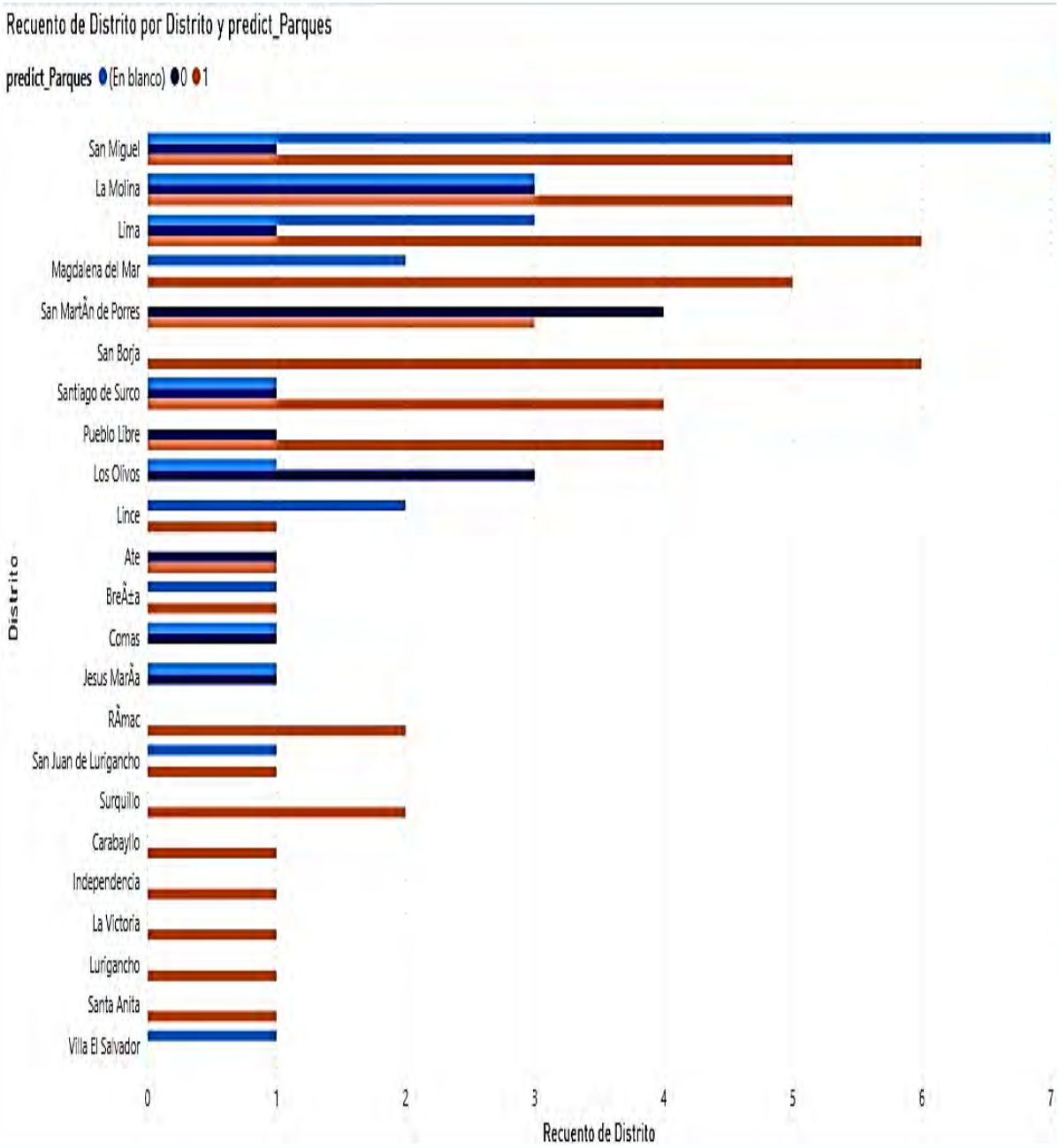
Métrica	Resultado
Precisión	0.857
Recall	0.893
F1-score	0.874
Accuracy	0.840

En este tipo de obras, el valor del indicador principal, el *F1-score*, es de 87.4%, lo cual indica que el modelo de predicción se cumple en un 87.4.7%, en tanto que un 12.6% no es predicho por el modelo en cuestión de parques. Mientras el valor de *F1-score* se acerque a 1 el resultado será mejor con respecto al valor de *F1-score* total (obras urbanas).

Para tener una visión de los datos analizados sobre parques, de acuerdo con las zonas a las que hacen referencia a partir de la ubicación obtenida del comentario de Twitter, la distribución por distritos en Lima se muestra en la Figura 11, donde se aprecian aquellas predicciones correspondientes a comentarios relacionados con parques. Los comentarios negativos tienen el valor de 0 y son de color azul, los positivos tienen el valor 1 y son de color naranja, y los comentarios que no corresponden a los dos tipos anteriores, de color celeste. Se observa que la mayor cantidad de comentarios proviene de los distritos de San Miguel, La Molina y Lima.

Figura 11

Distribución de Predicciones de Comentarios por Distritos de Lima en Parques



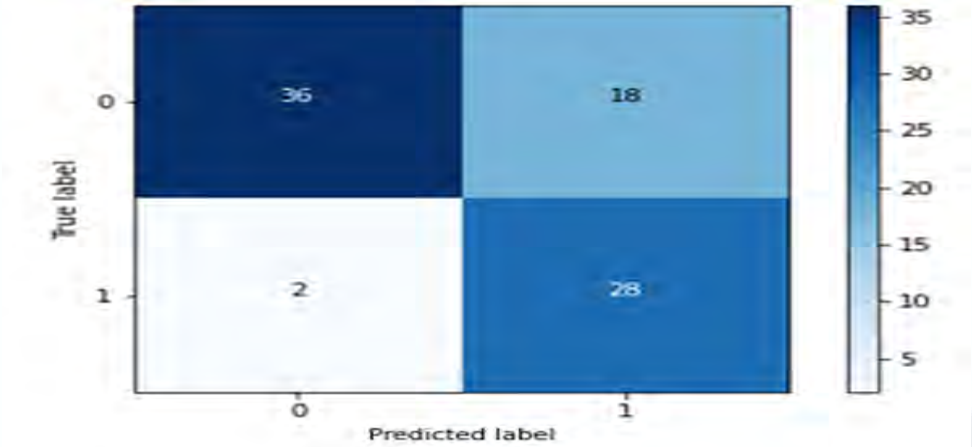
Nota. Elaborado con la plataforma Microsoft Power BI, por Microsoft ©, 2021 (https://powerbi.microsoft.com/es-es/).

4.3 Resultados de Predicción del Modelo en Veredas

Para la revisión de los resultados con las obras de veredas, se realizó un análisis similar a los anteriores; primero, la matriz de confusión y, luego, el cálculo de las cuatro métricas. La Figura 12 muestra el resultado de la matriz de confusión de veredas.

Figura 12

Matriz de Confusión Predicción del Modelo en Veredas



Nota. Adaptado de "La matriz de confusión y sus métricas," por J. Barrios, 2019 (https://www.juanbarrios.com/la-matriz-de-confusion-y-sus-metricas/).

Tal como se aprecia en la Figura 12, para el caso de las obras relacionadas con veredas, los valores de los cuadrantes de la diagonal principal son mayores. Esto quiere decir que el modelo ha predicho de forma correcta los verdaderos positivos y los verdaderos negativos. Para tener una mejor referencia, en la Tabla 21 se muestran las descripciones de la matriz de confusión de este tipo de obra.

Tabla 21

Descripción de Resultados Predicción del Modelo en Veredas

Cuadrante	Total	Descripción	Estado de predicción
TN: Verdaderos negativos	36	Número de comentarios negativos de Twitter referente a veredas, que el modelo ha predicho como negativos y que realmente son negativos.	Correcta
FP: Falsos positivos	18	Número de comentarios positivos de Twitter referente a veredas, que el modelo ha predicho como positivos pero que realmente son negativos.	Errónea
FN: Falsos negativos	2	Número de comentarios negativos de Twitter referente a veredas, que el modelo ha predicho como negativos pero que realmente son positivos.	Errónea
TP: Verdaderos positivos	28	Número de comentarios positivos de Twitter referente a veredas, que el modelo ha predicho como positivos y que realmente son positivos.	Correcta

La revisión de este análisis también comprende la observación de los resultados obtenidos en las cuatro métricas para la predicción de los comentarios relacionados con veredas, tal como se aprecia en la Tabla 22.

Tabla 22

Resultado de Métricas del Modelo de Predicción en Veredas

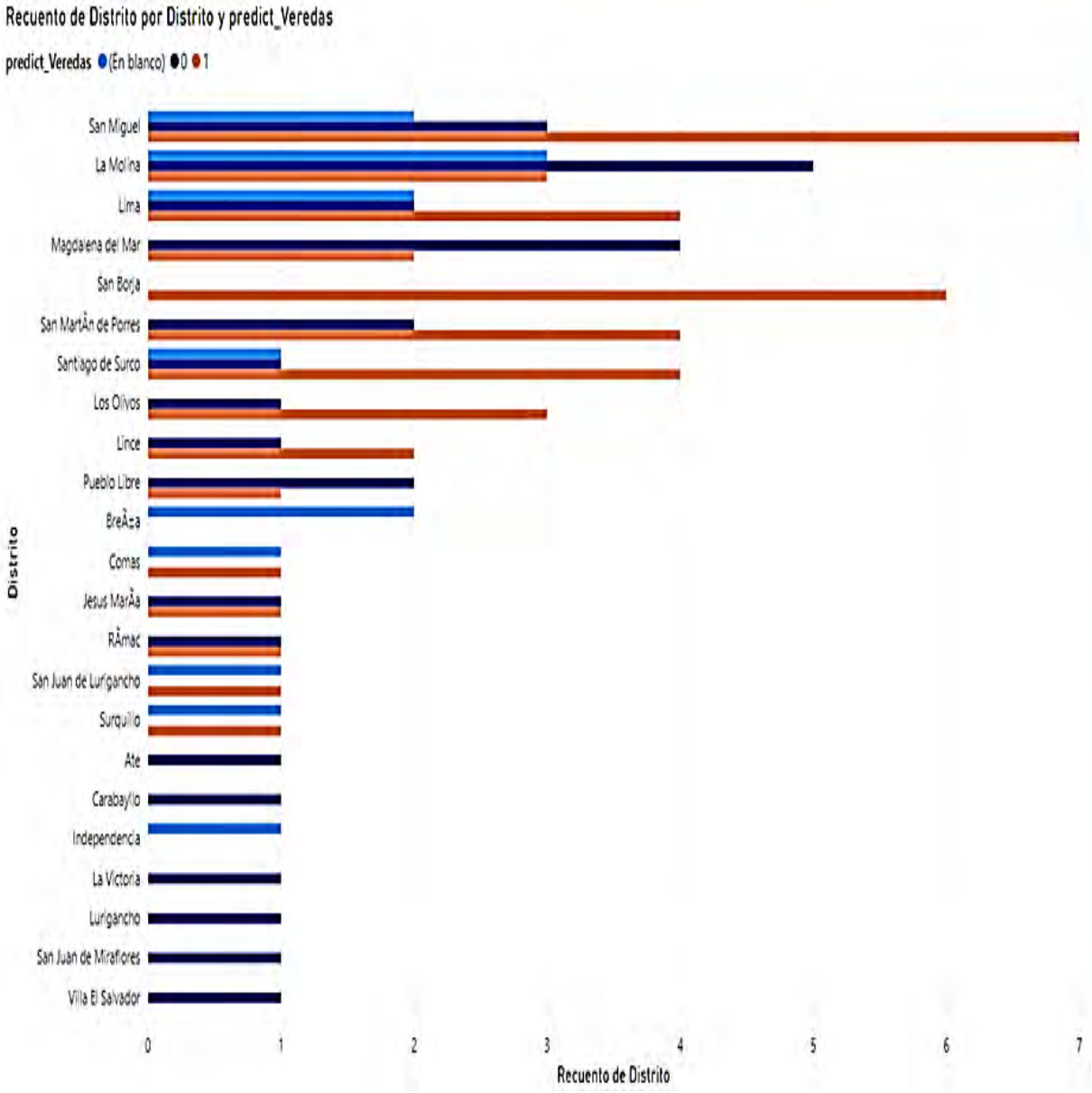
Métrica	Resultado
Precisión	0.609
Recall	0.933
F1-score	0.737
Accuracy	0.762

En este tipo de obras, el valor del indicador principal del presente estudio, el *F1-score*, es de 73.7%, que no es el mejor resultado con respecto al valor de *F1-score* total de 78% (obras urbanas). Esto se explica porque si bien el modelo de predicción se cumple a un 73.7%, se tiene un 26.3% que no es predicho en cuanto a las veredas, parques y pistas. Como ya se mencionó, mientras el valor de *F1-score* se acerque a 1 es mejor.

La revisión de los comentarios relacionados con veredas fue distribuida por distritos en la ciudad de Lima, para tener una visión amplia del resultado de la predicción del modelo sobre este tipo de obras en cada distrito del que se recibió información a través de Twitter. Tal como se aprecia en la Figura 13, aquellas predicciones de comentarios negativos son las representadas con 0 y de color azul; los comentarios positivos están representados con 1 y de color naranja, y los que no entran en estos dos tipos de comentarios son de color celeste. Es importante notar que en este tipo de obras los comentarios celestes se ven reducidos en comparación con las obras relacionadas con parques. Además, se puede observar que el distrito de San Borja no presenta comentarios negativos.

Figura 13

Distribución de Predicciones de Comentarios por Distritos de Lima en Veredas



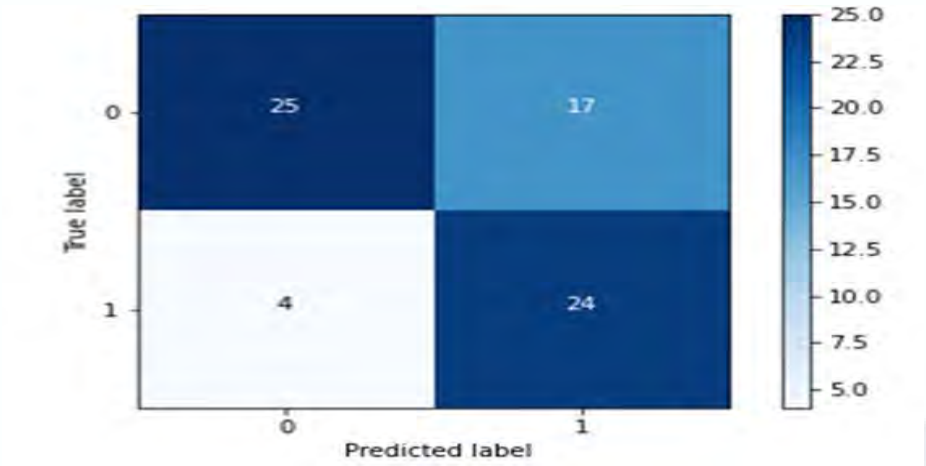
Nota. Elaborado con la plataforma Microsoft Power BI, por Microsoft ©, 2021 (https://powerbi.microsoft.com/es-es/).

4.4 Resultados de Predicción del Modelo en Pistas

La revisión de resultados para pistas es similar al de los dos tipos de obras anteriores; primero, se observa la matriz de confusión y, luego, el cálculo de las tres métricas. La Figura 14 muestra el resultado obtenido en la matriz de confusión para la predicción del modelo en pistas.

Figura 14

Matriz de Confusión Predicción del Modelo en Pistas



Nota. Adaptado de "La matriz de confusión y sus métricas," por J. Barrios, 2019 (https://www.juanbarrios.com/la-matriz-de-confusion-y-sus-metricas/).

En esta última matriz se observa un resultado con valores significativos en la diagonal principal, lo que demuestra que el modelo ha predicho verdaderos negativos y verdaderos positivos. El detalle de la matriz de confusión con su respectiva descripción por cuadrante se encuentra en la Tabla 23, que brindará un mejor entendimiento del resultado.

Tabla 23

Descripción de Resultados Predicción del Modelo en Pistas

Cuadrante	Total	Descripción	Estado de predicción
TN: Verdaderos negativos	25	Número de comentarios negativos de Twitter referente a pistas, que el modelo ha predicho como negativos y que realmente son negativos.	Correcta
FP: Falsos positivos	17	Número de comentarios positivos de Twitter referente a pistas, que el modelo ha predicho como positivos pero que realmente son negativos.	Errónea
FN: Falsos negativos	4	Número de comentarios negativos de Twitter referente a pistas, que el modelo ha predicho como negativos pero realmente son positivos.	Errónea
TP: Verdaderos positivos	24	Número de comentarios positivos de Twitter referente a pistas, que el modelo ha predicho como positivos y que realmente son positivos.	Correcta

Para completar la revisión de estos resultados, a continuación en la Tabla 24 se observan los valores calculados en las cuatro métricas para este tipo de obra, donde el valor del indicador principal, el *F1-score*, es de 70%, que es menor al resultado total de *F1-score* de parques, veredas y pistas. Cabe anotar que se tiene un 30% que el modelo no está prediciendo en cuestión de pistas. Por ello, mientras el valor de *F1-score* se acerque a 1 es mejor.

Tabla 24

Resultado de Métricas del Modelo de Predicción en Pistas

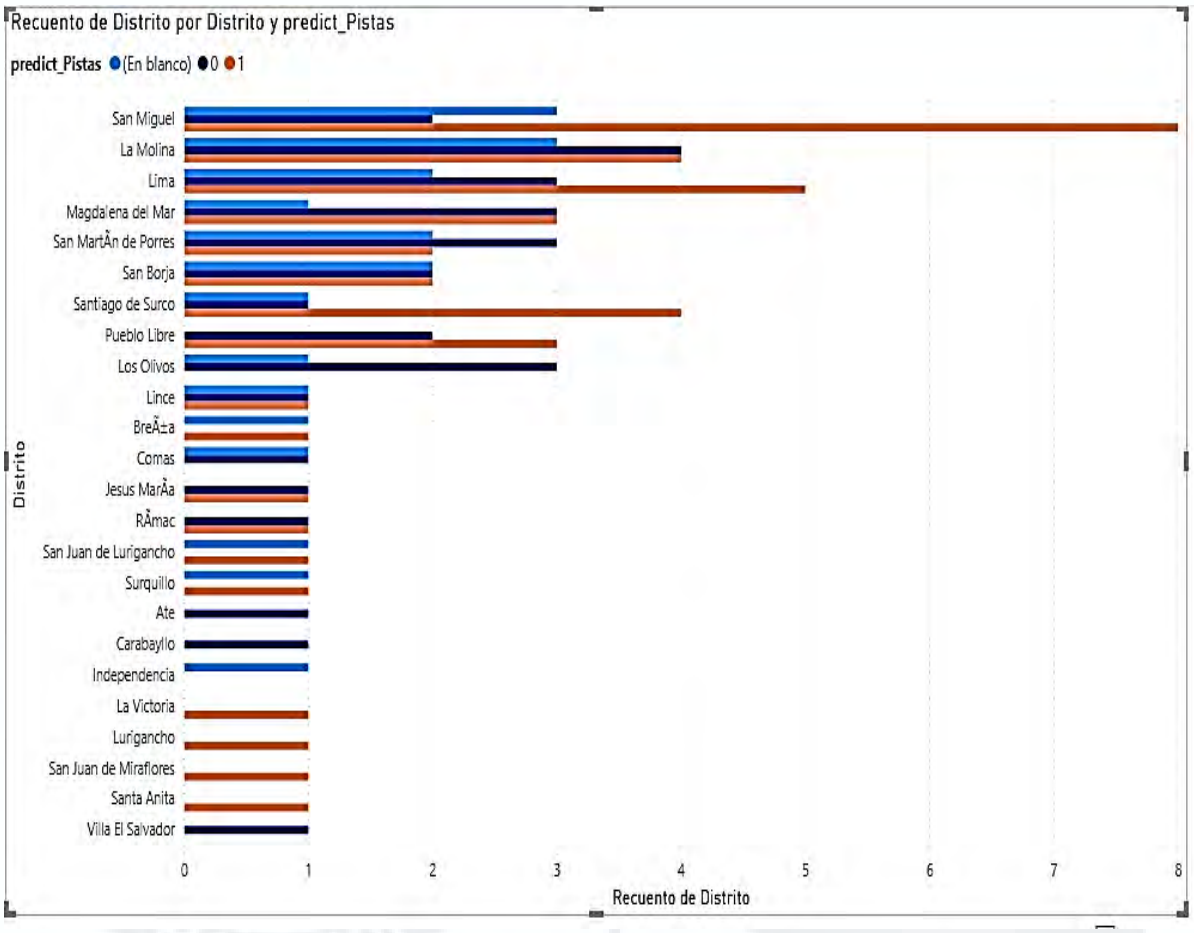
Métrica	Resultado
Precisión	0.585
Recall	0.857
F1-score	0.70
Accuracy	0.7

Para poder analizar las ubicaciones de los comentarios, se efectuó una distribución de las predicciones por los distritos de Lima con base en la información obtenida de Twitter. En la Figura 15, se muestra la distribución de las predicciones relacionadas con los comentarios de pistas.

De acuerdo con lo observado en la Figura 15, los comentarios negativos de pistas que están representados por color azul, en comparación con las otras dos obras de parques y veredas, se encuentran presentes en una gran proporción en la mayoría de los distritos. Se observa que a pesar de que algunos distritos sean de clase media alta, como en el caso de La Molina y San Borja, también presentan una representación elevada de comentarios negativos en referencia a pistas.

Figura 15

Distribución de Predicciones de Comentarios por Distritos de Lima en Pistas



Nota. Elaborado con la plataforma Microsoft Power BI, por Microsoft ©, 2021 (https://powerbi.microsoft.com/es-es/).

4.5 Análisis de Resultados

Para tener una mejor visión de los resultados obtenidos, en la Tabla 25 se resumen considerando las cuatro métricas utilizadas del total de la información analizada y dividida en los tres tipos de obras que fueron materia del presente trabajo de investigación.

Tabla 25

Resultado de Métricas del Modelo

Métrica	Total de obras	Parques	Veredas	Pistas
Precisión	0.69	0.857	0.609	0.585
Recall	0.890	0.893	0.933	0.857
F1-score	0.780	0.874	0.737	0.7
Accuracy	0.770	0.840	0.762	0.7

Como se mencionó en el inicio del capítulo, el indicador de éxito de la presente investigación es el *F1-score*, por tener clases desiguales. Para el caso general de las obras urbanas, el *F1-score* muestra un valor de 78%, que es mayor al *F1-score* de entrenamiento (76%). Este resultado resume la precisión y la sensibilidad del modelo.

En el caso de esta investigación, se tiene un *recall* alto y una precisión menor al *recall*. Esto demuestra que el modelo detecta bien los comentarios negativos, pero incluye algunos comentarios positivos (Barrios, 2019). Este valor es el esperado por tener la mayor cantidad de comentarios negativos (64%) frente a los comentarios positivos (35%), lo cual se debe a que la data de entrenamiento utilizada tenía mayor número de comentarios negativos.

Al comparar este resultado con investigaciones utilizando *datasets* con clases desiguales, el valor obtenido está dentro del rango comparable (Vakili et al., 2019). En el caso de parques, al evaluar por el tipo de obra urbana se observa que el modelo maneja bien la clase de comentarios negativos, debido a los resultados obtenidos de alto en *recall* (89.3%) y un resultado similar en *precisión* (85.7%). Este resultado se refleja en el *F1-score*, que es alto (87.4%), y está dentro del rango comparable (Valki et al., 2019).

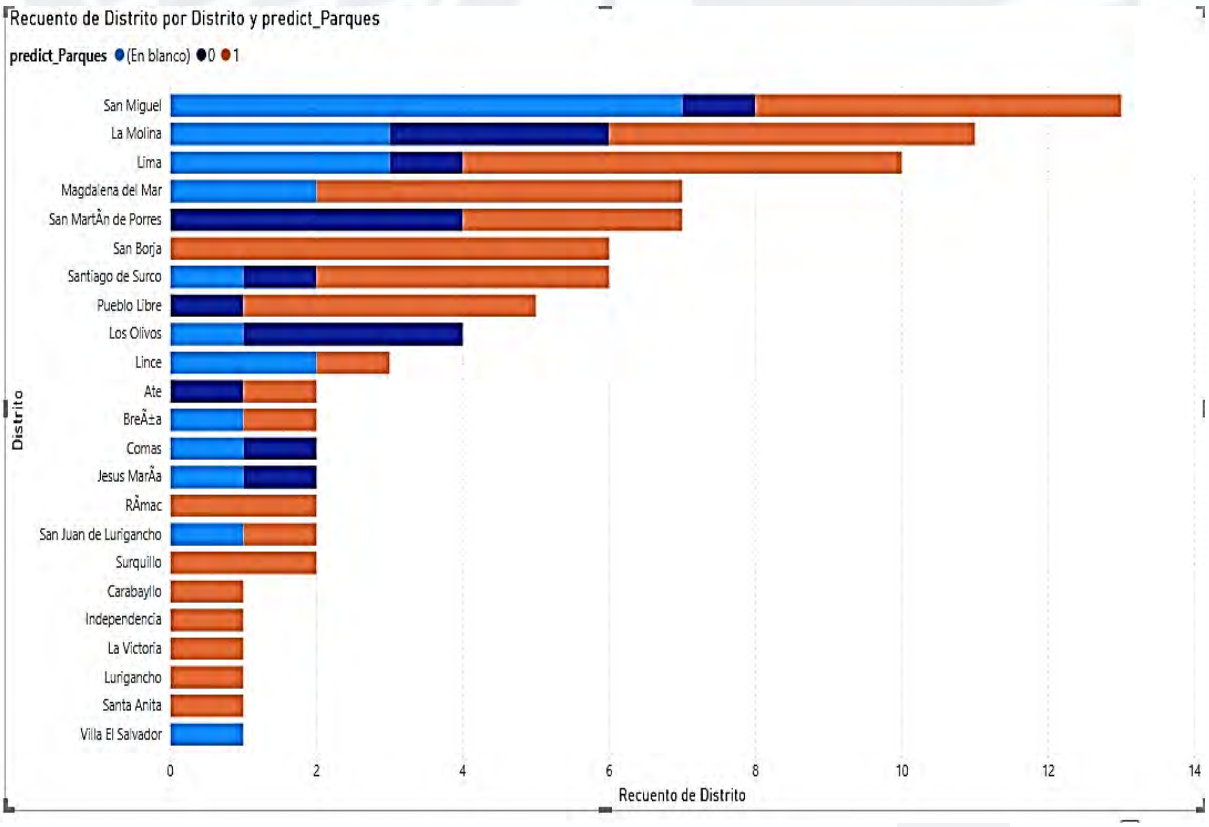
En cuanto a los casos de veredas y pistas, el modelo detectó bien la clase de comentarios negativos, pero incluye muestras de la otra clase de comentarios positivos. En veredas, el *recall* (93.3%) fue mayor que el valor obtenido en precisión (60.9%). Caso similar ocurrió con los resultados de pistas, donde el *recall* (85.7%) también fue mayor a su *precisión* (58.5%). Para ambos tipos de obras, sus valores en el *F1-score* son de 73.7% y 70%, respectivamente. Esto demuestra que el modelo detectó bien los comentarios negativos, pero incluye mayores comentarios positivos comparado con el resultado de los parques (Arce, 2019). El *F1-score* obtenido es cercano al rango comparable (Vakili, et al., 2019).

Sobre las zonas de la ciudad de Lima que presentan problemas relacionados con las obras que han sido materia de esta investigación, se analizarán las predicciones realizadas por

el modelo de acuerdo con cada una de estas obras. En la Figura 16 se muestra este resultado para el caso de los parques, donde las predicciones correspondientes a comentarios negativos aparecen de color azul; así, los distritos que tienen un mayor número de estos comentarios son Los Olivos con 75%, San Martín de Porres (57%), Ate y Comas (50%); esto en relación con el total de comentarios analizados por cada distrito. Cabe mencionar que San Borja presenta la totalidad de sus comentarios como positivos. Esta información puede ser llevada a la Municipalidad de Lima, para que sirva como un elemento más para la definición en la estrategia de sus obras relacionadas con el mejoramiento de parques en la ciudad de Lima, donde puedan, con el apoyo de los comentarios de la ciudadanía, definir una prioridad en aquellos distritos que tienen el mayor porcentaje de insatisfacción en estas obras. Este comentario está sustentado por el valor alto obtenido del *F1-score* (87.4%).

Figura 16

Distribución Acumulada de Predicciones de Comentarios por Distritos de Lima en Parques

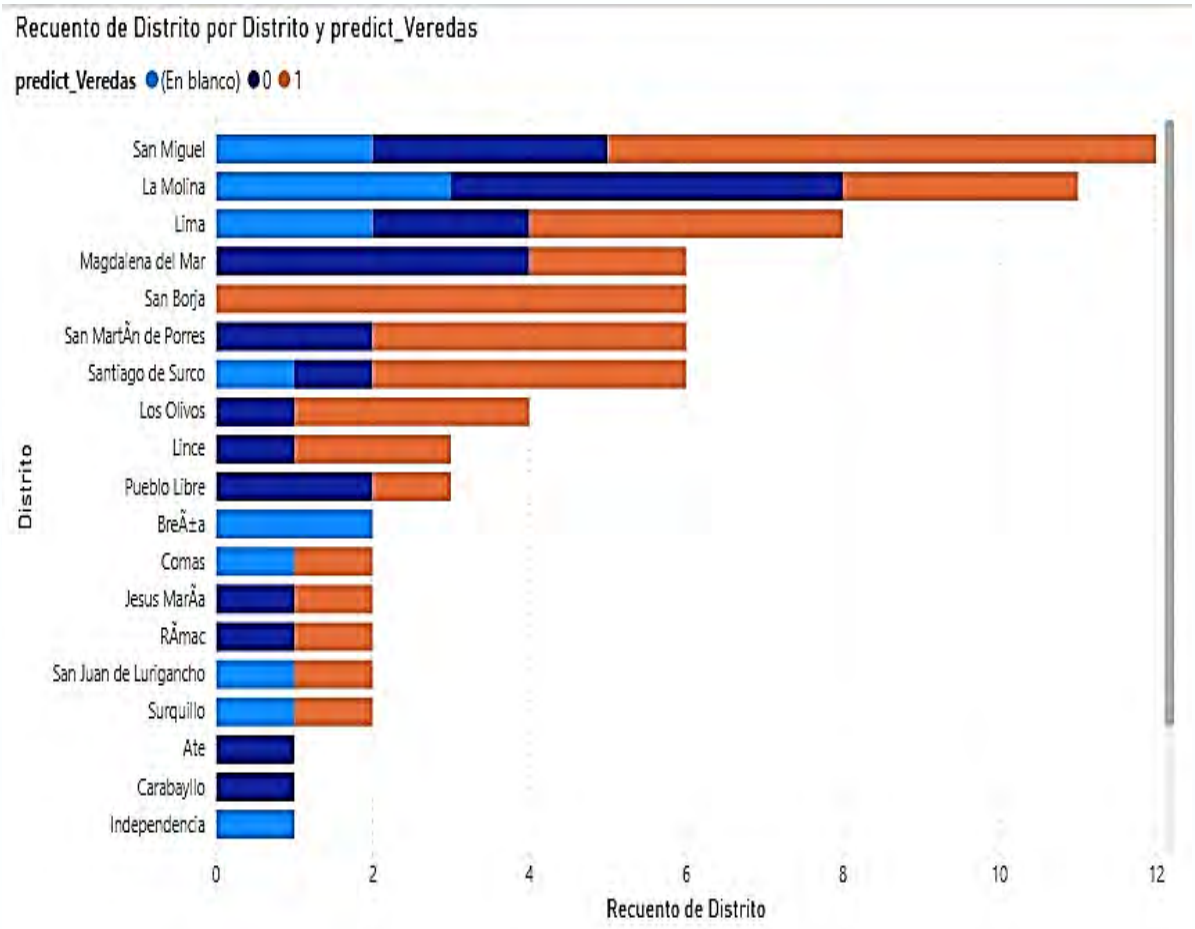


Nota. Elaborado con la plataforma Microsoft Power BI, por Microsoft ©, 2021 (https://powerbi.microsoft.com/es-es/).

Para el análisis de las veredas, se realizó una representación similar de forma gráfica, que ayudará a tener una visión completa de la información obtenida para estas obras. La Figura 17 muestra esta distribución.

Figura 17

Distribución Acumulada de Predicciones de Comentarios por Distritos de Lima en Veredas



Nota. Elaborado con la plataforma Microsoft Power BI, por Microsoft ©, 2021 (https://powerbi.microsoft.com/es-es/).

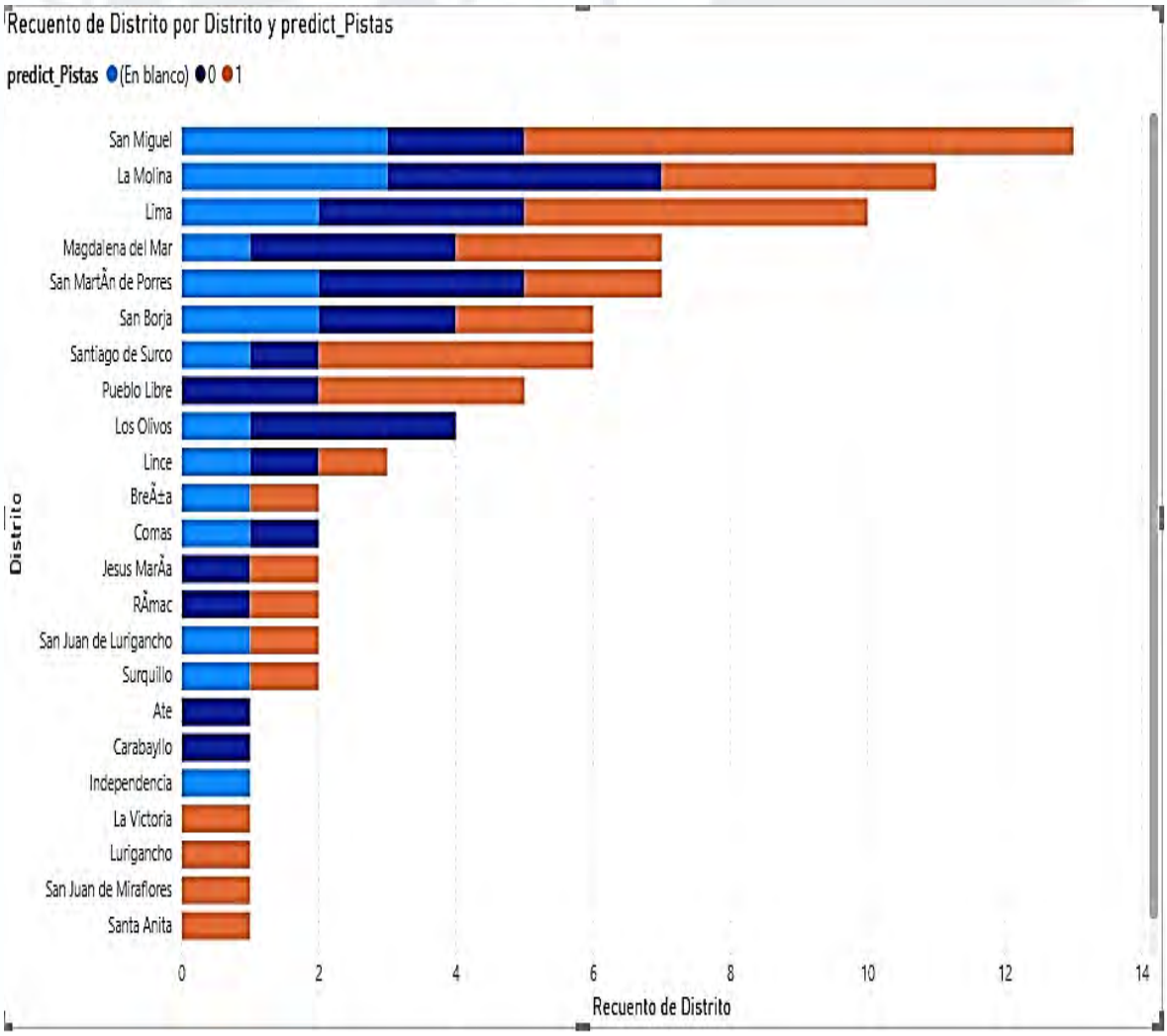
Se puede apreciar que los comentarios negativos son los indicados en color azul. Se resalta, asimismo, una observación clara en este gráfico que repite el comportamiento de los análisis relacionados con parques, en cuanto a que en el distrito de San Borja los comentarios son positivos en su totalidad, por lo que se podría indicar que este distrito viene manejando correctamente los trabajos de mantenimiento en este tipo de obras. Por el contrario, los distritos con mayor tasa de comentarios negativos en veredas son Magdalena del Mar y

Pueblo Libre con el 67%, La Molina con 45% y San Martín de Porres con 33%. Los resultados muestran una oportunidad de mejora para este tipo de obras de tránsito peatonal en los distritos mencionados. Con referencia al *F1-score* (73.7%), se tiene una incertidumbre alta de 27.3% sobre los porcentajes mostrados en los distritos de estudio, que puede afectar la toma de decisiones.

Por último, para el caso de las obras referidas a pistas, se ha empleado el mismo análisis por zonas en los distritos de Lima. En la Figura 18 se aprecian los resultados donde se realizó la distribución de las predicciones para este tipo de obras.

Figura 18

Distribución Acumulada de Predicciones de Comentarios por Distritos de Lima en Pistas



Nota. Elaborado con la plataforma Microsoft Power BI, por Microsoft ©, 2021 (https://powerbi.microsoft.com/es-es/).

En comparación con los otros dos tipos de obras, los comentarios negativos de color azul se encuentran presentes en la mayoría de los distritos, incluso en San Borja, que había demostrado tener un mejor resultado tanto en parques como en veredas, por lo que se podría indicar que esto es un problema a nivel de ciudad. Los distritos con mayor porcentaje de comentarios negativos en pistas son Los Olivos (67%), San Martín de Porres (60%), Magdalena (50%) y La Molina (44%). Esta información es de gran relevancia para el mejoramiento de este tipo de obras en los distintos distritos de la ciudad. Con referencia al *F1-score* (70%), se tiene una incertidumbre alta de 30% sobre los porcentajes mostrados en los distritos de estudio para este tipo de obras, que puede afectar la toma de decisiones.

De acuerdo con toda la información de predicciones revisadas en la presente investigación, deben validarse las respuestas a las preguntas planteadas, donde la primera pregunta hace referencia a si el uso de herramientas de *machine learning* permiten identificar los problemas que presenta la ciudadanía limeña sobre obras públicas de parques, veredas y pistas. La respuesta es sí; gracias al *machine learning* y al uso de información de redes sociales es posible recoger el sentir de los ciudadanos manifestado en estos medios de comunicación, las técnicas de aprendizaje automático permiten identificar los inconvenientes. Vale la pena mencionar que si se desea llegar a un mayor detalle en el análisis de los problemas, se pueden utilizar otras técnicas de análisis de texto para aprovechar la información de las redes sociales.

Referente a la segunda pregunta de investigación, respecto a en qué medida las herramientas analíticas pueden identificar las zonas de la ciudad donde se presentan problemas en este tipo de obras públicas, de acuerdo con la métrica del *F1-score* obtenida en la predicción se pueden identificar las zonas en un 78%. Este resultado obtenido con el modelo planteado, permite algunas mejoras a la calidad de predicción de modelo, pero

requerirá una mayor inversión de tiempo tanto para la recolección de datos como para el aprendizaje, además de considerar tener un *dataset* equilibrado.

Por último, la tercera pregunta planteada hace referencia a si el uso de la información obtenida de Twitter con herramientas de *machine learning* permite determinar las obras de parques, pistas y veredas con mayor número de quejas. Efectivamente, gracias al uso de los comentarios de Twitter y las predicciones realizadas por el modelo basado en *machine learning* se pudo determinar aquellas obras en distritos de la ciudad de Lima con mayor número de comentarios negativos. El resultado final depende mucho de los datos utilizados en el modelo; si no existen comentarios negativos sobre una obra, entonces no se podrá determinar la cantidad de quejas para ese distrito. El valor que brindan los datos es muy importante, pues permiten entrenar al modelo y, por tanto, su resultado mejorará mientras más datos tenga a disposición para su aprendizaje.

Capítulo V: Conclusiones y Recomendaciones

5.1 Conclusiones

El uso de tecnologías basadas en *machine learning* como el procesamiento de lenguaje natural con *support vector machine*, modelo utilizado en este trabajo de investigación, permitió identificar la necesidad de realizar obras urbanas en la ciudad de Lima con una precisión y sensibilidad del 78%, por lo que se concluye que es posible utilizar este tipo de tecnologías en el sector público para incluir la opinión de los ciudadanos, manifestada en las redes sociales, en la toma de decisiones de los organismos gubernamentales correspondientes, y así encaminar el desarrollo de la ciudad en el marco de una *smart city*.

Con la aplicación del modelo de *machine learning* utilizado se identificaron las zonas de Lima donde se presentan problemas en las obras públicas, con una precisión y sensibilidad de 87.4% para parques, 73.7% para veredas y 70% para pistas. Estos resultados son consecuencia de que la recolección de datos para la predicción contiene comentarios positivos, negativos y neutros, y para la investigación solo se consideraron aquellos positivos y negativos.

Los datos utilizados para el entrenamiento del modelo son de mucha importancia para obtener un mejor resultado. Para el trabajo de investigación, se utilizaron datos abiertos relacionados con distintos temas, con lo que se obtuvo en esta etapa de entrenamiento una precisión y sensibilidad del 76%, que es comparable a otros estudios que utilizaron un modelo de *support vector machine.*

La información obtenida a partir de los comentarios relacionados con los parques, veredas y pistas, debidamente analizada con el uso del modelo de *machine learning* planteado, permitió identificar las obras urbanas con mayor porcentaje de quejas. De acuerdo con los resultados obtenidos, las obras relacionadas con “pistas” muestran un mayor número de comentarios negativos, donde Los Olivos (67%) y San Martín de Porres (60%) son los distritos con mayor porcentaje de comentarios negativos.

Existe la capacidad de analizar toda la información relacionada con las obras urbanas tomando en consideración las expectativas de los ciudadanos mediante el uso de herramientas de *machine learning*, que permite disminuir el riesgo de ejecutar una obra mal hecha o dejarla inconclusa; por lo que se cumple con el objetivo del trabajo de investigación.

5.2 Recomendaciones

En el presente trabajo de investigación se utilizó el algoritmo de *support vector machine*, usado en documentos de investigación y en temas relacionados con el análisis de sentimientos. Para futuras investigaciones, se recomienda el uso de métodos combinados o híbridos como son los métodos SVM-*baileys* o SVM-CNN (redes neuronales), cuya ventaja se apreciará en mayores porcentajes de precisión y sensibilidad. Para este caso es necesaria una mayor inversión de equipos para el procesamiento, con lo que se reducirán los tiempos de entrenamiento.

Se recomienda que la recopilación de datos para investigaciones futuras se amplíe a redes sociales como Facebook, Instagram y Linkedin, considerando un periodo de recopilación de datos mayor a seis meses. Además, se pueden incluir diccionarios de palabras relacionadas con el tema de investigación para la etapa de preparación de datos y así obtener resultados de precisión más altos que los encontrados en esta investigación.

El presente trabajo de investigación puede ser replicado a otros problemas sociales que afecten a la ciudad de Lima, integrando este modelo a las plataformas de las entidades públicas, lo que servirá de entrada para la mejora de sus procesos.

En la elaboración de modelos que utilicen como fuente comentarios de redes sociales se deben considerar clasificaciones adicionales a los comentarios positivos o negativos, ya que los ciudadanos también realizan comentarios que escapan a esta clasificación, como pueden ser comentarios informativos, neutros, advertencias, entre otros, por lo que los

modelos por implementar deben ampliar clasificaciones relacionadas con este tipo de expresiones, a fin de reducir los falsos positivos que pueda presentar el modelo.

Se recomienda realizar un estudio que tenga como objetivo la elaboración de una base de datos pública con comentarios de redes sociales relacionados con obras públicas que se ejecutan en las ciudades del Perú, con los parámetros de tiempo y clasificación indicados en los párrafos anteriores, de tal manera que esta base de datos quede a disposición de las distintas entidades que requieran utilizarla para el entrenamiento de sus modelos, buscando un resultado de precisión y sensibilidad mayor al 78% obtenido en esta investigación.

Referencias

Aiala R., Chiruzzo, L., Etcheverry, M., & Castro, S. (2017, setiembre). RETUYT en TASS 2017: Análisis de sentimiento de tweets en español utilizando SVM y CNN. En *TASS 2017: Workshop on Semantic Analysis at SEPLN* (pp. 77-83). http://ceur-ws.org/Vol-1896/p9_retuyt_tass2017.pdf

Agencia Europea de Medio Ambiente. (2015). *El medio ambiente en Europa: Estado y perspectivas 2015.* Oficina de Publicaciones de la Unión Europea. https://www.eea.europa.eu/soer/2015/synthesis/el-medio-ambiente-en-europa

Aguilar, L. A., & Vásquez, Y. O. (2016). *Principal component analysis (PCA) para mejorar la performance de aprendizaje de los algoritmos support vector machine (SVM) y red neuronal multicapa (MLNN)* [Tesis de grado, Universidad Privada Antenor Orrego, Trujillo, Perú]. https://hdl.handle.net/20.500.12759/3398

Allam, Z., & Dhunny, Z. (2019, June). On big data, artificial intelligence and smart cities. *Elsevier, 89*, 80-91. https://doi.org/10.1016/j.cities.2019.01.032

Apolitano, J. (2019, 03 de junio). Las ciudades inteligentes y los retos que plantean. *El Pueblo*. https://www.elpueblo.pe/las-ciudades-inteligentes-y-los-retos-que-plantean/

Barrios, J. (2019, 26 de julio). La matriz de confusión y sus métricas. *Big Data.* https://www.juanbarrios.com/la-matriz-de-confusion-y-sus-metricas/

Bouskela, M., Casseb, M., Bassi, S., De Luca, C., & Facchina, M. (2016). *La ruta hacia las smart cities: Migrando de una gestión tradicional a la ciudad inteligente.* BID. https://publications.iadb.org/publications/spanish/document/La-ruta-hacia-las-smart-cities-Migrando-de-una-gesti%C3%B3n-tradicional-a-la-ciudad-inteligente.pdf

Candia, D. (2019). *Predicción del rendimiento académico de los estudiantes de la UNSAAC a partir de sus datos de ingreso utilizando algoritmos de aprendizaje automático* [Tesis de maestría, Universidad Nacional San Antonio de Abad, Cusco, Perú].

http://repositorio.unsaac.edu.pe/bitstream/handle/20.500.12918/4120/253T20191024_TC.pdf?sequence=1&isAllowed=y

Caragliu, A., & Del Bo, C. (2019). Smart innovative cities: The impact of smart city policies on urban innovation. *Technological Forecasting and Social Change, Elsevier, 142*(C), 373-383. https://doi.org/10.1016/j.techfore.2018.07.022

Carmona, E. (2016). *Tutorial sobre máquinas de vectores soporte (SVM).* https://www.researchgate.net/publication/263817587_Tutorial_sobre_Maquinas_de_Vectores_Soporte_SVM

Carrasco, L. G. (2019). *Métodos de análisis para identificar público objetivo que consuma productos de artesanía a través de Big Data Piura -2019* [Tesis de grado, Universidad César Vallejo, Piura, Perú]. https://repositorio.ucv.edu.pe/bitstream/handle/20.500.12692/55041/Carrasco_CLG-SD.pdf?sequence=1&isAllowed=y

Cohen, B. (2014, 20 de noviembre). The smartest cities in the world 2015: Methodology. *Fast Company*. https://www.fastcompany.com/3038818/the-smartest-cities-in-the-world-2015-methodology

ComexPerú. (2018, 28 de setiembre). Censos nacionales 2017. ¿Cuántos somos? *Semanario 955*. https://www.comexperu.org.pe/articulo/censos-nacionales-2017-cuantos-somos

ComexPerú. (2021, 26 de marzo). Cusco, Huancavelica y Ayacucho cuentan con el menor porcentaje de hogares con acceso a Internet. *Semanario ComexPerú 1068.* https://www.comexperu.org.pe/articulo/cusco-huancavelica-y-ayacucho-cuentan-con-el-menor-porcentaje-de-hogares-con-acceso-a-internet

Condori, A., & Valeriano, K. (2020). *Detection of suicidal intent in Spanish language social networks using Machine Learning* [Tesis de grado, Universidad Nacional de San Agustín, Arequipa, Perú].

http://repositorio.unsa.edu.pe/bitstream/handle/20.500.12773/12562/IScolaak_vavaky.pdf?sequence=1&isAllowed=y

Costa, V., & Dellunde, P. (2015). On free models for horn clauses over predicate fuzzy logics. *Artificial Intelligence Research and Development, 277*, 49-58. https://www.researchgate.net/publication/299135205_On_Free_Models_for_Horn_Clauses_over_Predicate_Fuzzy_Logics

Fandiño, M. I. (2005). *Le frazioni. Aspetti concettuali e didattici* [Las fracciones. Aspectos conceptuales y didácticos]. Pitagora.

Fernández, M. (2015). *La smart city como imaginario socio-tecnológico* [Tesis de doctorado, Universidad del País Vasco, Vizcaya, España]. http://polired.upm.es/index.php/ciur/article/viewFile/3498/3572

Gamarra, C. A., & Ríos, M. S. (2018). *Aplicación de técnicas de aprendizaje profundo para la clasificación y reconocimiento de objetos en imágenes* [Tesis de grado, Universidad Santo Tomás, Bogotá, Colombia]. https://repository.usta.edu.co/bitstream/handle/11634/10680/2018Gamarracamilo.pdf?sequence=1&isAllowed=y

García, J. (2019). *Implementación de un modelo computacional basado en reglas de clasificación supervisadas para la predicción de la deserción estudiantil en la Universidad Peruana Unión Filial Juliaca* [Tesis de grado, Universidad Peruana Unión, Juliaca, Perú]. https://repositorio.upeu.edu.pe/bitstream/handle/20.500.12840/1975/Jacob_Tesis_Licenciatura_2019.pdf?sequence=1&isAllowed=y

González, F. J., Navia, A., & Amor, A. M. (2017). Training support vector machines with privacy-protected data. *Pattern Recognition, 72,* 93-107. https://doi.org/10.1016/j.patcog.2017.06.016

Gordon, M. E. (2018). *Desarrollo de una herramienta de minería de datos para el análisis de influencia de cuentas automatizadas en temas de tendencia sobre la opinión de los usuarios de Twitter en Ecuador* [Tesis de maestría, Universidad Internacional SEK, Quito, Ecuador]. https://repositorio.uisek.edu.ec/bitstream/123456789/3023/2/Tesis%20Mario%20Gordon.pdf

Grández, M. A. (2017). *Aplicación de minería de datos para determinar patrones de consumo futuro en clientes de una distribuidora de suplementos nutricionales* [Tesis de grado, Universidad San Ignacio de Loyola, Lima, Perú]. http://repositorio.usil.edu.pe/bitstream/USIL/2763/1/2017_Granda_Aplicacion-de-mineria-datos.pdf

Grupo Tecma Red. (2015, 17 de noviembre). Informe sobre smart cities de PWC, IE Business School y Telefónica. *Esmart City.* https://www.esmartcity.es/2015/11/17/informe-sobre-smart-cities-de-pwc-ie-business-school-y-telefonica

Hernández, R., Fernández, C., & Baptista, P. (2014). *Metodología de la investigación* (6a ed.). McGraw-Hill.

Herrera, L. G. (2020). *Comparación de métodos para clasificar comentarios de lugares turísticos por medio de análisis de sentimiento* [Tesis de grado, Universidad de Lima, Lima, Perú]. https://repositorio.ulima.edu.pe/bitstream/handle/20.500.12724/12195/Herrera_Sarmiento_Luis_Guillermo.pdf?sequence=1&isAllowed=y

Hewlett Packard Enterprise. (2018). *¿Qué es la inteligencia artificial?* https://www.hpe.com/lamerica/es/what-is/artificial-intelligence.html

IESE Business School. (2020). *Índice IESE cities in motion 2020.* https://media.iese.edu/research/pdfs/ST-0542.pdf

International Standardization Organization. (2019). *ISO 37122: Sustainable cities and communities - Indicators for smart cities.* https://www.iso.org/obp/ui/#iso:std:iso:37122:ed-1:v1:en

Jerí, J. A., & & Sosa, Y. L. (2019). *Uso de indicadores big data para mejorar el nivel de ajuste de un modelo autorregresivo de arribos domésticos al Aeropuerto Internacional Jorge Chávez* [Tesis de grado, Universidad San Ignacio de Loyola, Lima, Perú]. http://repositorio.usil.edu.pe/bitstream/USIL/8765/1/2019_Jeri-Jong.pdf

Kalbandi, I., & Anuradha, J. (2015). A brief introduction on big data 5Vs characteristics and Hadoop technology. *Procedia Computer Science, 48*, 319-324. doi:10.1016/j.procs.2015.04.188

Laney, D. (2001). *3D Data Management: Controlling data volume, velocity, and variety.* META Group. https://studylib.net/doc/8647594/3d-data-management--controlling-data-volume--velocity--an...

Lima Cómo Vamos. (2019). *Lima y Callao según sus ciudadanos: Décimo informe urbano de percepción sobre calidad de vida en la ciudad.* IOP-PUCP. http://www.limacomovamos.org/wp-content/uploads/2019/11/Encuesta-2019_web.pdf

Lima: Pistas con huecos son un peligro latente para conductores. (2020, 11 de agosto). *Perú 21.* https://peru21.pe/lima/lima-pistas-huecos-son-peligro-latente-conductores-fotos-196125-noticia/

Liu, B. (2015). *Sentiment analysis: Mining opinions, sentiments, and emotions.* Cambridge University Press. https://doi.org/10.1017/CBO9781139084789

López, B. (2007). *Introducción a la inteligencia artificial.* Instituto Tecnológico de Nuevo Laredo. http://itnuevolaredo.edu.mx/takeyas/Articulos/Inteligencia%20Artificial/ARTICULO%20Introduccion%20a%20la%20Inteligencia%20Artificial.pdf

Maldonado, C., Mendoza, E., Noriega, R., Piedra, L., & Rodríguez, D. (2020). Determinación de los factores críticos para la transformación de un distrito de Lima Metropolitana en una smart city [Tesis de maestría, ESAN Graduate School of Business, Lima, Perú). https://repositorio.esan.edu.pe/bitstream/handle/20.500.12640/2205/2020_MATC_19-1_04_T.pdf?sequence=1&isAllowed=y

Mamani, D. I. (2019). *Modelo de minería de datos basado en factores asociados para la predicción de deserción estudiantil universitaria* [Tesis de grado, Universidad Nacional de Moquegua, Moquegua, Perú]. http://repositorio.unam.edu.pe/bitstream/handle/UNAM/94/T095_72389106_T.pdf?sequence=1&isAllowed=y

Manupati, V., Anand, R., Thakkar, J., Benyoucef, L., Garsia, F., & Tiwari, M. (2013). Adaptive production control system for a flexible manufacturing cell using support vector machine-based approach. *International Journal of Advanced Manufacturing Technology, 67*, 969-981. https://doi.org/10.1007/s00170-012-4541-1

Marín, P., & Díaz, A. (2015). Uso de Twitter por los partidos y candidatos políticos en las elecciones autonómicas de Madrid 2015. *Ámbitos, 1*(32), 1-6. https://www.redalyc.org/pdf/168/16845702009.pdf

McCarthy, J., Minsky, M., Rochester, N., & Shannon, C. (1955). A proposal for the Dartmouth summer research project on artificial intelligence. *AI Magazine, 27*(4), 12-14. https://doi.org/10.1609/aimag.v27i4.1904

Medhat, W., Hassan, A., & Korashy, H. (2014). Sentiment analysis algorithms and applications: A survey. *Ain Shams Engineering Journal, 5*(4), 1093-1113. https://doi.org/10.1016/j.asej.2014.04.011

Microsoft. (2021). *Microsoft Power BI.* https://powerbi.microsoft.com/es-es/

Ministerio de Transportes y Comunicaciones. (2015, noviembre). Soluciones inteligentes de movilidad urbana y green buildings. *Seminario Peruano-Alemán Smart City.* https://documen.site/download/soluciones-inteligentes-de-movilidad-urbana-y-green-buildings_pdf

Miranda, F. P. (2019). *Diseño de un proceso de alertas tempranas para disminuir las deserciones de los estudiantes de primer año en una institución de educación superior* [Tesis de maestría, Universidad de Chile, Santiago de Chile, Chile]. http://repositorio.uchile.cl/bitstream/handle/2250/172649/Dise%C3%B1o-de-un-proceso-de-alertas-tempranas-para-disminuir-las-deserciones.pdf?sequence=1

Mitchell, T. (1997). *Machine learning*. McGraw-Hill.

Montesinos, L. (2014). *Análisis de sentimientos y predicción de eventos en Twitter* [Tesis de grado, Universidad de Chile, Santiago de Chile, Chile]. http://repositorio.uchile.cl/bitstream/handle/2250/130479/cf-montesinos_lg.pdf?sequence=1&isAllowed=y

Moreno, J. (1940). *Fundamentos de sociometría*, Paidós.

Muñoz, L., Delgado, J., & Rodriguez, V. (2018). Measurement of air pollution with low-cost technology. In *13th Iberian Conference on Information Systems and Technologies* (pp. 1-5). doi: 10.23919/CISTI.2018.8399368

Nam, T., & Pardo, T. (2011). Conceptualizing smart city with dimensions of technology, people, and institutions. In *12th Annual International Digital Government Research Conference* (pp. 282-291). doi: 10.1145/2037556.2037602

Olarte, A., & Casaverde, A. (2020). A*nálisis masivo de datos en Twitter para identificación de opinión* [Tesis de grado, Universidad Nacional de San Antonio Abad del Cusco, Cusco, Perú].

http://repositorio.unsaac.edu.pe/bitstream/handle/20.500.12918/5252/253T20200108_TC.pdf?sequence=1&isAllowed=y

Organización de las Naciones Unidas. (s.f.). *Objetivos de desarrollo sostenible.* https://www.un.org/sustainabledevelopment/es/objetivos-de-desarrollo-sostenible/

Organización de las Naciones Unidas. (2018, 16 de mayo). Las ciudades seguirán creciendo, sobre todo en los países en desarrollo. *Noticias ONU.* https://www.un.org/development/desa/es/news/population/2018-world-urbanization-prospects.html

Pedregosa, F., Varoquaux, G., Gramfort, A., & Michel, M. (2011). *Scikit-learn: Machine learning in Python*. https://scikit-learn.org/stable/

Piva, C. (2016, November). Smart City Expo World Congress 2016. *TM Forum.* https://www.tmforum.org/events/smart-city-expo-world-congress-2016/

Rollins, J. B. (2015). *Metodología fundamental para la ciencia de datos.* IBM Corporation. https://www.ibm.com/downloads/cas/WKK9DX51

Rouse, M. (2019, agosto). La inteligencia artificial o IA. *Search Data Center.* https://searchdatacenter.techtarget.com/es/definicion/Inteligencia-artificial-o-AI

Russell, S., & Norvig, P. (2004). *Inteligencia artificial: Un enfoque moderno* (2a ed.). Pearson.

SAS Institute. (2018, September 5). Analytics experience 2018 explores AI, machine learning, IoT and more. SAS conference will include Hackathon aimed at fighting wildfires. *SAS Perú*. https://www.sas.com/es_pe/news/press-releases/locales/2018/analytics-experience-san-diego.html

Secretaría de Gobierno y Transformación Digital de la Presidencia del Consejo de Ministros. (2021). *Estrategia nacional de inteligencia artificial. Documento de trabajo para la participación de la ciudadanía 2021-2026*.

https://cdn.www.gob.pe/uploads/document/file/1899077/Estrategia%20Nacional%20de%20Inteligencia%20Artificial.pdf

Segura, L. (2019). *Evaluación de algoritmos de clasificación para el minado de opinión en Twitter* [Tesis de grado, Universidad Señor de Sipán, Pimentel, Perú]. https://repositorio.uss.edu.pe/handle/20.500.12802/6253

Sobrino, J. C. (2018). *Análisis de sentimientos en Twitter* [Tesis de maestría, Universidad Oberta de Catalunya, Barcelona, España]. http://openaccess.uoc.edu/webapps/o2/bitstream/10609/81435/6/jsobrinosTFM0618memoria.pdf

Sota, L. A. (2018). *Modelo de evaluación de ciudades, basado en el concepto de smart city* [Tesis de maestría, Universidad Cesar Vallejo, Trujillo, Perú]. https://repositorio.ucv.edu.pe/handle/20.500.12692/33830

Thakuriah, P. V., Tilahun, N. Y., & Zellner, M. L. (2017). Introduction to seeing cities through Big Data: Research, methods and applications in urban informatics. In *Seeing cities through Big Data* (pp. 1-9). Springer. https://doi.org/10.1007/978-3-319-40902-3_1

Torra, V. (2011). La inteligencia artificial. *Lychnos: Cuadernos de la Fundación General CSIC.* https://www.fgcsic.es/lychnos/es_es/articulos/inteligencia_artificial

Vakili, M., Ghamsari, M., & Rezaei, M. (2019, January). Performance analysis and comparison of Machine and Deep Learning algorithms for IoT data classification. *ResearchGate.* https://www.researchgate.net/publication/338853237_Performance_Analysis_and_Comparison_of_Machine_and_Deep_Learning_Algorithms_for_IoT_Data_Classification

Vidal, M., & Vialart, M. (2013, marzo). Redes sociales. *Revista Cubana de Educación Médica Superior 27*(1), 146-157. https://www.researchgate.net/publication/262782998_Redes_sociales

Vishnivetskaya, A., & Alexandrova, E. (2019). Smart city concept. Implementation practice. *IOP Conference Series: Materials Science and Engineering, 497*, 1-6. doi:10.1088/1757-899X/497/1/012019

Apéndice A: Formulario Comprensión del Negocio

Contexto:

Para nuestro caso, al tratarse de un tema público y no de una empresa en particular, debemos donde se debe detallar el bien público que se busca lograr con el desarrollo del trabajo de investigación. Otros campos que se consideran y se piden especificar son el alcance de las obras públicas, para nuestro caso, parques, pistas y veredas, también se pide delimitar el alcance geográfico pudiendo llegar hasta distritos, y por último se solicita un listado de palabras críticas que son las que ayudaran a definir el modelo.

Problema:

Existe una falta de sintonía entre las necesidades actuales de la ciudadanía y las obras públicas planteadas para la ciudad de Lima con referencia a los problemas que la ciudadanía presenta, donde no se consideran las urgencias y reclamos de los ciudadanos como apoyo para la definición de obras urbanas en Lima Metropolitana.

Objetivos:

Objetivo 1

Utilizar la tecnología basada en *Machine Learning* para la identificación de obras urbanas en la ciudad de Lima.

Objetivo 2

Identificar los problemas que presenta la ciudadanía limeña sobre obras públicas referentes a parques, veredas y pistas.

Utilizar la información registrada en las redes sociales de Twitter para identificar las zonas en la ciudad de Lima donde se presentan los problemas referidos a obras públicas específicamente parques, veredas y pistas

Objetivo 3

Determinar las obras públicas de parques, pistas y veredas, que presentan el mayor número de quejas y reclamos utilizando la información obtenida de las redes sociales como Twitter con herramientas analíticas basadas en *Machine Learning*

Requisitos de la solución

Requisito 1

Paso 1: Colectar Datos.

Paso 2: Preparar los datos

Paso 3: Elegir el modelo

Paso 4 Entrenar nuestra máquina

Paso 5: Evaluación

Paso 6: Parameter Tuning (configuración de parámetros)

Alcance de Obras Públicas:

El alcance de las obras es de Lima Metropolitana

Factores Críticos de Éxito:

RESPALDO DE LA ALTA DIRECCIÓN, COMPROMISO DEL EQUIPO, PERSONAL CALIFICADO, PRESUPUESTO.

Apéndice B: Matriz Enfoque Analítico

Problema	Objetivos	Objetivos analíticos	Enfoque analítico
Falencia en el mantenimiento de áreas verdes y pistas en la ciudad de Lima.	Identificar los problemas que presenta la ciudadanía limeña sobre obras públicas referentes a parques, veredas y pistas.	Predecir si un comentario de Twitter relacionado a una obra pública de parque, vereda o pista es "Positivo" o "Negativo".	Construcción, pruebas e implementación de un modelo de clasificación.
	Utilizar la información registrada en la red social Twitter para identificar las zonas en la ciudad de Lima donde se presentan los problemas referidos a obras públicas específicamente parques, veredas y pistas.	Realizar un análisis descriptivo sobre los comentarios de Twitter con predicciones negativas determinando los distritos de la ciudad de Lima que presentan problemas en obras públicas de parques, veredas y pistas.	Contrucción e implementación de un análisis descriptivo cualitativo.
	Determinar las obras públicas de parques, pistas y veredas, que presentan el mayor número de quejas y reclamos utilizando la información obtenida de la red social Twitter.	Realizar un análisis descriptivo sobre los comentarios de Twitter con predicciones negativas determinando las obras públicas con mayor número de comentarios negativos.	Contrucción e implementación de un análisis descriptivo cuantitativo.

Apéndice C: Matriz Requisitos de Datos

Enfoque Analítico	Fuente	Dato	Tipo
Construcción, pruebas e implementación de un modelo de clasificación.	Red Social Twitter	Comentarios de Twitter	Texto
	Red Social Twitter	Predicción del Modelo "Positivo" o "Negativo"	Texto
Contrucción e implementación de un análisis descriptivo cualitativo.	Red Social Twitter	Comentarios de Twitter	Texto
	Resultado del Modelo	Predicción del Modelo "Positivo" o "Negativo"	Texto
	Red Social Twitter	Ubicación de Twitter	Texto
Contrucción e implementación de un análisis descriptivo cuantitativo.	Red Social Twitter	Comentarios de Twitter	Texto
	Resultado del Modelo	Predicción del Modelo "Positivo" o "Negativo"	Texto
	Resultado del Modelo	Número de comentarios Negativos	Númerico
	Red Social Twitter	Obra del comentario Negativo	Texto

Apéndice D: Código Fuente y Resultados

Análisis de Sentimientos

Esta primera parte de este proyecto tiene como objetivo la deteccion de emosiones positivas onegativas (sentiment analysis) de los comentarios *tweets*, para nuestro caso en particular

Primero se usa la dataset con todos los temas relacionados: politicos, sociales, generales, etc ,pre-procesar y finalmente entrenar el modelo de una forma eficiente. link dataset:

```
 # primero vamos a conectar con el google drive
from google.colab import drive

# montamos el driver para el
accesodrive.mount('/gdrive')
```

```
Mounted at /gdrive
```

Lectura de los Datos

NEU(neutro), P(positivo), P+(muy positivo).

```
# mostrar la lista de archivos .xml
# esta lista esta dentro de la carpeta data_pistas# en el
google drive

!ls '/gdrive/My Drive/DATASETS/data_pistas'
```

```
data.csv                                    socialtv-tweets-train-tagged.csv
dataTwitter.csv                             socialtv-tweets-train-tagged.xml
dataTwitter.gsheet                          stompol-tweets-test.xml
general-tweets-test1k.xml stompol-tweets-train-tagged.csv general-tweets-
train-tagged.xml       stompol-tweets-train-tagged.xml       grid_search.pkl
                                            TASS2019_country_PE_
dev.xml
parques.csv                                 todo.csv
pistas.csv                                  tweets.txt
results                                     tw_faces4tassTrain1000rc.xml
socialtv-tweets-test.xml                    veredas.csv
```

Bueno, ya que tenemos la lista de archivos, acontinuacion vamos a ver el conenido de un archivo xml (por ejemplo "socialtv-tweets-train-tagged.xml")

```
# mostramos hasta la linea 30
!head -n 30 '/gdrive/My Drive/DATASETS/data pistas/socialtv-tweets-train-tagged.xml
```

```
<?xml version="1.0" encoding="UTF-8"?>
<tweets>
<tweet id="456544889786728451"><sentiment aspect="Afición" polarity="P">Los qu
<tweet id="456544890004852736">Diooooos que careron de <sentiment aspect="Juga
<tweet id="456544890231353345">Ganó el mejor. <sentiment aspect="Equipo-Real_M
<tweet id="456544890336206848"><sentiment aspect="Equipo-Real_Madrid" polarity
<tweet id="456544890533339136">@titelas Mañana <sentiment aspect="Jugador-Gare
<tweet id="456544890550099968">No digáis ahora que vaya robo, cuando el <senti
<tweet id="456544890654973952">Haha que risa los barcelonistas diciendo que as
<tweet id="456544890713673731">Enorme mi <sentiment aspect="Equipo-Real_Madrid
<tweet id="456544890877276160">Grande <sentiment aspect="Jugador" polarity="P"
<tweet id="456544891326062594">Vuestro odio es nuestra fuerza!!!!! <sentiment
<tweet id="456544891380568064">@El_Anto10: algo comico no le den la copa a <se
<tweet id="456544891460272128">"<sentiment aspect="Equipo-Real_Madrid" polarit
<tweet id="456544893645508608">Estadística oficial: <sentiment aspect="Jugador
<tweet id="456544893855211521">Menos mal que esta <sentiment aspect="Jugador-I
<tweet id="456544894203355136">Volver a pitar el himno, volver a tirarnos el b
<tweet id="456544894211723264">Y viva los tíos como <sentiment aspect="Jugador
<tweet id="456544894266269697">Nomas les pido <sentiment aspect="Equipo-Real_M
<tweet id="456544894345961474">El <sentiment aspect="Equipo-Real_Madrid" polar
<tweet id="456544894354333697">Ahora coge <sentiment aspect="Jugador-Sergio_Ra
<tweet id="456544894434045952"><sentiment aspect="Jugador-Gareth_Bale" polarit
<tweet id="456544894492753920">"En los ultimos dos clasicos tampoco jugo que p
<tweet id="456544894501146625">Para mí, <sentiment aspect="Jugador-Isco" polar
<tweet id="456544894517923840">@monterocnn @panqueka22 150 millones vs 100 Ney
<tweet id="456544894572445696">¿Sabéis qué? Que nosotros no jugábamos con <sen
<tweet id="456544894681493505">Adentro la tienen los cules bien adentro <senti
<tweet id="456544894706671616">@Borre7_: El palo de <sentiment aspect="Jugador
<tweet id="456544894715043841">Querido <sentiment aspect="Entrenador" polarity
<tweet id="456544894736023552">Para los que decís que el <sentiment aspect="Eq
```

Tipos de Archivos

```
# pandas para lectura en
framesimport pandas as
pd

#     max colum size 1000
pd.set_option('max_colwidth'
,1000)
```

```
# importamos libreria para lectura
xmlfrom lxml import objectify

# lee el archivo xml
xml = objectify.parse(open('/gdrive/My Drive/DATASETS/data_pistas/general-tweets-tr

# extraemos desde la
raiz xmlroot =
xml.getroot()
# definimos las etiquetas que queremos obtener
general_tweets_train = pd.DataFrame(columns=( content ,   polarity ,   agreement ))
# accedemos a todos los hijos                    de raiztweets = root.getchildren()
print('cantidad tweeets: ',len(tweets))   #mostrar cantidad#
recorremos todos los tweets uno por uno
for i in range(0,len(tweets)):
        tweet = tweets[i] # i-esimo tweet
        # organizamos en un dict (clave: valor) ejemplo: ["content": Salgo de VeoTV, row =
        dict(zip(['content', 'polarity', 'agreement'], [tweet.content.text, tweetrow_s = pd.Series(row) #
        cambiamos a vertical (como se muestra abajo) row_s.name = i # recorremos cada linea para
        guardar
        general_tweets_train = general_tweets_train.append(row_s)

# mostrar ejemplo
# head solo lista los 5 primeros
itemsgeneral_tweets_train.head()
```

cantidad tweeets: 7219

	Content	Polarity
Salgo de #VeoTV , que día más largoooooo...		NONE
0	@PauladeLasHeras No te libraras de ayudar me/nos. Besos y gracias	NEU DISAGREEMENT
1	@marodriguezb Gracias MAR	P AGREEENT
2	cuando se van sus corruptos. Intento no sacar conclusiones	N+ AGREEENT

Datos para el test

cargamos otro archivo con el mismo código

```
# IMPORTAMOS TASS DE FACE
xml = objectify.parse(open('/gdrive/My Drive/DATASETS/data_pistas/tw_faces4tassTrai#sample
tweet object
root = xml.getroot()
general_tweets_corpus_test = pd.DataFrame(columns=('content', 'polarity', 'agreementtweets =
```

```
root.getchildren()
print('cantidad tweeets: ',len(tweets))
for i in range(0,len(tweets)):
        tweet = tweets[i]
        # hacer un diccionario
        row = dict(zip(['content', 'polarity'], [tweet.content.text, tweet.sentiment.porow_s =
        pd.Series(row) # convierte del orizontal al vertical
        row_s.name = i
        general_tweets_corpus_test = general_tweets_train.append(row_s)

# ahora mostaramos los 5 primeros tweets para chekar si funciona
general_tweets_corpus_test.head()
```

```
cantidad tweeets:               1008
```

El segundo tipo de archivo que se carga es el siguiente, note que en realidad lo que cambia es en la creacion del dic y el acceso a los *tweets* pues en el segundo formato se tiene un lista del tip

```
# This is formatted as code
<tweet id="456544894706671616">@Borre7_: El palo de <sentiment aspect="Jugador-Neymar_Jr."
```

```
# definimos la carpteta donde se encuentra el archivo
data_dir = '/gdrive/My Drive/DATASETS/data_pistas/'
# creamos el objeto xml
xml = objectify.parse(open(data_dir+'socialtv-tweets-train-tagged.xml'))#creamos la
raiz
root =
xml.getr
oot()#
lectura
socialtv_tweets_train = pd.DataFrame(columns=('content', 'polarity'))tweets =
root.getchildren()
print('cantidad tweeets: ',len(tweets))
for i in range(0,len(tweets)):
        tweet = tweets[i]
        row = dict(zip(['content', 'polarity', 'agreement'], [' '.join(list(tweet.itertrow_s = pd.Series(row)
        row_s.name = i
        socialtv_tweets_train = socialtv_tweets_train.append(row_s)
socialtv_tweets_train.to_csv(data_dir+'socialtv-tweets-train-tagged.csv', index=Fal#mostrar los 5
ultimos
socialtv_tweets_train.tail()
```

```
cantidad tweets:  1773
```

Lectura de archivo tema politico xml

	content	polarity
1768	Hahahaha la cara de messi csm jajajajjaja	N

```
# lectura para el archivo content politico

xml = objectify.parse(open(data_dir+'stompol-tweets-train-tagged.xml'))#lectura
similar a los anteriores
root = xml.getroot()
stompol_tweets_corpus_train = pd.DataFrame(columns=('content', 'polarity'))tweets =
root.getchildren()
print('cantidad tweeets: ',len(tweets))
for i in range(0,len(tweets)):
    tweet = tweets[i]
    row = dict(zip(['content', 'polarity', 'agreement'], [' '.join(list(tweet.itertrow_s = pd.Series(row)
    row_s.name = i
    stompol_tweets_corpus_train = stompol_tweets_corpus_train.append(row_s)
stompol_tweets_corpus_train.to_csv(data_dir+'stompol-tweets-train-tagged.csv', inde# mostramos los
ultimos 5
stompol_tweets_corpus_train.tail()
```

```
cantidad tweeets:  784
```

@cs_hor @AntSoubrieCs @Albert_Rivera @Cs_Alcobendas_ Pues muchos NOhablaron
El PP ganaría la Comunidad de Madrid y el Ayuntamiento, pero necesitaría a@CiudadanosCs
¿Quién es Pedro Sánchez ? Mi artículo de los viernes en @lavozdegalicia
Hoy mucha propaganda del IBEX-35 en las encuestas.

Corpus total

en esta parte pasamos a unir los archivos xml, cargados individualmente: general,_train,general_test, face, socialtv, politicos.

```
# unimos los datos cargados
tweets_corpus = pd.concat([ general_tweets_train, socialtv_tweets_train,
general

# mostramos el tamaño total
print('size: ', tweets_corpus.shape)
```

```
size:  (16996, 3)
```

Como podemos notar en los archivos anteriores cargados previamente existe una columan convalores NaN vacios o nulos (en general_tweets_train), entonces procedemos a eliminar todos aesto que estan considerados como comentarios neutros, estos ademas aparecen junto al campo agreement = DISAGREEMENT por lo que dejamos de considerar estos, ademas podemosaprovechar en eliminar las URLS o todos aquellos que contengan "http. "

```
# binarizacion de los elementos
tweets_corpus = tweets_corpus.query('agreement != "DISAGREEMENT" and polarity != "N

#Eliminamos las urls
tweets_corpus = tweets_corpus[-tweets_corpus.content.str.contains('^http.*$')]

#mostramos lo que queda despues de
estotweets_corpus.shape

    (12546, 3)
```

Tokenización

Bueno listo ahora ya tenemos listo nuestro corpus de 12546 *tweets* y listos para ser entreandos y ahora toca el pre-procesamiento y esto empieza con la tokenizacion pero debemos descargar la librería nltk, definimos la lista de *stopwords* (palabras no deseadas), los signos de puntuación

```
# descargamos stopwords en
español import nltk
nltk.download("stopwords")

from nltk.corpus import stopwords
spanish_stopwords =
stopwords.words('spanish')# print
            stopwords en español
print('Stop words: ',spanish_stopwords)

# lista de puntuacion
from string import
punctuationnon_words
= list(punctuation)

# agregamos puntuacion en español
non_words.extend(['¿', '¡'])
non_words.extend(map(str,range(10)))
```

```
print('puntuaciones :',non_words)
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
Stop words:  ['de', 'la', 'que', 'el', 'en', 'y', 'a', 'los', 'del', 'se', 'la
puntuaciones : ['!', '"', '#', '$', '%', '&', "'", '(', ')', '*', '+', ',', '-
```

Steaming

Primero vamos a hacer el *steaming* que basicamente consiste en convertir las palabras a la forma básica (raíz o lemma) así que *steaming* es un proceso importante en el preprocesamiento.

```
# importamos todas las librerias de lematización y radicación

from sklearn.feature_extraction.text import CountVectorizer #comvesor a numerofrom
nltk.stem import SnowballStemmer #lematización
from nltk.tokenize import word_tokenize

# basado en http://www.cs.duke.edu/courses/spring14/compsci290/assignments/lab02.htstemmer =
SnowballStemmer('spanish')

# funcion que retorna los tokens en forma de lemas
def stem_tokens(tokens, stemmer):
    stemmed = []
    for item in tokens:
        stemmed.append(stemmer.ste
        m(item))
    return stemmed

# funcion general que
tokeniza y# llama a la
lematizacion
def tokenize(text):
    # eliminar espacios en blanco
    text = ''.join([c for c in text if c not in non_words])#
    tokenizar
    tokens =    word_tokenize(text)

    #
    lematizaci
    on(steam)
    try:
        stems = stem_tokens(tokens,
    stemmer)except Exception as e:
        ]
```

```
    return stems

    # [Haha, cara, nombre, cs ,jaja] [1]
```

Binarización

En esta parte lo que primero vamos a realizar es el filtro para no tomar en cuenta a los *tweets* etiquetados como NEU (neutros), y desestimarlos, así como finalmente convertiremos los valores de las etiquetas (Polarity) en "1 , 0", con N-,N: 0, y p,p+ : 1

```
# Vamos a tomar unicamente los tweets que son diferentes a NEU
tweets_corpus = tweets_corpus[tweets_corpus.polarity != 'NEU']

# damos el nuevo nombre de la
columna
tweets_corpus['polarity_bin'] = 0
# todos los P, P+ son = 1 tweets_corpus.polarity_bin[tweets_corpus.polarity.isin(['P', 'P+'])] = 1

# contamos la cantidad de valores
tweets_corpus.polarity_bin.value_counts(normalize=True)
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:7: SettingWithCopA value is
trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stimport sys
1       0.560018
0       0.439982
Name: polarity_bin, dtype: float64
```

Se muestra el resultado final de polarización

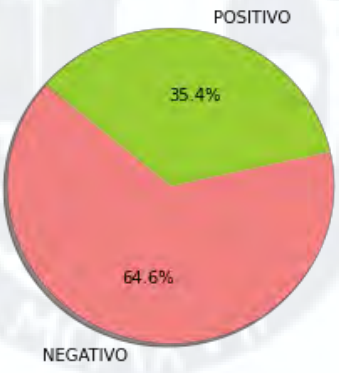
```
tweets_corpus.tail()
```

Diagrama de DataSet

El diagrama pastel muestra la cantidad que se tiene de comentarios postivos y negativos

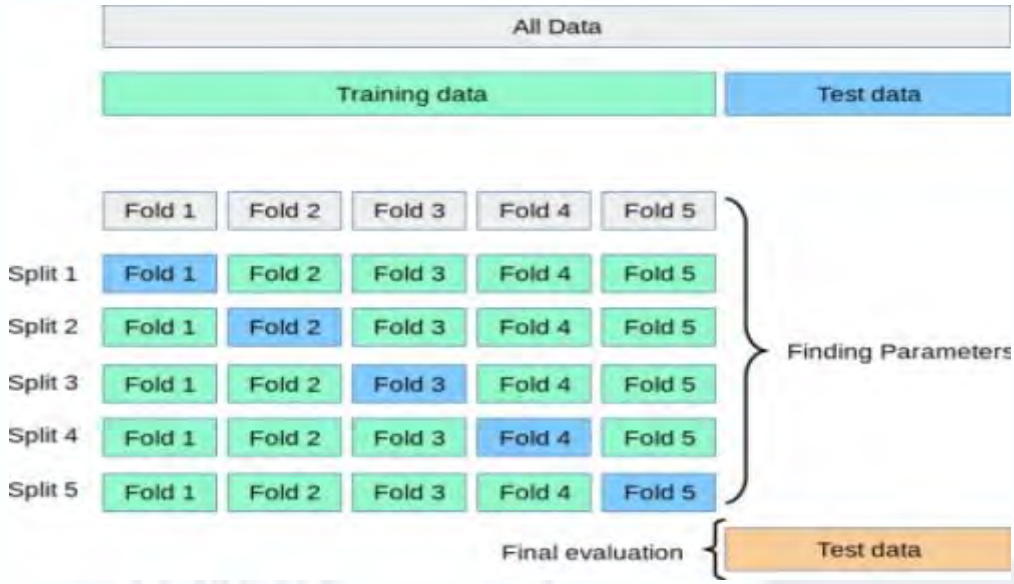
```
import matplotlib as plt
import matplotlib.pyplot as plt

sum_values = tweets_corpus.polarity_bin.value_counts()
sum_values
```

```
# mostrar valores en diagrama de
pastelserie = pd.Series([4973,2727])
serie.value_counts()
#serie

fig1, ax1 =
plt.subplots()
ax1.pie(serie ,
            labels=["NEGATIVO","POSITIVO"], colors=['lightcoral', 'yellowgreen'],
            shadow=True, startangle=140, autopct='%1.1f%%')

ax1.
axis
('eq
ual')
plt.s
how
()
```



Entrenamiento

Aquí primero se define el modelo SVM, la organización de ejecución (*pipeline*). Luego se define los parámetros que posiblemente tengan un mejor ajuste, que encontraremos con el método de validación cruzada el que encontrará el mejor parámetro que se muestra en el *score- cross-validation* más adelante,entonces estos valores son los que se usarán finalmente para el entrenamiento.



Hyperparametros count Vectorizer y SVM

Para tener un modelo que funcione eficientemente, un paso importante es la búsqueda de los parametros ideales para obtener una mejor efectividad, para esto usamos el *pipeline*, siguiendo el *link*, pero para esto nosotros especificamos los parámetros en *parameters* que son en realidad los hyper-parámetros para el countVectorizer: vect y linearSVCcls. que finalmente se efectuará con el método Cross-Validation

```
from sklearn.model_selection import train_test_split
from sklearn.svm import LinearSVC # SVM
from sklearn.pipeline import Pipeline #

from sklearn.model_selection import GridSearchCV



vectorizer = CountVectorizer(
                            analyzer =
                            'word',
                            tokenizer
                            =
                            tokenize,
                            lowercase
                            = True,
                            stop_words = spanish_stopwords)

pipeline = Pipeline([
        ('vect', vectorizer),#
```

```
        countVectorizer('cls',
        LinearSVC()), #SVM
])

# de acuerdo a https://scikit-learn.org/stable/auto_examples/model_selection/grid_s# configuracion de
los hypermarametros
# vect:
countVectori
zer# cls:
LinearSVC
parameters =
{
       'vect__max_df': (0.5, 1.9), # significa "ignorar los términos que aparecen en m'vect__min_df': (10,
  20,50), # significa "ignorar los términos que aparecen en 'vect__max_features': (500, 1000), # maximo
                                                                    de vocabulario necesarios para inic
        'vect__ngram_range': ((1, 1), (1, 2)),    # unigrams or bigrams
        'cls__C': (0.2,0.5,0.7),# significa "parametro de regularizacion,valor positivo'cls__loss': ('hinge',
        'squared_hinge'), # metricas de error para el ajuste de 'cls__max_iter': (500, 1000) # numero
        maximo de iteraciones para ajuste la line
}

#
["comentario","bueno","proyecto","final"]
#[[100011]]
#SVM

# instanciacion del metodo
grid_search = GridSearchCV(pipeline, parameters, n_jobs=-1 , scoring='roc_auc')

#efectuar cross-validation
grid_search.fit(tweets_corpus.content, tweets_corpus.polarity_bin)

        /usr/local/lib/python3.7/dist-packages/sklearn/feature_extraction/text.py:385:
            'stop_words.' %
        sorted(inconsistent))
        GridSearchCV(cv=None,
        error_score=nan,
                             estimator=Pipeline(memory=None,
                                                steps=[('vect',
                                                        CountVectorizer(analyzer='word',
                                                                        binary=Fal
                                                                        se,
                                                                        decode_err
                                                                        or='strict',
                                                                        dtype=<cla
                                                                        ss
                                                                        'numpy.in
                                                                        encoding='
                                                                        utf-8',
                                                                        input='cont
```

```
                                                                    ent',
                                                                    lowercase=
                                                                    True,
                                                                    max_df=1.
                                                                    0,
                                                                    max_featur
                                                                    es=None,
                                                                    min_df=1,
                                                                    ngram_ran
                                                                    ge=(1, 1),
                                                                    preprocesso
                                                                    r=None,
                                                                    stop_words
                                                                    =['de', 'la'
                                                          verbose=0))],
                                                v
                                erbose=False),
                                iid='deprecated', n_jobs=-1,
                                param_grid={'cls__C': (0.2, 0.5, 0.7),
                                            'cls__loss': ('hinge', 'squared_hinge'),
                                            'cls__max_iter': (500, 1000),
                                            'vect__max_df': (0.5, 1.9),
                                            'vect__max_features': (500, 1000),
                                            'vect__min_df': (10, 20, 50),
                                            'vect__ngram_range': ((1, 1), (1, 2))},
                                pre_dispatch='2*n_jobs', refit=True, return_train_score=False,
                                scoring='roc_auc', verbose=0)
```

Mostramos los mejores valores (hiperparámetros) encontrados con los que se obtiene un mejor *score*, que usaremos para el entrenamiento

```
# mostrar los mejores
parametros
grid_search.best_params_
```

```
{'cls__C': 0.2,
 'cls__loss':
 'squared_hinge','cls
 max_iter': 500,
 'vect__max_df': 0.5,
 'vect__max_features': 1000,
 'vect__min_df': 10,
 'vect__ngram_range': (1, 1)}
```

Guardamos los hiperparamentros

```
#guardar el modelo
from sklearn.externals import joblib
```

```
joblib.dump(grid_search, '/gdrive/My Drive/DATASETS/data_pistas/grid_search.pkl')
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/externals/joblib/__init__.py:15
  warnings.warn(msg, category=FutureWarning)
['/gdrive/My Drive/DATASETS/data_pistas/grid_search.pkl']
```

Count Vectorizer

Para poder ser capaz de administrar los *tweets*, primero necesitamos extraer la información del texto, esto se logra conseguir convirtiendo cada *token* en una matriz de unos y ceros onehotEncoding , por ejemplo si tenemos un *tweet* como: "Aprendizaje automático es muy genial" el módulo CountVectorizer va retornar:

	Tweet	Aprendizaje	Automático	Es	Muy	genial
1	Aprendizaje automático es muy genial	1	1	1	1	1
2	Aprendizaje automático es muy genial	1	1	1	0	1

De esta forma ahora vamos a trabajar con la representación numérica de los *tweets.*

```
# definimos el modelo de acuerdo a los hiper-paremtros encontrados
model = LinearSVC(C=.2, loss='squared_hinge',max_iter=1000,multi_class='ovr',
                  random_state=None,
)

# definimos el countVectorizer de acuerdo a los hiper-parametros
vectorizer = CountVectorizer(
    analyzer =
    'word',
    tokenizer
    =
    tokenize,
    lowercase
    = True,
    stop_words =
    spanish_stopwords,
    min_df = 50,
    max_df = 1.9,
    ngram_
    range=
    (1, 1),
    max_fe
    atures=
    1000
)
```

```
# realizamos la representacion de texto y el count vectorizer corpus_data_features =
vectorizer.fit_transform(tweets_corpus.content)#convertir los features (matriz
numerica), en un arreglo corpus_data_features_nd = corpus_data_features.toarray()

    /usr/local/lib/python3.7/dist-packages/sklearn/feature_extraction/text.py:385:'stop_words.' %
      sorted(inconsistent))
```

ROC_AUC

El área bajo la curva AUC_ROC es un métrica usada con el que vamos a medir el rendimiento del modelo

```
from sklearn.model_selection import cross_val_score

scores =
    cross_val
    _score(
    model,
    corpus_data_features_nd[0:len(tweets_corpus)], #[[10001000101]]
    y=tweets_corpus.polarity_bin
    , #[0]scoring='roc_auc',
    #scoring=make_scorer(F1-score, average='weighted', labels=[2]),cv=5
    )

scores.mean()

     0.811568396299801
```

Training

Esta parte ejecuta el entrenamiento, finalmente teniendo la configuracion necesaria y teniendo en cuenta el último *score* representa un buen rendimiento para los valores pre determinados con un *score* de %78 podemos prodecer ahora sí a entrenar directamente nuestro modelo.

```
# definimos directamente los parametros necesario con los
# cuales encotraos un mejor rendimiento para entrenarlo directamete

pipeline = Pipeline([
    ('vect',
    CountVectorize
    r(
            analyzer =
            'word',
            tokenizer
            =
            tokenize,
            lowercase
```

```
                    = True,
                    stop_words =
                    spanish_stopwords,
                    min_df = 50,
                    max_df = 1.9,
                    ngram_
                    range=
                    (1, 1),
                    max_fe
                    atures=
                    1000
                    )),
        ('cls', LinearSVC(C=.2, loss='squared_hinge',max_iter=1000,multi_class='ovr',
                          random_state=None,
])

#     para el training vamos a enternar y predecir
pipeline.fit(tweets_corpus.content, tweets_corpus.polarity_bin)


        /usr/local/lib/python3.7/dist-packages/sklearn/feature_extraction/text.py:385:'stop_words.' %
            sorted(inconsistent))
        Pipeline(memory=None,
                 steps=[('vect',
                         CountVectorizer(analyzer='word', binary=False,
                                         decode_error='strict',
                                         dtype=<class 'numpy.int64'>, encoding='utf-8'

('cls',input='content', lowercase=True, max_df=1.9,max_features=1000, min_df=50, ngram_range=(1, 1),
                                                                    preprocessor=None,
stop_words=['de', 'la', 'que', 'el', 'en', 'y
'a', 'los', 'del', 'se', 'las',
'por', 'un', 'para', 'con', 'no'.
                 'o', 'este', 'sí', 'porque', ...]strip_accents=None, token_pattern='(?u)\\b\\w\\w+\\b',
tokenizer=<function tokenize at 0x7fb3dcf0eb9vocabulary=None)),
                         LinearSVC(C=0.2, class_weight=None, dual=True,
                                   fit_intercept=True, intercept_scaling=1,
                                   loss='squared_hinge', max_iter=1000,
                                   multi_class='ovr', penalty='l2', random_state=None,
                                   tol=0.0001, verbose=0))],
                 verbose=False)
```

TEST

Para el *test* primero tenemos que tomar en cuenta que nuestras entradas serán *tweets* (publicaciones) reales tomadas de pistas y veredas, para esto necesitamos cargar el conjunto de *tweets* que está almacenado en un archivo csv.

TEST _PERU

pre-procesamos extraemos predecir matriz de confusión

```
# pandas para lectura en
framesimport pandas as
pd

#     max colum size 1000
pd.set_option('max_colwidth'
,1000)

# importamos libreria para lectura
xmlfrom lxml import objectify

# lee el archivo xml
xml = objectify.parse(open('/gdrive/My Drive/DATASETS/data_pistas/TASS2019_country_

# extraemos desde la
raiz xmlroot =
xml.getroot()
# definimos las etiquetas que queremos obtener
test_peru = pd.DataFrame(columns=('content', 'polarity', 'agreement'))# accedemos
a todos los hijos                                   de raiz
tweets = root.getchildren()
print('cantidad tweeets: ',len(tweets))    #mostrar cantidad#
recorremos todos los tweets uno por uno
for i in range(0,len(tweets)):
        tweet = tweets[i] # i-esimo tweet
        # organizamos en un dict (clave: valor) ejemplo: ["content": Salgo de VeoTV, row =
        dict(zip(['content', 'polarity'], [tweet.content.text, tweet.sentiment.porow s = pd.Series(row) #
        cambiamos a vertical (como se muestra abajo)

        row_s.name = i # recorremos cada linea para guardar
        test_peru = test_peru.append(row_s)

# mostrar ejemplo
# head solo lista los 5 primeros
itemstest_peru.head()
```

```
        cantidad tweeets:               498
```

	content	*polarity*	*agreement*
0	Así te paguen bien... Si es a última hora... No se podrá... Y hoy metocó servir, no es con pago económico, pero el pago me lo da el Rey		
1	Manolo: se llama H&M por Hombre y Mujer. Yo: ..pero.es una marcaamericana, no tendría sentido. Manolo: callate butch. Yo:		
2	Buen resumen de mi vida amorosa. - Lo he amado por años. - Cásatecon él, pé. - AJ, NO. No estoy para mantener a nadie. Que no joda.		

@BartanSoo12 Pero tú ya sabes por qué le digo así a Yixing ya sé que

```
# binarizacion de los elementos
test_peru = test_peru.query('polarity != "NONE"')

#Eliminamos las urls
test_peru = test_peru[-test_peru.content.str.contains('^http.*$')]

# Vamos a tomar unicamente los tweets que son diferentes a NEU# Vamos a tomar unicamente los tweets que son diferentes a NEU
test_peru = test_peru[test_peru.polarity != 'NEU']


# damos el nuevo nombre de la columnatest_peru['polarity_bin'] = 0
# todos los P, P+ son = 1 test_peru.polarity_bin[test_peru.polarity.isin(['P', 'P+'])] = 1test_peru.head(15)
```

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:15: SettingWithCoA value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stfrom ipykernel import kernelapp as app

	content	polarity	agreement	polarity_bin
0	Así te paguen bien... Si es a última hora... No se podrá... Y hoy me tocó servir, no es con pago económico, pero el pago me lo da el Rey	N	NaN	0
2	Buen resumen de mi vida amorosa. - Lo he amado por años. - Cásate con él, pé. - AJ, NO. No estoy para mantener a nadie. Que no joda.	N	NaN	0

```
test_peru.shape
```

(212, 4)

Ósea les juro por mi vida que estoy en el lugar más

```
tweets_test = test_peru['content']

#extraer unicamente los tweets
test_data_text = test_peru.iloc[:,0] # text

test_data_labels = test_peru.iloc[:,3] # labels
test_data_text.head()
#tweets_test_p = pd.DataFrame(tweets_test, columns = ['tweet','predict'])
#tweets_test_p.head()
```

```
0        Así te paguen bien... Si es a última hora... No se podrá... Y hoy me tocó
2              Buen resumen de mi vida amorosa. - Lo he amado por años. - Cásate con
5            Dicen q lo bueno dura poco Aveces pienso q es así y me pone triste,per
7         Ósea les juro por mi vida que estoy en el lugar más caro del mundo en el
9                        Y comenzó el día mas amado y esperado ja ja! "Lunes" a comenzar
Name: content, dtype: object
```

Como solo es necesario los comentario para la prediccion del modelo entonces solo extraemos esta parte

```
# mostrar los 5
primeros
test_data_label
s.head()
```

```
0    0
2    0
5    0
7    0
9    0
Name: polarity_bin, dtype: int64
```

Prediccion de tweets

Esta parte predice los comentarios (*tweets*) dando una lista de valores de acuerdo al ordenestablecido.

```
#
predict = pipeline.predict(test_data_text)
```

Mostramos la prediccion de las etiquetas, manteniendo el orden

```
predict # la ultima parte no predice correctamente
```

```
array([0, 1, 1, 0, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 0, 1, 0, 0, 0, 1,
       1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0,
       1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 1,
       1, 1, 0, 0, 1, 0, 1, 1, 1, 0, 1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 0, 1,
       0, 1, 0, 0, 0, 1, 1, 0, 1, 0, 1, 0, 0, 0, 1, 1, 1, 0, 1, 0, 1, 0,
       1, 0, 0, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 1,
       0, 0, 1, 0, 1, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 1, 1,
       1, 0, 1, 0, 0, 1, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0,
       1, 0, 1, 1, 0, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0,
       0, 0, 1, 0, 0, 1, 0, 1, 1, 1, 0, 1, 1, 0])
```

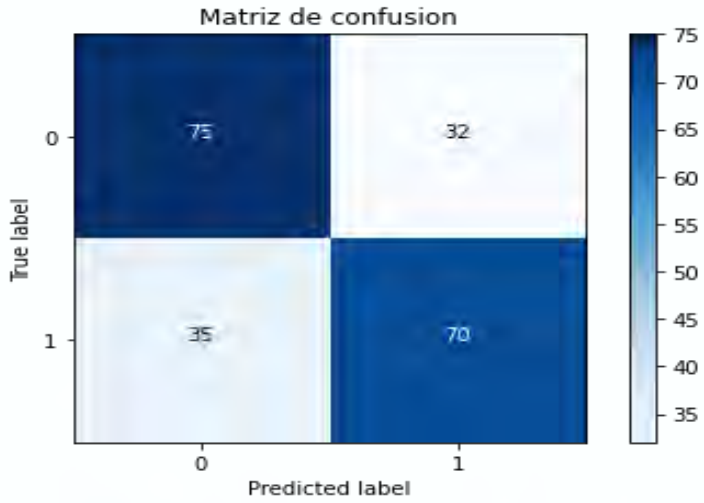
Métricas de Evaluación Matriz de Confusión

```
from sklearn import metrics

conf = metrics.confusion_matrix(test_data_labels,predict)f1 =
metrics.F1-score(test_data_labels,predict)

)
```

```
0.6763285024154588
[[75 32]
  [35 70]]
```

```
impor
t
nump
y as
np
impor
t
iterto
ols

classes = [0, 1]
# plot confusion matrix
plt.imshow(conf, interpolation='nearest', cmap=plt.cm.Blues)
plt.title("Matriz de confusion")
plt.colorbar()
tick_marks =
np.arange(len(classes))
plt.xticks(tick_marks, classes)
plt.yticks(tick_marks, classes)

fmt = 'd'
thresh = conf.max() / 2.
for i, j in itertools.product(range(conf.shape[0]), range(conf.shape[1])):plt.text(j, i,
    format(conf[i, j], fmt),
                horizontalalignment="center",
                color="white" if conf[i, j] > thresh else "black")
plt.tight_layout()
plt.ylabel('True label')
plt.xlabel('Predicted
label')
```

```
Text(0.5, 15.0, 'Predicted label')
```



TEST DATASET (Todo)

1. Cargamos los datos(todo: pistas, parques, veredas)

```
# cargar la dataset csv
tweets_test_parques = pd.read_csv('/gdrive/My Drive/DATASETS/data_pistas/todo.csv',#mostrar test
tweets_test_parques.head(10)
```

	Obra	*Marca temporal*	Indique su nombre	*Distrito*	*Tweets*	*Etiqueta*
0	Parques	7/07/2021 23:17	Fernando	Magdalena del Mar	Parque costado de la iglesia de la cúpula por la avenida sucre está descuidado y muy sucio	0
1	Parques	7/07/2021 23:23	Gerson Chumbimuni	Santiagode Surco	Hay muchos y están bien mantenidos	1
2	Parques	7/07/2021 23:26	Ana	San Borja	Bonitos	1
3	Parques	7/07/2021 23:27	Polo Santans	La Molina	Estan cuidados y con seguridad	1
4	Parques	7/07/2021 23:28	Ricardo	Lima	Requieren más cuidado	0
5	Parques	7/07/2021 23:30	PAblo	San Miguel	Regulares en presentación, son Limpios	1

```
# Vamos a tomar unicamente los tweets que son diferentes a NEU
tweets_test_parques = tweets_test_parques[tweets_test_parques.Etiqueta != 'Neutro']tweets test
parques = tweets test parques[tweets test parques.Etiqueta != 'neutro']# extraemos contenido texto
test_todo_text = tweets_test_parques.iloc[:,4]#
extraemos labels
test_todo_labels = tweets_test_parques.iloc[:,5]
#test_todo_text = tweets_test_parques['Tweets']
#test_todo_labels = tweets_test_parques['Etiqueta']
test_todo_text.head()
```

```
0    Parque costado de la iglesia de la cúpula por la avenida sucre está descu
1    Hay muchos y está
2
3
4
Name: Tweets, dtype: object
```

```
print(len(test_todo_la
bels))#convertir an
tipo int
test_todo_labels= test_todo_labels.astype(int)
test_todo_labels.head()
```

```
268
0    0
1    1
2    1
3    1
4    0
Name: Etiqueta, dtype: int64
```

```
predict = pipeline.predict(test_todo_text)
print(len(predict))
```

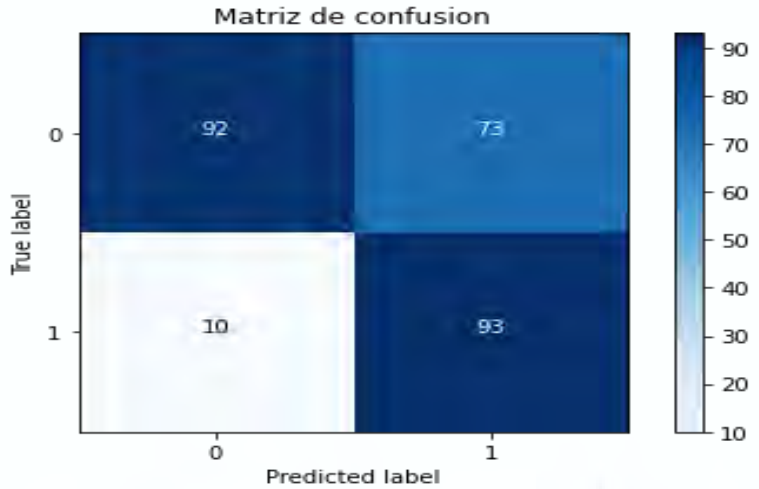
```
268
```

```
array([1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 0, 0, 0,
       0, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 0, 1, 1, 1,
       0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0,
       1, 0, 1, 1, 1, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1,
       0, 0, 1, 0, 0, 1, 0, 0, 1, 1, 0, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 1,
       1, 0, 1, 0, 0, 1, 1, 0, 1, 1, 1, 0, 0, 1, 0, 1, 1, 1, 1, 1, 1, 0,
       0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 1,
       0, 1, 0, 0, 0, 0, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 0,
       1, 1, 1, 0, 1, 0, 1, 0, 0, 1, 1, 0, 1, 0, 0, 1, 0, 0, 1, 1, 1, 0,
       0, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 0, 0, 1, 1, 0, 0, 0,
       0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 0, 0, 1, 0, 1, 1, 0, 0, 1,
       0, 1, 0, 0, 1, 1, 0, 0, 1, 0, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0,
       0, 1, 0, 1])
"""
```

```
conf = metrics.confusion_matrix(test_todo_labels,predict)f1 =
metrics.F1-score(test_todo_labels,predict)
print(f1) print(conf)
```

```
268
0.6914498141263942
[[92 73]
  [10 93]]
```

```
impor
t
nump
y as
np
impor
t
iterto
ols

classes = [0, 1]
# plot confusion matrix
plt.imshow(conf, interpolation='nearest', cmap=plt.cm.Blues)
plt.title("Matriz de confusion")
plt.colorbar()
tick_marks =
np.arange(len(classes))
plt.xticks(tick_marks, classes)
plt.yticks(tick_marks, classes)

fmt = 'd'
thresh = conf.max() / 2.
for i, j in itertools.product(range(conf.shape[0]), range(conf.shape[1])):plt.text(j, i,
    format(conf[i, j], fmt),
                horizontalalignment="center",
                color="white" if conf[i, j] > thresh else "black")

plt.tight_layout()
plt.ylabel('True label')
plt.xlabel('Predicted
label')
plt.savefig('/gdrive/My Drive/DATASETS/data_pistas/results/confusion_todo.png')
```



Guardar datos prediccion en archivo csv

```
# guardar dentro de un
dataframe
tweets_test_parques.shape
#out : (268, 6)

#agregamos una columna a nuestro archivo la prediccion
tweets_test_parques['predict'] =predict

# guardar el archivo
tweets_test_parques.to_csv('/gdrive/My Drive/DATASETS/data_pistas/results/result_to

# mostramos lo
agregado
tweets_test_parqu
es.head()
```

	Obra	*Marca temporal*	**Indique su nombre**	*Distrito*	*Tweets*	*Etiqueta*	*predict*
0	Parques	7/07/2021 23:17	Fernando	Magdalena del Mar	Parque costado de la iglesia de la cúpula por la avenida sucre está descuidado y muy sucio	0	1
1	Parques	7/07/2021 23:23	Gerson Chumbimuni	Santiagode Surco	Hay muchos y están bien mantenidos	1	1

TEST DATASET (Pistas)

1. Cargamos los comentarios tweets relacionados a pistas

```
# cargar la dataset csv
tweets_test_pistas = pd.read_csv('/gdrive/My Drive/DATASETS/data_pistas/pistas_2.cs#mostrar test
tweets_test_pistas.head()
```

	Obra	*Marca temporal*	**Indique su** *nombre*	**Distrito**	**Tweets**	**Etiqueta**
0	Pistas	7/07/2021 23:17	Fernando	Magdalena del Mar	Pista de la av. libertad estan rotas y el alcalde no las arregla	0
1	Pistas	7/07/2021 23:23	Gerson Chumbimuni	Santiagode Surco	Están en buenas condiciones	1

```
# Vamos a tomar unicamente los tweets que son diferentes a NEU tweets_test_pistas =
tweets_test_pistas[tweets_test_pistas.Etiqueta != 'Neutro']tweets_test_pistas =
tweets_test_pistas[tweets_test_pistas.Etiqueta != 'neutro']# extraemos contenido texto
test_pistas_text = tweets_test_pistas.iloc[:,4]#
extraemos labels
test_pistas_labels = tweets_test_pistas.iloc[:,5]
#convertir an tipo int
test_pistas_labels= test_pistas_labels.astype(int)

test pistas text.head()
```

```
0    Pista de la av. libertad estan rotas y el alcalde no las arregla
1                                                 Están en buenas condiciones
2                                                       Faltan reparar
3                                                Muy pocas estan en mal estado
6                                                      Pistas en mal estado
Name: Tweets, dtype: object
```

```
from sklearn

import metrics#

predicion para

pistas
predict = pipeline.predict(test_pistas_text)

# matriz de confusion para pistas
conf = metrics.confusion_matrix(test_pistas_labels,predict)f1 =
metrics.F1-score(test_pistas_labels,predict)

        0.6956521739130435
        [[25 17]
         [ 4 24]]


impor
t
nump
y as
np
impor
t
iterto
ols

# DIAGRAMA PARA PISTAS

classes = [0, 1]
# plot confusion matrix
plt.imshow(conf, interpolation='nearest', cmap=plt.cm.Blues)
plt.title("Matriz de confusion")
plt.colorbar()
tick_marks =
np.arange(len(classes))
plt.xticks(tick_marks, classes)
plt.yticks(tick_marks, classes)

fmt = 'd'
thresh = conf.max() / 2.
for i, j in itertools.product(range(conf.shape[0]), range(conf.shape[1])):plt.text(j, i,
        format(conf[i, j], fmt),
                        horizontalalignment="center",
                        color="white" if conf[i, j] > thresh else "black")

plt.tight_layout()
```

```
plt.ylabel('True label')
plt.xlabel('Predicted
label')
plt.savefig('/gdrive/My Drive/DATASETS/data_pistas/results/confusion_PISTAS.png')
```


```
#agregamos una columna a nuestro archivo la prediccion
tweets_test_pistas['predict'] =predict


# guardar el archivo
tweets_test_pistas.to_csv('/gdrive/My Drive/DATASETS/data_pistas/results/result_PIS

# mostramos lo
agregado
tweets_test_pistas.
head()
```

	Obra	Marca temporal	Indique su nombre	Distrito	Tweets	Etiqueta	predict
0	Pistas	7/07/2021 23:17	Fernando	Magdalena del Mar	Pista de la av. libertad estan rotas y el alcalde	0	0
1	Pista	7/07/2021 23:23	Gerson Chumbimuni	Santiago de Surco	Están en buenas condiciones	1	1
2	Pistas	7/07/2021 23:26	Ana	San Borja	Faltan reparar	0	0

TEST DATASET (Parques)

1. Cargamos los comentarios tweets relacionados a parques

```
# cargar la dataset csv
tweets_test_parque = pd.read_csv('/gdrive/My Drive/DATASETS/data_pistas/parques.csv#mostrar test
tweets_test_parque.head()
```

```
# Vamos a tomar unicamente los tweets que son diferentes a NEU tweets_test_parque =
tweets_test_parque[tweets_test_parque.Etiqueta != 'Neutro']tweets_test_parque =
tweets_test_parque[tweets_test_parque.Etiqueta != 'neutro']# extraemos contenido texto
test_parques_text = tweets_test_parque.iloc[:,4]#
extraemos labels
test_parques_labels = tweets_test_parque.iloc[:,5]
#convertir an tipo int
test_parques_labels= test_parques_labels.astype(int)

test_parques_text.head()
```

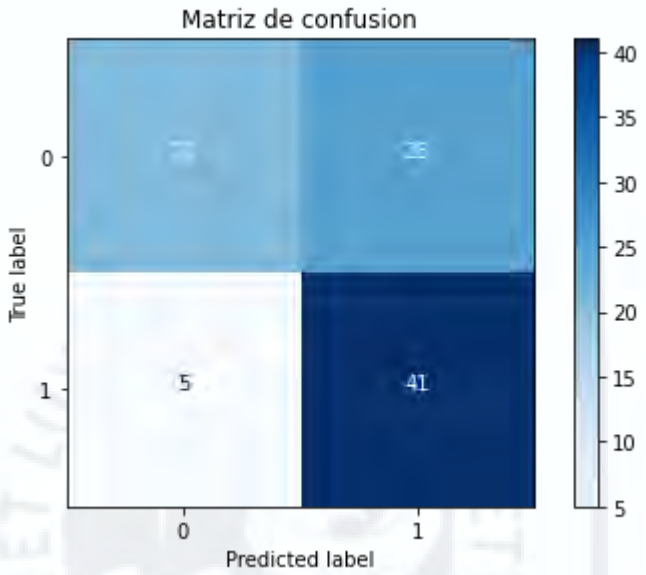
```
0    Parque costado de la iglesia de la cúpula por la avenida sucre está descu
1    Hay muchos y está
2
```

```
# predicion para pistas
predict = pipeline.predict(test_parques_text)

# matriz de confusion para pistas
conf = metrics.confusion_matrix(test_parques_labels,predict)f1 =
metrics.F1-score(test_parques_labels,predict)
P
```

```
0.7321428571428572
[[21 25]
 [ 5 41]]
```

```
# DIAGRAMA PARA PARQUES

classes = [0, 1]
# plot confusion matrix
plt.imshow(conf, interpolation='nearest', cmap=plt.cm.Blues)
plt.title("Matriz de confusion")
plt.colorbar()
tick_marks =
np.arange(len(classes))
plt.xticks(tick_marks, classes)
plt.yticks(tick_marks, classes)

fmt = 'd'
thresh = conf.max() / 2.
for i, j in itertools.product(range(conf.shape[0]), range(conf.shape[1])):plt.text(j, i,
    format(conf[i, j], fmt),
                horizontalalignment="center",
                color="white" if conf[i, j] > thresh else "black")
```

```
plt.tight_layou
t()
plt.ylabel('Tru
e label')
plt.xlabel('Predicted label')
plt.savefig('/gdrive/My Drive/DATASETS/data_pistas/results/confusion_PARQUES.png')
```




```
#agregamos una columna a nuestro archivo la prediccion
tweets_test_parque['predict'] =predict


# guardar el archivo
tweets_test_parque.to_csv('/gdrive/My Drive/DATASETS/data_pistas/results/result_PAR

# mostramos lo
agregado
tweets_test_par
que.head()
```

	Obra	*Marca temporal*	Indique su nombre	*Distrito*	*Tweets*	*Etiqueta*	*predict*
0	Parques	7/07/2021 23:17	Fernando	Magdalena del Mar	Parque costado de laiglesia de la cúpula por la avenida sucre está descuidado y muy sucio	0	1

TEST DATASET (Veredas)

1. Cargamos los comentarios tweets relacionados a veredas

```
# cargar la dataset csv
tweets_test_veredas = pd.read_csv('/gdrive/My Drive/DATASETS/data_pistas/veredas.cs#mostrar test
tweets_test_veredas.head()
```

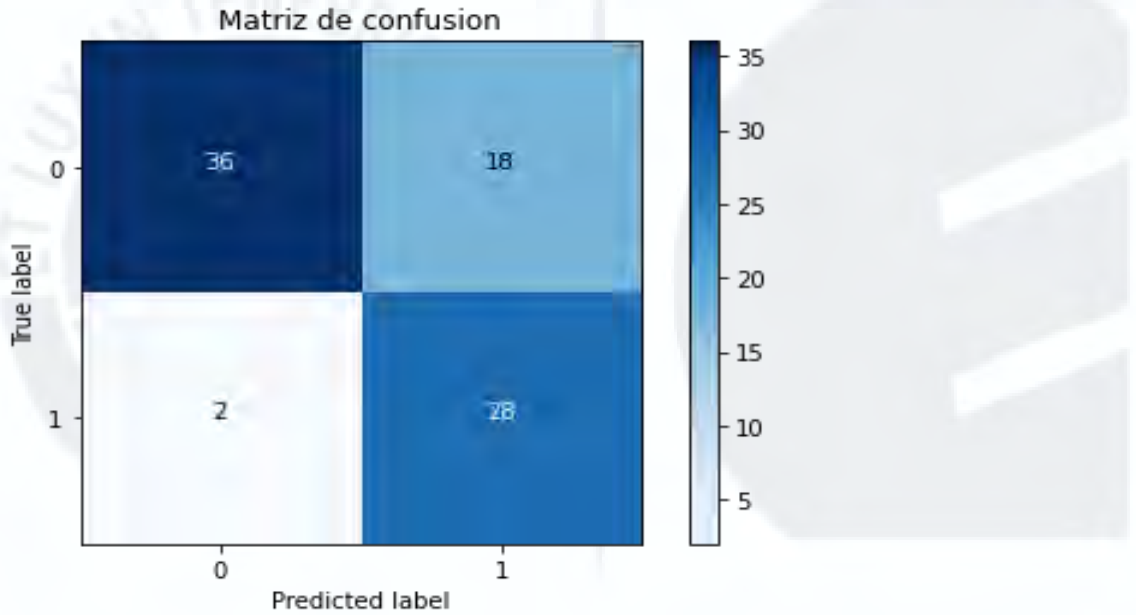
0	Veredas	7/07/2021	Gerson	Santiago	Están en buenas condiciones	1

```
# Vamos a tomar unicamente los tweets que son diferentes a NEU
tweets_test_veredas = tweets_test_veredas[tweets_test_veredas.Etiqueta != 'Neutro']
tweets_test_veredas = tweets_test_veredas[tweets_test_veredas.Etiqueta != 'neutro'] # extraemos
contenido texto
test_veredas_text = tweets_test_veredas.iloc[:,4]#
extraemos labels
test_veredas_labels = tweets_test_veredas.iloc[:,5]
#convertir an tipo int
test_veredas_labels= test_veredas_labels.astype(int)

test_veredas_text.head()

# predicion para pistas
predict = pipeline.predict(test_veredas_text)

# matriz de confusion para pistas
conf = metrics.confusion_matrix(test_veredas_labels,predict)f1 =
metrics.F1-score(test_veredas_labels,predict)
)
```

```
0.7368421052631579
[[36 18]
 [ 2 28]]
```

```
# DIAGRAMA PARA PARQUES

classes = [0, 1]
# plot confusion matrix
plt.imshow(conf, interpolation='nearest', cmap=plt.cm.Blues)
plt.title("Matriz de confusion")
plt.colorbar()
tick_marks =
np.arange(len(classes))
plt.xticks(tick_marks, classes)
plt.yticks(tick_marks, classes)

fmt = 'd'
```

```
thresh = conf.max() / 2.
for i, j in itertools.product(range(conf.shape[0]), range(conf.shape[1])):


        plt.text(j, i, format(conf[i, j], fmt),
                         horizontalalignment="c
                         enter",
                         color="white" if conf[i, j] > thresh else "black")

plt.tight_layout()
plt.ylabel('True label')
plt.xlabel('Predicted
label')
plt.savefig('/gdrive/My Drive/DATASETS/data_pistas/results/confusion_VEREDAS.png')
```




```
#agregamos una columna a nuestro archivo la prediccion
tweets_test_veredas['predict'] =predict


# guardar el archivo
tweets_test_veredas.to_csv('/gdrive/My Drive/DATASETS/data_pistas/results/result_VE

# mostramos lo
agregado
tweets_test_vered
as.head()
```

	Obra	Marca temporal	Indique su nombre	Distrito	*Tweets*	Etiqueta	*Predict*
0	Veredas	7/07/2021 23:17	Fernando	Magdalena del Mar	Las veredas en la av leoncio prado son muy pequeñas y no tienen la bajada para inválidos	0	0

Apéndice E : Data Comentarios de Twitter

Obra	Distrito de residencia	Comentario sobre los parques, pistas y veredas
Parques	Magdalena del Mar	Parque costado de la iglesia de la cúpula por la avenida sucre está descuidado y muy sucio
Parques	Santiago de Surco	Hay muchos y están bien mantenidos
Parques	San Borja	Bonitos
Parques	La Molina	Estan cuidados y con seguridad
Parques	Lima	Requieren más cuidado
Parques	San Miguel	Regulares en presentación, son limpios
Parques	San Martín de Porres	Mal cuidados
Parques	Lince	Bonitos y seguros
Parques	La Molina	Buenos
Parques	Rímac	Los parques Muy descuidados no hay mantenimiento departe de la municipalidad de mi distrito
Parques	La Molina	Me gustan pq son muy seguidos y grandes
Parques	Los Olivos	Están descuidados
Parques	Breña	pésimos y descuidados
Parques	Lima	Mi parque San Antonio, Bellavista Callao, siempre atendido por la municipalidad, juegos para niños y grande.
Parques	San Miguel	Faltan tachos de basura
Parques	Santa Anita	decentes
Parques	San Miguel	No voy a parques
Parques	Lurigancho	Algunos se encuentras en buen estado de mantenimiento y otros no.
Parques	Santiago de Surco	Están bien cuidados
Parques	Jesus María	Tienen un cuidado aceptable
Parques	Santiago de Surco	Están horribles ratas, garrapatas pulgas
Parques	La Molina	los parques que estan cerca de mi c asa, les falta mas juegos para ninos pequeñs
Parques	Santiago de Surco	Están bonitos
Parques	La Molina	FALTA DE MANTENIMIENTO
Parques	San Miguel	Limpios y ordenados
Parques	Carabayllo	El municipio se está encargando de mantenerlos limpios y los riegan constantemente
Parques	La Victoria	ROBOS EN LA NOCHE
Parques	Jesus María	Los parques del distrito son pequeños, limpios y con muchos árboles.
Parques	Santiago de Surco	Son muy bonitos
Parques	Los Olivos	Falta mantenimiento en algunos parques
Parques	Pueblo Libre	Limpios
Parques	Pueblo Libre	Cheveres
Parques	La Molina	Son parques grandes pero muy cuidados, falta mayor cuidado y mantenimiento
Parques	Lima	Recibe atención se la municipalidad
Parques	Rímac	Están mejorando su mantenimiento
Parques	San Martín de Porres	Falta de limpieza y mantenimiento.
Parques	Pueblo Libre	Regulares
Parques	Magdalena del Mar	Descuidados

Parques	Magdalena del Mar	Reciben poco mantenimiento, falta riego. Falta programa de arborización en la Av Brasil.
Parques	Magdalena del Mar	En buen estado
Parques	Villa El Salvador	No existen, están mal tratados
Parques	Lima	Buen Mantenimiento
Parques	San Miguel	regularmente conservados
Parques	San Borja	Se encuentran bien cuidados y limpios
Parques	La Molina	La Molina es uno de los pocos distritos con muchas áreas verdes
Parques	San Miguel	Bonitos, bien conservados, muchos restos de perritos.
Parques	San Borja	Bien cuidados en su mayoría
Parques	Lima	Algunos descuidados
Parques	Surquillo	Requieren mas mantenimiento
Parques	Magdalena del Mar	En buen estado
Parques	Lince	Ordenados, limpios y con seguridad
Parques	Lima	Pueden mejorar y verse más verdes
Parques	San Borja	Hay muchos y la mayoría bien cuidados.
Parques	Lima	Esta en buenas condiciones
Parques	Independencia	Los vecinos son los que se ocupan del mantenimiento de los parques
Parques	La Molina	Son muy escazos
Parques	Los Olivos	mal estado
Parques	San Martín de Porres	Actualmente están bien cuidados cerca a mi Domicilio
Parques	San Miguel	algunos parque están en buenas condiciones, bien arreglados
Parques	Ate	Bien cortados y cuidados
Parques	La Molina	Se encuentran en buen estado y reciben mantenimiento constante
Parques	San Miguel	Hay abundantes áreas verdes, sin embargo, en algunas zonas se debe mejorar el mantenimiento y el riego.
Parques	Santiago de Surco	ESTAN MUY BIEN CUIDADAS
Parques	Lima	Es mayormente para perros, llenos de excremento. Inadecuado para los niños. Solo hay uno, el cuircuito del agua. Igual es usado para perros en las mañanas.
Parques	Lince	Grandes, amplios y de mucha arboleda, pero no con el mejor mantenimiento.
Parques	San Martín de Porres	pocos
Parques	San Martín de Porres	En la zona solo hay tierra y polvo
Parques	Pueblo Libre	Poca limpieza
Parques	Comas	No están en buenas condiciones, no hay servicio de limpieza y sin iluminación por las noches eso hacen que personas de mal vivir estén robando y peor que no hay seguridad de parte de la policía ni de la municipalidad.
Parques	Pueblo Libre	La infraestructura es antigua.
Parques	San Miguel	En buen estado, los riegan con frecuencia
Parques	La Molina	Falta de riego y mantenimiento
Parques	La Molina	Hay gran cantidad de parques, tienen juegos y la gente pasea a sus mascotas constantemente
Parques	San Martín de Porres	Existen pocos parques y lejos de mi vivienda
Parques	Lima	Modernizados varios de ellos y en camino a serlo, otros.
Parques	Ate	LA MAYORIA ESTAN DESCUIDADOS
Parques	San Miguel	Les falta un poco de atención, pero son bonitos.
Parques	Breña	Muy pocos, casi no conozco parques en el distrito.
Parques	San Juan de Lurigancho	es un poco escaso la labor de mantenimiento de los parques, como también se sugiere contar con mas áreas verdes en el distrito
Parques	Comas	FALTA MANTENIMIENTO EN LOS PARQUES
Parques	Lima	Le falta cuidado y vigilancia
Parques	San Martín de Porres	Los parques están abandonados y urgen un mejor mantenimiento e implementar mas áreas recreativas
Parques	Magdalena del Mar	En buen estado

Parques	San Miguel	bien iluminados
Parques	San Borja	Bonitos
Parques	San Miguel	Se encuentran en buen estado
Parques	San Borja	Bastantes y bien cuidados
Parques	San Juan de Lurigancho	Más seguridad
Parques	Magdalena del Mar	Falta mayor limpieza, los perros dejan sus escrementos
Parques	San Juan de Miraflores	Se encuentran en remodelacion
Parques	Los Olivos	Los parques están abandonados, la gente no cuida los parques
Parques	Surquillo	La mayoria de parques donde vivo se encuentran en buen estado pues. La municipalidad realiza el mantenimiento regularmente.
Parques	San Miguel	Son amplios, algunos tienen lozas deportivas, todos deberian tener 1
Pistas	Magdalena del Mar	Pista de la av. libertad estan rotas y el alcalde no las arregla
Pistas	Santiago de Surco	Están en buenas condiciones
Pistas	San Borja	Faltan reparar
Pistas	La Molina	Muy pocas estan en mal estado
Pistas	Lima	Requieren cuidado
Pistas	San Miguel	Regulares, no hay muchos huecos
Pistas	San Martín de Porres	Pistas en mal estado
Pistas	Lince	Aceptables vivo por 2 de mayo
Pistas	La Molina	En buen eatado
Pistas	Rímac	Las pistas. Están mejor q las veredas
Pistas	La Molina	La av.la molina (carretera a cineguilla)es muy complicada para el peaton y transporte
Pistas	Los Olivos	Están en mal estado, en especial las pistas de las avenidas principales
Pistas	Breña	destrozadas
Pistas	Lima	En mal estado por sectores sin mantenimiento.
Pistas	San Miguel	Reparación urgente de las pistas, señalización para evitar accidentes en ciclistas y autos.
Pistas	Santa Anita	un asco
Pistas	San Miguel	Debe mejorar es asfaltado
Pistas	Lurigancho	Solo la pista central que se va la centro se encuentra en mediado estado de mantenimiento sin embargo de al rededores de los distritos la mayoría están averiadas.
Pistas	Santiago de Surco	Normalmente atienden y hacen mantenimiento a las pistas
Pistas	Jesus María	Estan en condiciones aceptables
Pistas	Santiago de Surco	En refacción
Pistas	La Molina	algunas estan en mal estado, sovretodo por la zona del corregidor
Pistas	Santiago de Surco	Algunas en muy mal estado
Pistas	La Molina	LLENO DE BACHES
Pistas	San Miguel	Cerradas en determinadas ocasiones
Pistas	Carabayllo	En algunas lugares deben volver a pavimentarlo
Pistas	La Victoria	LLENO DE HUECOS
Pistas	Jesus María	A finales del año pasado las pistas aledañas a mi domicilio recibieron mantenimiento y actualmente se encuentran en perfecto estado.
Pistas	Santiago de Surco	Están bien, pero podrían estar mejor
Pistas	Los Olivos	Falta de mantenimiento y pistas dañadas en algunas zonas.
Pistas	Pueblo Libre	Algunas en mal estado
Pistas	Pueblo Libre	Pulentas

Pistas	La Molina	Están de mal en peor el Alcade no hace nada por mejorarlas
Pistas	Lima	Estado transitable
Pistas	Rímac	Falta mejorar las pistas y su mantenimiento
Pistas	San Martín de Porres	incompletas
Pistas	Pueblo Libre	Regulares
Pistas	Magdalena del Mar	Destruidas
Pistas	Magdalena del Mar	Moderadamente aceptables, salvo excepciones
Pistas	Magdalena del Mar	en mal estado por tanta construcción
Pistas	Villa El Salvador	En. Mal estado
Pistas	Lima	Falta mantenimiento
Pistas	San Miguel	no estan en buen estado
Pistas	San Borja	Hace poco han hecho mantenimiento y todas están muy bien
Pistas	La Molina	Si bien la mayoría esta en buen estado, hay algunas zonas donde deben ser mejoradas
Pistas	San Miguel	bien conservados
Pistas	San Borja	Bien asfaltadas, pero en algunas partes falta mejorar las señalizaciones con la ciclovia
Pistas	Lima	Mejorando últimamente
Pistas	Surquillo	No hay una buena señalización
Pistas	Magdalena del Mar	Requieren arreglo regularmente. Se dañan rapido
Pistas	Lince	Algunas en mal estado y sin señalizar
Pistas	Lima	Puede pintarse la señalización más frecuentemente
Pistas	San Borja	De los que conozco, la mayoría en buen estado.
Pistas	Lima	Se encuentra en buenas condiciones
Pistas	Independencia	Están deterioradas, mi calle parece avenida, los carros para cortar camino pasan por mi calle (porque a dos cuadras esta el mercado y los ambulantes ocupan la av. principal)
Pistas	La Molina	No tienen mantenimiento constante
Pistas	Los Olivos	mal estado
Pistas	San Martín de Porres	Se encuentran en correcto estado, sin embargo Zonas cercanas aún no poseen pista
Pistas	San Miguel	La pista de la Av. Libertad debe ser arreglada
Pistas	Ate	Algunas son buenas otras no
Pistas	La Molina	El 30 % de las pistas se encuentran dañadas y con huecos
Pistas	San Miguel	Les falta mantenimiento y muchos casos refacción.
Pistas	Santiago de Surco	EN LOS ULTIMOS AÑOS SE REALIZÓ EL MEJORAMIENTO SOLICITADO (BENAVIDES Y CAMINOS DEL INCA)
Pistas	Lima	Generamente las paran rompiendo ya sea por mejoras de desagües, pero no las dejan igual.
Pistas	Lince	Como la mayor parte de Lima, baches, huecos, mantenimientos inconclusos.
Pistas	San Martín de Porres	algunas en mal estado con huecos y baches
Pistas	San Martín de Porres	Falta mantenimiento
Pistas	Pueblo Libre	Falta mantenimiento
Pistas	Comas	Pésimo, están en muy malas condiciones, por eso es que pasan accidentes y los vehículos se malogran constantemente.
Pistas	Pueblo Libre	Se encuentran en buen estado.
Pistas	San Miguel	En buen estado

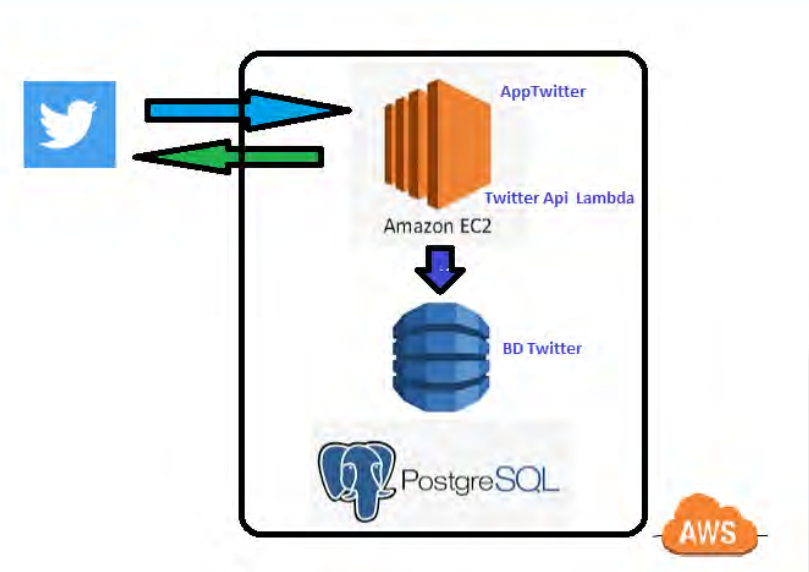
Pistas	La Molina	Falta de señalización
Pistas	La Molina	La mayoría están en buen estado, sin embargo hay zonas que podrían mejorar
Pistas	San Martín de Porres	Tienen muchos baches, y huecos
Pistas	Lima	En mejora, varias de ellas en el Cercado de Lima.
Pistas	Ate	ESTAN MUY DETERIODADAS
Pistas	San Miguel	La mayoría están en buen estado.
Pistas	Breña	En general están en regular estado la mayoria, muchos baches y pocas pistas en buen estado.
Pistas	San Juan de Lurigancho	la av santa rosa, la av lima entre otras av principales del distrito, es necesarios tapar los baches o realizar un nuevo asfaltado, se cuenta con muchos de estos inconvenientes en dichas avenidas.
Pistas	Comas	FALTA MANTENIMIENTO EN LAS PISTAS
Pistas	Lima	Están en buen estado.
Pistas	San Martín de Porres	En lamentable estado
Pistas	Magdalena del Mar	Requieren arreglo regularmente. Se dañan rapido
Pistas	San Miguel	Con varios desperfectos
Pistas	San Borja	Faltan reparar
Pistas	San Miguel	Falta mejorar Av La Paz
Pistas	San Borja	La mayoría en buen estado
Pistas	San Juan de Lurigancho	Algunas de partes con hueco
Pistas	Magdalena del Mar	Falta reparación
Pistas	San Juan de Miraflores	Parcialmente en buen estado.
Pistas	Los Olivos	Algunas están en muy mal estado
Pistas	Surquillo	Se necesita un mejor mantenimiento en las pistas, existen varias que se encuentran averidas, rotas o con baches.
Pistas	San Miguel	Muchos baches en algunas, como av independencia, lima, la paz, bolognesi.
Veredas	Magdalena del Mar	Las veredas en la av leoncio prado son muy pequeñas y no tienen la bajada para inválidos
Veredas	Santiago de Surco	Están en buenas condiciones
Veredas	San Borja	En buen estado
Veredas	La Molina	Muy pocas estan en mal estado
Veredas	Lima	Está bien
Veredas	San Miguel	En mal estado
Veredas	San Martín de Porres	Veredas en mal estado
Veredas	Lince	En buen estado
Veredas	La Molina	Muy pocas
Veredas	Rímac	En mi distrito las veredas están la mayoría rotas o dejan trabajos sin concluir y afecta al transitar peatonalmente
Veredas	La Molina	No hay veredas hay mas jardines y es dificil para el peaton
Veredas	Los Olivos	Son usados para comercio ambulatorio
Veredas	Breña	sucias, con mucho excremento de perro.
Veredas	Lima	Mal estado por sectores, no hay mantenimiento.
Veredas	San Miguel	En muchas calles de San Miguel hay veredas rotas, lo cuál puede ocasionar accidentes en personas mayores o niños.
Veredas	Santa Anita	en algunas zonas estan bien, en otras están rotas
Veredas	San Miguel	Debe mejorar

Veredas	Lurigancho	Falta realizar el mantenimiento de las veredas de las calles principales de mi distrito.
Veredas	Santiago de Surco	Hacen mantenimiento a las veredas, se encuentran cuidadas
Veredas	Jesus María	Podrían ser un poco más amplias.
Veredas	Santiago de Surco	Rotas
Veredas	La Molina	las veredas estan bien, pero deben haber pistas para ciclistas
Veredas	Santiago de Surco	Pueden mejorar
Veredas	La Molina	NO HAY VEREDAS
Veredas	San Miguel	En determinadas zonas sin mantenimiento
Veredas	Carabayllo	Con poco mantenimiento
Veredas	La Victoria	SUCIAS Y FALTA DE REPARACION
Veredas	Jesus María	La mayor parte del distrito tiene veredas en buen estado.
Veredas	Santiago de Surco	Están ok. Pero podrían mejorar y tener mayor acceso para personas con discapacidad
Veredas	Los Olivos	En buen estado.
Veredas	Pueblo Libre	Algunas en mal estado
Veredas	Pueblo Libre	Pollito?
Veredas	La Molina	Faltan veredas o requieren mantenimiento o no existen en algunos casos.
Veredas	Lima	Regular.
Veredas	Rímac	Están en buen estado pero se pueden mejorar en algunos sitios
Veredas	San Martín de Porres	incompletas
Veredas	Pueblo Libre	Regulares
Veredas	Magdalena del Mar	Mal reparadas
Veredas	Magdalena del Mar	La 31 y 32 de la Av Brasil están impresentables
Veredas	Magdalena del Mar	en estado regular
Veredas	Villa El Salvador	Falta colocar en todas las zonas que se necesita
Veredas	Lima	Falta algo mantenimiento
Veredas	San Miguel	regulares
Veredas	San Borja	Se encuentran en buen estado
Veredas	La Molina	Si bien la mayoría esta en buen estado, hay algunas zonas donde deben ser mejoradas
Veredas	San Miguel	Bien conservados
Veredas	San Borja	Bien aunque han estado modificando bastante en parte de avenidas principales sin un objetivo claro
Veredas	Lima	Mejorando
Veredas	Surquillo	Cumplen con su función
Veredas	Magdalena del Mar	En condición relativamente aceptable
Veredas	Lince	Algunas en mal estado
Veredas	Lima	Los bordes de algunas veredas (que van a las pistas) están dañadas. Debería usarse cemento anti resbalones (lluvia)
Veredas	San Borja	Veo que se realizan trabajos para mejorarlas
Veredas	Lima	Esta en buenas condiciones
Veredas	Independencia	Las veredas si estan bien, no hay problema
Veredas	La Molina	Faltan veredas
Veredas	Los Olivos	mal estado
Veredas	San Martín de Porres	No se encuentra homogéneo.

Veredas	San Miguel	Deben arregladas para evitar torcederas de tobillos o traspiés.
Veredas	Ate	Algunas son bonitas otras están bien cuidadas
Veredas	La Molina	Se encuentran en buen estado y cumplen las necesidades peatonales.
Veredas	San Miguel	En algunas zonas no hay veredas.
Veredas	Santiago de Surco	NECESITAN RENOVACION
Veredas	Lima	Generalmente son destruidas por las constructoras. Cuando terminan los edificios no las reconstruyen o dejan igual. Además de dejar rebabas de fierros que generan tropiezos para las personas.
Veredas	Lince	En la mayoría de los casos en estado regular a bueno.
Veredas	San Martín de Porres	hay zonas donde no hay veredas
Veredas	San Martín de Porres	Invadidas por ambulantes
Veredas	Pueblo Libre	Falta mantenimiento
Veredas	Comas	Muy peligrosas, al estar en malas condiciones las personas sufren muchos accidentes.
Veredas	Pueblo Libre	Se encuentran en buen estado.
Veredas	San Miguel	Deberian estar en buen estado. Algunas estan rotas y esto puede ocasionar accidentes a cualquier persona, sobretodo al adulto mayor.
Veredas	La Molina	En mal estado la mayoría
Veredas	La Molina	Podrían estar mejor, es necesario darles mantenimiento porque están agrietadas, también están sucias a veces debido a que los que pasean perros no recogen las excretas.
Veredas	San Martín de Porres	Veredas rotas en inconclusas
Veredas	Lima	Sé de refacciones hechas en algunas calles del Cercado de Lima.
Veredas	Ate	ESTAN SUCIAS Y DESCUIDADAS
Veredas	San Miguel	requiren de mantenimiento pero la mayoria se ve bien.
Veredas	Breña	Están en regular estado, muchas partes invadido por comercios.
Veredas	San Juan de Lurigancho	no he tenido inconveniente con las veredas en las zonas donde resido e visto, los inmuebles y/o predios respetan el distanciamiento adecuado para el transito del ciudadano.
Veredas	Comas	ESTAN EN BUENAS CONDICIONES
Veredas	Lima	Están en buen estado
Veredas	San Martín de Porres	arreglarlas
Veredas	Magdalena del Mar	En condición relativamente aceptable
Veredas	San Miguel	Con varios desperfectos como huecos
Veredas	San Borja	En buen estado
Veredas	San Miguel	Muy estrechas
Veredas	San Borja	La mayoría en buen estado
Veredas	San Juan de Lurigancho	Buenas
Veredas	Magdalena del Mar	Falta mantenimiento
Veredas	San Juan de Miraflores	Algunas están inconclusas o dañadas por el tiempo de uso.
Veredas	Los Olivos	Caminar por la vereda es un poco peligros por la delincuencia
Veredas	Surquillo	De igual forma que en las pistas, las veredas de algunas calles se encuentran averidas, rotas o con baches.
Veredas	San Miguel	Las andan reparando

Apéndice F: Extracción de Datos Twitter

Para la extracción de datos se utilizó los servicios de Amazon. Se alquiló un RDS en postgresSQL y un servidor que apunte hacia Twitter. En este servidor se desarrolló dos códigos llamados AppTwiter y Twitter Api Lambda. En el RDS se recolectó la data y se utilizó el código BD_Twitter



AppTwitter

Este programa encuentra la palabra o texto lo delimita. Este lo manda a Twitter para buscar las interacciones con esta palabra en la localización de Perú.

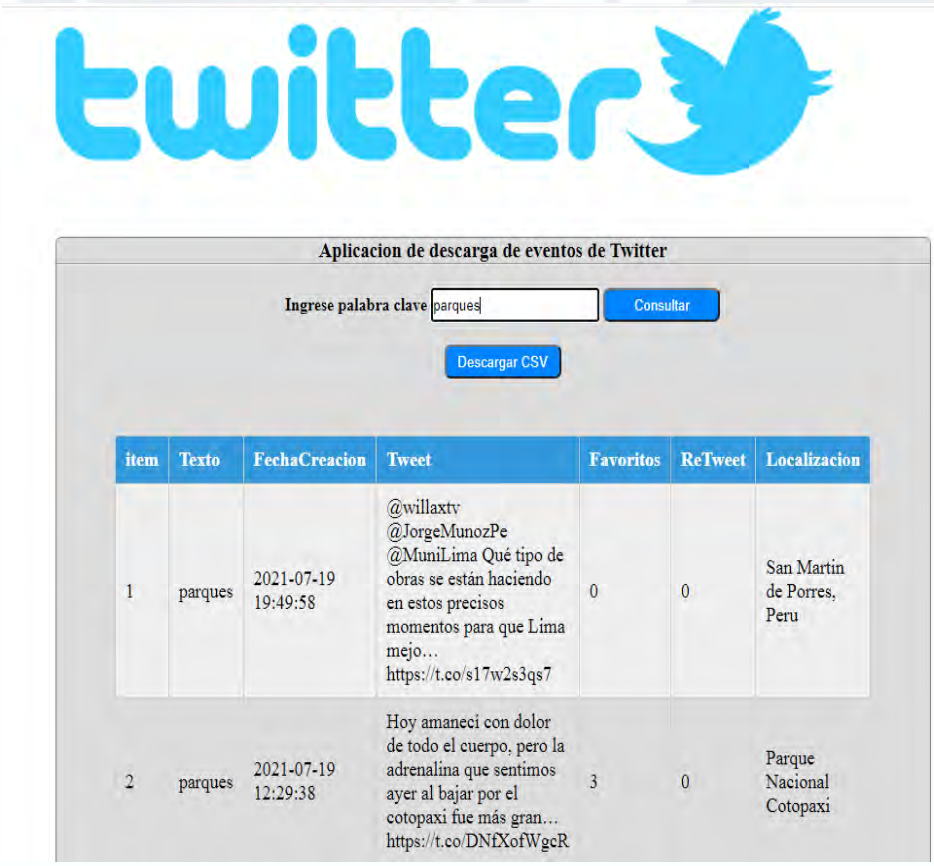
```
function getData() {

    const ul = document.getElementById('authors');
    const url = 'https://a7c4hzn0s4.execute-api.us-east-1.amazonaws.com/v1/';

    const palabra = document.getElementById('txtpalabra');

    if(palabra.value == ''){
        alert('ingrese palabra a buscar');
        return;
    }

    let req = {
        word: palabra.value
    }


    // request options
    const request = new Request(url, {
        method: 'POST',
```

```
        body: JSON.stringify(req),
        headers: new Headers({
            'Content-Type': 'application/json'
        }),

    });


    let loader = `<div class="boxLoading">cargando...</div>`;
    document.getElementById('content').innerHTML = loader;

    fetch(request)
        .then((resp) => resp.json())
        .then(function (data) {
            console.log(data);
            let dataTwitter = data.body;
            document.getElementById('content').innerHTML="";
            document.getElementById("content").appendChild(buildTable(dataTwitter));

        })
        .catch(function (error) {
            console.log(error);
        });


}

function postData() {

    const url = 'https://randomuser.me/api';

    let data = {
        name: 'Sara'
    }

    var request = new Request(url, {
        method: 'POST',
        body: data,
        headers: new Headers()
    });

    fetch(request)
        .then(function () {
            // Handle response we get from the API
        })

}

function createNode(element) {
    return document.createElement(element);
```

```
}

function append(parent, el) {
   return parent.appendChild(el);
}

//
https://programacion.net/articulo/como_exportar_una_tabla_html_a_csv_mediante_javascript
_1742
function descagar(){
   alert('descargando...');
}

function buildTable(data) {
   var table = document.createElement("table");
   table.className="gridtable";
   var thead = document.createElement("thead");
   var tbody = document.createElement("tbody");
   var headRow = document.createElement("tr");

["item","Texto","FechaCreacion","Tweet","Favoritos","ReTweet","Localizacion"].forEach(f
unction(el) {
     var th=document.createElement("th");
     th.appendChild(document.createTextNode(el));
     headRow.appendChild(th);
   });
   thead.appendChild(headRow);
   table.appendChild(thead);

   var cont=1;
   data.forEach(function(el) {
      // console.log('el',el);
      // console.log('location::', el.location);

     var tr = document.createElement("tr");

     var td = document.createElement("td");
     td.appendChild(document.createTextNode(cont))
     tr.appendChild(td);

     var td = document.createElement("td");
     td.appendChild(document.createTextNode(el.word))
     tr.appendChild(td);

     var td = document.createElement("td");
     td.appendChild(document.createTextNode(el.created_at))
     tr.appendChild(td);

     var td = document.createElement("td");
     td.appendChild(document.createTextNode(el.tweet))
```

```
      tr.appendChild(td);

      var td = document.createElement("td");
      td.appendChild(document.createTextNode(el.favorite_count))
      tr.appendChild(td);

      var td = document.createElement("td");
      td.appendChild(document.createTextNode(el.retweet_count))
      tr.appendChild(td);

      var td = document.createElement("td");
      td.appendChild(document.createTextNode(el.location))
      tr.appendChild(td);


    //   for (var o in el) {
    //     var td = document.createElement("td");
    //     td.appendChild(document.createTextNode(el[o]))
    //     tr.appendChild(td);
    //   }

      tbody.appendChild(tr);
      cont++;
    });
    table.appendChild(tbody);
    return table;
}


function downloadCSV(csv, filename) {
    var csvFile;
    var downloadLink;

    // CSV file
    csvFile = new Blob([csv], {type: "text/csv"});

    // Download link
    downloadLink = document.createElement("a");

    // File name
    downloadLink.download = filename;

    // Create a link to the file
    downloadLink.href = window.URL.createObjectURL(csvFile);

    // Hide download link
    downloadLink.style.display = "none";

    // Add the link to DOM
    document.body.appendChild(downloadLink);
```

```
    // Click download link
    downloadLink.click();
}


function exportTableToCSV(filename) {
    var csv = [];
    var rows = document.querySelectorAll("table tr");

    for (var i = 0; i < rows.length; i++) {
        var row = [], cols = rows[i].querySelectorAll("td, th");

        for (var j = 0; j < cols.length; j++)
            row.push(cols[j].innerText);

        csv.push(row.join(";"));
    }

    // Download CSV file
    downloadCSV(csv.join("\n"), filename);
}
```



Aplicacion de descarga de eventos de Twitter

Ingrese palabra clave parques Consultar

Descargar CSV

item	Texto	FechaCreacion	Tweet	Favoritos	ReTweet	Localizacion
1	parques	2021-07-19 19:49:58	@willaxtv @JorgeMunozPe @MuniLima Qué tipo de obras se están haciendo en estos precisos momentos para que Lima mejo… https://t.co/s17w2s3qs7	0	0	San Martin de Porres, Peru
2	parques	2021-07-19 12:29:38	Hoy amaneci con dolor de todo el cuerpo, pero la adrenalina que sentimos ayer al bajar por el cotopaxi fue más gran… https://t.co/DNfXofWgcR	3	0	Parque Nacional Cotopaxi

Twitter Api Lambda

Es el proceso que hace la consulta con el servicio API hacia Twiter. En este archivo se colocan los parámetros de localización, el texto a buscar y la fecha.

```
import tweepy
import psycopg2
import boto3
import csv
import io
import uuid
import json
from psycopg2.extras import execute_values

api_key = 'qOGIDKF06MdKy8XLXve8dewi3'
api_secret_key = '5BtpUi4mQOgfj16rDnJgW6vQ1Uxy14ojDg4OWvieAvRe7B59f3'
access_token = '346297120-C9b4npiOw2sh7dUYn2GDnCwhnR129xJCGpMEk1b6'
access_token_secret = 'LAq4XhEX0gmN1EvAhN3v2BsHsX135V5xk0JVInxy6Jg9g'
aws_key_id = 'AKIAXH37C4D6QIAPU6GX'
aws_secret_key_id = 'y3veVVLmLDO1Ea1eDKvxvMZs6kqjHomy7YAnu0M+'
bucket = 'twitter-data-extraction'

database = 'tweetsdb'
user = 'postgres'
password = '12345678'
host = 'twitter-extraction.cp23czrhpmva.us-east-1.rds.amazonaws.com'
db_table = 'tweets_dev'


s3_client = boto3.client('s3', aws_access_key_id=aws_key_id,
aws_secret_access_key=aws_secret_key_id)

def insert_to_db(objs):

    if not objs:
        print('No objects to insert.')
        return

    connection_string = f"dbname='{database}' user='{user}' password='{password}'
host='{host}'"
    connection = psycopg2.connect(connection_string)
    cursor = connection.cursor()
    columns = ','.join(list(objs[0].keys()))
    insert_query = f"INSERT INTO {db_table}({columns}) values %s"
    values = [[value for value in obj.values()] for obj in objs]
    execute_values(cursor, insert_query, values)
    connection.commit()
    cursor.close()
    connection.close()
```

```
def create_csv(objs, word):

    if not objs:
        print('No objects to create doc.')
        return

    _f = io.StringIO()
    field_names = list(objs[0].keys())
    key = f"{word}-run_id-{uuid.uuid4().hex}"
    with _f:
        writer = csv.DictWriter(_f, fieldnames=field_names)
        writer.writeheader()
        writer.writerows(objs)
        _f.seek(0)
        s3_client.put_object(Bucket=bucket, Key='tweets-info/' + key + '.csv', Body=_f.read())

def main(event, context):

    auth = tweepy.OAuthHandler(api_key, api_secret_key)
    auth.set_access_token(access_token, access_token_secret)
    api = tweepy.API(auth)

    tweet_objs = list()
    last_id = 0
    q = event['word']
    base_params = {
        'q': q,
        'result_type': 'popular',
        'count': 100,
        'include_entities': False,
        'geocode': '-9.1813525,-75.002365,1000km',
        'lang': 'es'
    }

    while True:

        if last_id > 0 and last_id is not None:
            base_params.update({'since_id': last_id})

        tweets = api.search(**base_params)
        if not tweets: break
        for tweet in tweets:
            obj = {
                'word': q,
                'created_at': str(tweet.created_at),
                'tweet': tweet.text,
                'favorite_count': tweet.favorite_count,
                'retweet_count': tweet.retweet_count,
                'location': tweet.place.full_name
```

```
            }

            tweet_objs.append(obj)
            print(obj)
            last_id = tweet.id

    print(len(tweet_objs))

    create_csv(tweet_objs, q)
    insert_to_db(tweet_objs)

    return tweet_objs


if __name__ == '__main__':
    main({'word': 'elecciones'}, None)
```

BD Twiter

Es el lugar de almacenamiento (Base de datos) que devuelve el API cada vez que se le hace la consulta.

```
-- Table: public.tweets_metrics

-- DROP TABLE public.tweets_metrics;

CREATE TABLE IF NOT EXISTS public.tweets_metrics
(
    _id integer NOT NULL DEFAULT nextval('tweets_metrics__id_seq'::regclass),
    word character varying(50) COLLATE pg_catalog."default",
    created_at timestamp without time zone,
    tweet text COLLATE pg_catalog."default",
    favorite_count integer,
    retweet_count integer,
    location character varying(50) COLLATE pg_catalog."default",
    CONSTRAINT tweets_metrics_pkey PRIMARY KEY (_id)
)

TABLESPACE pg_default;

ALTER TABLE public.tweets_metrics
    OWNER to postgres;
```