Maestría en

Ciencia de Datos y Máquinas de Aprendizaje con Mención en Inteligencia Artificial

Trabajo Previo a la Obtención de Título de Magister en Ciencia de Datos y Máquinas de Aprendizaje con Mención en Inteligencia Artificial

AUTORES:

Granda Gómez John Michael

Lema Auz Bryan Germán

Llerena Mena Alex Fernando

Villarruel Cerón Wagner Rodrigo

TUTORES

Alejandro Cortés López
Iván Galo Reyes Chacón

TEMA:

Modelo Híbrido de Segmentación y Predicción de Flujo Vehicular en la Avenida “Las Aguas” en Guayaquil - Ecuador, utilizando K-Means y Random Forest

Quito - Ecuador

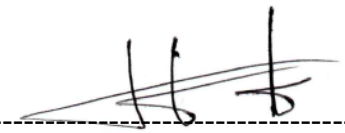
julio – 2025

REINVENTEMOS EL FUTURO

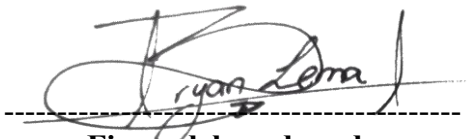
Certificación de Autoría

Nosotros, **Granda Gómez John Michael, Lema Auz Bryan Germán, Llerena Mena Alex Fernando y Villarruel Cerón Wagner Rodrigo,** declaramos bajo juramento que el trabajo aquí descrito es de nuestra autoría; que no ha sido presentado anteriormente para ningún grado o calificación profesional y que se ha consultado la bibliografía detallada.

Cedemos nuestros derechos de propiedad intelectual a la Universidad Internacional del Ecuador (UIDE), para que sea publicado y divulgado en internet, según lo establecido en la Ley de Propiedad Intelectual, su reglamento y demás disposiciones legales.

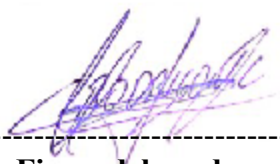


Firma del graduando
Granda Gómez John Michael



Firma del graduando
Lema Auz Bryan Germán

Firma del graduando
Llerena Mena Alex Fernando



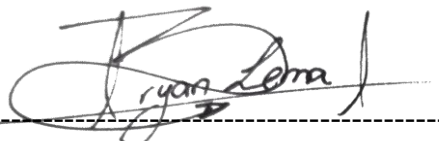
Firma del graduando
Villarruel Cerón Wagner Rodrigo

Autorización de Derechos de Propiedad Intelectual

Nosotros, **Granda Gómez John Michael, Lema Auz Bryan Germán, Llerena Mena Alex Fernando y Villarruel Cerón Wagner Rodrigo**, en calidad de autores del trabajo de investigación titulado ***Modelo Híbrido de Segmentación y Predicción de Flujo Vehicular en la Avenida "Las Aguas" en Guayaquil - Ecuador, utilizando K-Means y Random Forest***, autorizamos a la Universidad Internacional del Ecuador (UIDE) para hacer uso de todos los contenidos que nos pertenecen o de parte de los que contiene esta obra, con fines estrictamente académicos o de investigación. Los derechos que como autores nos corresponden, lo establecido en los artículos 5, 6, 8, 19 y demás pertinentes de la Ley de Propiedad Intelectual y su Reglamento en Ecuador.

D. M. Quito, julio 2025

Firma del graduando
Granda Gómez John Michael



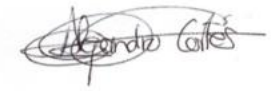
Firma del graduando
Lema Auz Bryan Germán

Firma del graduando
Llerena Mena Alex Fernando

Firma del graduando
Villarruel Cerón Wagner Rodrigo

Aprobación de Dirección y Coordinación del Programa

Nosotros, **Alejandro Cortés López** y **Reyes Chacón Iván Galo**, declaramos que: **Granda Gómez John Michael, Lema Auz Bryan Germán, Llerena Mena Alex Fernando y Villarruel Cerón Wagner Rodrigo**, son los autores exclusivos de la presente investigación y que ésta es original, auténtica y personal de ellos.



Alejandro Cortés López
Director de la Maestría en Ciencia de Datos y Máquinas de Aprendizaje con Mención en Inteligencia Artificial -EIG

Reyes Chacón Iván Galo
Coordinador de la Maestría en Ciencia de Datos y Máquinas de Aprendizaje con Mención en Inteligencia Artificial - UIDE

Dedicatoria

A mi madre, pilar fundamental en mi vida y fuente de motivación y empuje, a mi padre que me guía desde el cielo y con su fuerza me hizo quien soy como hombre, a mi esposa por su apoyo constante en cada paso y a mis hijas que con sus sonrisas iluminan cada día.

Granda Gómez John Michael

A mi padre, madre y hermanas, por forjar un camino lleno de sueños y aspiraciones, saber que siempre puedo dar un poco más y continuar resiliente ante cualquier dificultad. A mi gran amor, Victoria Gordón, fuiste, eres y serás mi mayor fortaleza e inspiración. Y la infaltable mención especial para Rocco, Kata y Thomas, tres seres que no han permitido que me rinda con su amor incondicional.

Lema Auz Bryan Germán

A mi familia, en especial a mi abuela Elvia Mena, por su amor incondicional, por creer en mí incluso en los momentos más difíciles y por ser mi mayor fuente de inspiración.

Llerena Mena Alex Fernando

A mi futura esposa, Tatiana Arcos, tu compañía ha sido fundamental para mantenerme motivado y enfocado. Muchas gracias por enseñarme a ver la vida de distintos colores y ser la cómplice fundamental en cada una de nuestras aventuras.

Villarruel Cerón Wagner Rodrigo

Agradecimientos

Mi más profundo agradecimiento a mis padres, mi madre que con su guía y empuje siempre me impulsa a seguir creciendo en todos los aspectos de mi vida, mi padre que con su fuerza y carácter cimento las bases para nunca darme por vencido y perseverar ante cualquier situación.

A mi esposa por su apoyo constante y ejemplo de mejoramiento continuo, sin su ayuda y comprensión el alcanzar esta meta no hubiese sido posible, a mis hijas por ser luz y alegría en cada día, con sus sonrisas y juegos hacen que cada día sea único.

Granda Gómez John Michael

En primer lugar, en un sentido un poco egoísta, quiero agradecer mi esfuerzo a mi mismo por lograr culminar este éxito en el ámbito académico con dedicación y mucho esfuerzo.

A mis padres por sus esfuerzos al brindarme su apoyo y motivación para lograr mis títulos académicos a pesar de cualquier conflicto.

Al amor de mi vida, por su apoyo incondicional durante todo este proceso. Sabemos que no fue fácil y hubo más contras que pros pero aquí estamos.

Lema Auz Bryan Germán

Mi más sincero agradecimiento a mi director de tesis, por su orientación constante, su exigencia académica y por la constante motivación de no conformarme con lo básico.

A los docentes que con su experiencia enriquecieron mis conocimientos en inteligencia artificial, máquinas de aprendizaje y ciencia de datos.

Y, por supuesto, a mi familia, por su paciencia, confianza y apoyo en todo momento.

Llerena Mena Alex Fernando

Me siento muy agradecido con mi madre, Mónica Cerón, por ser una madre tan maravillosa y presente en cada uno de mis pasos, brindarme su guía y sabiduría.

A mis hermanos por brindarme sus consejos y comprensión en cada momento de mi vida.

A personas muy importantes en mi vida: Liliana Cerón, Diego Cadena, Stéfano Cruz, Santiago Aguirre, Carolina García y Ulises Maigua. Por brindarme tanto cariño, apoyo incondicional, risas compartidas y vivencias que las llevo en mi corazón.

Villarruel Cerón Wagner Rodrigo

Resumen

El enfoque de este proyecto es el análisis y predicción del tráfico vehicular, una necesidad que se puede notar en las urbes que se encuentran en constante crecimiento o cambio las cuales están en un proceso de mejorar en la gestión de la movilidad urbana de manera constante. El objetivo principal es desarrollar un modelo híbrido de segmentación que permita identificar los patrones de tráfico y ayude a predecir el flujo vehicular.

La metodología propuesta se basa en la combinación de técnicas de agrupamiento y aprendizaje supervisado. Se utiliza el algoritmo K-Means para segmentar los datos de tráfico, identificando patrones; diferenciando estados como el flujo fluido como lento o congestionado. También, se incorpora un modelo de Random Forest para la predicción del flujo vehicular. La fortaleza de Random Forest para la predicción no lineal, lo hace una elección adecuada para el análisis de los datos de tráfico. Además, se integran factores contextuales como variables climáticas, las cuales han demostrado que influyen en el comportamiento del tráfico y ayudan tener un mejor modelo predictivo.

El modelo híbrido de K-Means y Random Forest proporcionó la segmentación de patrones de tráfico y en la predicción del flujo vehicular, junto con la capacidad de identificar y predecir condiciones de tráfico, estos datos resultantes permitirán realizar propuestas de mejora en el tráfico vehicular.

En conclusión, la integración de K-Means y Random Forest no solo provee un análisis de las condiciones de tráfico y su predicción, sino que también tiene el potencial para generar propuestas de mejoras en el ámbito de movilidad urbana.

Palabras Claves: Tráfico, K-Means, Random Forest, Predicción, Modelo Híbrido

Abstract

The focus of this project is the analysis and prediction of vehicular traffic—an increasingly pressing need in urban areas undergoing constant growth or transformation. These cities are continuously striving to improve the management of urban mobility. The primary objective is to develop a hybrid segmentation and prediction model capable of identifying traffic patterns and forecasting traffic flow.

The proposed methodology is based on the integration of clustering and supervised learning techniques. Specifically, the K-Means algorithm is employed to segment traffic data, enabling the identification of distinct traffic states such as free-flowing, slow-moving, or congested conditions. In parallel, a Random Forest model is incorporated to perform traffic flow prediction. Given its strength in handling non-linear relationships and complex interactions, Random Forest is a suitable choice for modeling traffic behavior.

Additionally, the model integrates contextual factors such as weather variables, which have been shown to significantly influence traffic dynamics. Including these variables contributes to building a more accurate and robust predictive model.

The hybrid K-Means and Random Forest approach allows for both the segmentation of traffic patterns and the prediction of traffic flow, as well as the identification of specific traffic conditions. The insights generated from this model can inform targeted proposals for traffic improvement measures.

In conclusion, the integration of K-Means and Random Forest not only facilitates the analysis and prediction of traffic conditions but also offers the potential to support evidence-based decision-making for enhancing urban mobility systems.

Keywords: Traffic, K-Means, Random Forest, Prediction, Hybrid Model

Índice General

Índice de Tablas

Índice de Figuras

Capítulo I

Introducción

La movilidad urbana es un tema crítico en las ciudades modernas, donde los avances tecnológicos buscan no solo ser más eficientes en la construcción de vías más eficientes, sino también mejorar los medios de transporte terrestre y fomentar un uso adecuado de cada uno de ellos. Es importante considerar una adecuada información sobre la planificación del tráfico y la comunicación de los riesgos inherentes al tránsito. El flujo vehicular se refiere a la cantidad de vehículos que transitan por una vía en un determinado periodo de tiempo. Su adecuada gestión es esencial para reducir la congestión vehicular, cuellos de botella, riesgos de siniestros viales con el fin de mejorando la calidad de vida de los ciudadanos (OPS, 2022).

Según La Organización Mundial de la Salud (UNIDAS, 2024) se estima que 1.3 millones de personas mueren cada año a nivel mundial por siniestros de tránsito, lo sugiere la necesidad de implementar estrategias que no solo se enfoquen en la seguridad vial, sino que también se enfoque la optimización del flujo vehicular urbano. La seguridad vial y sus consecuencias tiene afectación en pérdidas irreparables para sus familias o personas lesionadas de forma permanente sufriendo limitaciones para el resto de su vida (OMS, 2023).

De acuerdo al Reporte de Siniestralidad Guayaquil 2022, se estima que hay un parque automotor de 693.161 vehículos, es decir, hay 250 automotores por cada mil habitantes (2.772.896). El 32% del parque automotor corresponde a motocicletas. Precisamente, los conductores de estos vehículos son los más vulnerables. Las estadísticas nos indican que 3 de cada 5 fallecidos por accidentes de tránsito fueron motociclistas (ECUAVISA, 2023).

La Avenida “Las Aguas”, una de las zonas más transitadas en la ciudad de Guayaquil – Ecuador. Es un claro ejemplo de los desafíos modernos que enfrenta la movilidad urbana. En esta vía, es frecuente los accidentes de tránsito, lo que ha provocado a los moradores de sectores aledaños piden a las autoridades un mayor control, debido a que varios conductores

viajan a alta velocidad, ocasionando accidentes de tránsitos y en casos más lamentables atropellamientos (METRO, 2022).

Uno de los principales factores en la movilidad viene dado por las condiciones climáticas, mismas que afectan la seguridad vial, la velocidad de los automotores y como efecto domino el flujo del tráfico. Diversos factores tales como las lluvias, con mayor o menor intensidad, inundaciones, temperaturas extremas, niebla y vientos fuertes alteran las condiciones de las carreteras, la visibilidad y el comportamiento de los conductores. En Guayaquil, Ecuador, una ciudad costera vulnerable al cambio climático, las inundaciones, las lluvias intensas y las altas temperaturas son desafíos clave que impactan la movilidad urbana, especialmente en avenidas como Las Aguas, ubicada en el sector de Urdesa. Esta avenida, una vía residencial y comercial importante, experimenta congestión significativa durante eventos climáticos adversos, lo que incrementa el estrés de los conductores y el riesgo de accidentes, afectando el flujo vehicular.

1.1 Efectos Generales de las Condiciones Climáticas en el Flujo Vehicular

A continuación, se citarán las condiciones climáticas que tienen mayor impacto en la movilidad urbana, flujo vehicular, y su impacto o relación en el aumento o disminución del mismo, los apartados citados se analizan en rasgos generales tanto aquellos que tienen mayor índice de ocurrencia en el caso de estudio tales como lluvias o temperaturas extremas, así como aquellos con índice de ocurrencia menor o nulo tales como niebla o vientos fuertes.

- **Lluvia y Superficies Resbaladizas**

La lluvia reduce la visibilidad y la adherencia de los neumáticos, disminuyendo la velocidad promedio de los vehículos y el flujo vehicular, dando lugar a congestionamientos. Las carreteras mojadas aumentan el riesgo de accidentes (FasterCapital 2024), este riesgo intuitivamente lleva a los conductores a ser más cautelosos causando, como consecuencia, que el volumen de tráfico incremente durante dicho comportamiento climático, al

comportamiento descrito se añade que como producto de la intensidad de las precipitaciones, se originen con mayor facilidad siniestros de tránsito que a su vez deriva en mayor congestión vehicular, aumentando temporalmente el flujo en ciertas áreas debido a ambos escenarios.

- **Niebla y Reducción de la Visibilidad**

La niebla limita la visibilidad, obligando a los conductores a reducir la velocidad. Autoescuela Quinta Avenida (2024) indica que la niebla incrementa el riesgo percibido, reduciendo el número de vehículos en circulación. Aunque menos común en Guayaquil, la niebla puede afectar áreas periféricas, contribuyendo al estrés de los conductores.

- **Vientos Fuertes**

Los vientos fuertes afectan la estabilidad de los vehículos, especialmente los pesados, reduciendo el flujo vehicular. Los vientos extremos incrementan el riesgo de accidentes (FasterCapital 2024), este comportamiento climático, aunque menos común en la urbe Guayaquileña, puede llevar al cierre de vías, posibles accidentes por la falta de pericia de los conductores o incluso, dependiendo de la severidad, congestionamiento vehicular, es necesario recalcar que como efecto paralelo en condiciones climáticas adversas el flujo tiende a disminuir.

- **Temperaturas Extremas**

Las temperaturas altas causan sobrecalentamiento de motores, mientras que las bajas pueden generar hielo en regiones frías. Gorayeb (2024) destaca que las temperaturas extremas afectan la infraestructura vial y la disposición de los conductores a viajar. En Guayaquil, las altas temperaturas y la humedad contribuyen al estrés y la fatiga, aumentando el riesgo de accidentes.

- **Cambios en la Demanda de Transporte**

Las condiciones climáticas guardan relación directa sobre la demanda del transporte

público, en condiciones climáticas adversas el transeúnte, para efectos de movilidad, buscará salvaguardar su propio transporte y dará uso del transporte público. La Asociación Automotriz del Perú (2024) reporta que las buenas condiciones climáticas incrementaron el flujo de vehículos pesados en 2024, mientras que las condiciones adversas reducen los viajes no esenciales. En Guayaquil, las lluvias intensas disminuyen el flujo de vehículos ligeros, en contraparte acorde a lo indicado, dicho use se ve abocado al transporte público, sin embargo, los accidentes, producto de dichas condiciones adversas, pueden generar congestión.

1.2 Contexto Climático de Guayaquil y la Avenida Las Aguas

Guayaquil, con aproximadamente 2.6 millones de habitantes, enfrenta desafíos debido a su ubicación costera y su vulnerabilidad al cambio climático. Según el Banco de Desarrollo de América Latina (CAF, 2017), la ciudad recibe un promedio de 791 mm de precipitaciones anuales, con inundaciones como el principal impacto climático. La Avenida Las Aguas, ubicada en el sector de Urdesa, es una vía residencial y comercial que conecta áreas clave del norte de Guayaquil. Soporta un promedio de 4,000-6,000 vehículos por hora en horas pico, sin embargo, su capacidad se ve afectada por lluvias intensas, inundaciones y altas temperaturas, que incrementan el estrés de los conductores y el riesgo de accidentes. A continuación, se citan los escenarios climáticos presentes en la ciudad de Guayaquil que afectan directamente al incremento o decremento, dependiendo del escenario, del flujo vehicular y adicional, por falta de pericia o cuidado de los conductores, posibles accidentes que agravan las condiciones de movilidad.

- **Inundaciones y Congestión Vehicular**

Las inundaciones son el principal factor climático que afecta la Avenida Las Aguas. Hernández (2017) indica que las lluvias intensas generan acumulación de agua en intersecciones como las de la Av. Las Aguas con las calles Ilanes y Bálsamos, reduciendo la velocidad promedio a 10-15 km/h incrementando la congestión vehicular y como

consecuencia el flujo se ve afectado. Estas inundaciones, causadas por un sistema de drenaje insuficiente, así como posibles convergencias con otros factores naturales tales como aguajes, provocan cierres parciales y desvíos, reduciendo el flujo vehicular en un 20-30% durante las horas pico (Ziad & Verdezoto, 2020). Los accidentes causados por el estrés y la baja visibilidad, así como la falta de pericia de los conductores bajo estas condiciones, generan congestión adicional, aumentando temporalmente el flujo de vehículos detenidos.

- **Lluvias Intensas, Estrés y Accidentes**

Las lluvias intensas, acompañadas de posibles tormentas eléctricas, muy comunes durante el periodo de mayores precipitaciones y humedad (enero-abril), inciden en el comportamiento, pericia y rango de maniobra de los conductores en la Avenida Las Aguas. Según Ziad y Verdezoto (2020), la acumulación de agua y el tráfico de vehículos comerciales y de transporte público generan atascos prolongados. El estrés causado por las lluvias, sumado con la dificultad de circular en una vía congestionada, aumenta notablemente la posibilidad de errores al momento de conducir. Hassan et al. (2021) destacan que las condiciones climáticas adversas, como la lluvia, incrementan el estrés psicológico, lo que se traduce en una mayor incidencia de accidentes, especialmente colisiones traseras. En la Avenida Las Aguas, estos accidentes pueden bloquear carriles, aumentando el flujo de vehículos detenidos.

- **Temperaturas Altas, Humedad y Fatiga**

Guayaquil experimenta temperaturas de 25-30°C, en promedio, con alta humedad, llegando a tener incluso sensación térmica superior a 34°C, esta condición climática afecta a los conductores en la Avenida Las Aguas. Hidalgo Sánchez (2017) remarca que el calor contribuye a la fatiga en avenidas comerciales, aumentando el riesgo de accidentes. Según Wickens et al. (2015), las altas temperaturas y la humedad elevan el estrés térmico, afectando la atención y el tiempo de reacción. En la Avenida Las Aguas, al igual que en el resto de

avenidas de la ciudad, este estrés puede llevar a maniobras imprudentes producto del estrés acumulado, tales como cambios de carril bruscos, carencia de reacción ante semaforización o señalética de tránsito, entre otros, que generan colisiones que pueden ir desde menores hasta graves y como producto congestión, incrementando el flujo de vehículos detenidos.

- **Impacto en el Comercio Local**

La Avenida Las Aguas es un centro comercial y residencial con restaurantes, tiendas y oficinas. Las inundaciones y las altas temperaturas reducen el flujo vehicular y el acceso de clientes, afectando los ingresos de los negocios. El Municipio de Guayaquil (2019) reporta que las lluvias intensas disminuyen la afluencia de vehículos en zonas comerciales, mientras que el estrés de los conductores reduce la disposición de los consumidores a visitar la avenida durante condiciones adversas.

- **Contaminación y Emisiones Vehiculares**

Las condiciones climáticas influyen en las emisiones vehiculares en la Avenida Las Aguas, la lluvia, así como las altas temperaturas conllevan a los conductores de vehículos livianos a hacer uso de los sistemas de enfriamiento y por consecuencia a incrementar la contaminación por monóxido de carbono (CO). Ziad y Verdezoto (2020) indican que las altas temperaturas durante la temporada húmeda incrementan las emisiones de monóxido de carbono (CO) y óxidos de nitrógeno (NOx), esta contaminación ambiental, producto del incremento en las emisiones, puede llevar a restricciones en la movilidad, medidas que han sido adoptadas en otras ciudades del mundo. La contaminación ambiental, producto de las emisiones vehiculares mencionadas, es otro agravante para accidentes causados por el estrés de los conductores que derivan en congestión adicional.

- **Vulnerabilidad de la Infraestructura Vial**

La infraestructura vial de la Avenida Las Aguas es vulnerable a las lluvias intensas debido a un sistema de drenaje obsoleto. Según la CEPAL (2014), el cambio climático agrava

las inundaciones urbanas, afectando la capacidad de las carreteras. Las calles aledañas en Urdesa, cuya infraestructura es vulnerable incluso en mayor medida a la Avenida Las Aguas, también sufren inundaciones incluso en mayor medida que la Avenida principal por la falta de mantenimiento del drenaje, limitando las rutas alternativas y concentrando el tráfico en la avenida.

El impacto de las condiciones climáticas en la Avenida Las Aguas refleja los desafíos estructurales y climáticos de Guayaquil. Las inundaciones, incrementadas por el cambio climático, reducen el flujo vehicular en un 20-30% durante lluvias intensas, pero los accidentes causados por el estrés de los conductores generan picos de congestión, aumentando temporalmente el flujo de vehículos detenidos.

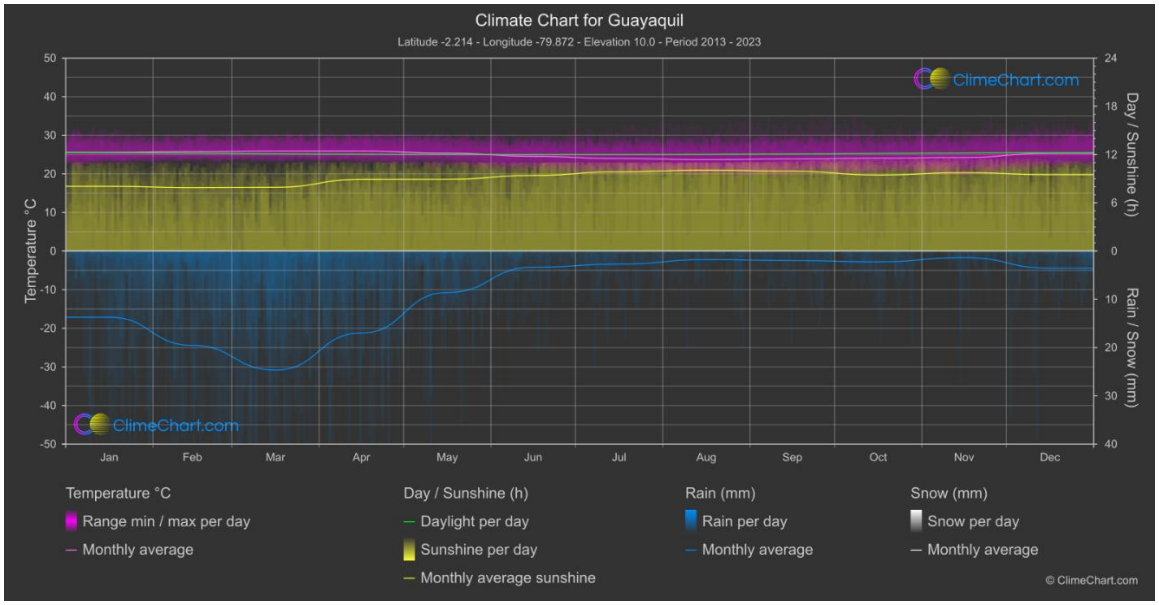
Las altas temperaturas y la humedad agravan el estrés térmico, especialmente en una vía como la Avenida Las Aguas, donde los conductores enfrentan tráfico denso y presión para estacionarse. Este estrés, combinado con la fatiga, incrementa la probabilidad de accidentes, como colisiones por alcance, que bloquean la vía y generan congestión. Comparada con otras avenidas, como la Av. Pedro Menéndez Gilbert, la Avenida Las Aguas es más vulnerable debido a su menor ancho y la falta de rutas alternativas efectivas en Urdesa.

La falta de un sistema de drenaje moderno es un factor crítico. Las intersecciones propensas a inundaciones, como Bálsamos, no solo reducen el flujo vehicular, sino que también afectan la economía local al limitar el acceso a los comercios. La modernización del drenaje, combinada con semáforos inteligentes y monitoreo en tiempo real, podría ayudar a mitigar estos impactos en las vias. Además, campañas de educación vial que sean planficadas a partir de modelos predictivos ayudarían a una mejora en la conducción en condiciones adversas y de esta manera también lograr minimizar los accidentes y optimizar el flujo vehicular.

Las condiciones climáticas descritas se pueden observar en la figura1.

Figura 1

Clima en Guayaquil



Fuente: (ClimeChart, 2023)

La movilidad urbana y la seguridad vial son desafíos que se encuentran constantemente en ciudades con alta densidad de tráfico como Guayaquil. La Avenida “Las Aguas”, es una arteria de tránsito vehicular principales dentro de la ciudad, por lo cual presenta un alto flujo de vehículos de distinto tipo y como en toda avenida principal también se presencian la incidencia de siniestros viales. La predicción de accidentes puede basarse en análisis estadísticos simples o actualmente con el desarrollo de herramientas tecnológicas y los avances computacionales, se pueden usar modelos de aprendizaje automático para dichas predicciones. Sin embargo, algunos de estos enfoques no consideran contextos físicos para la predicción del tráfico, como son los factores climáticos.

Para abordar este tipo de desafíos que se encuentran en las ciudades en constante movilidad urbana en especial la avenida objeto de estudio, es crucial implementar estrategias que integren contextos físicos como el clima para poder generar una mejor predicción. De tal manera, que al predecir estos flujos vehicular se pueda proponer planes que puedan ayudar a

una movilidad más segura y eficiente.

1.3 Planteamiento del Problema e Importancia del Estudio

1.3.1 *Tema del Proyecto*

Modelo híbrido de segmentación y predicción de siniestros viales en la avenida “Las Aguas” en Guayaquil - Ecuador, utilizando k-means y random forest.

1.3.2 Importancia del Estudio

Se han realizado esfuerzos para mejorar la movilidad urbana en Guayaquil, como es la regulación de tiempos de semáforos, cambio de sentido de flujo vehicular en ciertas zonas, pero aún no se ha identificado una estrategia puntual que permita identificar patrones y factores que incrementan el flujo vehicular en zonas específicas de la ciudad. La Avenida “Las Aguas”, siendo una de las vías más transitadas de la ciudad, presenta características específicas las cuales tienen que ser consideradas para una adecuada gestión y control de tráfico.

Este estudio busca cubrir estos aspectos mediante un enfoque que permita segmentar los datos de tráfico en grupos homogéneos según características específicas como franjas horarias, condiciones climáticas, y otros factores contextuales como es el clima. La segmentación adecuada de los datos permitirá identificar áreas de mayor congestión poder realizar la predicción del flujo vehicular.

La pregunta principal que se plantea es:

¿Cómo se pueden identificar y analizar los patrones de flujo vehicular en la Avenida “Las Aguas” en Guayaquil?

¿Qué tan efectivo sería un enfoque que combine el análisis de datos segmentados con la predicción del tráfico para mejorar la movilidad urbana en la avenida de interés?

¿Cómo se pueden utilizar los datos segmentados para predecir el flujo vehicular teniendo en cuenta condiciones climáticas?

1.4 Formulación del Problema

¿Cómo segmentar los datos de tráfico de la Avenida “Las Aguas” para identificar patrones que permitan predecir el flujo vehicular, utilizando un modelo híbrido basado en K-Means y Random Forest?

La segmentación de los datos en grupos homogéneos y la predicción específica dentro de esos segmentos mediante Random Forest podría mejorar las estrategias de gestión actuales.

1.5 Sistematización del Problema

- ¿Cómo segmentar los datos de tráfico en la Avenida “Las Aguas” en grupos homogéneos usando K-Means?
- ¿Qué tan efectivo es el uso de un modelo híbrido de segmentación de grupos y predicción en comparación con modelos tradicionales?
- ¿Cómo la combinación de K-Means y Random Forest puede mejorar la predicción del flujo vehicular?

1.6 Justificación e Importancia de la Investigación

La gestión del flujo vehicular es importante para mejorar la calidad de vida de los ciudadanos. De igual manera, es fundamental que las ciudades implementen medidas de control que permitan que las calles y carreteras sean un lugar seguro y eficiente no solo para los conductores, sino también para los usuarios más vulnerables, como los peatones, los ciclistas, y los motociclistas (OPS, 2022).

De acuerdo con el informe más reciente de la OMS 2023, la cifra anual de defunciones por accidentes de tránsito ha descendido ligeramente. No obstante, 1.19 millones de personas fallecen cada año por esta causa. Es por esto, que es importante desarrollar estrategias de educación vial y gestión del tráfico que promuevan un entorno más seguro y fluido.

Bajo este escenario, se pretende realizar un análisis predictivo y preventivo para determinar de manera adecuada el flujo vehicular en la Avenida “Las Aguas”. Así mismo, considerando que una gestión adecuada del tráfico puede mejorar la calidad de vida de los ciudadanos, que permitirán tener un control adecuado de estrategias que se podrían implementar. De tal manera, se abordarán ciertos conceptos y criterios que nos permitirán desarrollar el modelo predictivo.

En este contexto, se pretende hacer uso de la técnica K-Means que es un algoritmo de clustering no supervisado que nos permite dividir un conjunto de datos en grupos homogéneos, llamados clusters. El objetivo de K-Means es poder segmentar los datos del flujo vehicular de la Avenida “Las Aguas” en subsegmentaciones que representen patrones similares entre los datos. Estos patrones podrían estar relacionados con factores como la hora del día, el clima, el volumen de tráfico, o el tipo de vía.

1.6.1 Rol de K-Means

Identificación de patrones ocultos, dado que los datos de tráfico son complejos y multifacéticos, K-Means ayuda a descubrir grupos de datos con características comunes que pueden estar asociadas a un flujo vehicular elevado. Por tanto, estos grupos pueden estar relacionados por franjas horarias específicas, condiciones climáticas adversas o zonas de mayor congestión vehicular.

1.6.2 Segmentación de los Datos

Cuando aplicamos el algoritmo K-Means, los datos se dividen en un determinado K clusters. Esto nos permite segmentar las áreas y momentos más críticos en subgrupos. Por ejemplo, tener cluster con una alta densidad vehicular o con condiciones climáticas adversas. Así mismo, esta segmentación es crucial para nuestro modelo debido a que nos permite crear un modelo predictivo más preciso, debido a que no se tratan a los datos de manera homogénea, por el contrario se adaptarán los riesgos específicos de cada grupo identificado.

1.6.3 Random Forest

Random Forest es un algoritmo de aprendizaje supervisado que utiliza múltiples árboles de decisión para generar predicciones robustas dentro de nuestro modelo. Cada árbol de decisión se entrena considerando una muestra aleatoria del conjunto de datos, esto nos permite manejar la variabilidad y reducir el sobreajuste. Este algoritmo es ideal para modelos de predicción en donde existen relaciones no lineales entre las variables, como es el caso del flujo vehicular, que dependen de múltiples factores (tráfico, clima, hora del día, etc.).

1.6.4 Rol de Random Forest

Con el rol de Random Forest dentro de nuestro modelo se busca generar una predicción dentro de cada segmento. Después se entrena dentro de cada segmento (cluster) para que el modelo aprenda patrones de flujo específicos para cada tipo de contexto (por ejemplo: alta densidad vehicular, días lluviosos, días soleados, etc.). Cuando se presta atención a los subgrupos homogéneos puede mejorar significativamente la precisión de las predicciones del modelo.

1.6.5 Manejo de Variables Complejas

El flujo vehicular tiene una amplia variedad de factores que pueden modificar el tráfico, muchos de los cuales pueden no ser lineales o interdependientes entre ellos. Al utilizar el algoritmo Random Forest se puede generar múltiples árboles de decisión, que son capaces de manejar esta complejidad y la interacción entre nuestras variables. De tal forma, mejorar la precisión predictiva del modelo del flujo vehicular entre cada segmento.

1.6.6 Estimación de la Importancia de las Variables

Random Forest también nos permite medir la importancia relativa de cada variable en la predicción de nuestro Dataset. De tal forma, que es crucial para entender qué factores son los más influyentes en el flujo vehicular en cada segmento, podría ser útil para realizar recomendaciones de políticas públicas enfocadas para reducir y mejorar la calidad del tráfico

vehicular.

1.7 Impacto de la Investigación

Tener una adecuada planificación urbana para movilidad vehicular puede influir en el bienestar social de una ciudad, disminuir la contaminación de las ciudades grandes y mejorar la calidad de vida de las personas. Así mismo, el uso de algoritmos de aprendizaje automático, específicamente K-Means y Random Forest ayudará a la segmentación de grupos con características similares y una predicción de flujo vehicular adecuada. En este caso, nuestro enfoque particular es en la Avenida “Las Aguas” en la ciudad Guayaquil – Ecuador, la cual sería el área de estudio y pronostico.

El poder combinar dos algoritmos como K-Means y Random Forest nos permite abordar el problema de la gestión vehicular desde dos frentes: segmentación de datos y predicción del modelo. Primero, K-Means nos ayuda a identificar grupos homogéneos que comparten patrones de tráfico similares. Segundo, Random Forest es útil para predecir el comportamiento del flujo vehicular dentro de cada uno de esos subgrupos. De tal manera, que mejora la precisión de las predicciones del modelo al considerar las características específicas de cada subgrupo.

Previo a lo mencionado, este enfoque nos permite obtener beneficios técnicos en términos de exactitud y robustez del modelo. Además, tiene un alto potencial de impacto en el sector urbano de la Avenida “Las Aguas” que es altamente transitada. La capacidad de segmentar y predecir de esta manera permitirá presentar a las autoridades estrategias basadas en datos que ayuden al tránsito focalizado. Dentro de las posibles soluciones se podría ajustar dinámicamente los tiempos de los semáforos, redistribuir recursos de control vehicular o emitir alertas preventivas en zonas y horarios críticos. Así mismo, el modelo puede integrarse en plataformas inteligentes de tráfico y flujo vehicular, permitiendo así tomar decisiones políticas orientadas a la movilidad sostenible y la implementación de sistemas inteligentes de

transporte en ciudades de alta densidad vehicular como Guayaquil.

1.7.1 Rol Conjunto de Algoritmos

Una previa segmentación de datos mediante el uso de K-Means antes de la predicción mejora la calidad de las predicciones realizadas por Random Forest. La idea es tener un entrenamiento para cada subgrupo que podría considerar patrones subyacentes importantes, es decir, se entrena un modelo específico para cada segmento de tráfico identificado. De tal forma, que se reduce la heterogeneidad de los datos, esto implica que haciendo uso del algoritmo Random Forest pueda tener predicciones más precisas que se interpreten a la realidad.

1.7.2 Mejora en la Precisión y Robustez

Mediante una propuesta híbrida de algoritmos de segmentación y predicción el modelo puede ser más robusto que un modelo tradicional, debido a que se evita generalizar de manera errónea datos heterogéneos. En este caso, el algoritmo K-Means actúa como un previo filtro que proporciona a Random Forest datos más limpios y segmentados dependiendo el horario de tiempo y clima. Por tanto, mejora la capacidad predictiva del modelo con mayor exactitud el flujo vehicular pronosticado.

1.8 Alcance de la Investigación

En el presente proyecto se busca desarrollar un modelo híbrido de segmentación y predicción del flujo vehicular en la avenida “Las Aguas” en la ciudad de Guayaquil - Ecuador. A través de técnicas como K-Means y Random Forest, se analizarán los datos del flujo vehicular en esta zona en específico para identificar patrones y mejorar las predicciones sobre el comportamiento del tráfico vehicular en esta avenida principal de la ciudad. El alcance de este estudio se encuentra limitada a esta avenida, y los resultados obtenidos serán aplicables únicamente a este contexto específico. La avenida seleccionada se ilustra en la figura 2.

Figura 2

Ruta Seleccionada: Avenida las Aguas en Guayaquil

Fuente: (Google Maps, 2025)

1.9 Objetivo General

Desarrollar un modelo híbrido basado en K-Means y Random Forest para segmentar y predecir el flujo vehicular en la avenida “Las Aguas” en la ciudad de Guayaquil - Ecuador, considerando herramientas útiles para la toma de decisiones del flujo vehicular y la movilidad urbana.

1.10 Objetivos Específicos

- Segmentar los datos de tráfico en la Avenida “Las Aguas” mediante K-Means, identificando zonas y franjas horarias con mayor congestión vehicular, basándose en patrones de tráfico y condiciones climáticas.
- Entrenar un modelo con Random Forest dentro de cada segmento identificado, con el fin de mejorar la precisión de la predicción de siniestros viales dentro de contextos específicos.
- Evaluar el desempeño del modelo híbrido integrando las predicciones de cada segmento generado por K-Means y Random Forest, analizando su aplicabilidad en

la predicción de siniestros viales en función de las diferentes condiciones de tráfico.

Capítulo II

Revisión de Literatura

2.1 Estado del Arte

En el campo de los modelos para el tráfico, se ha notado un aumento en las investigaciones que usan una mezcla de enfoques. Específicamente, se están combinando técnicas de agrupamiento y de aprendizaje supervisado para estudiar cómo se comporta el tráfico en las zonas urbanas. Esto se hace con la finalidad de entender y predecir con mayor exactitud la movilidad de los vehículos en la ciudad.

Los modelos que se analizaron tienen un gran potencial para mejorar la gestión del tráfico inteligente. Al clasificar los flujos de vehículos (por hora o por nivel de congestión) y predecir momentos críticos, las autoridades podrían planear acciones preventivas. Esto incluye desde cambiar las fases de los semáforos hasta enviar avisos a los conductores.

Los resultados de varios estudios sugieren que una gestión basada en aprendizaje automático puede reducir las altas velocidades de los vehículos y controlar los tiempos de viaje, sin causar retrasos adicionales. Además, al identificar los puntos de alto riesgo, como cruces peligrosos o periodos pico con muchos accidentes, se facilita una mejor asignación de recursos, como más iluminación o señalización.

En general, la literatura especializada indica que integrar técnicas como K-Means y Random Forest con datos del contexto puede ofrecer un análisis técnico muy sólido. Esto, a su vez, podría traer beneficios importantes para la sociedad, como una movilidad más fluida, menos congestión y una posible reducción de accidentes en las ciudades densamente pobladas.

Por ejemplo, un estudio realizado en una ciudad de India utilizó K-Means para segmentar los datos de tráfico y luego aplicó Random Forest para predecir la congestión en cada segmento. Los resultados mostraron que el modelo híbrido podía identificar momentos

críticos de congestión en áreas específicas, lo que permitió a las autoridades implementar medidas de gestión del tráfico más efectivas (Khan et al., 2020). En Ecuador, se ha realizado un estudio sobre la "Aplicación de Técnicas de Aprendizaje Automático para la Predicción del Tráfico en Ciudades Ecuatorianas", donde se implementaron modelos híbridos que combinaron K-Means y Random Forest para predecir congestiones en tiempo real, logrando mejorar la precisión de las predicciones en un 30% en comparación con modelos tradicionales (Martínez & López, 2022).

Martínez y López (2022) también han realizado las aplicaciones técnicas de aprendizaje automático en la predicción del tráfico en ciudades ecuatorianas. En esta investigación se muestra resultados positivos en la mejora de la precisión de las predicciones, lo que sugiere que el uso de modelos híbridos puede ser una estrategia efectiva para abordar de manera afectiva la movilidad urbana en el país. Por tanto, los estudios mencionados subrayan la relevancia de la investigación en este campo y la necesidad de seguir explorando nuevas técnicas y enfoques para mejorar la calidad de vida de los ciudadanos.

A continuación, se detallan otros estudios que se han considerado como relevantes en el tema de interés.

2.1.1 Data-driven Insights: Unravelling Traffic Dynamics with K-means Clustering and Vehicle Type Differentiation

Sohail et al. (2024) aplicaron K-Means a datos reales urbanos (conteos, velocidades y tipos de vehículos) para identificar clusters de tráfico distintos como se muestra en la figura 3. El objetivo del estudio es aplicar el algoritmo de agrupamiento k-means para analizar datos de tráfico diferenciando entre tipos de vehículos, con el fin de identificar patrones de comportamiento vial y mejorar la gestión del tráfico urbano. Como resultado, lograron clasificar distintos patrones según el tipo de vehículo y el momento del día, lo que permitió detectar zonas críticas de congestión y comportamientos atípicos.

Figura 3

Desglose de Estadísticas por Clúster

Day	Cluster 0			Cluster 1		
	Traffic count	Centroid traffic count	Centroid traffic velocity	Traffic count	Centroid traffic count	Centroid traffic velocity
Monday	48	24	52	48	22	63
Tuesday	42	20	42	42	21	59
Wednesday	48	23	54	48	22	67
Thursday	42	16	43	42	15	57
Friday	42	22	56	42	22	56
Saturday	42	23	53	42	21	66
Sunday	42	25	55	42	22	68

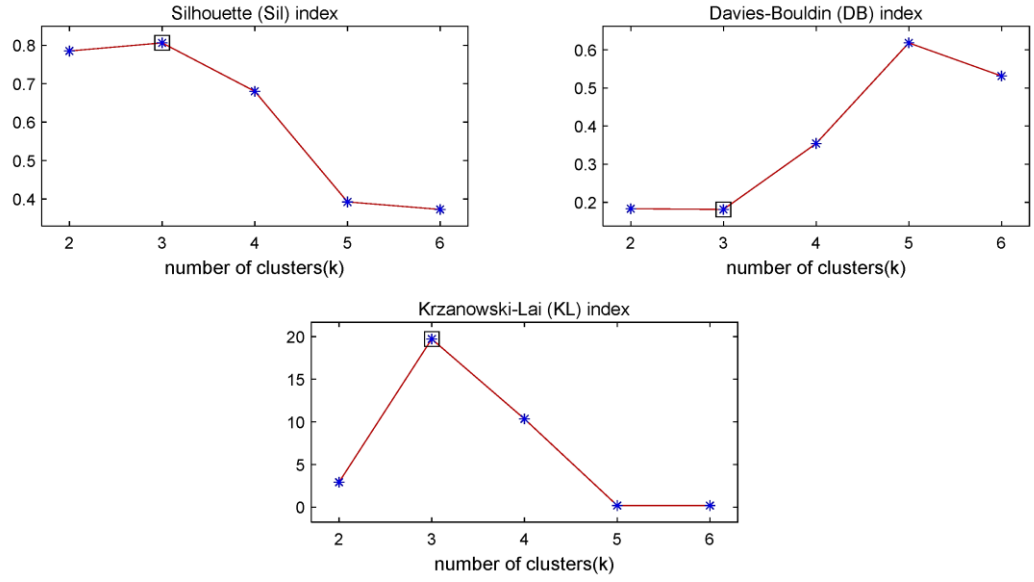
Fuente: (Sohail, 2024)

2.1.2 A Hybrid Method for Traffic State Classification Using K-Medoids Clustering and Self-Tuning Spectral Clustering

Según Qiang Shang, Yang Yu y Tian Xie (2022), un enfoque híbrido que combina el algoritmo de agrupamiento k-medoids con el clustering espectral autoajustable ayuda a mejorar la clasificación del estado del tráfico, la evaluación de clusters se puede observar en la figura 4.

Figura 4

Indicadores de Evaluación para Diferentes Números de Clústeres.



Fuente: (Shang, Yu & Xie, 2022)

Esta combinación surge como respuesta a las limitaciones de los métodos tradicionales, que suelen depender de parámetros fijos o de puntos de detección individuales, lo que reduce su aplicabilidad en entornos urbanos complejos. Los resultados, validados con datos reales de sensores en California, muestran que este enfoque supera a métodos previos en precisión y consistencia, destacando su capacidad para adaptarse a múltiples escalas.

2.1.3 Activity Pattern Generation Using Machine Learning Techniques for Iransport Demand Models

Ajala (2024) comparó Random Forest, árboles de decisión y redes neuronales en clasificación de actividades de viaje en Cuenca-Ecuador. El objetivo principal del estudio es desarrollar una metodología para entrenar y evaluar técnicas de aprendizaje automático aplicadas a la generación de patrones de actividad, con el fin de integrarlos en modelos de demanda de transporte. Para ello, se plantean cuatro tareas de clasificación: elección del medio de transporte, motivo del desplazamiento, destino y hora de inicio del viaje. En la figura 5 se puede observar el rendimiento de cada modelo propuesto por el autor basado en la precisión en el tiempo de inicio de la clasificación de desplazamiento. Concluyó que el mejor desempeño lo obtuvo Random Forest en todas las tareas de clasificación (transporte, destino, motivo, hora).

Figura 5

Comparación de Rendimiento Basado en la Clasificación de Desplazamiento

Model	Accuracy
Random Forest Classifier using GSCV	0.7548
Random Forest Classifier using RSCV	**0.8010**
Decision Tree Classifier using GSCV	0.7450
Decision Tree Classifier using RSCV	0.7983
Artificial Neural Network	0.7194

Fuente: (Ajala, 2024)

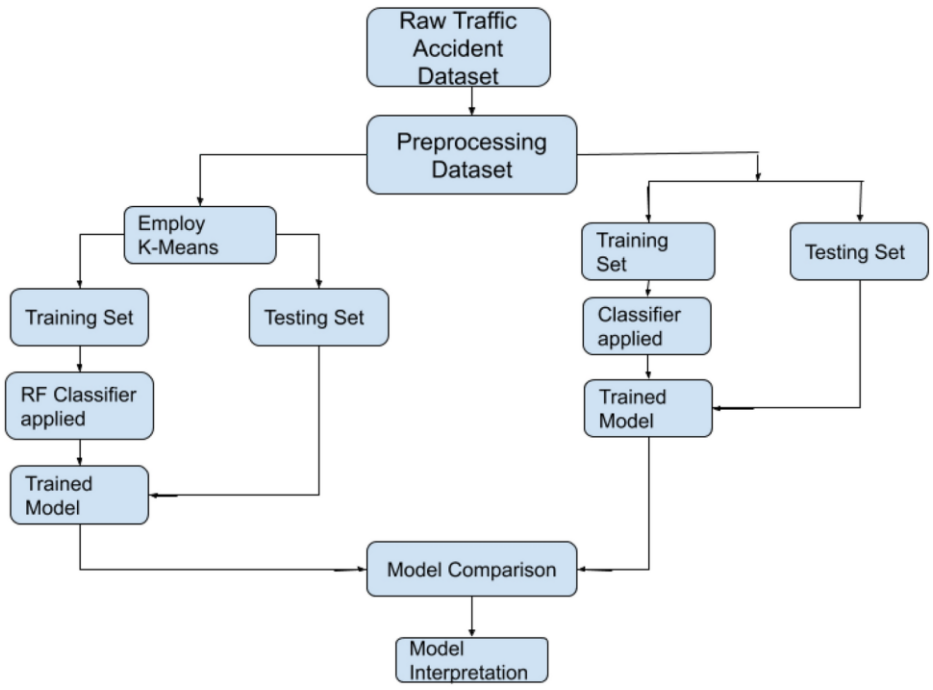
2.1.4 Road Accident Prediction and Model Interpretation using a Hybrid K-means and Random Forest Algorithm Approach

Yassin y Pooja (2020) propusieron exactamente este flujo para predecir la severidad de accidentes viales: usaron K-Means para extraer información oculta (creando una nueva variable de distancia-cluster óptima) y luego Random Forest para clasificar la gravedad del accidente. El modelo híbrido propuesto demostró una precisión del 99.86 %, superando significativamente a otros métodos tradicionales como SVM, KNN y regresión logística. El modelo propuesto por los autores se puede observar en la figura 6.

Además, permitió identificar factores clave que contribuyen a la severidad de los accidentes, como la experiencia del conductor, la edad, las condiciones de luz y los años de servicio del vehículo. Estos hallazgos no solo validan la eficacia del enfoque híbrido, sino que también ofrecen información valiosa para agencias de transporte y aseguradoras interesadas en diseñar estrategias de seguridad vial más efectivas.

Figura 6

Diagrama de Flujo del Modelo Propuesto para Predecir Accidentes de Tráfico



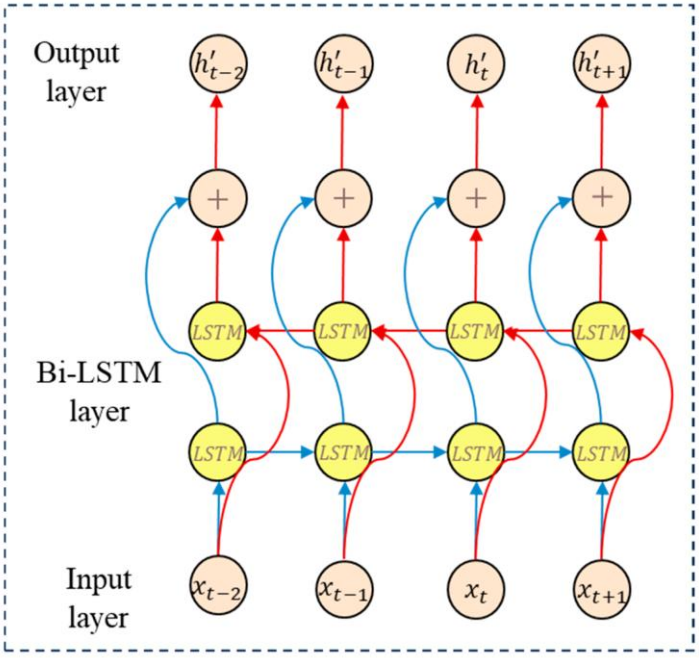
Fuente: (Yassin & Pooja, 2020)

2.1.5 Short-Term Traffic Flow Prediction Considering Weather Factors Based on Optimized Deep Learning Neural Networks: Bo-GRA-CNN-BiLSTM

Wang et al. (2025) muestran que incluir variables como la temperatura, precipitaciones, visibilidad, calidad del aire, mejora sustancialmente la predicción del flujo vehicular. Han demostrado que un modelo de CNN+BiLSTM optimizado con análisis relacional gris que incorpora múltiples factores climáticos, el modelo BiLSTM usado en su estudio se muestra en la figura 7, reduce los errores de predicción comparado con modelos sin datos meteorológicos. Esto evidencia que datos meteorológicos no deben omitirse: su integración como features pesados permite capturar las tendencias integradas entre tráfico y clima, mejorando la precisión predictiva.

Figura 7

Estructura de Red BiLSTM.



Fuente: (Wang et al, 2025)

2.2 Movilidad Urbana

La movilidad urbana se define como la capacidad de las personas para moverse dentro de un entorno urbano utilizando diversos modos de transporte, incluyendo automóviles,

bicicletas, transporte público y caminatas. El aumento del tráfico vehicular y la necesidad de una gestión eficiente para mejorar la calidad de vida de los ciudadanos es un tema importante a tener en consideración para la planificación de ciudades modernas. Tener un adecuado control de movilidad urbana es esencial no solo para reducir la congestión del tráfico, sino también para mejorar la calidad del aire, promover la sostenibilidad ambiental y aumentar la calidad de vida de los ciudadanos (Martínez & López, 2022).

El concepto de movilidad urbana considera varios aspectos, dentro de los cuales incluye a la accesibilidad, conectividad y sostenibilidad. La accesibilidad se refiere a la facilidad con la que los individuos pueden llegar a sus destinos, mientras que la conectividad implica la interrelación entre diferentes modos de transporte. La sostenibilidad, por su parte, se refiere a la capacidad de satisfacer las necesidades actuales sin comprometer la capacidad de las futuras generaciones para satisfacer sus propias necesidades (Litman, 2020). Por lo tanto, la movilidad urbana debe ser vista como un sistema complejo que requiere un enfoque multidisciplinario para su análisis y mejora.

Entre los desafíos en la movilidad urbana los más comunes son la congestión del tráfico, la infraestructura inadecuada, la contaminación del aire y la seguridad vial. La congestión del tráfico es uno de los problemas más visibles y frustrantes para los ciudadanos, ya que puede resultar en pérdidas de tiempo significativas y un aumento del estrés (World Health Organization, 2021). De igual manera, la congestión contribuye a la contaminación del aire, lo que representa un riesgo para la salud pública. Según la Organización Mundial de la Salud (2021).

2.3 Python

Es catalogado como lenguaje de programación con características de alto nivel, interpretado y de propósito general, de autoría de Guido van Rossum. Soporta paradigmas como programación orientada a objetos y funcional. Incluye bibliotecas como NumPy,

Pandas y Scikit-learn entre otras. Su sintaxis clara y legible lo hace ideal para ciencia de datos, aprendizaje automático, desarrollo web y automatización. (Van Rossum & Drake, 2009).

Este lenguaje de programación es considerado fundamental en la ciencia de datos, teniendo como sus principales fortalezas su simplicidad y la disponibilidad de bibliotecas especializadas como NumPy y Pandas. Su practicidad y flexibilidad facilita a los investigadores el poder desarrollar soluciones adaptables y personalizadas, teniendo como principal limitante el rendimiento en aplicaciones de alta computación.

2.4 NumPy

Es la biblioteca de Python para realizar cálculos numéricos, cuenta con soporte para arreglos multidimensionales y diferentes funciones matemáticas optimizadas. Es fundamental para procesar datos numéricos en investigación. Permite operaciones como álgebra lineal y estadísticas con alta eficiencia gracias a su implementación en C. Es la base de bibliotecas como Pandas y Scikit-learn. (Harris et al., 2020).

La biblioteca de Python, NumPy, es la base sobre la cual se han creado muchas bibliotecas de ciencia de datos, todo esto gracias a su capacidad para realizar operaciones vectorizadas, obteniendo como resultado la reducción en el tiempo de cómputo. Su implementación es crítica en la preparación de los datos para modelos de aprendizaje automático, todo lo anteriormente descrito teniendo como consideración esencial que se debe poseer conocimientos previos de álgebra lineal para su uso eficiente.

2.5 Pandas

Es una biblioteca de Python para manipulación y análisis de datos, la cual se encuentra fundamentada sobre las bases de la biblioteca NumPy. Soporta filtrado, agrupación, fusión y manejo de datos faltantes como características. Proporciona estructuras como DataFrames y Series para el manejo de datos tabulares y series temporales. Es ideal

para datos heterogéneos de bases de datos o CSV (McKinney, 2010).

Pandas es una de las bibliotecas de Python indispensable para la limpieza y exploración de datos para el correcto análisis de los mismos con uso aplicado en investigaciones científicas. Entre sus principales fortalezas se tienen su interfaz intuitiva, siendo ideal para manejar datos heterogéneos, sin embargo, en contraparte, puede ser menos eficiente con conjuntos de datos masivos, bajo este último escenario se tienen alternativas como Polars más adecuadas para este tipo de datos.

2.6 DataFrames

Es una estructura bidimensional en Pandas, similar a una tabla de base de datos. Es útil para limpieza y análisis de datos. Organiza datos en filas y columnas etiquetadas, permitiendo manipular datos de diferentes tipos. Se integra con otras bibliotecas de Python (McKinney, 2017).

Los DataFrames son una estructura primordial y clave para la facilidad en la manipulación de datos en entornos académicos, todo esto gracias a su similitud con herramientas de uso conocido como Excel o SQL. Es necesario remarcar que en su manejo es requerido cuidado en términos de memoria al realizar trabajos con datasets grandes.

2.7 Matplotlib

Es una biblioteca de Python con enfoque esencial en visualización de datos, que permite crear gráficos estáticos, interactivos y animados. Ofrece una alta personalización para visualizaciones científicas. Permite generar, entre otros, histogramas, gráficos de líneas y diagramas de dispersión. Es la base fundamental de otras herramientas como Seaborn (Hunter, 2007).

La biblioteca de Python Matplotlib es una herramienta poderosa para la creación y visualización, gracias a su alta personalización. Una de sus principales limitantes al momento de su uso es su sintaxis, la cual puede llegar a ser compleja para principiantes, lo

anteriormente mencionado es lo que ha permitido el impulso del uso de otras bibliotecas como Seaborn para tareas más comunes.

2.8 Seaborn

Es una biblioteca de visualización basada en Matplotlib, su enfoque se encuentra centrado principalmente en gráficos estadísticos. Requiere menos código que Matplotlib facilitando su uso para principiantes. Facilita la creación de mapas de calor, gráficos de violín y diagramas de dispersión. Es ideal para análisis exploratorios (Waskom, 2021).

Seaborn tiene como base fundamental su simplicidad en la creación de visualizaciones estadísticas, bajo este concepto su uso es ideal para análisis exploratorios bajo visualización. Su dependencia de Matplotlib en su concepción, permite personalizaciones avanzadas, sin embargo su uso puede limitarse al tratarse de gráficos no estadísticos.

2.9 Bases de datos

Es un sistema organizado para almacenar, gestionar y recuperar datos. Es considerado esencial para el manejo de grandes volúmenes de datos. Puede ser de características relacional (MySQL) o no relacional (MongoDB), usando lenguajes como SQL. Soporta operaciones de consulta y análisis (Elmasri & Navathe, 2015).

Las bases de datos son consideradas esenciales para el correcto manejo de grandes volúmenes de datos en investigaciones. La correcta elección entre las variables en modelos relacionales y no relacionales va a depender básicamente de la estructura de los datos que se manejaran, siendo las bases relacionales las más adecuadas para datos estructurados.

2.10 API

Una API (Interfaz de Programación de Aplicaciones) es un conjunto de reglas para la comunicación entre sistemas. Es considerado clave para la integración de datos externos en proyectos. Facilita el intercambio de datos o funcionalidades entre aplicaciones. Se usa ampliamente en ciencia de datos (Fielding, 2000).

Las APIs son esenciales en el enfoque de integración de datos externos en diferentes proyectos como de investigación, datos tales como datos en tiempo real. El realizar un correcto diseño es primordial con el enfoque de garantizar escalabilidad y, principalmente, seguridad en aplicaciones de machine learning.

2.11 JSON

JSON (JavaScript Object Notation) es en esencia un formato ligero enfocado en el intercambio de datos, cuya base fundamental son pares clave-valor. Es compatible con múltiples lenguajes. Es bastamente usado en APIs y almacenamiento de datos semiestructurados. Es fácil de leer y escribir (Bray, 2014).

JSON en concepción es considerado ideal para el manejo de datos semiestructurados en proyectos, principalmente, de ciencia de datos fundamentalmente debido a su simplicidad y compatibilidad con diferentes lenguajes. Su principalmente limitante recae en su reducidad eficiencia para datos binarios o muy grandes.

2.12 Joblib

Es una biblioteca de Python con uso en serialización eficiente de objetos y ejecución paralela. Optimiza flujos de trabajo en ciencia de datos. Su uso se fundamenta en guardar y cargar modelos de machine learning. Es compatible con Scikit-learn (Varoquaux et al., 2020).

Joblib es la biblioteca de Python valiosa en el apartado de la optimización de flujos de trabajo en machine learning, con principal énfasis en el apartado de guardar modelos grandes. Su facilidad en el manejo lo convierte en el aliado ideal para investigadores, sin embargo su dependencia en el uso de pickle puede llegar a plantear posibles riesgos de seguridad.

2.13 Os

Es un módulo de Python que permite la interacción con el sistema operativo. Faculta el gestionar archivos, directorios y variables de entorno. Su utilidad se ve plasmada en automatizar tareas en proyectos de datos. Es parte de la biblioteca estándar de Python (Van

Rossum & Drake, 2009).

El módulo OS es fundamental y esencial para la automatización de tareas de gestión de archivos en proyectos de datos, sin embargo al usarse se debe tener cuidado con el fin de garantizar la portabilidad entre diferentes sistemas operativos.

2.14 Pydantic

Es una biblioteca de Python para validación de datos y gestión de configuraciones. Su uso es común en APIs y aplicaciones modernas. Usa anotaciones de tipo para garantizar integridad de datos. Se integra con FastAPI (King, 2022).

Pydantic es fundamental en el uso de aplicaciones que tienen como requerimiento datos validados, como APIs. Su integración y complemento con FastAPI optimiza la robustez de los sistemas, la curva de aprendizaje inicial puede llegar a ser un desafío para principiantes.

2.15 FastAPI

Es un framework de Python para construir APIs web rápidas. Su eficiencia se ve reflejada en aplicaciones de machine learning. Basado en OpenAPI y JSON Schema, soporta tipado y validación automática. Usa programación asíncrona (Ramírez, 2022).

Es considerado ideal para el desarrollo de APIs en proyectos de machine learning gracias a su velocidad y facilidad de uso. Su integración con Pydantic permite garantizar datos consistentes, sin embargo, es necesario poseer conocimientos de programación asíncrona.

2.16 PCA

PCA (Análisis de Componentes Principales) es una técnica de reducción de dimensionalidad. Tiene como objetivo el facilitar la visualización y modelado en datasets complejos. Transforma datos en un nuevo sistema de coordenadas, maximizando la varianza. Es común en ciencia de datos (Jolliffe, 2002).

Es considerado fundamental en el apartado de reducir la complejidad en datasets de alta dimensionalidad, ayudando a la visualización y la facilidad en el modelado. Como principal desventaja que se podría generar se encuentra la posibilidad de perder información interpretativa al revisar datos en dimensiones menores.

2.17 KMeans

Es un algoritmo de clustering no supervisado que permite la agrupación de datos en k clústeres. Es ampliamente usado en análisis exploratorios. Entre sus inputs se requiere especificar el número de clústeres (MacQueen, 1967).

Es muy utilizado con el fin mayor de poder segmentar datos, sin embargo su sensibilidad a la inicialización de centroides y el tener que asignar como input la correcta asignación de k puede limitar su aplicabilidad.

2.18 Random Forest Classifier

Es un algoritmo supervisado que combina múltiples árboles de decisión. Es robusto para tareas de clasificación. Entre sus principales características se tiene que mejora la precisión y reduce el sobreajuste. Se usa en Scikit-learn (Breiman, 2001).

Es considerado, entre sus principales fortalezas, robusto y versátil enfocado en tareas de clasificación, sin embargo en contraparte su interpretabilidad puede llegar a ser limitada y con la posibilidad de ser computacionalmente costoso en datasets grandes.

2.19 Scikit-learn

Es una biblioteca de Python para aprendizaje automático. Es accesible y ampliamente usada en investigación. Ofrece herramientas para preprocesamiento, modelado, evaluación y optimización de algoritmos. Soporta múltiples algoritmos (Pedregosa et al., 2011).

Scikit-learn es una herramienta integral gracias a su vasta gama de algoritmos y principalmente a su facilidad de uso. Como principal contraparte se puede mencionar que no está totalmente optimizada para grandes volúmenes de datos.

2.19.1 PowerTransformer

Es una clase de Scikit-learn que aplica transformaciones de potencia (Box-Cox o Yeo-Johnson). Mejora el rendimiento de modelos de machine learning. Normaliza datos no gaussianos y estabiliza la varianza. Es parte del preprocesamiento de datos (Pedregosa et al., 2011).

Su principal utilidad se encuentra basada en el preprocesar datos en modelos de machine learning, consiguiendo de esta manera el mejorar su rendimiento al normalizar distribuciones. Sin embargo, su efectividad se basa en la naturaleza de los datos y requiere pruebas cuidadosas.

2.19.2 Classification_report

Es una función de la biblioteca Scikit-learn que genera un informe de métricas. Evalúa el rendimiento del modelo. Incluye entre sus métricas de evaluación precisión, recall y F1-score para problemas de clasificación. Es útil para análisis detallados (Pedregosa et al., 2011).

Esta función es fundamental para la evaluación de modelos de clasificación en un marco integral, sin embargo requiere interpretación cuidadosa en clases desbalanceadas.

2.19.3 Confusion_matrix

La matriz de confusión muestra las predicciones correctas e incorrectas de un modelo de clasificación. Organiza los resultados por clase para evaluar el rendimiento. Es una herramienta clave en machine learning. Se usa en Scikit-learn (Fawcett, 2006).

La matriz de confusión es una herramienta clave para entender errores específicos del modelo, sin embargo en contraparte su interpretación puede llegar a ser compleja en problemas multiclase.

2.19.4 Accuracy_score

Mide la proporción de predicciones correctas en un modelo de clasificación. Es una

métrica simple para evaluar rendimiento. Se implementa en Scikit-learn. Es útil en problemas balanceados (Pedregosa et al., 2011).

Es una métrica simple y útil, pero puede ser engañosa en Datasets desbalanceados, donde métricas como F1-score son más informativas.

2.19.5 Precision_score

Mide la proporción de predicciones positivas correctas en clasificación. Es crucial cuando los falsos positivos son costosos. Se implementa en Scikit-learn. Evalúa la calidad de las predicciones (Pedregosa et al., 2011).

La precisión es crítica en aplicaciones donde los falsos positivos son costosos, pero debe complementarse con otras métricas para una evaluación completa.

2.19.6 Recall_score

Mide la proporción de casos positivos reales identificados correctamente. Es vital cuando los falsos negativos son críticos. Se usa en Scikit-learn. Evalúa la sensibilidad del modelo (Pedregosa et al., 2011).

El recall es vital en problemas donde los falsos negativos son críticos, como en diagnósticos médicos, pero puede sacrificar precisión.

2.19.7 F1_score

Es la media armónica de precisión y recall en clasificación. Proporciona un balance entre ambas métricas. Es útil en Dataset desbalanceados. Se implementa en Scikit-learn (Pedregosa et al., 2011).

F1-score es ideal para evaluar modelos en Dataset desbalanceados, pero no captura todos los aspectos del rendimiento del modelo.

2.19.8 R2_score

Mide la proporción de varianza explicada por un modelo de regresión. Evalúa la calidad del ajuste del modelo. Es común en Scikit-learn. Es útil para problemas lineales

(Pedregosa et al., 2011).

R2_score es útil para evaluar modelos de regresión, pero puede ser engañoso en modelos no lineales o con datos ruidosos.

2.19.9 Train_test_split

Es una función de Scikit-learn que divide datos en conjuntos de entrenamiento y prueba. Evita el sobreajuste al evaluar modelos. Permite especificar proporciones de división. Es esencial en machine learning (Pedregosa et al., 2011).

Esta función es esencial para evitar el sobreajuste, pero la elección de proporciones y la aleatoriedad deben manejarse cuidadosamente.

2.20 Conjunto de Entrenamiento

El conjunto de entrenamiento es la porción de datos usada para entrenar un modelo de machine learning. Ajusta los parámetros del modelo para aprender patrones. Es crucial para la generalización. Se usa con train_test_split (Goodfellow et al., 2016).

Un conjunto de entrenamiento representativo es considerado clave para poder generalizar el modelo, sin embargo el contar con datos sesgados o insuficientes pueden llegar a limitar su eficacia.

2.21 Conjunto de Validación

El conjunto de validación es un subconjunto de datos para ajustar hiperparámetros. Sirve principalmente para optimizar el rendimiento del modelo. Tiene como objetivo principal el evaluar el modelo durante el entrenamiento sin sesgar la prueba final. Se usa en machine learning (Goodfellow et al., 2016).

Este conjunto es fundamental con el objetivo de optimizar modelos sin llegar a sesgar la evaluación final, sin embargo su tamaño debe balancearse correctamente con el conjunto de entrenamiento.

2.22 Conjunto de Prueba

El conjunto de prueba evalúa el rendimiento final de un modelo de machine learning. Tiene como objetivo el garantizar una evaluación objetiva. Es un subconjunto reservado, no usado en entrenamiento. Es esencial para validar generalización (Goodfellow et al., 2016).

El conjunto de prueba nos otorga como resultado una evaluación objetiva, sin embargo debe ser representativo y no debe ser utilizado durante el entrenamiento para evitar posibles sesgos.

2.23 Machine Learning

El aprendizaje automático (machine learning) es una rama de la inteligencia artificial. Se usa en clasificación, regresión y clustering. Permite a los sistemas aprender de datos sin programación explícita. Transforma la investigación de datos (Mitchell, 1997).

El machine learning transforma la investigación al conseguir automatizar el descubrimiento de patrones, sin embargo su éxito en la implementación depende principalmente de datos de calidad y algoritmos adecuados.

2.24 Tipos de Algoritmos de Machine Learning

Los algoritmos de machine learning se dividen en supervisados, no supervisados, semi-supervisados y por refuerzo. Incluyen métodos, entre otros, como regresión y clustering. Cada tipo aborda diferentes problemas según los datos disponibles. Son clave en ciencia de datos (Bishop, 2006).

La diversidad de algoritmos da la facultad de abordar problemas variados, sin embargo es crucial la elección del tipo adecuado, para dicha elección se requiere un entendimiento profundo del problema y los datos.

2.24.1 Aprendizaje Supervisado

El aprendizaje supervisado usa datos etiquetados para entrenar modelos. Es común en problemas con datos anotados. Predice variables objetivo en tareas como clasificación y

regresión. Se implementa en Scikit-learn (Bishop, 2006).

Es ideal para problemas con datos etiquetados, pero su dependencia en la asignación de etiquetas de calidad puede llegar a ser un desafío en aplicaciones reales.

2.24.2 Aprendizaje no Supervisado

El aprendizaje no supervisado tiene como objetivo el identificar patrones en datos no etiquetados. Es principalmente útil para análisis exploratorios. Incluye técnicas como clustering y reducción de dimensionalidad. No requiere etiquetas previas (Bishop, 2006).

Es en esencia útil para exploración de datos, como principal punto a considerar se tiene que la falta de etiquetas dificulta la interpretación de los resultados.

2.24.3 Aprendizaje Semi-supervisado

El aprendizaje semi-supervisado combina datos etiquetados y no etiquetados. Es fundamentalmente útil en escenarios con datos escasos. Mejora el rendimiento cuando las etiquetas son limitadas. Combina técnicas supervisadas y no supervisadas (Chapelle et al., 2006).

Este tipo de aprendizaje es valioso en escenarios cuando se cuenta con datos escasos, su implementación puede llegar a ser compleja en parte debido a la necesidad de balancear ambos tipos de datos.

2.24.4 Aprendizaje por Refuerzo

El aprendizaje por refuerzo entrena agentes para tomar decisiones secuenciales. Es comúnmente usado en robótica y juegos. Maximiza una recompensa en un entorno dinámico. Requiere definición de recompensas (Sutton & Barto, 2018).

Es un tipo de aprendizaje poderoso principalmente para problemas de optimización dinámica, sin embargo su entrenamiento puede llegar a ser computacionalmente intensivo y sensible esencialmente a la definición de recompensas.

2.25 Hiperparámetros del Algoritmo de Aprendizaje

Los hiperparámetros son parámetros configurables antes del entrenamiento. Incluyen n_estimators y max_depth. Controlan principalmente el comportamiento y rendimiento del modelo. Su optimización es clave en machine learning (Goodfellow et al., 2016).

La optimización de hiperparámetros es fundamental enfocado en el poder maximizar el rendimiento, pero puede llegar a ser un proceso costoso que requiere técnicas como por ejemplo búsqueda en cuadrícula o bayesiana.

2.25.1 N_estimators

Es un hiperparámetro en algoritmos como RandomForest. Aumenta la robustez del modelo. Define en concepto el número de árboles de decisión en el ensamblado. Se usa en Scikit-learn (Breiman, 2001).

Este hiperparametro es un importante dado que un mayor número de árboles mejora la robustez, pero en contraparte aumenta el costo computacional, en base a lo indicado se requiere un balance cuidadoso.

2.25.2 Max_depth

Es un hiperparámetro que limita la profundidad de árboles en algoritmos como RandomForest. Es fundamental para conseguir balancear la complejidad. Controla el sobreajuste del modelo. Se usa en Scikit-learn (Breiman, 2001).

Limitar el hiperparámetro max_depth tiene como objetivo prevenir el sobreajuste, sin embargo valores demasiado bajos tienden a reducir finalmente la capacidad predictiva del modelo.

2.25.3 Class_weight

Es un hiperparámetro que asigna pesos a clases en clasificación. Maneja desbalanceos en los datos. Mejora el rendimiento en clases minoritarias. Se implementa en Scikit-learn (Pedregosa et al., 2011).

Es útil para mejorar el rendimiento en clases minoritarias, pero su ajuste requiere un entendimiento claro de las prioridades del problema.

Capítulo III

Desarrollo

3.1 Metodología

Para el desarrollo del presente estudio, se llevó a cabo un proceso sistemático de recopilación y análisis de datos enfocados en el comportamiento del tráfico vial en la Avenida Las Aguas-Guayaquil.

Esta investigación se desarrolló bajo un enfoque cuantitativo, complementado con elementos cualitativos, lo cual cubre la necesidad de comprender los patrones numéricos de tráfico y el contexto físico correspondiente a las variables climáticas de este entorno vial. Se aplicó un diseño no experimental, transversal y explicativo, mediante el uso de técnicas algoritmos de aprendizaje automático.

Para el presente estudio, la segmentación de grupos del tráfico es útil el algoritmo K-Means, debido a que se utiliza para identificar áreas con alta congestión y patrones de tráfico. Por ejemplo, se puede segmentar una ciudad en diferentes grupos según el volumen de tráfico, la velocidad promedio y otras variables relevantes. Por medio de esta segmentación es que se puede tomar decisiones informadas sobre cómo optimizar la infraestructura y los servicios de transporte. De tal manera, el algoritmo K-Means ayuda a identificar patrones en los que se producen un aumento de flujo vehicular, lo cual es crucial para la planificación de semáforos y la gestión de flujos vehiculares (Kumar & Singh, 2020).

El algoritmo K-Means se usa como una técnica de agrupación ampliamente utilizado en el análisis de datos que tiene como objetivo dividir un conjunto de datos en K grupos basados en características similares (MacQueen, 1967). Este algoritmo es particularmente útil para la movilidad urbana, debido a que permite segmentar grupos de tráfico en función de los patrones de comportamiento de los ciudadanos y de las condiciones climáticas. El funcionamiento de este algoritmo implica la asignación de puntos de datos a grupos en

función de la distancia a los centroides de cada grupo. Este proceso se repite iterativamente hasta que los centroides no cambian significativamente, lo que indica que se ha alcanzado una convergencia (Hastie, Tibshirani, & Friedman, 2009).

Una de las principales ventajas de aplicar el algoritmo K-Means es su simplicidad y rapidez en comparación con otros algoritmos de agrupamiento. Sin embargo, también tiene limitaciones, como es la necesidad de especificar el número de grupos de antemano y su sensibilidad a los valores atípicos (Jain, 2010). No obstante, de estas limitaciones el algoritmo K-Means sigue siendo una herramienta valiosa para la segmentación de datos para determinar una segmentación adecuada del flujo vehicular.

De igual manera para la predicción del flujo vehicular, el algoritmo Random Forest se ha utilizado para estimar las condiciones del flujo vehicular, la velocidad del tráfico y la congestión en tiempo real (Ali et al., 2021). Por ejemplo, un estudio realizado en una ciudad de Estados Unidos demostró que Random Forest superó a otros modelos tradicionales en la predicción de la congestión del tráfico, logrando una mejora del 25% en la precisión de las predicciones (Chen et al., 2019). Este tipo de resultados resalta la eficacia de Random Forest como herramienta para la gestión del tráfico en sectores urbanos.

Por tanto, se considera Random Forest porque es un algoritmo de aprendizaje supervisado que se utiliza para múltiples árboles de decisión que nos permiten realizar predicciones. Este algoritmo se caracteriza por su robustez frente al sobreajuste y su capacidad para manejar grandes conjuntos de datos con múltiples variables (Breiman, 2001). Así mismo, Random Forest funciona creando un bosque de árboles de decisión. En donde, cada árbol se entrena utilizando una muestra aleatoria de los datos y una selección aleatoria de características. Las predicciones se realizan promediando las predicciones de todos los árboles en el bosque, lo que mejora la precisión general del modelo.

Así mismo, Random Forest tiene la capacidad para manejar datos faltantes y su

resistencia al ruido, lo que lo convierte en una opción ideal para el análisis del flujo vehicular, donde los datos pueden ser inconsistentes (Zhou & Troyanskaya, 2005). De igual manera, el algoritmo Random Forest proporciona una medida de la importancia de las características, lo que permite a los analistas identificar qué variables tienen el mayor impacto en las predicciones. Esto es relativamente útil para la movilidad urbana, donde múltiples factores pueden influir en el flujo vehicular, como las condiciones meteorológicas, el tiempo del día y condiciones adversas de la vía.

Al combinar el algoritmo K-Means y Random Forest ayuda a mejorar la precisión de las predicciones. Al segmentar los datos en grupos homogéneos, se reduce la variabilidad en los datos, lo que permite a Random Forest hacer predicciones más robustas (Hodge & Austin, 2004). Por ejemplo, al analizar el flujo vehicular en una ciudad, K-Means puede identificar diferentes patrones de tráfico en áreas urbanas y rurales, residenciales, comerciales e industriales. Esta segmentación permite a Random Forest realizar predicciones más precisas al tener en cuenta las características específicas de cada segmento.

El poder realizar una combinación de algoritmos como K-Means y Random Forest permite un enfoque más efectivo para la segmentación y predicción del tráfico. Al segmentar los datos primero con K-Means, luego Random Forest puede hacer predicciones más precisas al trabajar con grupos homogéneos (Sharma & Singh, 2019). La combinación de estos algoritmos facilita la identificación de patrones específicos de datos del tráfico que pueden no ser evidentes en un análisis global. Este enfoque híbrido se ha convertido en una práctica común en el análisis de datos de tráfico, debido a que permite a los analistas aprovechar las fortalezas de ambos algoritmos.

3.2 Fuentes de Datos

La recolección de datos incluyó fuentes oficiales. Se ocupo los datos de la Autoridad de Tránsito Municipal (ATM). Adicionalmente, se integraron registros climáticos históricos,

obtenidos de plataformas meteorológicas confiables como NOAA Climate Data Online u Open Weather.

En la figura 8 se presenta el flujo promedio de automotores diarios para el año 2024 en la Avenida Las Aguas. La información se desglosa detalladamente en dos secciones: la primera muestra el promedio de vehículos por tramo de hora, diferenciando las direcciones Norte-Sur y Sur-Norte, así como los días laborables (lunes a viernes) de los fines de semana. La segunda sección resume el promedio total de vehículos por día para los mismos periodos. Este conjunto de datos sirve como una fuente de información clave para comprender los patrones de tráfico y movilidad en la mencionada avenida, mostrando variaciones significativas entre las horas pico y el resto del día, además de las diferencias entre la semana y el fin de semana.

Figura 8

Flujo Promedio de Vehículos Diarios

FLUJO PROMEDIO DE AUTOMOTORES DIARIOS (AÑO 2024)

Av. Las Aguas

Tramo de hora	Promedio por tramo de hora			
	Lunes a Viernes		Sábado y Domingo	
	NORTE-SUR	SUR-NORTE	NORTE-SUR	SUR-NORTE
00:00 - 01:00	192	193	434	437
01:00 - 02:00	100	87	276	277
02:00 - 03:00	52	52	196	208
03:00 - 04:00	39	47	144	164
04:00 - 05:00	59	71	119	143
05:00 - 06:00	239	177	161	147
06:00 - 07:00	1 356	906	454	321
07:00 - 08:00	2 181	1 761	859	617
08:00 - 09:00	2 387	1 857	1 159	872
09:00 - 10:00	1 965	1 577	1 359	1 071
10:00 - 11:00	1 729	1 490	1 461	1 188
11:00 - 12:00	1 681	1 535	1 486	1 300
12:00 - 13:00	1 827	1 716	1 491	1 358
13:00 - 14:00	1 825	1 716	1 525	1 441
14:00 - 15:00	1 837	1 708	1 469	1 391
15:00 - 16:00	1 830	1 692	1 398	1 322
16:00 - 17:00	1 800	1 777	1 368	1 289
17:00 - 18:00	2 061	2 048	1 370	1 289
18:00 - 19:00	1 929	1 988	1 333	1 242
19:00 - 20:00	1 501	1 511	1 317	1 206
20:00 - 21:00	1 217	1 184	1 174	1 072
21:00 - 22:00	1 082	1 041	1 054	971
22:00 - 23:00	764	797	819	792
23:00 - 00:00	442	473	533	554

Promedio por día		
Sector referencial	Lunes a Viernes	Sábado a Domingo
NS_Av Las Aguas Sector Delfos	30 095	23 026
SN_Av Las Aguas Sector Delfos	27 405	20 726

Fuente: (ATM,2025)

Esta información, que proviene de la ATM, es la base que se utilizará. A partir de ella,

se generarán más datos para entrenar los modelos que se van a desarrollar.

3.3 Preprocesamiento y Selección de Datasets

La información recolectada fue organizada en una base de datos estructurada, a partir de la cual se definieron los datasets de entrada para el modelo híbrido propuesto.

3.3.1 Dataset de Segmentación (K-Means)

Incluyó variables como: hora del día, volumen vehicular, tipo de vehículo, velocidad promedio, condiciones climáticas y tipo de día (laboral o fin de semana). Este conjunto fue utilizado para identificar patrones y clústeres representativos del comportamiento del tráfico.

3.3.2 Dataset de Predicción (Random Forest)

A partir de la segmentación anterior, se seleccionaron subconjuntos etiquetados con niveles de riesgo o congestión para alimentar el modelo supervisado. Se incluyeron como variables predictoras: número de vehículos, tiempo promedio de cruce, número de incidentes reportados por hora, intensidad del tráfico y saturación por carril.

El preprocesamiento incluyó limpieza de datos, imputación de valores faltantes, codificación de variables categóricas y normalización, siguiendo buenas prácticas de ciencia de datos para garantizar consistencia y confiabilidad de los modelos construidos.

3.4 Estrategia de Predicción del Flujo Vehicular

Debido a la naturaleza dinámica y no lineal de los datos tanto del flujo vehicular como de las condiciones climáticas se ha decidido dividir la manera en la que se clasificará el flujo vehicular. Teniendo en cuenta que el flujo vehicular está compuesto netamente de valores numéricos, es necesario un algoritmo de agrupamiento para categorizar los valores que contengan valores similares. Por lo tanto, el algoritmo Kmeans ha sido seleccionado para realizar la tarea de agrupamiento y categorización de estos numéricos.

Dado que el resultado del algoritmo Kmeans puede asignar una etiqueta textual a los valores del flujo vehicular, esta salida será de utilidad para el entrenamiento de un modelo de

clasificación que será utilizado para predecir el flujo vehicular de manera textual.

3.5 Modelo de Agrupamiento

El modelo agrupamiento seleccionado ha sido Kmeans debido a sus múltiples beneficios. Como bien lo menciona Ikotun (2023), esto debido a su a amplia aplicabilidad del algoritmo en muchas áreas. Además, las ventajas más significativas del algoritmo son su simplicidad de implementación y baja complejidad computacional.

Sin embargo, como también lo menciona Ikotun (2023), el algoritmo K-means presenta varios desafíos que afectan negativamente su rendimiento de agrupamiento. Uno de ellos se presenta durante el proceso de inicialización del algoritmo, debido a que el usuario debe especificar previamente el número de clústeres en el conjunto de datos, mientras que los centroides iniciales se seleccionan aleatoriamente. Además, que el hecho de encontrar el número de óptimo de clústeres puede llegar a convertirse en una de las tareas más desafiantes.

3.6 Modelo de Clasificación

El modelo de clasificación seleccionado ha sido Random Forest debido a sus múltiples beneficios en cuanto al manejo de grandes cantidades de información. Como lo menciona Zhu (2020): “el algoritmo Random Forest es flexible, tiene una amplia gama de aplicaciones y ofrece un buen rendimiento en una gran cantidad de conjuntos de datos. Además, es inmune a supuestos estadísticos estrictos y no requiere un preprocesamiento complejo, lo que le permite manejar conjuntos de datos extensos con alta dimensionalidad y valores faltantes.

No obstante, Random Forest presenta dificultades con variables categóricas de alta cardinalidad, datos desbalanceados, predicciones en series temporales, y en la interpretación de variables, además de ser sensible a la configuración de hiperparámetros.”

3.6.1 Análisis Exploratorio de Datos

Para el análisis exploratorio de datos se han utilizado tres enfoques:

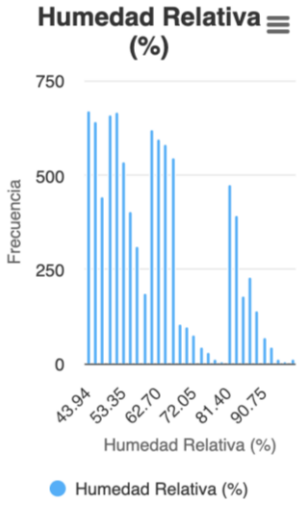
- Análisis de distribución de frecuencias para cada característica utilizando histogramas
- Análisis de distribución de datos para cada característica utilizando diagramas de caja
- Análisis de correlación de características utilizando una matriz de correlación

3.6.2 Análisis de Distribución de Frecuencias

El histograma de humedad relativa mostrado en la figura 9, presenta una gran cantidad de registros entre el 50 y 70%, por lo que se determina que fue común detectar un clima templado o ligeramente húmedo durante el periodo de medición debido a las condiciones ambientales del lugar.

Figura 9

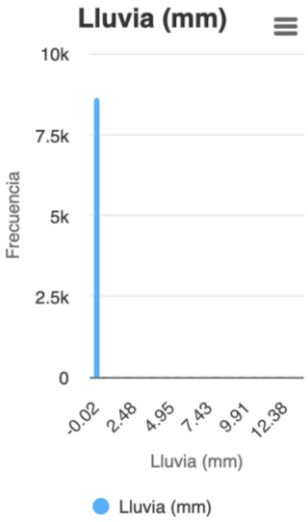
Histograma de Humedad Relativa



El histograma de lluvias detallado en la figura 10, muestra que no existen eventos significativos en la mayoría del conjunto de datos por lo que se determina que el entorno climático se encontró seco la mayor parte del tiempo. Se debe tomar en cuenta esta información para las secciones a continuación ya que la presencia de lluvia puede ser un factor muy relevante al medir y predecir el flujo vehicular.

Figura 10

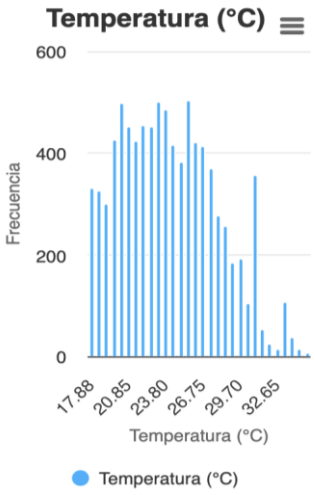
Histograma de Lluvias



El histograma de temperatura mostrado en la figura 11, esta medido en grados centígrados, muestra que en su mayoría se encuentra entre los 20 y 29 grados centígrados debido a las condiciones ambientales. Cabe recalcar que existen picos desde los 30 grados en adelante que podrían significar eventos aislados donde existió un aumento de calor extremo.

Figura 11

Histograma de Temperatura

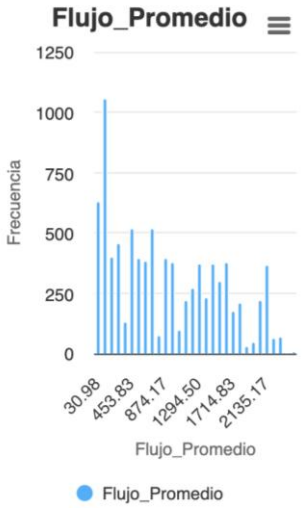


El histograma de flujo promedio mostrado en la figura 12, presenta la existencia una alta concentración entre 30 y 450 unidades vehiculares, por lo que se puede determinar que es

común presenciar un flujo vehicular moderado mientras que los flujos vehiculares altos son poco comunes, pero tienen una presencia significativa.

Figura 12

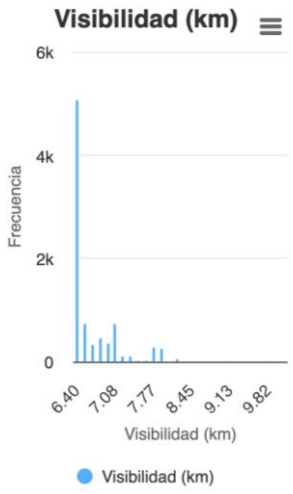
Histograma de Flujo Vehicular Promedio



El histograma de visibilidad detallado en la figura 13, muestra una gran concentración en el punto 6.40, esto puede deberse a las condiciones climáticas constantes o la misma funcionalidad del sensor utilizado. Para el resto de los eventos se determina que la falta de visibilidad se puede determinar cómo eventos aislados o muy raros.

Figura 13

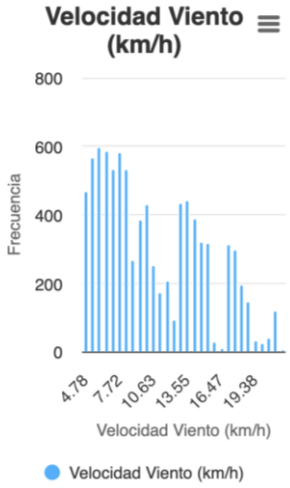
Histograma de Visibilidad



El histograma de velocidad del viento detallado en la figura 14, muestra una concentración desde los 5 a 13 kilómetros por hora por lo que se determina que en la mayoría de los casos la velocidad del viendo fue baja o moderada. Para el resto de las variaciones se consideran eventos meteorológicos puntuales durante la medición.

Figura 14

Histograma de Velocidad del Viento

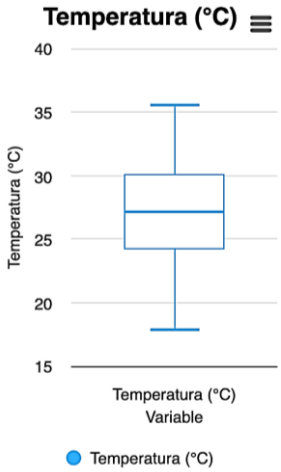


3.6.3 Análisis de Distribución de Datos

El diagrama de caja de la temperatura detallad en la figura 15, muestra que los datos se encuentran entre los 24 y 30 grados centígrados, teniendo una mediana de 27 grados centígrados por lo que se determina que existe en su mayoría un clima cálido. Además, existe una temperatura mínima de aproximadamente 18 grados centígrados y una máxima de 37 grados centígrados haciendo que los datos tengan una gran amplitud térmica. Cabe resaltar que no existen valores fuera de los límites por lo que se descarta la presencia de valores atípicos y una distribución relativamente uniforme. Se puede notar que no existen valores atípicos entre los límites indicados.

Figura 15

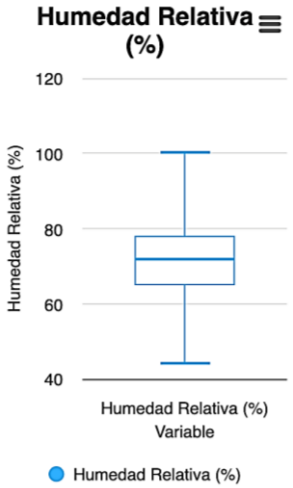
Distribución de Temperatura



El diagrama de caja de la humedad relativa mostrado en la figura 16, indica que los datos en su mayoría están entre los 65% y 80%, teniendo una mediana aproximada de 72% por lo que se determina que durante la medición predominó el clima húmedo debido a las condiciones climáticas de la región costa en general. Se puede notar que no existen valores atípicos entre los límites indicados.

Figura 16

Distribución de Humedad Relativa

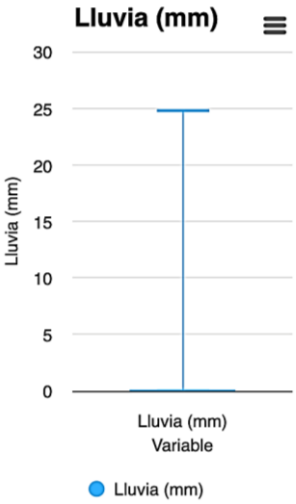


El diagrama de caja de la lluvia mostrado en la figura 17, indica que la variación de

los datos es muy baja ya que en su mayoría se encuentran en cero, por lo que se determina que existieron pocos eventos de lluvia y para el resto de los casos, se los considera como eventos climáticos aislados. Se puede notar que no existen valores atípicos entre los límites indicados.

Figura 17

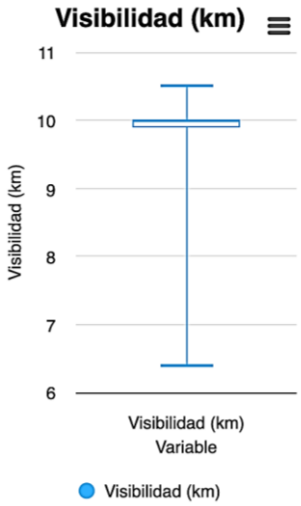
Distribución de Lluvia



El diagrama de caja de la visibilidad detallado en la figura 18, muestra que en su mayoría existe un rango de hasta 10km, por lo que se determina que no existe un mayor conflicto de visibilidad. El rango intercuartílico es muy estrecho, por lo que se determina que no existe una gran variabilidad. Sin embargo, es importante mencionar que existe un valor de aproximadamente 6.4 Km en el rango de la visibilidad que puede ser atribuido a eventos climáticos aislados posiblemente por lluvia, contaminación o niebla. Se puede notar que no existen valores atípicos entre los límites indicados.

Figura 18

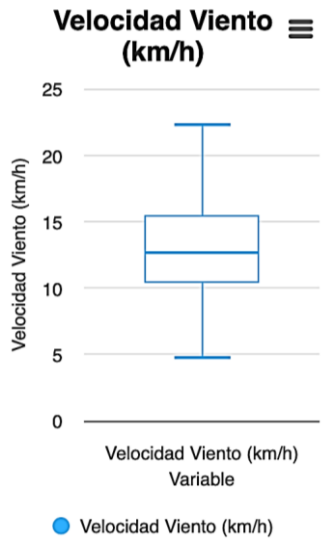
Distribución de Visibilidad



El diagrama de caja de la velocidad del viento mostrado en la figura 19, indica que existe en su mayoría una variación de entre los 10KM/h hasta los 16Km/h, con una mediana de aproximadamente 13 Km/h lo que sugiere que existe un rango de viento de ligero a moderado. Además, el valor mínimo se aproxima a los 5Km/h y el máximo a 23Km/h siendo estos casos de eventos climáticos aislados. Se puede notar que no existen valores atípicos entre los límites indicados.

Figura 19

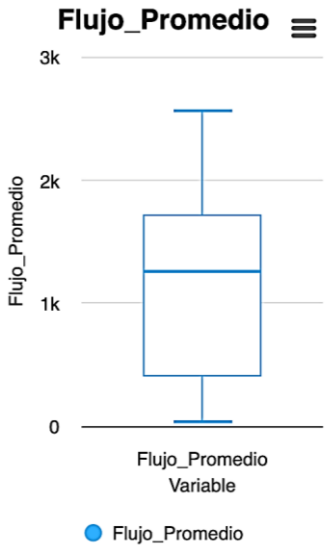
Distribución de Velocidad del Viento



El diagrama de caja del flujo promedio detallado en la figura 20, muestra una variación entre los 400 y 1750 vehículos lo que determina un flujo moderado en la mayoría de los casos, con una mediana de 1300 vehículos.

Figura 20

Distribución de Flujo Promedio



Además, se muestra un rango de amplitud de entre los cero y 2700 vehículos, esto para casos en los que existe un flujo vehicular muy alto debido a condiciones climáticas extremas o accidentes y en el caso de flujo vehicular muy bajo debido a la hora pudiendo ser estas horas de la madrugada en la que el flujo vehicular es muy poco habitual.

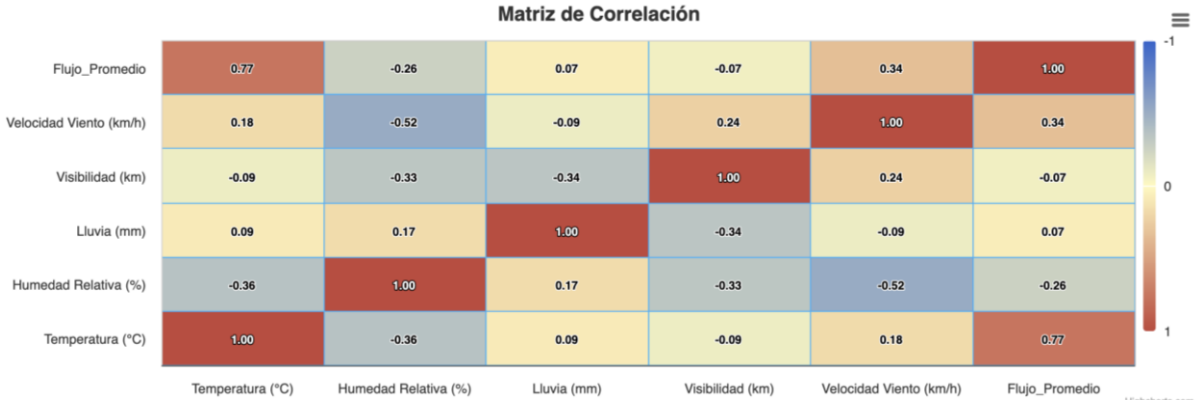
3.7 Análisis de Correlación de Variables

La matriz de correlación detallada en la figura 21, muestra la relación lineal entre el flujo promedio y las variables climáticas. Aquí se muestra que la temperatura tiene una fuerte correlación positiva con el flujo promedio teniendo un valor de 0.77, un valor bastante alto, lo que indica que, al aumentar la temperatura, el flujo vehicular llegaría a aumentar también.

Por otro lado, se observa que la humedad tiene una correlación moderada con el flujo vehicular teniendo un valor de -0.36, lo que indica que a menos humedad existe menos flujo vehicular.

Figura 21

Matriz de Correlación de Variables



En cuanto a las variables lluvia y visibilidad con valores de 0.09 y -0.09 respectivamente muestra una correlación baja, por lo que se puede determinar que al menos en este lugar estas variables no están completamente relacionadas con el flujo vehicular.

Finalmente, se muestra una correlación negativa entre las variables humedad y viento con un valor en común de -0.52 lo que puede indicarse como eventos atmosféricos ya que, al aumentar el viento, disminuye la humedad y viceversa.

En términos generales se determina que la variable que tiene mayor significancia al predecir el flujo promedio, mientras que el resto de las variables tienen efectos muy bajos en la predicción.

3.8 Preprocesamiento de Datos

Para el preprocesamiento de datos ha sido necesaria la identificación de valores faltantes. Estos se han dado por errores en la recolección de los datos o sencillamente al no existir un valor adecuado para ese momento. Estos valores han sido reemplazados por el valor promedio de los valores existentes para no crear valores atípicos innecesarios.

3.9 Escalamiento de Datos

Durante el análisis exploratorio de datos se ha notado que las variables a pesar de ser numéricas tienen rangos diferentes, por lo tanto, ha sido necesario implementar un

escalamiento con el fin de normalizar los valores y evitar que aquellos valores que sepan sumamente superiores dominen en el proceso de entrenamiento del modelo de aprendizaje automático.

Además, esta normalización es importante para que los algoritmos PCA y KMeans puedan utilizar estar características de manera equitativa y puedan detectar los patrones en el conjunto de datos, que bien pudieran pasar desapercibidos si se mantuviesen los datos originales.

Dicho lo anterior, se ha implementado el método de escalamiento Power Transformer que como está definido por Scikit Learn (*Preprocessing Data*, s. f.) es un método que se basa en transformaciones monolíticas de las características, dicho de otra manera, este método conserva el rango de los valores para cada una de las características.

Este método soporta dos tipos de transformadores, Yeo-Johnson y Box-Cox. En este proyecto se ha implementado el transformador Yeo-Johnson ya que ha sido el más adecuado considerando que se tratan de datos no lineales como se ha podido comprobar en secciones anteriores. Este método es en particular útil para estos casos ya que como sus autores lo mencionan (Yeo, 2000) “la transformación propuesta es capaz de manejar tanto valores positivos como negativos y proporciona una familia continua de transformaciones que pueden mejorar la normalidad o la simetría para una amplia gama de tipos de datos” (p. 955) por lo que corregirá el conflicto con los datos no lineales.

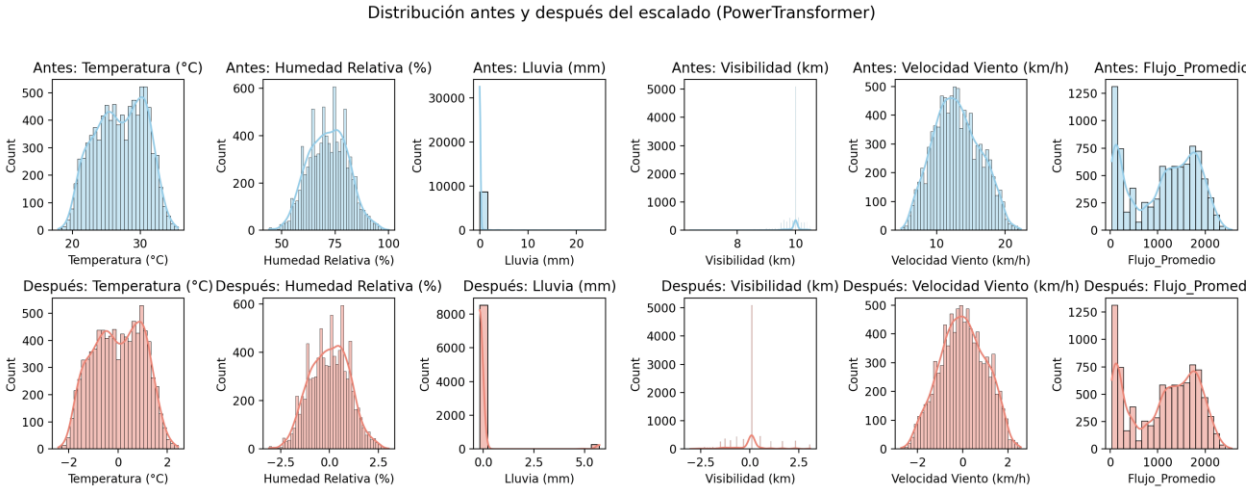
En la figura 22 muestra el escalamiento obtenido al aplicar el transformador Yeo-Johnson. Se puede notar los siguientes cambios:

- Temperatura: normaliza el rango de 20 a 30 unidades a un rango de -2 a 2 unidades
- Humedad relativa: normaliza el rango de 50 a 100 unidades a un rango de -2.5 a 2.5 unidades
- Lluvia: normaliza el rango de 0 a 20 unidades a un rango de 0 a 5 unidades

- Visibilidad: normaliza el rango de 0 a 10 unidades a un rango de -2.5 a 2.5 unidades
- Velocidad Viento: normaliza el rango de 0 a 20 unidades a un rango de -2.5 a 2.5 unidades

Figura 22

Distribución del Escalado de Datos

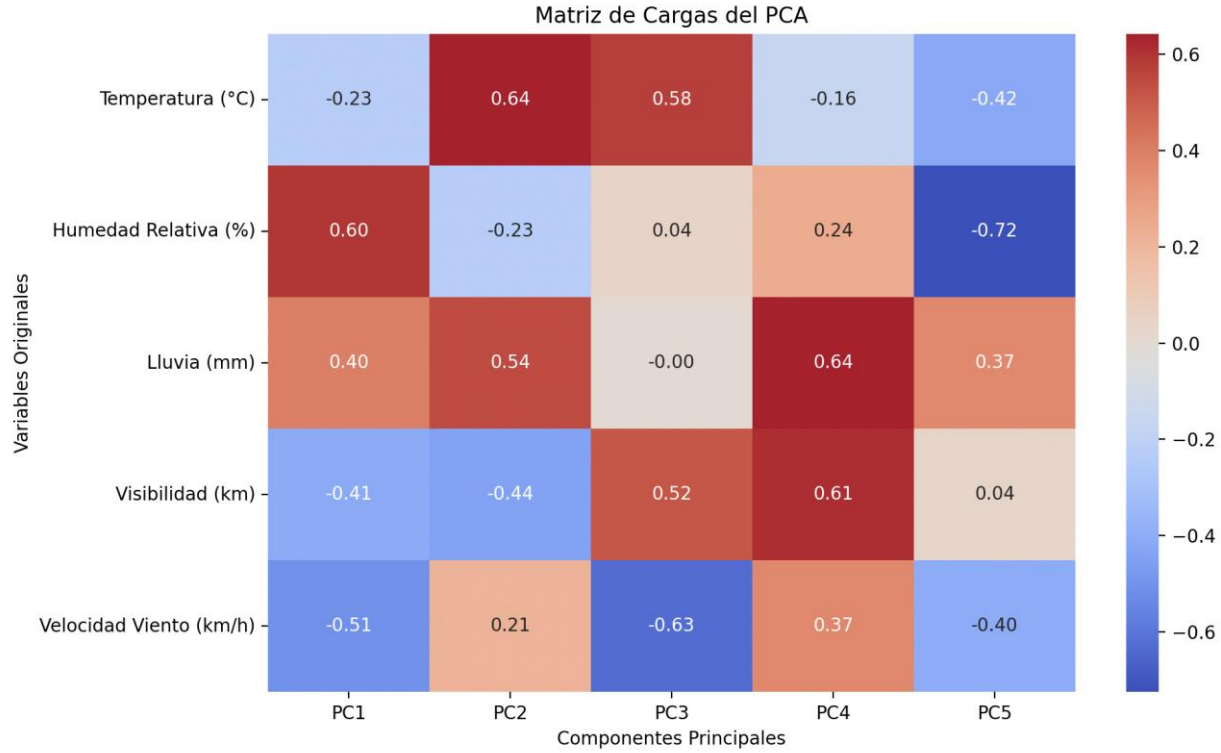


3.10 Reducción de Dimensionalidad

Debido a que se utilizará el algoritmo de agrupamiento KMeans, es importante evitar que los clústeres que se generen tiendan a sesgos, por lo que se aplicará la técnica de reducción de dimensionalidad Principal Component Analysis desde ahora referido como PCA. Este enfoque ha sido considerado ya que como lo menciona Ding (2004) en su estudio de la aplicación de PCA en el algoritmo de agrupamiento KMeans: “sobre la reducción de dimensionalidad, el resultado ofrece nuevas perspectivas sobre la efectividad observada de las reducciones de datos basadas en PCA, más allá de la explicación convencional centrada en la reducción de ruido. En particular, demuestra que el PCA, mediante la descomposición en valores singulares (SVD), proporciona la mejor aproximación lineal de baja dimensión de los datos.”.

Figura 23

Reducción de Dimensionalidades PCA



En la figura 23 se muestra el resultado de la aplicación del método de reducción de dimensionalidad PCA, del que se puede determinar que:

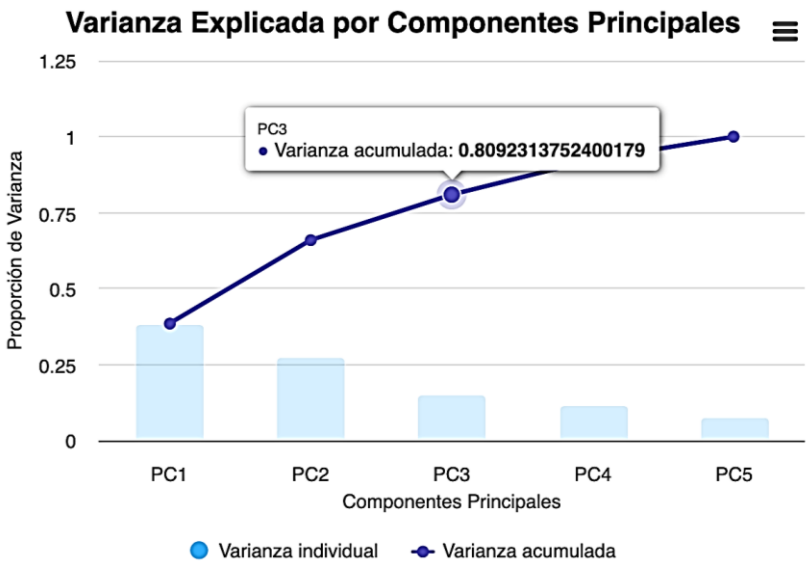
- Componente PC1: este componente está influenciado en su mayoría por la Humedad Relativa en conjunto con Lluvia, en contraste con la Velocidad del viento y la Visibilidad que tienen puntajes más bajos. Este componente refleja la clara distinción entre los climas húmedos y mojados frente a los días secos en los que el viento y la visibilidad son más comunes.
- Componente PC2: este componente está influenciado por la Temperatura y la Lluvia, en contraste con Visibilidad y Humedad con los puntajes más bajos. En este componente se puede determinar la presencia de un clima cálido o templado con la presencia de lluvia donde no necesariamente se reduce la visibilidad.

- Componente PC3: este componente está influenciado en su mayoría por la Tempratura y la Visibilidad, en contraste con un fuerte valor negativo de la Velocidad del Viento. En este componente se puede determinar las condiciones ambientales comparando entre días calurosos y con vientos reducidos o días fríos con vientos más fuertes.
- Componente PC4: este componente está influenciado fuertemente por la Lluvia y la Visibilidad y moderadamente con la Velocidad del Viento. En este componente se puede determinar la presencia de Lluvia sin afectar la visibilidad y con presencia de viento moderada.
- Componente PC5: este componente está influenciado en su mayoría por valores negativos como son la Humedad, Temperatura y Velocidad del Viento. En este componente se puede determinar la existencia de Lluvia en días secos.

Ahora, con los componentes obtenidos es posible verificar la varianza acumulada para cada uno de los componentes.

Figura 24

Varianza Individual y Acumulada



En la figura 24 es importante destacar que la varianza acumulada a partir del tercer

componente llega a ser óptimo, ya que contiene la mayor parte de la información de conjunto de datos original. Esto permite reducir la complejidad del modelo especialmente para realizar el agrupamiento de la información con KMeans.

3.11 Selección de la Data para Cada Modelo

3.11.1 Datos para Cada Modelo

Como se ha indicado en secciones anteriores, se han aplicado técnicas de escalamiento y reducción de dimensionalidad. Es importante definir que esto ha sido necesario para definir la entrada de cada algoritmo, teniendo en cuenta que siempre será necesario conservar la mayor cantidad de información para no sesgar el entrenamiento de los modelos.

Para el modelo K-Means, será necesario utilizar los componentes obtenidos desde la reducción de dimensionalidad, ya que este método nos permite un mejor rendimiento en el cuanto a agrupación.

Para el modelo Random Forest será necesario utilizar los datos obtenidos desde el escalamiento, ya que aplicar los datos de PCA podría llevar a pérdida de la información y sobre ajustaría innecesariamente el modelo.

3.11.2 Clase Objetivo

Cabe recalcar la importancia de la selección de la clase objetivo para cada uno de los modelos, dado que los datos que serán predichos por el modelo K-Means serán utilizados para el entrenamiento del modelo Random Forest.

Dado que los datos de Flujo Promedio han sido seleccionados como la variable objetivo y los valores son solamente numéricos, no es posible realizar una clasificación entre categorías de flujo vehicular. Por lo tanto, se ha utilizado el algoritmo K-Means para realizar el agrupamiento de estos valores numéricos y obtener cuatro clases a partir de los tres primeros componentes principales obtenidos de PCA.

El análisis de distribución de frecuencia del flujo promedio se ha determinado que pueden existir cuatro categorías las cuales están mostradas en la tabla 1, considerando la existencia de mayor cantidad de datos en el flujo regular.

Tabla 1

Categorías Usadas en el Modelo de Predicción

Categoría
Muy Bajo
Bajo
Medio
Alto

Con este resultado, es posible entrenar el modelo Random Forest para predecir la categoría del flujo vehicular utilizando los datos de condiciones climáticas.

3.12 Entrenamiento del Modelo

3.12.1 K-Means

El entrenamiento del modelo será realizado a partir de los tres primeros componentes obtenidos desde la reducción de dimensionalidad, se utiliza un rango amplio desde uno hasta once clústeres, como se muestra en la figura 25, para verificar el número óptimo de clústeres extrayendo las coordenadas de su centroide.

Figura 25

Determinación del Número Óptimo de Clústeres

```
X_pca_reducido = X_pca[:, :3]
ks = range(1, 11)
inercias = [KMeans(n_clusters=k, random_state=33).fit(X_pca_reducido).inertia_ for k in ks]
```

Una vez obtenidas las coordenadas de cada uno de los centroides se encuentra el valor

óptimo utilizando el algoritmo del codo. Este cálculo parte de la identificación del punto inicial y el punto final de la línea que representa el codo, siendo el punto inicial el primer valor del rango utilizado en conjunto con la primera inercia obtenida y el punto final el último valor del rango en conjunto con la última inercia obtenida. La figura 26 muestra la obtención del coordenadas entre clúster y centroide.

Figura 26

Obtención de Coordenada entre el Clúster y el Centroide

```
x1, y1 = ks[0], inercias[0]
x2, y2 = ks[-1], inercias[-1]
```

Ahora, se itera cada uno de los valores del rango utilizado para obtener las inercias y se extrae el centroide de cada punto. Al obtener la coordenada entre el clúster y el centroide es posible encontrar la distancia entre la recta determinada por el punto inicial y el final y este punto. Para ello se utiliza la fórmula de la Distancia Perpendicular la cual está definida por la ecuación 1.

$$\text{Distancia}= \frac{|(y2-y1)*x0-(x2-x1)*y0+x2*y1+x1*y2|}{\sqrt{(y2-y1)^2+(x2-x1)^2}} \qquad (1)$$

Figura 27

Obtención de Distancias entre Centroides

```
distancias = []
for i in range(len(ks)):
    x0, y0 = ks[i], inercias[i]
    num = abs((y2 - y1)*x0 - (x2 - x1)*y0 + x2*y1 - y2*x1)
    den = ((y2 - y1)**2 + (x2 - x1)**2)**0.5
    distancias.append(num / den)

k_optimo = ks[np.argmax(distancias)]
```

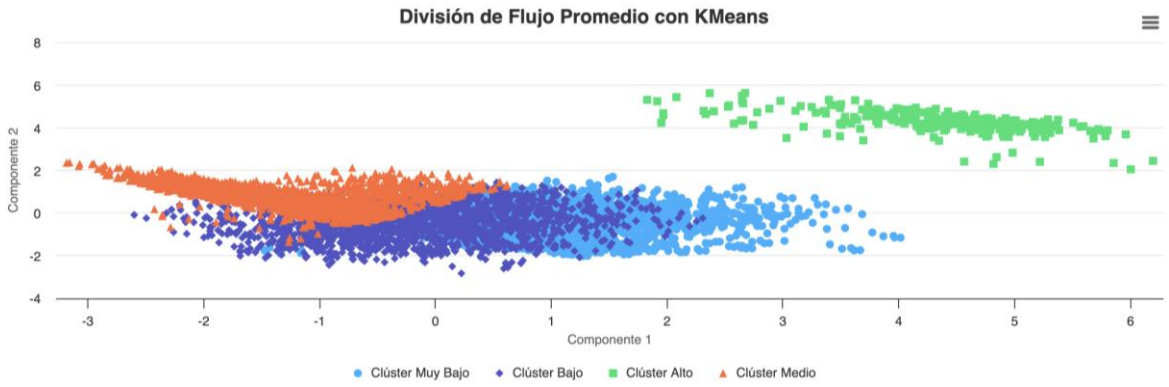
Así se puede obtener la distancia de cada uno de los centroides en referencia con la recta, como se muestra en la figura 27, es entonces que se puede determinar que el número de clústeres óptimo viene siendo aquel punto que se encuentra más alejado de la recta entre los

puntos iniciales y finales.

Una vez obtenido número de clústeres óptimo se entrena el modelo K-Means para clasificar el flujo vehicular promedio en categorías. Dado que el valor de cada clúster del entregado por el modelo no necesariamente representa una categoría, es necesario asignar una categoría a partir de los valores de los centroides de cada clúster. En la figura 28 se visualizan los clústeres de Flujo de Tráfico Mediante K-Means.

Figura 28

Clústeres en el Modelo K-Means



Utilizando el conjunto de datos original, se asigna los clústeres a cada registro y para asignar una etiqueta correspondiente al flujo promedio, se realiza el cálculo de la media agrupada por clúster, es así que, al ordenarlo de mayor a menor. La tabla 2 muestra los resultados obtenidos.

Tabla 2

Número de Clústeres por Categoría

Categoría	**Clúster**
Muy bajo	2
Bajo	0
Medio	3
Alto	1

Ahora es posible asignar una categoría a los valores predichos por KMeans para el flujo vehicular promedio. En la figura 29 se muestra la segmentación de datos de trafico mediante K-means y su etiquetado.

Figura 29

Segmentación de Datos de Tráfico usando K-Means y Etiquetado de Clústeres

```
k_optimo = ks[np.argmax(distancias)]
kmeans = KMeans(n_clusters=k_optimo, random_state=33, n_init=10)
y_kmeans = kmeans.fit_predict(X_pca_reducido)
data['Cluster'] = y_kmeans
categorias = ['Muy Bajo', 'Bajo', 'Medio', 'Alto']

cluster_means = data.groupby('Cluster')['Flujo_Promedio'].mean().sort_values(ascending=False)
mapping = {cluster_means.index[i]: label for i, label in enumerate(categorias)}
y_etiquetado = data['Cluster'].map(mapping)
```

3.13 Random Forest

El entrenamiento del modelo será realizado a partir del resultado obtenido del escalamiento del conjunto de datos original para evitar perder información valiosa en el entrenamiento y la asignación de las etiquetas obtenidas desde K-Means en la clase objetivo del flujo vehicular promedio.

Se define el modelo Random Forest con los parámetros de la tabla 3.

Tabla 3

Parámetros para el Modelo Random Forest

Parámetro	Valor	Justificación
n_stimators	100	Valor óptimo entre costo y rendimiento del modelo.
max_depth	20	Previene el sobreajuste del modelo considerando que entre más profundidad tenga, más inferencia sobre los datos realiza.
random_state	33	Valor asignado para reproducibilidad.

Es importante mencionar la división de los datos en entrenamiento y prueba, para lo

cual se ha utilizado el teorema de Pareto dividiendo el conjunto de datos en 80% será parte del conjunto de entrenamiento y el 20% será parte del conjunto de pruebas.

Con el modelo definido se entrena el conjunto de datos especializado se realiza la validación teniendo como resultado las métricas de la figura 30.

Figura 30

Matriz de Confusión del Random Forest

Capítulo IV

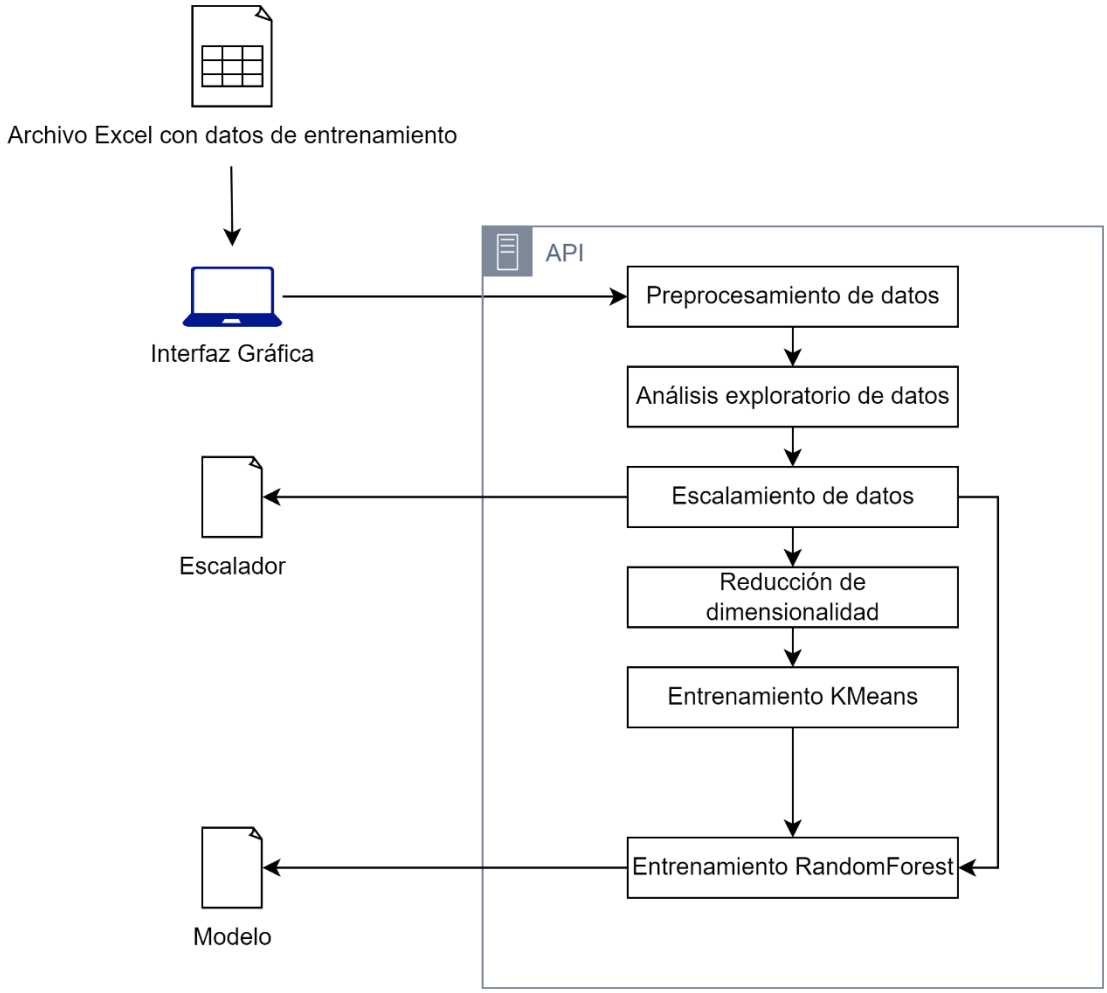
Análisis de Resultados

4.1 Esquema Conceptual del Proyecto

En la figura 31 se muestra el análisis funcional del flujo implementado en el desarrollo del modelo híbrido de segmentación y predicción de flujo vehicular en la avenida de “Las Aguas”. Se ha implementado una interfaz gráfica que permite la importación de un archivo Excel con los datos de entrenamiento para este modelo híbrido.

Figura 31

Flujograma del Modelo Hibrido



Una vez el archivo ha sido importado, se enviará a un servidor FastAPI donde se ha expuesto un API en la que se leen los datos tabulares y se realiza el entrenamiento del

modelo. Las fases del modelo son las siguientes:

- Preprocesamiento de datos: se refinan los datos detectando y eliminando valores faltantes y seleccionando las variables más importantes para el entrenamiento del modelo.
- Análisis exploratorio de datos: se realiza un análisis detallado de la información por frecuencia, valores y correlación para determinar la relación entre las variables y próximos pasos.
- Escalamiento de datos: se realiza el escalamiento de datos seleccionando el método que mejor se acople a los datos no lineales utilizados. De esta etapa se exporta un archivo físico que será utilizado para el escalamiento de los nuevos datos en la fase de predicción.
- Reducción de dimensionalidad: se realiza la reducción de dimensionalidad para asegurar que el método K-means no tienda a sesgar la información al crear el agrupamiento de clúters.
- Entrenamiento del modelo de agrupamiento K-means: se realiza el entrenamiento del modelo K-means para realizar la categorización de la variable objetivo utilizando el resultado de la reducción de dimensionalidad.
- Entrenamiento del modelo de clasificación Random Forest: se realiza el entrenamiento del modelo Random Forest utilizando el resultado del escalamiento, ya que necesitamos la información original del conjunto de datos y el resultado del agrupamiento con K-means. De aquí se exporta un archivo físico con el modelo que será utilizado para la predicción de nuevos datos.

  El Diagrama de funcionalidad completo se muestra en el apéndice F.

4.2 Resultados del Modelo de Agrupamiento

El modelo de agrupamiento K-means que se implementó en el código se utilizó para

poder identificar los patrones y realizar la respectiva clusterización basados en características como temperatura, humedad, lluvia, visibilidad y velocidad del viento.

Previo al uso del algoritmo de K-means, los datos se proceden a revisar y pre-procesar con el fin de eliminar posibles valores nulos, a su vez se procede a escalar las características para normalizarlas, con el uso de un escalador, y, con el uso de PCA, se procede a reducir la dimensionalidad. Como producto del uso de PCA se seleccionan las cuatro características principales permitiendo tener la mayor variación de los datos en un espacio de menor dimensión y como consecuencia facilitando su agrupamiento.

El algoritmo evalúa diferentes cantidades de clusters (de 1 a 10) y permitió calcular la inercia (suma de las distancias al cuadrado de los puntos a sus centroides) para cada uno. Posteriormente, se procede a utilizar el método del codo para poder identificar el número óptimo de clusters. El método del codo tiene como objetivo el poder medir la distancia de cada punto de inercia a una línea trazada entre el primer y último punto del rango de clusters, y se procede a seleccionar el valor de `k` que maximiza dicha distancia. Este análisis nos permite asegurar la correcta elección del número de clusters que, como consecuencia, tenga un balance entre la compacidad de los grupos y, a la vez, mantener simple el modelo.

Posterior a la correcta elección del número óptimo de clústers, se procede a la aplicación del algoritmo de K-means con dicho valor, utilizando un estado aleatorio fijo (`random_state=33`) para garantizar su reproducibilidad y 10 inicializaciones (`n_init=10`) para mejorar la estabilidad de los resultados. Los datos se proceden a dividir en `k_optimo` grupos, previamente determinado, y cada punto se procede a asignar a un clúster basado en su cercanía a los centroides calculados.

Los centroides de los clústers resultantes se proceden a analizar según su magnitud (norma euclidiana), y como consecuencia, se ordenan de menor a mayor con el fin de poder asignar las etiquetas cualitativas previamente elegidas: "Muy Bajo", "Bajo", "Medio", "Alto"

y "Muy Alto". El mapeo realizado nos permitió clusterizar basándonos en las condiciones climáticas, por índice de severidad, que se encuentran representadas por las correspondientes características.

Con el fin de poder visualizar los resultados obtenidos del agrupamiento, se hizo uso de la función `visualizar_entrenamiento`, que nos permitió observar la distribución de los datos, distintivos visualmente por colores conforme a las etiquetas que se asignaron previamente, pudiendo evaluar, mediante visualización, la respectiva separación de los clústers. Como resultado de todo lo realizado, en pasos posteriores fue posible usar las etiquetas generadas por K-means como variable objetivo (`y_etiquetado`), con el fin de usarlas como inputs para poder realizar el entrenamiento del modelo de clasificación Random Forest, en resumen, el agrupamiento realizado nos sirvió como el paso intermedio para el uso de un algoritmo supervisado.

En resumen, el modelo K-means implementado permitió segmentar los datos que se poseían de las condiciones climáticas en clústers, adicional con el uso de la correcta reducción de dimensionalidad, por medio de PCA, y el respectivo uso del método del codo, fue posible la optimización en la elección del número óptimo de clústers a ser usados. Las etiquetas elegidas para la asignación permitieron facilitar el análisis de los resultados obtenidos, adicional el agrupamiento obtenido nos permitió poder observar los patrones presentes en los datos y tener una base sólida para el posterior uso del modelo de clasificación de Random Forest.

4.3 Resultados del Modelo de Clasificación

Dentro de los modelos de clasificación se eligió el modelo de Random Forest, con el fin de conseguir la predicción de la cantidad de flujo en base a las etiquetas previamente definidas ("Muy Bajo", "Bajo", "Medio", "Alto", "Muy Alto") basadas en los datos procesados y agrupados previamente con K-means.

El modelo de clasificación Random Forest se ajustó por medio del uso de los hiperparámetros configurándolo con 100 árboles (`n_estimators=100`), una profundidad máxima de 20 (`max_depth=20`) y pesos de clase balanceados (`class_weight='balanced'`), este último hiperparámetro nos permite asegurar que el modelo no favorezca categorías mayoritarias, mejorando la predicción de clases menos representadas. Adicional a los hiperparámetros mencionados anteriormente, al igual que en K-means, el parámetro `random_state=33` nos asegura reproducibilidad. La configuración de hiperparámetros mencionada tuvo el enfoque de poder capturar patrones complejos a la vez de brindarnos la posibilidad de poder reducir las probabilidades de un sobreajuste, a la vez que se brinda importancias equitativas a todas las características.

Se procedió al entrenamiento del modelo con los datos de entrenamiento (`X_train`, `y_train`) y posterior evaluación en el conjunto de prueba (`X_test`), generando posteriormente predicciones (`y_pred`). Finalmente, se procedió a la visualización de los resultados mediante una matriz de confusión y las métricas de clasificación tales como Accuracy, Precision, Recall y F1 score obteniendo porcentajes mayores al 97%.

El modelo Random Forest implementado nos permitió clasificar el flujo en base a las etiquetas previamente definidas, tomando como input inicial los clústers previamente generados por K-means. Su configuración balanceada y su integración con los inputs, previamente preprocesados de manera robusta con escalamiento y PCA, lo hacen ideal para el objetivo primario de la predicción en clasificación del flujo en base a las condiciones climáticas.

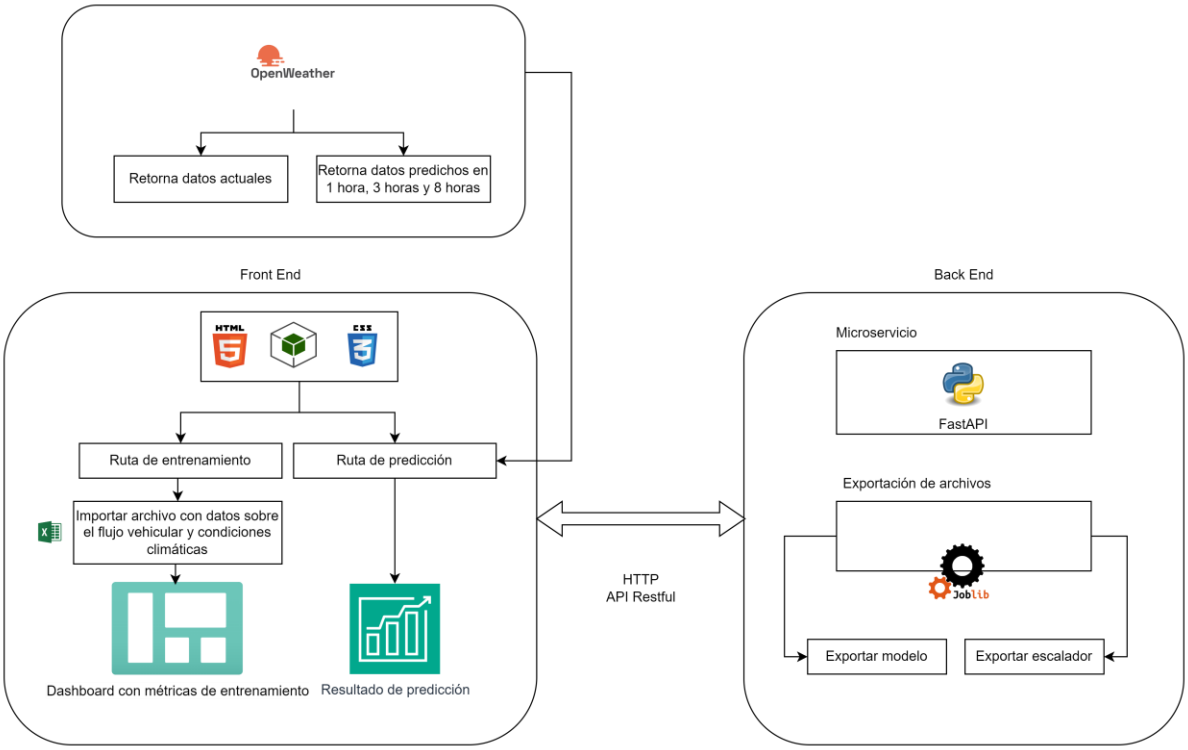
4.4 Diagrama de Arquitectura del Sistema

La arquitectura se muestra en la figura 32, donde se puede observar como se integra el procesamiento de datos con la capacidad de predicción mediante inteligencia artificial. Asi como el backend, apoyado por la librería Scikit-learn que es el cerebro del sistema: gestiona

el ciclo de vida de un modelo de machine learning. A través de una interfaz web amigable, pueden cargar conjuntos de datos históricos para entrenar el modelo, con la posibilidad de visualizar inmediatamente los resultados y métricas clave en un dashboard. Una vez entrenado, el modelo se guarda para su uso.

Figura 32

Arquitectura General del Sistema Híbrido



4.5 Backend

El backend ha sido desarrollando utilizando FastAPI debido a su ligeza y gran capacidad en el desarrollo de proyectos de aprendizaje automático. La estructura del proyecto ha sido adaptada un microservicio en la que cada módulo tiene su funcionamiento encapsulado tanto en clases como en directorios.

Se han expuesto tres rutas, una para el entrenamiento del modelo denominada /entrenar, la segunda para la predicción de nuevos datos usando un modelo previamente entrenado denominada /predecir y la última ruta para consultar las métricas obtenidas en la

fase de entrenamiento. Para la primera ruta, el servidor recibirá datos tabulares en un archivo tipo Excel donde se encuentran las características: ‘Temperatura (°C)', 'Humedad Relativa (%)', 'Lluvia (mm)', 'Visibilidad (km)' y 'Velocidad Viento (km/h)' de las cuales se creará un escalador y un modelo en archivos físicos para ser utilizados por la segunda ruta expuesta. La segunda recibirá los datos de las características utilizadas para el entrenamiento con y los archivos creados durante el entrenamiento. Finalmente, la tercera ruta retorna en formato JSON las métricas obtenidas en el análisis exploratorio de datos, escalamiento de datos, reducción de dimensionalidad, agrupamiento con K-means y resultado del entrenamiento del modelo Random Forest.

Es importante mencionar que el microservicio ha sido configurado con CORS para que pueda ser consumido por la interfaz gráfica. Además, tiene una conexión con una base de datos Postgress para almacenar las métricas obtenidas durante la fase de entrenamiento.

4.6 Frontend

El frontend ha sido desarrollando utilizando React y estilos con Tailwind, se han seleccionado estas librerías con el fin de capturar eventos en tiempo real y brindar un interfaz estéticamente agradable y amigable para el usuario.

Esta interfaz está dividida en dos rutas, la primera está enfocada en el entrenamiento del modelo. Aquí se podrá importar un archivo tipo Excel para que pueda ser enviada al servidor, una vez la información ha sido procesada por el servidor, se consultarán las métricas obtenidas.

Esta primera ruta cuenta con un Dashboard dividido en secciones para apreciar de mejor manera las métricas obtenidas en cada paso del entrenamiento y los resultados de la clasificación final. Es así que se pueden observar enumeradas las secciones:

1. EDA: Análisis Exploratorio de Datos
2. Reducción de Dimensionalidad

3. Categorización de Flujo Promedio con K-means
4. Clasificación con Random Forest

Finalmente, en la segunda ruta se presentará un panel en el que se mostrarán dos cuatro botones, el primero para obtener la predicción del flujo vehicular usando datos en tiempo real extraídos desde OpenWeather y el resto de los botones para obtener la predicción del flujo vehicular usando datos predichos por la plataforma OpenWeather en la siguiente hora, en las siguientes tres horas y las siguientes ocho horas.

4.7 Comunicación entre Frontend y Backend

Para la comunicación entre la interfaz de usuario y el servidor, se utilizará el protocolo HTTP, este protocolo es muy utilizado comúnmente debido a su facilidad en la conexión, gran cantidad de librerías expuestas y su fácil manejo de datos de respuesta en formato JSON.

Capítulo V

Conclusiones y Recomendaciones

5.1 Conclusiones

En el presente proyecto se desglosa en cada paso como realizar una predicción del flujo vehicular en la Avenida “Las Aguas” en la ciudad de Guayaquil – Ecuador, utilizando técnicas avanzadas de aprendizaje automático. De tal forma, podemos abordar las siguientes conclusiones:

Primero, consideramos el análisis exploratorio de datos en esta investigación, la cual nos ha permitido obtener una mayor comprensión del comportamiento del flujo vehicular en la Avenida “Las Aguas” en Guayaquil - Ecuador. Dentro del cual, podemos mencionar a los histogramas y diagramas de caja que nos indican que la mayoría de los registros de flujo vehicular se concentran entre 30 y 450 unidades. Es decir, que el flujo vehicular se encuentra en un nivel moderado de unidades la mayoría de los días. Sin embargo, también se identificó una alta concentración de flujo vehicular, aunque son menos comunes, pueden ocurrir en situaciones específicas. Tales como condiciones climáticas o eventos especiales dentro de la ciudad.

Teniendo en consideración lo anterior, podemos concluir que las autoridades del municipio de Guayaquil deberían prestar mayor importancia a la concentración de flujo vehicular dentro de la ciudad debido a que pueden resultar en congestiones significativas que pueden afectar el bienestar y calidad de vida de las personas. Así mismo, al identificar los patrones del flujo vehicular facilita tomar decisiones estratégicas para implementar medidas preventivas. Dentro de las cuales podemos mencionar, regulación del tráfico vehicular, planificación de rutas alternas, mensajes de concientización y control vehicular.

De igual manera, es importante mencionar la fuerte correlación positiva entre la temperatura y el flujo vehicular, con un coeficiente de correlación de 0.77. Este resultado nos

indica que un incremento en la temperatura está directamente relacionado con un aumento del flujo vehicular. Este fenómeno podría deberse al hecho de que, en climas más cálidos o con grandes distancias las personas son más propensas a utilizar sus vehículos para poder desplazarse. A diferencia de caminar o usar un transporte público.

Por otro lado, también se encuentra particularmente relevante la presencia de humedad al tener una correlación negativa con el flujo vehicular de -0.36. Este hallazgo sugiere que, a medida que la humedad aumenta el flujo vehicular tiende a disminuir. Es decir, en los días más húmedos las personas optan por no salir o utilizar menos sus vehículos, esto puede deberse a las condiciones climáticas adversas que se puedan presentar.

En base a lo mencionado, podemos identificar que las condiciones climáticas son cruciales para comprender cómo afecta el comportamiento del flujo vehicular en la Avenida “Las Aguas”. Una condición climática donde se presente mayor temperatura puede ser un factor que contribuya a un flujo vehicular más constante, mientras que en los días lluviosos podrían causar interrupciones significativas en el tráfico debido a las condiciones climáticas adversas.

En nuestro estudio consideramos el uso de K-Means para agrupar los datos del flujo vehicular, lo que nos permitió clasificar en cuatro categorías de congestión vehicular: muy bajo, bajo, medio y alto. La presente clasificación es de vital importancia para identificar el control y gestión vehicular de acuerdo a las condiciones presentes. Después, al considerar el uso de Random Forest permite realizar predicciones precisas sobre el flujo vehicular en función de las variables climáticas y los horarios de congestión. Con lo mencionado previamente, permite a las autoridades anticipar y decidir de manera oportuna los diferentes niveles de congestión vehicular presentes en la ciudad. Así mismo, poder optimizar la planificación vehicular en distintas circunstancias presentes del día, ya sea condiciones climáticas o categorías de congestión vehicular.

Los resultados de nuestro modelo han demostrado ser significativos para mejorar la precisión de predicción del flujo vehicular utilizados en este estudio al considerar la implementación de técnicas de preprocesamiento de datos como la normalización y la reducción de dimensionalidad mediante el PCA (Análisis de Componentes Principales). De igual manera, el uso de los algoritmos K-Means y Random Forest fueron capaces de clasificar de manera adecuada el flujo vehicular en categorías significativas, facilitando la identificación de patrones en los datos y poder interpretarlos en la vida real.

Los presentes hallazgos de esta investigación tienen implicaciones significativas para proponer un control o estrategias para la planificación urbana y la gestión del flujo vehicular en Guayaquil. El poder entender cómo las condiciones climáticas afectan el flujo vehicular puede ayudar a las autoridades a implementar estrategias más efectivas para el control del tráfico y mejorar la infraestructura vial.

Finalmente, se puede mejorar la planificación vehicular urbana existente, debido a que permite tomar decisiones informadas sobre la ubicación de Avenidas principales en específico. Por tanto, para la presente investigación se consideró la Avenida “Las Aguas”, sin embargo, con los hallazgos presentes se puede extrapolar el modelo hibrido de clasificación y predicción a nuevas avenidas o avenidas principales de la ciudad que presenten un alto flujo vehicular. Así mismo, brindar un conocimiento de concientización sobre el uso del transporte público en días de alta temperatura.

5.2 Recomendaciones

Se recomienda crear un sistema de monitoreo continuo de las variables climáticas (temperatura, humedad, lluvia, visibilidad y velocidad del viento) en las Avenidas principales de las ciudades. Esto debido a que, poder tener data histórica y en tiempo real sobre las condiciones climáticas permitirá calibrar y mejorar los modelos predictivos, lo que a su vez beneficiará la gestión del flujo vehicular en zonas específicas. Además, el monitoreo continuo

puede ayudar a identificar tendencias entre el clima y el flujo vehicular, lo que puede permitir tomar decisiones estratégicas basadas en datos teniendo en cuenta las variaciones climáticas que ayuden a mejorar la planificación urbana sostenible.

Así mismo, es importante continuar desarrollando y mejorando los modelos predictivos, como la inclusión de más variables que puedan influir en el flujo vehicular como variables demográficas. La inclusión de variables como la densidad poblacional, eventos especiales en la ciudad como días festivos y cambios en la infraestructura vial podría mejorar significativamente la precisión de las predicciones.

Finalmente, se recomienda continuar con investigaciones que permitan mejorar la predicción del flujo vehicular y su relación con variables climáticas. La colaboración entre varias instituciones y autoridades municipales pueden generar estrategias prácticas y útiles que puedan mejorar el flujo vehicular y la planificación urbana en las principales avenidas de la ciudad de Guayaquil - Ecuador.

Bibliografía

Ali, A., Khan, M., & Ahmed, F. (2021). *Traffic flow prediction using machine learning: A review.* Journal of Traffic and Transportation Engineering, 8(2), 123-135. https://doi.org/10.1016/j.jtte.2020.12.001

Asociación Automotriz del Perú. (2025, 24 de febrero). *Mayor dinamismo económico impulsó un crecimiento del 4% en el flujo vehicular durante 2024*. https://aap.org.pe

Autoescuela Quinta Avenida. (2024, 11 de septiembre). *Impacto de las condiciones climáticas en la seguridad vial durante los exámenes de conducir*. https://autoescuelaquintaavenida.es

Banco de Desarrollo de América Latina. (2017). *Diagnóstico y proyección de vulnerabilidades frente a la variabilidad y cambio climático en la ciudad de Guayaquil*. https://scioteca.caf.com

Bishop, C. M. (2006). *Pattern recognition and machine learning.* Springer.

Bray, T. (2014). *The JavaScript Object Notation (JSON) data interchange format.* RFC 7159. https://doi.org/10.17487/RFC7159

Breiman, L. (2001). *Random forests.* Machine Learning, 45(1), 5–32. https://doi.org/10.1023/A:1010933404324

CEPAL. (2014). *La economía del cambio climático en América Latina y el Caribe: paradojas y desafíos del desarrollo sostenible*. https://www.cepal.org

Chapelle, O., Schölkopf, B., & Zien, A. (2006). *Semi-supervised learning.* MIT Press.

Chien, S., Ding, Y., Wei, C., & Wei, C. (2002). *Dynamic bus arrival time prediction with artificial neural networks.* Journal of Transportation Engineering, 128(5), 400-406.

ClimeChart (2023). *Comprehensive Climate Chart of Guayaquil, Ecuador: Monthly Averages and Weather Trends*. https://www.climechart.com/en/climate-chart/guayaquil/ecuador

Ding, C., & Gov. (n.d.). *K-means Clustering via Principal Component Analysis*. https://icml.cc/Conferences/2004/proceedings/papers/262.pdf

Ecuavisa. (2023). *¿Cuáles son las vías con más accidentes mortales en Guayaquil?* https://www.ecuavisa.com/noticias/guayaquil/cuales-son-las-vias-con-mas-accidentes-mortales-en-guayaquil-NA5165197

Elmasri, R., & Navathe, S. B. (2015). *Fundamentals of database systems (7th ed.).* Pearson.

FasterCapital. (2024, 5 de junio). *AAR y condiciones climáticas: comprensión del impacto en los accidentes*. https://fastercapital.com

Fawcett, T. (2006). *An introduction to ROC analysis.* Pattern Recognition Letters, 27(8), 861–874. https://doi.org/10.1016/j.patrec.2005.10.010

Fielding, R. T. (2000). *Architectural styles and the design of network-based software architectures (Doctoral dissertation).* University of California.

Gobierno Autónomo Descentralizado Provincial del Guayas. (2012). *Diagnóstico de la vulnerabilidad sectorial de la provincia del Guayas frente al cambio y la variabilidad climática.* https://www.researchgate.net

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning.* MIT Press.

González, J., Pérez, M., & Rodríguez, A. (2023*). Modelo Híbrido de Segmentación y Predicción de Siniestros Viales en la Avenida Las Aguas en Guayaquil*. Revista de Ingeniería de Tráfico, 12(3), 45-60.

Gorayeb & Associates. (2024, 27 de junio). *Impacto del clima en temas de seguridad en construcción.* https://www.gorayeb.com

Harris, C. R., Millman, K. J., van der Walt, S. J., et al. (2020). *Array programming with NumPy.* Nature, 585(7825), 357–362. https://doi.org/10.1038/s41586-020-2649-2

Hassan, H. M., Abdel-Aty, M., & Oloufa, A. A. (2021). *The effect of weather conditions on road safety: A review.* Transportation Research Part A: Policy and Practice, 145, 143-156. https://doi.org/10.1016/j.tra.2020.12.012

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction,* (2nd ed.). Springer.

Hernández, M. (2017). *Diagnóstico y proyección de vulnerabilidad frente a la variabilidad y cambio climático en la ciudad de Guayaquil*. https://es.slideshare.net

Hidalgo Sánchez, R. F. (2017). *Contaminación sonora por tráfico vehicular en la avenida Juan Tanca Marengo-Guayaquil* [Tesis de licenciatura, Universidad de Guayaquil]. https://revistas.ecotec.edu.ec

Hodge, V. J., & Austin, J. (2004). *A survey of outlier detection methodologies. Artificial Intelligence Review,* 22(2), 85-126. https://doi.org/10.1023/B:AIRE.0000045509.59194.9d

Hunter, J. D. (2007). *Matplotlib: A 2D graphics environment.* Computing in Science & Engineering, 9(3), 90–95. https://doi.org/10.1109/MCSE.2007.55

Ikotun, A. M., Ezugwu, A. E., Abualigah, L., Abuhaija, B., & Heming, J. (2023). *K-means clustering algorithms: A comprehensive review, variants analysis, and advances in the era of big data.* Information Sciences, 622, 178–210. https://doi.org/10.1016/j.ins.2022.11.139

iRAP. (2025, 9 de mayo). *Climate change and road safety*. https://irap.org

Jain, A. K. (2010). *Data clustering: 50 years beyond K-means.* Pattern Recognition Letters, 31(8), 651-666. https://doi.org/10.1016/j.patrec.2009.09.011

Jin, W., & Wang, J. (2019). *Big data analytics for traffic accident prediction: A case study in Beijing*, *China*. Journal of Traffic and Transportation Engineering (English Edition), 6(5), 504-514.

Jolliffe, I. T. (2002). *Principal component analysis (2nd ed.).* Springer.

King, S. (2022). *Pydantic: Data validation and settings management using Python type hints.* Python Package Index. https://pydantic-docs.helpmanual.io

Kumar, A., & Singh, S. (2020). *K-Means clustering for traffic analysis: A case study*. International Journal of Traffic and Transportation Engineering, 9(1), 15-23. https://doi.org/10.11648/j.ijtte.20200901.12

Litman, T. (2020). *Evaluating transportation equity.* Victoria Transport Policy Institute. https://www.vtpi.org/equity.pdf

MacQueen, J. (1967). *Some methods for classification and analysis of multivariate observations.* In Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability (Vol. 1, pp. 281-297). University of California Press.

Martínez, R., & López, S. (2022). *Aplicación de Técnicas de Aprendizaje Automático para la Predicción del Tráfico en Ciudades Ecuatorianas*. Journal of Urban Mobility, 10(2), 25-38.

McKinney, W. (2010). *Data structures for statistical computing in Python*. Proceedings of the 9th Python in Science Conference, 56–61. https://doi.org/10.25080/Majora-92bf1922-00a

McKinney, W. (2017). *Python for data analysis: Data wrangling with Pandas, NumPy, and IPython (2nd ed.).* O'Reilly Media.

METRO. (2022). *Fuerte accidente de tránsito en Av. Las Aguas, 3 vehículos chocaron.* https://www.metroecuador.com.ec/noticias/2022/03/09/fuerte-accidente-de-transito-en-av-las-aguas-3-vehiculos-chocaron/

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

Mokhtarian, P. L., & Chen, C. (2004). *Tangible and intangible impacts of road pricing on driving behavior: A case study of San Francisco Bay Area*. Transportation Research Part A: Policy and Practice, 38(4), 271-295.

Municipio de Guayaquil. (2019). *Informe técnico de cubiertas vegetales para edificaciones en Guayaquil.* https://www.guayaquil.gob.ec

OMS. (2023). *A pesar de los notorios progresos, la seguridad vial sigue siendo un problema apremiante para el mundo.* https://www.who.int/es/news/item/13-12-2023-despite-notable-progress-road-safety-remains-urgent-global-issue

OPS. (2022). *Seguridad Vial*. https://www.paho.org/es/temas/seguridad-vial.

Pedregosa, F., et al. (2011). *Scikit-learn: Machine learning in Python.* Journal of Machine Learning Research, 12, 2825–2830.

Preprocessing data. (s. f.). *Scikit-learn.* https://scikit-learn.org/stable/modules/preprocessing.html

Ramírez, S. (2022). *FastAPI: High performance, easy to learn, fast to code.* Python Package Index. https://fastapi.tiangolo.com

Shao, Z., & Kim, Y. (2016). *A big data approach for traffic accident prediction using machine learning techniques*. Transportation Research Part C: Emerging Technologies, 64, 94-107.

Sharma, A., & Singh, P. (2019). *Hybrid approach of K-Means and Random Forest for traffic prediction.* Journal of Intelligent Transportation Systems, 23(4), 345-357. https://doi.org/10.1080/15472450.2018.1474240

Shinar, D. (2007). *Traffic safety and human behavior*. Elsevier.

Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction (2nd ed.).* MIT Press.

Tarko, A. P., & Mannering, F. L. (2007). *The effects of traffic flow and road geometry on the frequency of accidents: A case study of the US highway system*. Accident Analysis & Prevention, 39(5), 954-962.

Van Rossum, G., & Drake, F. L. (2009). *Python 3 reference manual.* CreateSpace.

Varoquaux, G., et al. (2020). *Joblib: Running Python functions as pipeline jobs.* Python Package Index. https://joblib.readthedocs.io

Waskom, M. (2021). *Seaborn: Statistical data visualization.* Journal of Open Source Software, 6(60), 3021. https://doi.org/10.21105/joss.03021

Wickens, C. D., Lee, J. D., Liu, Y., & Gordon-Becker, S. (2015). *An introduction to human factors engineering (2nd ed.).* Pearson Education.

World Health Organization. (2021). *WHO global status report on road safety 2021*. https://www.who.int/publications/i/item/9789241565438

Xie, X., & Ding, Y. (2011). *Data mining techniques for traffic accident prediction and analysis*. Proceedings of the International Conference on Intelligent Transportation Systems, 115-121.

Yeo, I. (2000). *A new family of power transformations to improve normality or symmetry*. Biometrika, 87(4), 954-959. https://doi.org/10.1093/biomet/87.4.954

Zhou, Q., & Troyanskaya, O. (2005). *Statistical modeling of high-dimensional data.* Bioinformatics, 21(4), 780-788. https://doi.org/10.1093/bioinformatics/bti016

Zhu, Tongtian. (2020). *Analysis on the Applicability of the Random Forest*. Journal of Physics: Conference Series. 1607. 012123. 10.1088/1742-6596/1607/1/012123.

Ziad, T., & Verdezoto, A. (2020). *Análisis del congestionamiento vehicular para el mejoramiento de vía principal en Guayaquil-Ecuador*. https://www.redalyc.org

Anexo A

Código de Rutas

```
from fastapi import APIRouter, UploadFile, File
from app.services.model_service import entrenar_modelo, predecir_modelo
from fastapi import APIRouter, HTTPException
from pydantic import BaseModel
from app.gateway.db import CreateDatabaseConnection
from app.models.metric import Metric
import pandas as pd
router = APIRouter()
class InputData(BaseModel):
    Temperatura: float
    Humedad: float
    Lluvia: float
    Visibilidad: float
    Viento: float
@router.post("/entrenar")
async def entrenar_api(file: UploadFile = File(...)):
    return entrenar_modelo(file)
@router.post("/predecir")
async def predecir_api(data: InputData):
    try:
        df = pd.DataFrame([{
            'Temperatura (°C)': data.Temperatura,
            'Humedad Relativa (%)': data.Humedad,
            'Lluvia (mm)': data.Lluvia,
            'Visibilidad (km)': data.Visibilidad,
```

```
            'Velocidad Viento (km/h)': data.Viento
        }])
        resultado = predecir_modelo(df)
        return {"prediccion": resultado}
    except Exception as e:
        raise HTTPException(status_code=400, detail=f"Error al procesar los datos: {str(e)}")
@router.get("/metrics")
def get_metrics():
    db = CreateDatabaseConnection()
    records = db.session.query(Metric).all()
    result = []
    for row in records:
        result.append({
            "id": row.id,
            "TX_NAME": row.TX_NAME,
            "CD_TYPE": row.CD_TYPE,
            "TX_METRIC": row.TX_METRIC  # esto es un JSON string
        })
    db.close_conn()
    return result
```

Anexo B

Código del servicio

```
from app.repositories.data_repository import cargar_datos_excel
from app.utils.preprocessing import preprocess_data, escalar_datos, aplicar_pca,
        eliminar_outliers_iqr_por_clase
from app.utils.visualization import generar_distribuciones, graficar_pca, visualizar_entrenamiento,
        visualizar_clasificacion
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.cluster import KMeans
import numpy as np
import pandas as pd
import joblib
import os
BASE_DIR = os.path.abspath(os.path.join(os.path.dirname(__file__), ".."))
print(BASE_DIR)
model_dir = os.path.join(BASE_DIR, "static")
os.makedirs(model_dir, exist_ok=True)
model_paths = {
    "scaler": os.path.join(model_dir, "scaler.pkl"),
    "pca": os.path.join(model_dir, "pca.pkl"),
    "clf": os.path.join(model_dir, "random_forest.pkl"),
    "mapping": os.path.join(model_dir, "mapping.npy")
}
def entrenar_modelo(file):
    data = cargar_datos_excel(file.file)
    carateristicas = ['Temperatura (°C)', 'Humedad Relativa (%)', 'Lluvia (mm)', 'Visibilidad (km)',
```

```
        'Velocidad Viento (km/h)']
#Preprocesamiento de datos
#   - Obtener variables X y Y sin valores nulos
data, X, y = preprocess_data(data, carateristicas)
#Análisis EDA
#   - Distribución de características
#   - Dispersión de datos en características
#   - Matriz de correlación entre características
generar_distribuciones(pd.concat([X, y], axis=1))
#División y escalamiento de datos
#   - División 80/20 según la regla de paretto
#   - Random State: 33 -> según el promedio de edades de todos los integrantes del equipo
X_escalado, scaler = escalar_datos(X)
# import matplotlib.pyplot as plt
# import seaborn as sns
# df_escalado = pd.DataFrame(X_escalado, columns=X.columns)
# # Concatenamos para graficar
# original = pd.concat([X, y], axis=1)
# transformado = pd.concat([df_escalado, y], axis=1)
# # Graficar
# fig, axs = plt.subplots(2, len(original.columns), figsize=(15, 6))
# fig.suptitle('Distribución antes y después del escalado (PowerTransformer)', fontsize=14)
# for i, col in enumerate(original.columns):
#     sns.histplot(original[col], ax=axs[0, i], kde=True, color="skyblue")
#     axs[0, i].set_title(f'Antes: {col}')
#     sns.histplot(transformado[col], ax=axs[1, i], kde=True, color="salmon")
#     axs[1, i].set_title(f'Después: {col}')
# plt.tight_layout(rect=[0, 0, 1, 0.95])
```

```
# plt.show()
#Redimensionamiento de características
#   - Aplica PCA a las características en X_train_escalado y X_test_escalado
X_pca, pca = aplicar_pca(X_escalado, X.columns)
graficar_pca(X_pca, pca)
# X_pca_reducido = X_pca[:, :2]  # Solo las 2 primeras componentes
# kmeans = KMeans(n_clusters=2, random_state=33, n_init=20)
# y_kmeans = kmeans.fit_predict(X_pca_reducido)
X_pca_reducido = X_pca[:, :3]
ks = range(1, 11)
inercias = [KMeans(n_clusters=k, random_state=33).fit(X_pca_reducido).inertia_ for k in ks]
# Línea entre el primer y último punto
x1, y1 = ks[0], inercias[0]
x2, y2 = ks[-1], inercias[-1]
distancias = []
for i in range(len(ks)):
    x0, y0 = ks[i], inercias[i]
    num = abs((y2 - y1)*x0 - (x2 - x1)*y0 + x2*y1 - y2*x1)
    den = ((y2 - y1)**2 + (x2 - x1)**2)**0.5
    distancias.append(num / den)
k_optimo = ks[np.argmax(distancias)]
kmeans = KMeans(n_clusters=k_optimo, random_state=33, n_init=10)
y_kmeans = kmeans.fit_predict(X_pca_reducido)
data['Cluster'] = y_kmeans
categorias = ['Muy Bajo', 'Bajo', 'Medio', 'Alto']
cluster_means = data.groupby('Cluster')['Flujo_Promedio'].mean().sort_values(ascending=False)
mapping = {cluster_means.index[i]: label for i, label in enumerate(categorias)}
y_etiquetado = data['Cluster'].map(mapping)
```

```
    #Visualiza la distribución de los clusters obtenidos con el mejor centroide en 10 iteraciones
    visualizar_entrenamiento(X_pca, y_etiquetado)
    X_train, X_test, y_train, y_test = train_test_split(X_escalado, y_etiquetado, test_size=0.3,
        random_state=33)
    modelo_clasificador = RandomForestClassifier(n_estimators=100, max_depth=20,
        random_state=33)
    modelo_clasificador.fit(X_train, y_train)
    y_pred = modelo_clasificador.predict(X_test)
    visualizar_clasificacion(y_test, y_pred)
    joblib.dump(scaler, model_paths["scaler"])
    joblib.dump(modelo_clasificador, model_paths["clf"])
    return {"mensaje": "Modelos entrenados y guardados exitosamente"}
def predecir_modelo(df):
    X = df.fillna(df.mean())
    scaler = joblib.load(model_paths["scaler"])
    clf = joblib.load(model_paths["clf"])
    X_scaled = scaler.transform(X)
    y_pred = clf.predict(X_scaled)
    return y_pred.tolist()[0]
```

Anexo C

Código de repositorio

```
import pandas as pd
def cargar_datos_excel(file):
    return pd.read_excel(file, sheet_name='Datos_2024')
```

Código de modelos

```
#Archivo: predict_input.py
from pydantic import BaseModel
from typing import List
class PredictRequest(BaseModel):
    features: List[float]  # 5 valores: Temp, Humedad, Lluvia, Visibilidad, Viento
#Archivo: train_input.py
from pydantic import BaseModel
class TrainRequest(BaseModel):
    file_path: str
    sheet_name: str
#Archivo: metrics.py
from sqlalchemy import Column, Integer, Text
from sqlalchemy.ext.declarative import declarative_base
Base = declarative_base()
class Metric(Base):
    __tablename__ = "metrics"
    __table_args__ = {"schema": "public"}
    id = Column(Integer, primary_key=True)
    TX_NAME = Column(Text, nullable=False)
    CD_TYPE = Column(Integer, nullable=False)
    TX_METRIC = Column(Text, nullable=False)
```

Anexo D

Código Gateway

```
#Archivo gateway/db.py
import json
import pandas as pd
from sqlalchemy import create_engine, text
from sqlalchemy.orm import sessionmaker
class CreateDatabaseConnection:
    def __init__(self):
        self.__DATABASE_URL = "postgresql+psycopg2://postgres:postgres@localhost:5432/tesis"
        self.__engine = create_engine(self.__DATABASE_URL)
        Session = sessionmaker(bind=self.__engine)
        self.session = Session()
    def call_fill_metrics(self, id_, name, cd_type, metric_json):
        self.session.execute(
            text("CALL fill_metrics(:id, :name, :cd_type, :metric)"),
            {"id": id_, "name": name, "cd_type": cd_type, "metric": metric_json}
        )
    def close_conn(self):
        self.session.commit()
        self.session.close()
#Archivo gateway/app.py
from fastapi import FastAPI
from app.routes.traffic_routes import router as traffic_router
from fastapi.middleware.cors import CORSMiddleware
app = FastAPI(title="Microservicio de Flujo Vehicular")
app.include_router(traffic_router, prefix="/flujo")
```

```
app.add_middleware(
    CORSMiddleware,
    allow_origins=["*"],  # or ["*"] para permitir todos los orígenes (no recomendado en producción)
    allow_credentials=True,
    allow_methods=["*"],  # Permite todos los métodos (GET, POST, etc)
    allow_headers=["*"],  # Permite todos los headers
)
```

Anexo E

Código de utilidades

```
#Archivo: preprocessing.py
import pandas as pd
import numpy as np
from sklearn.preprocessing import PowerTransformer
from sklearn.decomposition import PCA
def preprocess_data(data, features):
    data['Flujo_Promedio'] = (
        data['Flujo_Vehicular_NorteSur'] + data['Flujo_Vehicular_SurNorte']
    ) / 2
    X = data[features].copy().fillna(data[features].mean())
    y = data['Flujo_Promedio'].fillna(data['Flujo_Promedio'].mean())
    return data, X, y
def escalar_datos(X):
    scaler = PowerTransformer(method='yeo-johnson')
    X_escalado = scaler.fit_transform(X)
    return X_escalado, scaler
def aplicar_pca(X, varianza=0.95):
    pca = PCA(n_components=varianza)
    X_pca = pca.fit_transform(X)
    return X_pca, pca
def eliminar_outliers_iqr_por_clase(X, y):
    df = pd.DataFrame(X)
    y_series = pd.Series(y)
    mask_total = pd.Series([True] * len(df))
    for clase in np.unique(y):
```

```
        indices_clase = y_series[y_series == clase].index
        df_clase = df.loc[indices_clase]
        Q1 = df_clase.quantile(0.25)
        Q3 = df_clase.quantile(0.75)
        IQR = Q3 - Q1
        mask = ~((df_clase < (Q1 - 1.5 * IQR)) | (df_clase > (Q3 + 1.5 * IQR))).any(axis=1)
        if mask.sum() >= 3:
            mask_total.loc[indices_clase] = mask
    return df[mask_total].values, y_series[mask_total].values
#Archivo: visualization.py
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
import numpy as np
import json
from app.gateway.db import CreateDatabaseConnection
from sklearn.metrics import (
    classification_report,
    confusion_matrix,
    accuracy_score,
    precision_score,
    recall_score,
    f1_score,
    r2_score
)
counter = 1
def generar_distribuciones(data):
    global counter
```

```
# data.hist(bins=30, figsize=(10, 8))
# plt.suptitle("Distribuciones de características")
# plt.tight_layout()
# plt.show()
# plt.figure(figsize=(10, 6))
# sns.boxplot(data=data)
# plt.title("Dispersión de datos por variable")
# plt.xticks(rotation=45)
# plt.show()
# corr_matrix = data.corr()
# plt.figure(figsize=(8, 6))
# sns.heatmap(corr_matrix, annot=True, cmap='coolwarm', fmt=".2f")
# plt.title("Matriz de correlación")
# plt.show()
db = CreateDatabaseConnection()
for col in data.select_dtypes(include='number').columns:
    counts, bins = pd.cut(data[col], bins=30, retbins=True, labels=False)
    hist_data = pd.value_counts(counts, sort=False).tolist()
    bin_edges = bins.tolist()
    payload = {
        "variable": col,
        "bin_edges": bin_edges,
        "frequencies": hist_data
    }
    json_metric = json.dumps(payload)
    db.call_fill_metrics(counter, col, 1, json_metric)
    counter += 1
```

```
    for col in data.select_dtypes(include='number').columns:
        values = data[col].dropna().sort_values()
        q1 = values.quantile(0.25)
        med = values.quantile(0.5)
        q3 = values.quantile(0.75)
        payload = {
            "variable": col,
            "min": float(values.min()),
            "q1": float(q1),
            "median": float(med),
            "q3": float(q3),
            "max": float(values.max())
        }
        json_metric = json.dumps(payload)
        db.call_fill_metrics(counter, col, 2, json_metric)
        counter += 1
    corr = data.corr()
    corr_payload = {
        "variables": corr.columns.tolist(),
        "matrix": corr.values.tolist()
    }
    json_metric = json.dumps(corr_payload)
    db.call_fill_metrics(counter, "Matriz de Correlación", 3, json_metric)
    db.close_conn()
    counter += 1
def visualizar_entrenamiento(x,y ):
    # Visualización de los clústeres
    global counter
```

```
    # plt.figure(figsize=(6, 4))
    # sns.scatterplot(x=x[:, 0], y=x[:, 1], hue=y, palette='Set2')
    # plt.title("Clústeres por KMeans")
    # plt.show()
    db = CreateDatabaseConnection()
    json_metric = json.dumps({
        "type": "scatter",
        "x": x[:, 0].tolist(),
        "y": x[:, 1].tolist(),
        "labels": y.tolist(),
        "palette": "Set2"
    })
    db.call_fill_metrics(counter, "Clusters KMeans", 4, json_metric)
    db.close_conn()
    counter += 1
def visualizar_clasificacion(y_test, y_pred):
    global counter
    db = CreateDatabaseConnection()
    accuracy = accuracy_score(y_test, y_pred)
    precision = precision_score(y_test, y_pred, average='weighted', zero_division=0)
    recall = recall_score(y_test, y_pred, average='weighted')
    f1 = f1_score(y_test, y_pred, average='weighted')
    metricas = json.dumps({
        "accuracy": accuracy,
        "precision": precision,
        "recall": recall,
        "f1": f1
    })
```

```
    db.call_fill_metrics(counter, "Reporte de clasificación", 7, metricas)
    counter += 1
    # print(f"Accuracy: {accuracy:.4f}")
    # print(f"Precision (weighted): {precision:.4f}")
    # print(f"Recall (weighted): {recall:.4f}")
    # print(f"F1 Score (weighted): {f1:.4f}")
    # print("Matriz de confusión:")
    # sns.heatmap(confusion_matrix(y_test, y_pred), annot=True, fmt='d', cmap='Blues')
    # plt.xlabel("Predicho")
    # plt.ylabel("Real")
    # plt.title("Matriz de Confusión - RandomForest sobre etiquetas de KMeans")
    # plt.show()
    matrix = confusion_matrix(y_test, y_pred)
    # Convertir a lista para JSON
    matrix_json = json.dumps({
        "labels": sorted(list(set(y_test) | set(y_pred))),  # opcionalmente explícito
        "matrix": matrix.tolist()
    })
    db.call_fill_metrics(counter, "Matriz de Confusión - RandomForest", 8, matrix_json)
    db.close_conn()
    counter += 1
    counter = 1
def graficar_pca(X_train_pca, pca):
    global counter
    db = CreateDatabaseConnection()
    X_train_pca_df = pd.DataFrame(X_train_pca, columns=[f'PC{i+1}' for i in
            range(X_train_pca.shape[1])])
    pca_metric = json.dumps({
```

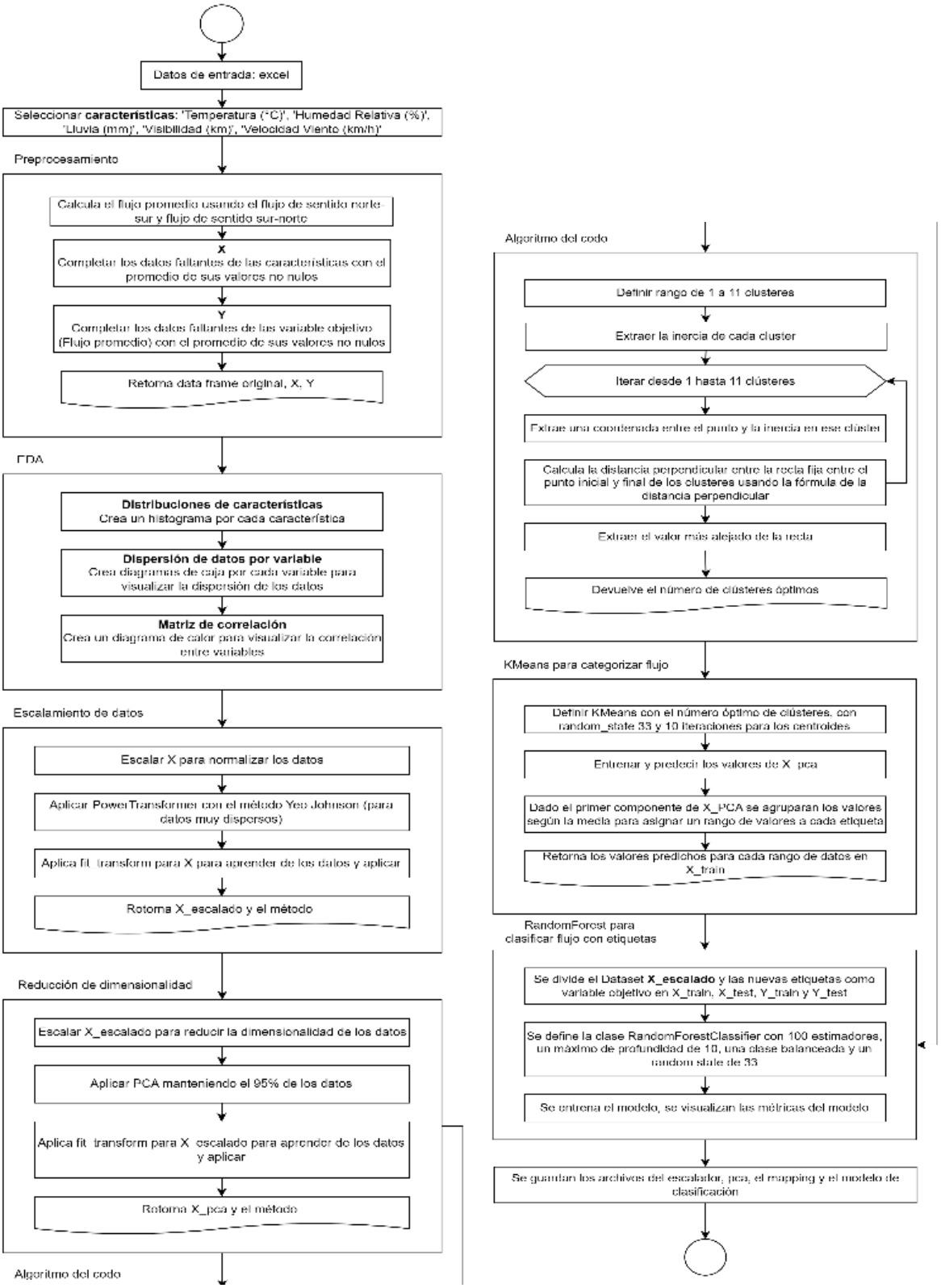
```
    "type": "pca_histplot",
    "columns": list(X_train_pca_df.columns),
    "data": X_train_pca_df.values.tolist()
})
db.call_fill_metrics(counter, "Distribución PCA", 5, pca_metric)
counter += 1
individual = pca.explained_variance_ratio_.tolist()
acumulada = np.cumsum(pca.explained_variance_ratio_).tolist()
varianza_acumulada = json.dumps({
    "type": "pca_varianza",
    "individual": individual,
    "acumulada": acumulada
})
db.call_fill_metrics(counter, "Varianza Explicada PCA", 6, varianza_acumulada)
db.close_conn()
counter += 1
# sns.histplot(X_train_pca_df, kde=True, element="step")
# plt.title("Distribución PCA")
# plt.show()
# explained_variance = pca.explained_variance_ratio_
# plt.figure(figsize=(10, 5))
# plt.bar(range(1, len(explained_variance) + 1), explained_variance, alpha=0.7, label='Varianza
      individual')
# plt.step(range(1, len(explained_variance) + 1), explained_variance.cumsum(), where='mid',
      label='Varianza acumulada')
# plt.xlabel('Componente Principal')
# plt.ylabel('Proporción de Varianza')
# plt.title('Varianza Explicada por Componentes Principales')
```

```
    # plt.legend()
    # plt.show()
#Archivo: evaluation.py
from sklearn.metrics import classification_report, accuracy_score, confusion_matrix
import matplotlib.pyplot as plt
import seaborn as sns
def evaluar_clasificacion(y_true, y_pred):
    print("Accuracy:", accuracy_score(y_true, y_pred))
    print("\nClassification Report:\n", classification_report(y_true, y_pred))
def mostrar_matriz_confusion(y_true, y_pred, etiquetas=None):
    cm = confusion_matrix(y_true, y_pred)
    plt.figure(figsize=(8, 6))
    sns.heatmap(cm, annot=True, fmt='d', cmap='Blues',
                xticklabels=etiquetas, yticklabels=etiquetas)
    plt.xlabel('Predicción')
    plt.ylabel('Real')
    plt.title('Matriz de Confusión')
    plt.show()
```

Código Aplicación

```
from app.gateway.app import app
import uvicorn
if __name__ == "__main__":
    uvicorn.run("app.gateway.app:app", host="0.0.0.0", port=8000, reload=True)
```

Anexo F


Datos de entrada: excel
Seleccionar características: 'Temperatura (°C)', 'Humedad Relativa (%)', 'Lluvia (mm)', 'Visibilidad (km)', 'Velocidad Viento (km/h)'
Preprocesamiento
Calcula el flujo promedio usando el flujo de sentido norte-sur y flujo de sentido sur-norte
X
Completar los datos faltantes de las características con el promedio de sus valores no nulos
Y
Completar los datos faltantes de las variable objetivo (Flujo promedio) con el promedio de sus valores no nulos
Retorna data frame original, X, Y
EDA
Distribuciones de características
Crea un histograma por cada característica
Dispersión de datos por variable
Crea diagramas de caja por cada variable para visualizar la dispersión de los datos
Matriz de correlación
Crea un diagrama de calor para visualizar la correlación entre variables
Escalamiento de datos
Escalar X para normalizar los datos
Aplicar PowerTransformer con el método Yeo Johnson (para datos muy dispersos)
Aplica fit_transform para X para aprender de los datos y aplicar
Retorna X_escalado y el método
Reducción de dimensionalidad
Escalar X_escalado para reducir la dimensionalidad de los datos
Aplicar PCA manteniendo el 95% de los datos
Aplica fit_transform para X_escalado para aprender de los datos y aplicar
Retorna X_pca y el método
Algoritmo del codo
Algoritmo del codo
Definir rango de 1 a 11 clusteres
Extraer la inercia de cada cluster
Iterar desde 1 hasta 11 clústeres
Extrae una coordenada entre el punto y la inercia en ese clúster
Calcula la distancia perpendicular entre la recta fija entre el punto inicial y final de los clusteres usando la fórmula de la distancia perpendicular
Extraer el valor más alejado de la recta
Devuelve el número de clústeres óptimos
KMeans para categorizar flujo
Definir KMeans con el número óptimo de clústeres, con random_state 33 y 10 iteraciones para los centroides
Entrenar y predecir los valores de X_pca
Dado el primer componente de X_PCA se agruparan los valores según la media para asignar un rango de valores a cada etiqueta
Retorna los valores predichos para cada rango de datos en X_train
RandomForest para clasificar flujo con etiquetas
Se divide el Dataset X_escalado y las nuevas etiquetas como variable objetivo en X_train, X_test, Y_train y Y_test
Se define la clase RandomForestClassifier con 100 estimadores, un máximo de profundidad de 10, una clase balanceada y un random state de 33
Se entrena el modelo, se visualizan las métricas del modelo
Se guardan los archivos del escalador, pca, el mapping y el modelo de clasificación