

Análisis Descriptivo y Predictivo Basado en Datos del Tráfico Vehicular en Antofagasta: Un Enfoque a partir de Reportes de Conductores

Richard Peña Bonifaz

19 de octubre de 2025

Dedicatoria

A todas las personas que estuvieron conmigo y me alentaron en los momentos más difíciles.

Reconocimientos

A los profesores apasionados por su trabajo, que no solo transmiten conocimiento, sino también la pasión por el saber y la aplicación de la materia que imparten.

Índice

1. Resumen	1
2. Introducción	2
2.1. Descripción del problema	2
2.2. Metodología	3
2.3. Objetivo general	6
2.4. Objetivos específicos	6
2.5. Alcances	6
3. Marco teórico	7
3.1. Gestión de la seguridad vial	7
3.2. Indicadores de bondad y ajuste	8
3.3. Ciencia de datos para el análisis de comportamiento	11
3.4. Evaluación de los modelos de Machine Learning	12
3.5. Algoritmos de clasificación: Random Forest, Logistic Regression y XG- Boost	13
3.6. Análisis geoespacial	14
3.7. Aprendizaje automático	15
3.8. Investigaciones previas	16
4. Desarrollo	19
4.1. Filtrado temporal y agrupación de eventos	20
4.2. Eventos en el espacio y tiempo	21
4.3. Modelo de correlación lineal	24
4.4. Prueba de hipótesis de correlación lineal	24
4.5. Modelo de correlación exponencial inversa	28

4.6. Análisis de correlación no paramétrica de Spearman	32
4.7. Frecuencia por zona	33
4.8. Selección del modelo de Machine Learning	36
4.9. Generación de eventos negativos simulados	39
4.10. Resultados del modelo seleccionado	41
4.11. Curva de aprendizaje del modelo	42
4.12. Despliegue del modelo y panel de visualización	43
5. Discusiones	45
5.1. Patrones espaciales y temporales del tráfico	45
5.2. Desempeño y comparación de modelos predictivos	45
5.3. Aporte del análisis geoespacial	46
5.4. Implicancias prácticas y futuras aplicaciones	46
5.5. Limitaciones del estudio	47
6. Conclusiones	48
7. Recomendaciones	49
A. Transformación de coordenadas UTM	A
B. Código fuente: Filtrado temporal y agrupación de eventos	D
C. Código fuente: Generación de eventos negativos	F
D. Estructura general del sistema y despliegue	H

Índice de figuras

1.	Pipeline general de datos	5
2.	Flujo de información en dashboard	5
3.	Jerarquía de control de riesgos (NIOSH, 2024).	8
4.	Diagrama de la base de datos	19
5.	Distribución espacial de eventos en Antofagasta	22
6.	Eventos según hora del día en Antofagasta	23
7.	Correlación lineal entre accidentes y congestión	24
8.	Residuos del modelo lineal para días de semana y fines de semana . . .	27
9.	Gráficos Q-Q de los residuos del modelo lineal	27
10.	Correlación exponencial inversa entre accidentes y congestión	29
11.	Residuos del modelo exponencial inverso para días de semana y fines de semana	31
12.	Gráficos Q-Q de los residuos del modelo exponencial inverso	31
13.	Identificación de cuadrantes en mapa de Antofagasta	34
14.	Accidentes por segmento	35
15.	Curvas ROC	39
16.	Curvas Precision-Recall	39
17.	Curva de aprendizaje del modelo <i>XGBoost</i> : desempeño en entrenamien- to con validación cruzada estratificada de 20 particiones.	43
18.	Vista general del <i>dashboard</i> desarrollado para el monitoreo y predicción del tráfico vehicular.	44

Índice de cuadros

1. Comparativa de modelos ajustados con GridSearchCV según el área bajo las curvas ROC y Precision–Recall (PR AUC). 38

1. Resumen

El proyecto desarrollado tuvo como propósito la creación de una herramienta efectiva para analizar y predecir, con alta confiabilidad, el comportamiento del tráfico vehicular en la ciudad de Antofagasta, utilizando datos provenientes de la plataforma Waze Cities. Esta plataforma, alimentada por su comunidad de usuarios, proporciona información en tiempo real que permite obtener una visión detallada de los eventos de tráfico en la ciudad. La investigación resultante generó información relevante para la gestión del tráfico, facilitando la toma de decisiones por parte de las autoridades locales, con miras a mejorar la seguridad vial y optimizar la eficiencia del flujo vehicular. A partir del análisis y explotación de estos datos, fue posible identificar patrones y tendencias que, al ser integrados en la planificación urbana, permiten optimizar rutas críticas, reducir la congestión y disminuir la probabilidad de accidentes.

Se utilizó el algoritmo `GridSearchCV` para seleccionar los mejores parámetros del modelo acorde a su curva ROC-AUC final. Se probaron tres modelos de clasificación: `RandomForestClassifier`, `LogisticRegression` y `XGBBoost`, siendo este último el modelo seleccionado debido a que obtuvo el mayor rendimiento, con un F1-score de 78.6 % y volumen bajo la curva ROC (ROC-AUC) de 84.3 %. Se utilizaron 52,672 datos para el entrenamiento, 26,336 datos de eventos reportados por conductores (luego de ser filtrados) y 26,336 datos generados de forma ficticia como no-ocurrencia de eventos para el balanceo de clases, debido a la naturaleza de los datos al contener solo registros de ocurrencias.

2. Introducción

2.1. Descripción del problema

Antofagasta, una ciudad con más de 106,000 vehículos en circulación (CONASET, 2023), enfrenta desafíos significativos en la gestión de su tráfico vehicular. Durante el año 2023, se registraron 1,715 accidentes, los cuales resultaron en 31 fallecidos y 102 heridos graves (CONASET, 2023). Tuvo un crecimiento de su parque automotor de 0.8 % y 3.5 % en los años 2022 y 2023 respectivamente según datos del INE (INE, 2023).

La infraestructura vial limitada cuenta con dos arterias principales para poder atravesar la ciudad, la Avenida Edmundo Perez Zujovic y la Avenida Pedro Aguirre Cerda, La segunda solo presente en una parte del recorrido, por lo que para el resto, se deben utilizar diferentes alternativas en similares alturas de la ciudad, ninguna de estas continuas. Esto posiciona a la Avenida Edmundo Perez Zujovic, la cual en el sector sur de la ciudad se encuentra con Avenida Grecia y Avenida Ejército, como la única vía continua que atraviesa la ciudad completamente. Esta particularidad dada las características geográficas de Antofagasta sumada a la alta concentración de vehículos, genera una alta congestión y riesgo de accidentes, especialmente en las zonas anteriormente mencionadas.

Actualmente, no existe algún sistema de monitoreo en tiempo real que permitan gestionar el tráfico de manera proactiva. Los sistemas de monitoreo tradicionales, como son la gestión de semáforos y el estudio de vías, es valioso para la administración del flujo vehicular y la estandarización de las intersecciones. Sin embargo, tiene limitaciones por ser estático para la vía en donde el diseño fue realizado. Este tipo de gestión tiene la limitación de no permitir una visión global del flujo vehicular, como también la poca flexibilidad ante los cambios que se generan en el entorno (Auld & Mohammadian,

2009).

Explorar nuevas fuentes de datos, como los eventos vehiculares en la ciudad, o la gestión a través de visión por computadora, son herramientas que permiten gestionar de forma eficiente y efectiva el flujo vehicular, proporcionando datos en tiempo real, también ofrecen opciones de automatización y predictibilidad (C. Chen et al., 2015). Contar con grandes volúmenes de datos permite desarrollar modelos inteligentes para la gestión del tráfico vehicular.

Waze recopila datos de eventos reportados por los usuarios, los cuales cuentan con tres aspectos principales; el tiempo, la ubicación, y el tipo de evento. Esta categorización permite poder desarrollar modelos que detecten patrones y puedan estimar la probabilidad de eventos futuros, basándose en datos del pasado. Esta opción entrega una visión global de la ciudad, para poder detectar focos de atención en cuanto a la ocurrencia de accidentes y congestión vehicular. El uso de datos colaborativos, como los reportados por usuarios en Waze, ha demostrado ser una fuente válida para el análisis y predicción de patrones de tráfico urbano, permitiendo detectar focos críticos de congestión y accidentes (Ferreira et al., 2017).

La plataforma Waze Cities, permite a ciudades poder obtener datos para gestionar el tráfico vehicular con herramientas inteligentes, esto conlleva a mejorar la seguridad y la eficiencia en el tráfico vehicular, usando una fuente de dato ya existente (Waze, 2024b).

2.2. Metodología

Se realiza un análisis geoespacial con el objetivo de identificar puntos críticos, como vías principales, calles secundarias y zonas de alto tráfico. Este análisis se llevó a cabo utilizando GeoPandas. Los resultados se presentan mediante técnicas de visualización que permiten interpretar las tendencias y puntos de interés de manera efectiva.

El pipeline de datos incluye una base de datos relacional para almacenarlos, un flujo ETL (Extract, Transform, Load) para procesarlos y un cliente web para visualizarlos. Se utilizó PostgreSQL (PostgreSQL Global Development Group, 2025) como base de datos SQL, Memcached (Memcached Development Team, 2025) como base de datos de caché, APScheduler para la programación de tareas y Dash (Plotly Technologies Inc., 2025) como herramienta de visualización. Este enfoque permitió automatizar el flujo de datos y garantizar la actualización constante de la información.

Para el proceso de entrenamiento del modelo, se llevó a cabo un balanceo de clases, debido a la naturaleza de los datos, solo contiene datos de la ocurrencia de eventos, esto genera sesgos en el modelo y un sobre-ajuste que hace que el modelo tenga un rendimiento disminuido (He & Garcia, 2009).

Adicionalmente, se entrenó un modelo de clasificación para determinar la probabilidad de ocurrencia de accidentes en diferentes puntos de la ciudad. El modelo seleccionado fue XGBoost (T. Chen & Guestrin, 2016), el cual fue seleccionado utilizando técnicas de validación cruzada y optimización de hiperparámetros (Géron, 2019). Para la selección de variables se utilizó GridSearchCV (Géron, 2019; Pedregosa et al., 2011), una técnica de búsqueda de hiperparámetros que permite encontrar la mejor combinación de variables para el modelo, se compararon modelos de regresión logística, árboles de decisión y XGBoost.

Para la gestión del modelo, en cuanto a su implementación, mantenimiento y actualización, se programó APScheduler, para ejecutar tareas periódicas que permiten gestionar el ciclo de vida de los modelos de aprendizaje automático, desde su entrenamiento hasta su despliegue en producción y versionado.

El servidor se desarrolló en Rust (Rust Team, 2024), un lenguaje de programación enfocado en la eficiencia y seguridad en el uso de memoria. Se utilizó Memcached para generar un caché de los datos y de esta forma poder servirlos de manera más eficiente

y rápida (Fitzpatrick, 2004), debido al volumen de datos, se priorizó la unicidad de los mismos, evitando generar copias, para no aumentar el uso de memoria.

Para el procesado de los datos en el cliente —Dash—, se generan múltiples workers que permiten servir la aplicación de forma eficiente, y se utiliza solo una instancia en memoria de los datos, compartida por todos los workers. Para mantener la integridad de los datos, se implementa un Mutex para controlar el acceso de lectura a los mismos desde las tareas asíncronas (Ramalho, 2015).

La perspectiva general del flujo de datos se muestra en la Figura 1 y el flujo desde la API de Waze hasta el dashboard se puede observar en la Figura 2

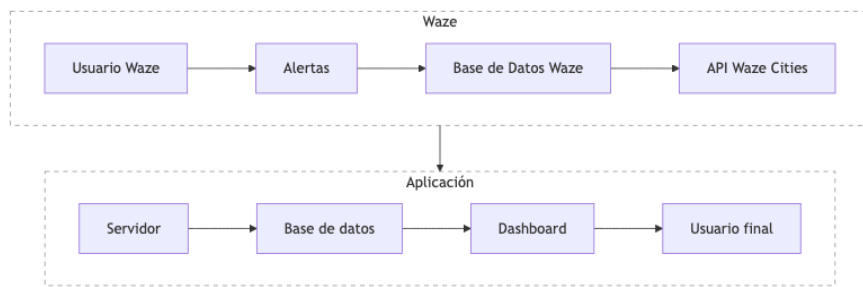


Figura 1: Pipeline general de datos

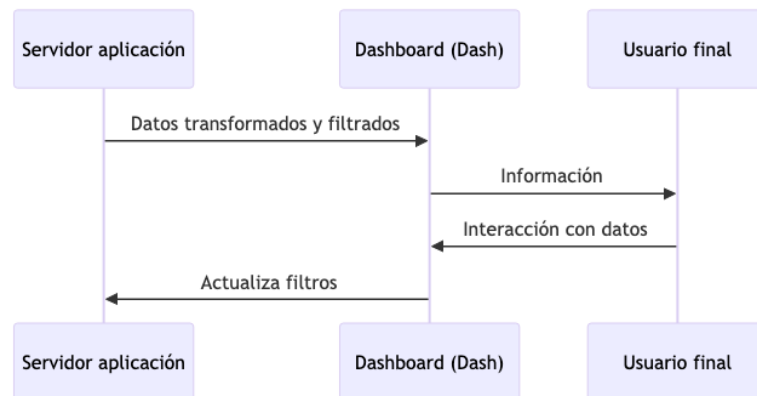


Figura 2: Flujo de información en dashboard

2.3. Objetivo general

Analizar y modelar el comportamiento del tráfico vehicular en Antofagasta mediante datos de la plataforma Waze, con el fin de desarrollar una herramienta predictiva que contribuya a la gestión eficiente y segura del tráfico urbano.

2.4. Objetivos específicos

- Realizar un análisis descriptivo de los datos recolectados para identificar patrones y tendencias relevantes en el comportamiento del tráfico.
- Presentar información visualmente comprensible y útil para la gestión vial, facilitando la implementación de políticas y acciones basadas en datos.
- Proponer una herramienta exploratoria para apoyar la toma de decisiones en la gestión vial

2.5. Alcances

Los datos utilizados corresponden al periodo desde octubre de 2024 hasta abril de 2025, en el sector urbano de la ciudad de Antofagasta, desde la zona norte urbana actual, sector Costa Laguna, hasta la zona sur, salida a camino de la minería, excluyendo el camino a puerto Coloso.

El área de estudio está delimitada por las coordenadas geográficas 23.35°S a 23.65°S de latitud y 70.36°O a 70.45°O de longitud, correspondientes al sistema de referencia EPSG:4326 (WGS84).

3. Marco teórico

El tráfico vehicular en entornos urbanos presenta un comportamiento complejo e impredecible, lo que dificulta su gestión eficiente. No obstante, el avance de las tecnologías móviles y la popularidad de aplicaciones como Waze permiten disponer de datos en tiempo real generados por los propios usuarios. Este proyecto se apoya en técnicas de análisis de datos y aprendizaje de máquinas (Machine Learning) para convertir esta información en herramientas útiles para la gestión vial. La utilización de datos geoespaciales, junto con la automatización de los procesos de recolección, análisis y visualización, constituye una solución costo-efectiva para mejorar la planificación del tráfico (Barceló & Casas, 2005).

3.1. Gestión de la seguridad vial

Existen diferentes estrategias para la gestión vehicular, desde sistemas manuales como la dirección del tránsito por carabineros; o más automatizados, como la gestión inteligente de semáforos. En términos de seguridad, para evitar la ocurrencia de accidentes, resulta conveniente aplicar la jerarquía de control de riesgo (Figura 3). Esta busca abordar desde la medida de mayor impacto, hasta la de menor impacto en la exposición al riesgo. Se divide en dos grupos principales, las barreras duras y las barreras blandas (o administrativas). El primer grupo consta de la eliminación del agente de riesgo, la sustitución por uno de menor exposición o medidas ingenieriles, que buscan administrar la exposición con algún rediseño u organización. Las medidas administrativas constan de diferentes estrategias que puedan cambiar el comportamiento, el cuidado o la visibilidad de la exposición, por último, los elementos de protección personal, que buscan mitigar el daño ante algún posible accidente (NIOSH, 2024).

La eliminación de la necesidad de transporte, puede ser abordada con estrategias como el fomento del teletrabajo para los puestos que lo permitan, o haciendo que las perso-

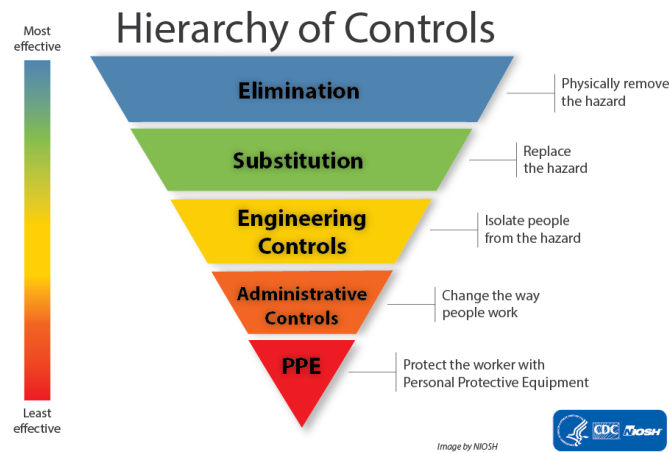


Figura 3: Jerarquía de control de riesgos (NIOSH, 2024).

nas se encuentren más cerca de sus trabajos, eliminando la necesidad de transporte en vehículo. La sustitución se genera fomentando el uso de transporte público, el uso de medios alternativos de transporte, como las bicicletas o el uso compartido de vehículos. Las medidas de ingeniería buscan optimizar el flujo vehicular, puede ser a través del rediseño del plano urbano, el diseño de vehículos más seguros, monitoreo a través de radares, cámaras y GPS, planificación de rutas y nuevas vías de tránsito. Como medidas administrativas se encuentran la educación vial, la señalización y los límites de velocidad. En cuanto a elementos de protección personal, se posiciona el uso de cinturón de seguridad, la instalación de airbags, barreras anti-vuelco y barreras de contención en las vías, entre otras.

3.2. Indicadores de bondad y ajuste

Para evaluar la relación entre variables cuantitativas y validar los supuestos estadísticos de los modelos aplicados, se emplearon indicadores clásicos de ajuste y asociación. En particular, se utilizaron pruebas de hipótesis, el coeficiente de correlación de Pearson y el coeficiente de correlación de Spearman, herramientas fundamentales en el análisis

estadístico inferencial (Devore, 2011; Field, 2018; Montgomery & Runger, 2018).

La **prueba de hipótesis** permite evaluar si existe suficiente evidencia en los datos para rechazar una afirmación nula sobre una población. En el caso de correlaciones, la hipótesis nula establece que no existe relación entre las variables ($H_0 : \rho = 0$), mientras que la hipótesis alternativa plantea la existencia de una correlación ($H_1 : \rho \neq 0$). El valor-p obtenido indica la probabilidad de observar un estadístico igual o más extremo bajo el supuesto de que la hipótesis nula es verdadera. Si el valor-p es inferior al nivel de significancia (α), se rechaza la hipótesis nula (Devore, 2011).

El **coeficiente de correlación de Pearson** (r) mide la intensidad y dirección de la relación lineal entre dos variables cuantitativas. Su valor varía entre -1 y 1 , donde valores cercanos a 1 indican una correlación positiva fuerte, valores cercanos a -1 una correlación negativa fuerte, y valores próximos a 0 una relación débil o inexistente. Este indicador se define como:

$$r = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2 \sum_{i=1}^n (y_i - \bar{y})^2}} \quad (1)$$

donde x_i y y_i representan los valores observados, y \bar{x} y \bar{y} sus respectivas medias (Montgomery & Runger, 2018).

No obstante, el coeficiente de Pearson asume linealidad y normalidad en las variables, condiciones que no siempre se cumplen en fenómenos reales como el tráfico vehicular. Por ello, se complementó el análisis mediante el estudio de los **residuos** del modelo, a fin de verificar la validez de los supuestos de normalidad, independencia y homocedasticidad. Los residuos, definidos como las diferencias entre los valores observados y los estimados, deben distribuirse aleatoriamente en torno a cero y sin patrones sistemáticos para garantizar que el modelo explique adecuadamente la relación entre las variables (Montgomery & Runger, 2018).

La evaluación de la normalidad de los residuos se realizó mediante la **prueba de D’Agostino–Pearson**, la cual combina medidas de asimetría y curtosis para contrastar la hipótesis de normalidad. Si el valor-p resultante es mayor que el nivel de significancia ($\alpha = 0.05$), no se rechaza la hipótesis nula, lo que indica que los residuos se distribuyen aproximadamente de forma normal y las desviaciones del modelo pueden considerarse aleatorias.

De forma complementaria, se incorporó el **coeficiente de correlación de Spearman** (ρ_s), una medida no paramétrica que evalúa la relación monótona entre dos variables a partir del orden de sus valores, sin requerir supuestos de normalidad ni homocedasticidad (Field, 2018). Este coeficiente se define como la correlación de Pearson aplicada sobre los rangos de las observaciones:

$$\rho_s = 1 - \frac{6 \sum_{i=1}^n d_i^2}{n(n^2 - 1)} \quad (2)$$

donde d_i representa la diferencia entre los rangos de cada par de observaciones y n el número total de datos. Al basarse en rangos, esta medida es más robusta frente a valores atípicos y distribuciones no normales, siendo especialmente útil cuando la relación entre las variables es creciente o decreciente, pero no estrictamente lineal.

Finalmente, para evaluar el ajuste de los modelos de regresión —tanto lineales como no lineales (exponencial inverso)— se emplearon indicadores de **bondad de ajuste**, como el coeficiente de determinación (R^2) y el error cuadrático medio (ECM). El primero mide la proporción de la variabilidad explicada por el modelo, mientras que el segundo cuantifica la magnitud promedio del error entre los valores observados y los predichos. En conjunto, estas métricas permiten analizar el grado de precisión del modelo y su capacidad de generalización.

En síntesis, el enfoque adoptado en este estudio combina métodos **paramétricos** y **no paramétricos**, junto con un análisis exhaustivo de residuos, para garantizar la validez

estadística y la robustez de las conclusiones. Este marco metodológico mixto permite que la inferencia sobre la relación entre congestión y accidentes se base tanto en fundamentos teóricos como en evidencia empírica contrastada.

3.3. Ciencia de datos para el análisis de comportamiento

Entender el comportamiento del tráfico vehicular, permite desarrollar medidas ingenieriles que disminuyan la probabilidad de accidentes, gestionando diversas variables como la congestión vehicular, los horarios con mayor probabilidad de accidentes y los sectores con mayor frecuencia de accidentes. La ciencia de datos permite, a través del uso de diferentes herramientas estadísticas, predecir con un grado de certeza conocido, la probabilidad de que ocurra un accidente tomando en cuenta variables que puedan ser determinantes para la ocurrencia de los mismos.

Usando datos de eventos pasados, es posible generar modelos que, a partir de una entrada, entreguen una salida determinada según ciertos parámetros, esto se conoce como análisis predictivo. El análisis predictivo consiste en entrenar modelos con datos históricos para anticipar resultados futuros, utilizando distintas técnicas de aprendizaje automático, las cuales pueden clasificarse en aprendizaje supervisado, no supervisado y por refuerzo (Bishop, 2006; Géron, 2019; Murphy, 2012).

En el aprendizaje supervisado, el modelo recibe tanto los datos de entrada como los de salida esperada, lo que le permite aprender patrones y realizar predicciones sobre nuevos datos con base en ese entrenamiento. El aprendizaje no supervisado utiliza datos de entrada, buscando estructuras o patrones en los mismos sin una clasificación específica, dejando la interpretación a cargo del especialista. Finalmente, en el aprendizaje por refuerzo, el modelo recibe una entrada, ejecuta una acción o predicción, y ajusta su comportamiento en función de una retroalimentación positiva o negativa, con el objetivo de maximizar el rendimiento a lo largo del tiempo.

Adicionalmente, los modelos pueden clasificarse según el tipo de salida que generan: modelos de regresión, que entregan valores numéricos continuos, y modelos de clasificación, que asignan las observaciones a una o más categorías previamente definidas. En estos últimos, el resultado se expresa como una distribución de probabilidades, y se considera como salida final la categoría con mayor probabilidad.

En este estudio se empleó un modelo supervisado de clasificación, con el objetivo de estimar la probabilidad de ocurrencia de un accidente, considerando dos posibles resultados: “ocurre” o “no ocurre”.

3.4. Evaluación de los modelos de Machine Learning

Para evaluar el rendimiento del modelo se utilizaron métricas ampliamente empleadas en problemas de clasificación binaria, tales como la exactitud (*accuracy*), la precisión (*precision*), la recuperación (*recall*) y la medida F1 (*F1-score*) (Manning et al., 2008). La exactitud representa el porcentaje total de predicciones correctas sobre el total de casos evaluados. La precisión corresponde a la proporción de casos positivos correctamente identificados por el modelo, es decir, el número de verdaderos positivos dividido por el total de predicciones positivas. Por su parte, la recuperación —también conocida como *recall*— mide la capacidad del modelo para identificar correctamente todos los casos positivos reales, dividiendo los verdaderos positivos por el total de casos positivos reales. La medida F1, en tanto, es la media armónica entre precisión y recuperación, y proporciona una visión equilibrada del desempeño del modelo cuando es necesario considerar tanto los errores por omisión como los errores por comisión.

Además, se construyó una matriz de confusión, la cual permite observar el número de aciertos y errores cometidos en cada clase —“ocurre” y “no ocurre”— lo que facilita una evaluación más detallada del comportamiento del modelo. Este análisis es clave para comprender sus limitaciones y fortalezas, especialmente en aplicaciones reales re-

lacionadas con la predicción de accidentes de tránsito.

Las fórmulas utilizadas para el cálculo de estas métricas son las siguientes:

$$\text{Exactitud (Accuracy)} = \frac{TP + TN}{TP + TN + FP + FN}$$

$$\text{Precisión (Precision)} = \frac{TP}{TP + FP}$$

$$\text{Recuperación (Recall)} = \frac{TP}{TP + FN}$$

$$\text{Medida F1 (F1-score)} = 2 \cdot \frac{\text{Precisión} \cdot \text{Recuperación}}{\text{Precisión} + \text{Recuperación}}$$

donde TP son los verdaderos positivos, TN los verdaderos negativos, FP los falsos positivos y FN los falsos negativos.

3.5. Algoritmos de clasificación: Random Forest, Logistic Regression y XGBoost

Para abordar problemas de clasificación binaria, como la predicción de ocurrencia o no ocurrencia de eventos de tráfico, se utilizaron algoritmos de aprendizaje supervisado. En particular, se emplearon los métodos *Random Forest*, *Logistic Regression* y *XGBoost*, los cuales han demostrado un alto rendimiento en tareas de clasificación estructurada (T. Chen & Guestrin, 2016; Friedman et al., 2001; Hosmer et al., 2013).

Random Forest es un algoritmo basado en el principio del ensamblado de modelos, específicamente mediante la construcción de múltiples árboles de decisión. Cada árbol es entrenado sobre una muestra aleatoria del conjunto de datos (con reemplazo), y en cada nodo del árbol se selecciona una subdivisión óptima a partir de un subconjunto aleatorio de características. Esta técnica, conocida como *bagging* (bootstrap aggregating), permite reducir la varianza del modelo y mejorar su capacidad de generalización. La predicción final se obtiene mediante votación mayoritaria entre todos los árboles del

conjunto (Breiman, 2001).

Logistic Regression es uno de los modelos estadísticos más utilizados para tareas de clasificación binaria. Se basa en la estimación de la probabilidad de pertenencia a una clase mediante la función logística o sigmoide, que transforma una combinación lineal de las variables predictoras en valores comprendidos entre 0 y 1. Este enfoque permite interpretar los coeficientes del modelo como efectos marginales sobre la probabilidad de ocurrencia del evento, manteniendo una alta interpretabilidad y simplicidad computacional. No obstante, su desempeño puede verse limitado ante relaciones no lineales o interacciones complejas entre las variables (Hosmer et al., 2013).

XGBoost (*Extreme Gradient Boosting*) es un algoritmo basado en el método de *boosting*, donde los árboles son construidos de manera secuencial. A diferencia del *bagging*, el *boosting* busca corregir los errores cometidos por los modelos anteriores, ajustando cada nuevo árbol a los residuos del conjunto anterior. XGBoost optimiza una función de pérdida regularizada utilizando técnicas avanzadas como la poda previa, el manejo eficiente de valores faltantes y una implementación paralelizada, lo que lo convierte en uno de los algoritmos más eficientes y precisos en clasificación tabular (T. Chen & Guestrin, 2016).

Estos algoritmos permiten capturar relaciones tanto lineales como no lineales entre las variables predictoras y la variable objetivo, siendo adecuados para trabajar con datos heterogéneos, como los recopilados desde fuentes de tráfico urbano. En la etapa de desarrollo se aplicaron los modelos para comparar su desempeño predictivo sobre el conjunto de datos estudiado.

3.6. Análisis geoespacial

Waze entrega las coordenadas geoespaciales de los eventos de tráfico en el sistema de referencia **EPSG:4326**. Este sistema corresponde al datum geodésico global **WGS84**

(**World Geodetic System 1984**), ampliamente utilizado en aplicaciones de posicionamiento global (GPS) y cartografía digital (IOGP, 2024b; NIMA, 2000). En EPSG:4326, las coordenadas están expresadas en grados decimales de latitud y longitud, lo cual es adecuado para visualización, pero no para realizar cálculos métricos directos como distancias o áreas.

Para llevar a cabo análisis espaciales cuantitativos sobre los eventos reportados por Waze —incluyendo cálculos de distancia, densidad y agrupación geográfica— fue necesario transformar dichas coordenadas a un sistema proyectado con unidades métricas.

La transformación se realizó utilizando la proyección Transversa de Mercator (Transverse Mercator), en su implementación como zona UTM 19 Sur (EPSG:32719). Esta proyección es conforme y está basada en el elipsoide WGS84, con un meridiano central en $\lambda_0 = -69^\circ$, un factor de escala $k_0 = 0.9996$, y un falso este de 500 000 m. Para el hemisferio sur, además, se aplica una corrección vertical de 10 000 000 m a la coordenada norte. Esto se detalla en el Apéndice A.

Esta transformación se aplicó mediante la función `to_crs` de la librería GeoPandas, la cual utiliza internamente la librería Pyproj como interfaz de la librería PROJ (GeoPandas Development Team, 2024; OSGeo, 2023; Snow et al., 2024).

La elección de EPSG:32719 permite minimizar la distorsión local, ya que el sistema UTM divide el globo en zonas longitudinales estrechas, optimizando la precisión en cada región. Al ser una proyección conforme y métrica, asegura que las distancias, áreas y densidades calculadas sobre el plano proyectado sean válidas y consistentes con el Sistema Internacional de Unidades (IOGP, 2024a).

3.7. Aprendizaje automático

A medida que la cantidad de datos aumenta, el modelo requiere ser reentrenado para incorporar los nuevos registros y así actualizar sus parámetros. Este proceso de aprendiza-

je continuo permite que el modelo se adapte a las variaciones que puedan presentarse en los patrones de tráfico vehicular a lo largo del tiempo. Para asegurar esta adaptabilidad, se programó una frecuencia de reentrenamiento mensual, lo que implica la incorporación de aproximadamente 9,244 nuevos datos al modelo cada mes. El reentrenamiento periódico resulta fundamental para mantener la vigencia y precisión del modelo, ya que permite integrar las tendencias más recientes y responder oportunamente a los cambios en las condiciones del tráfico (Gama et al., 2014).

Para la gestión de los modelos se utilizó MLFlow, un framework que permite gestionar las versiones de los modelos y almacenarlos en una base de datos, asegurando la persistencia de los mismos. También permite almacenar las diferentes métricas de cada versión del modelo (MLflow Development Team, 2025).

Para la gestión del modelo, en cuanto a su implementación, mantenimiento y actualización, se programó con APScheduler, una librería de tareas asíncronas en Python (Grönholm, 2025).

3.8. Investigaciones previas

La literatura sobre análisis y predicción del tráfico urbano integra tres líneas principales: (i) *fuentes de datos colaborativas y en tiempo real* para caracterizar el estado de la red; (ii) *modelos predictivos* (estadísticos y de aprendizaje automático) para estimar riesgo/ocurrencia de incidentes; y (iii) *métodos geo-espaciotemporales* para identificar y explicar patrones de concentración. De forma transversal, se estudian estrategias para enfrentar *datos desbalanceados*, *evaluación adecuada de clasificadores* y *actualización continua* ante *concept drift*.

Datos colaborativos y analítica en tiempo real. El uso de plataformas de reporte ciudadano (como Waze) y fuentes heterogéneas ha mostrado ser efectivo para caracterizar congestión y siniestralidad a escala urbana, permitiendo alimentar gemelos digitales y

tableros operacionales con latencias bajas (C. Chen et al., 2015; Ferreira et al., 2017; Waze, 2024b). Trabajos con simuladores meso/microscópicos (p. ej., AIMSUN) han servido para validar escenarios y políticas de control bajo supuestos realistas de demanda y oferta (Barceló & Casas, 2005).

Modelos predictivos de incidentes. En predicción de accidentes y eventos de tráfico, se han utilizado modelos desde regresiones generalizadas (p. ej., binomial negativa para frecuencias) hasta *ensemble* (Random Forest, Gradient Boosting/XGBoost) y redes neuronales, reportándose ventajas consistentes de los métodos no lineales frente a relaciones complejas entre variables (Berhanu et al., 2024; T. Chen & Guestrin, 2016; Lord & Mannering, 2010; Van Lint et al., 2005). Dado el interés práctico por interpretabilidad, técnicas de explicación pos-hoc como *SHAP* permiten descomponer la contribución de variables y favorecer decisiones de gestión con trazabilidad (Lundberg & Lee, 2017).

Análisis espacial y espaciotemporal. La identificación de *hotspots* de siniestralidad se ha abordado mediante estimación de densidad por núcleo (KDE) en red vial y pruebas de autocorrelación espacial (Moran's I, LISA) y de clústeres locales (Getis-Ord G_i^*), así como agrupamiento espaciotemporal (ST-DBSCAN) para capturar persistencia temporal y proximidad espacial de incidentes (Anselin, 1995; Birant & Kut, 2007; Getis & Ord, 1992; Moran, 1950; Xie & Yan, 2013). Estas herramientas complementan la modelación predictiva al ofrecer una lectura geométrica de la exposición y permiten priorizar intervenciones.

Desbalance de clases y generación de no-eventos. En problemas de seguridad vial es común el desbalance positivo/negativo. La literatura recomienda tanto *reponderación* de clases como *resampling* (p. ej., SMOTE) y, cuando no existen registros negativos explícitos, la generación controlada de no-eventos a partir de la combinación de rejillas temporales y segmentación espacial, cuidando la consistencia con el proceso generati-

vo (Chawla et al., 2002; Goedertier et al., 2009). En este trabajo se adopta una estrategia 1:1 de eventos simulados negativos, coherente con enfoques previos para clasificar ocurrencia/no-ocurrencia bajo restricciones espaciotemporales.

Evaluación de clasificadores en escenarios desbalanceados. Además de la curva ROC–AUC, la literatura recomienda la curva Precision–Recall (PR–AUC) por su mayor sensibilidad a la clase positiva cuando es minoritaria (Saito & Rehmsmeier, 2015). La comparación de modelos debe considerar ambas métricas y, cuando sea posible, análisis de *calibración* de probabilidades y matrices de costo específicas del dominio.

Actualización y deriva de concepto. Los patrones de tráfico presentan estacionalidad y cambios estructurales (obras viales, demografía, hábitos de movilidad), por lo que los modelos requieren políticas explícitas de reentrenamiento y monitoreo de desempeño para mitigar *concept drift* (Gama et al., 2014). La práctica de versionar modelos y registrar métricas/h_params facilita trazabilidad y gobierno del ciclo de vida.

En síntesis, el estado del arte respalda el uso de fuentes colaborativas, aprendizaje automático no lineal e instrumentos geo-espaciotemporales para comprender y predecir eventos de tráfico. Este estudio se alinea con dichas evidencias: integra datos colaborativos (Waze), compara algoritmos (Regresión Logística, Random Forest, XGBoost), aplica métodos espaciales y enfrenta el desbalance mediante no-eventos simulados y métricas apropiadas (ROC–AUC y PR–AUC).

4. Desarrollo

Se diseña una estrategia de recolección de datos que garantiza información con un nivel de certeza conocido. La API de Waze Cities, que proporciona información en tiempo real sobre eventos activos, fue la fuente principal de datos. Para lograr un volumen de datos representativo, se implementó un servidor encargado de recopilar y almacenar esta información de manera continua. El servidor fue desarrollado en Rust, utilizando la librería reqwest (McArthur, 2024) para extraer los datos desde la API de Waze Cities. Se recolectan los eventos de tráfico y se almacenan en una base de datos PostgreSQL. Este servidor se ejecuta de manera continua actualizando la información cada 2 minutos. La estructura de la base de datos se muestra en la Figura 4.

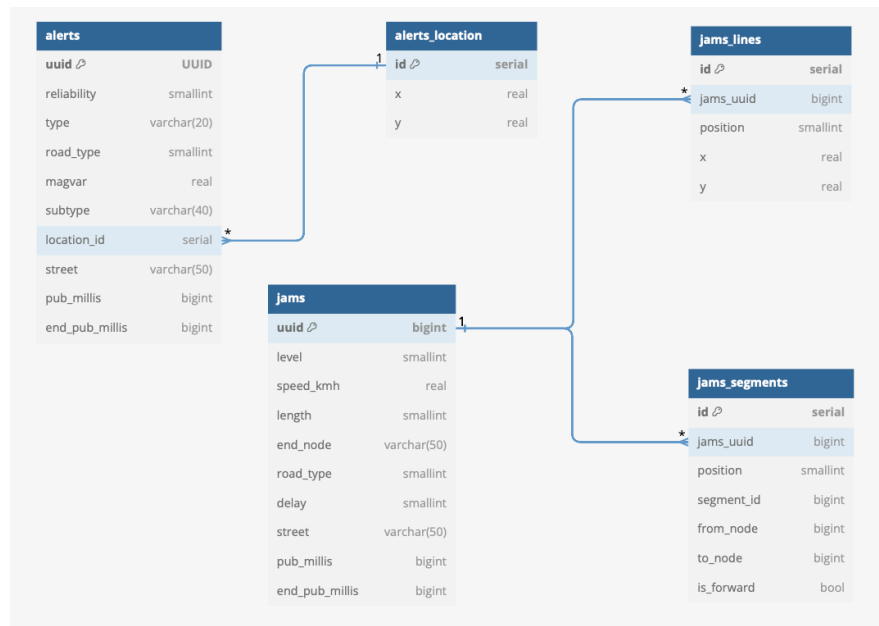


Figura 4: Diagrama de la base de datos

Existen dos estructuras principales, las alertas (alerts) y las congestiones (jams), el primero registra todos los eventos reportados por usuarios, tales como peligros, accidentes, congestión o rutas cerradas. En el caso de los eventos de congestión, son datos

generados automáticamente por Waze, los cuales usando la geolocalización, estiman congestiones en diferentes puntos de la ciudad. Para este proyecto, se utiliza la información de las alertas. Sin embargo, se almacenaron los datos de congestión, para un posterior análisis, que está fuera del alcance de este estudio. Esta estructura se definió con base en los datos relevantes y completos que proporciona la API, los tipos de datos disponibles pueden ser consultados en la documentación de Waze (Waze, 2024a).

Desde octubre de 2024 hasta abril de 2025, se capturan 53,159 alertas, que al ser filtradas (ver Subsección 4.1), finalmente se obtienen 26,336, las cuales se analizarán a lo largo de este estudio.

4.1. Filtrado temporal y agrupación de eventos

Las alertas pueden contener duplicados, debido a que diferentes usuarios pueden reportar el mismo evento. Para evitar esto, se filtraron los datos agrupándolos por segmento (Subsección 4.7) y periodos de tiempo, eliminando grupos duplicados en una ventana de tiempo de 120 minutos, es decir, todo evento que se reportó con el mismo tipo, en el mismo segmento, dentro de un periodo de dos horas, será considerado como un solo evento. Este procedimiento, resumido en el Algoritmo 1, permite homogeneizar la granularidad temporal de los eventos y asegurar la consistencia tipológica de los atributos (group, type). El código fuente completo se presenta en el Apéndice B.

De este modo, el filtro actúa como un mecanismo de depuración de ruido observacional, preservando un único registro representativo por evento real. Aunque esta consolidación produce una ligera disminución en las métricas de desempeño (ROC AUC y F1-score), el efecto se considera metodológicamente favorable, ya que aumenta la consistencia temporal y evita el sesgo hacia zonas con mayor densidad de usuarios o reportes. En consecuencia, el modelo resultante refleja con mayor precisión la ocurrencia efectiva de incidentes de tráfico, en lugar de su frecuencia de reporte.

Algoritmo 1 Filtrado temporal y agrupación de eventos

Require: Conjunto de datos D , tamaño de intervalo Δt (en minutos)

Ensure: Conjunto filtrado y agrupado D'

```
1:  $D' \leftarrow \emptyset$ 
2: if  $D = \emptyset$  or no existe columna pub_millis then
3:   return  $D'$ 
4: end if
5:  $D' \leftarrow$  Copia de  $D$ 
6:  $paso \leftarrow \Delta t \times 60000$  ▷ Conversión de minutos a milisegundos
7: for all evento  $\in D'$  do
8:   if pub_millis no es entero then
9:     evento.pub_millis  $\leftarrow$  Redondear(evento.pub_millis /  $10^6$ )
10:  end if
11:  evento.interval_start  $\leftarrow \lfloor \text{evento.pub\_millis} / \text{paso} \rfloor \times \text{paso}$ 
12: end for
13: Convertir interval_start a formato fecha-hora (UTC)
14: Asegurar tipo entero en group y tipo texto en type
15: Eliminar duplicados en [interval_start, group, type]
16: Eliminar columna auxiliar interval_start
17: Convertir pub_millis a formato fecha-hora (zona local)
18: Reiniciar índices de  $D'$ 
19: return  $D'$ 
```

4.2. Eventos en el espacio y tiempo

En la Figura 5 se observa la distribución espacial de los distintos tipos de eventos en la ciudad. Desde una perspectiva general se aprecian concentraciones en determinadas zonas de la ciudad, las cuales serán analizadas en futuros gráficos.

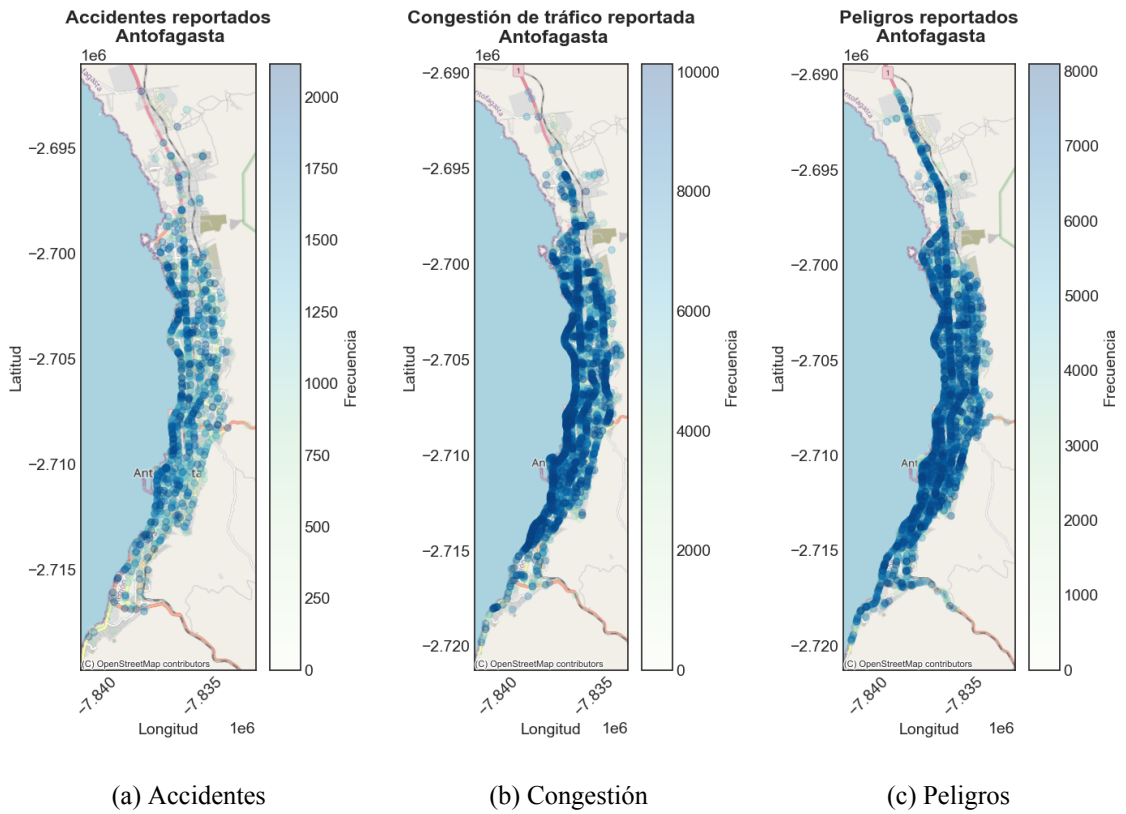
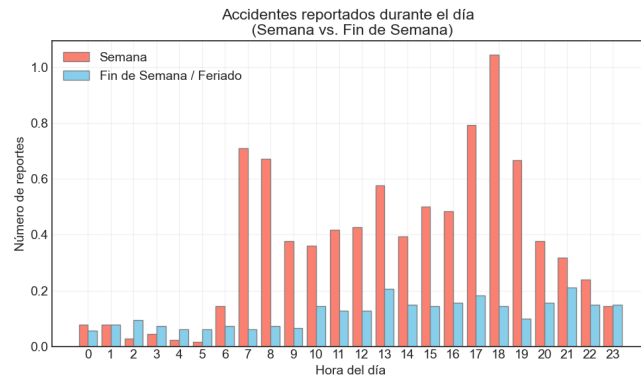
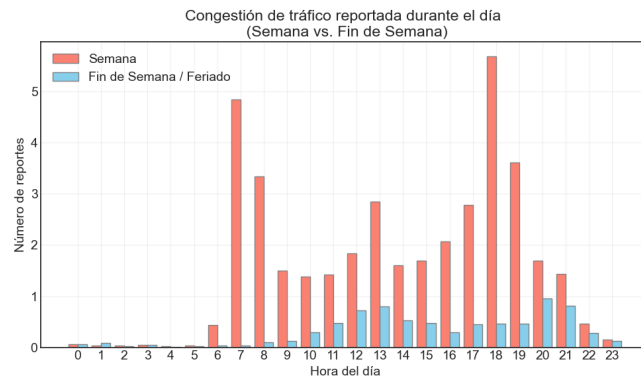


Figura 5: Distribución espacial de eventos en Antofagasta

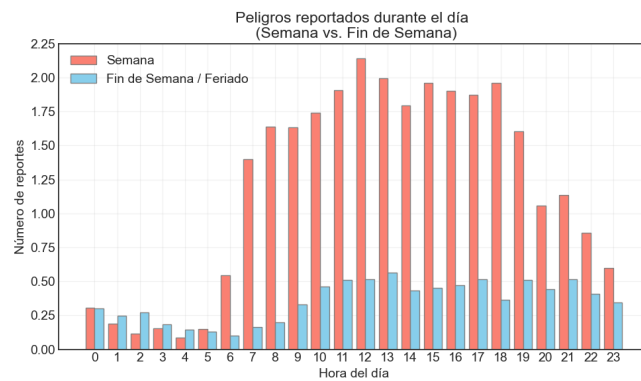
Para el análisis temporal, los eventos se agruparon por hora del día, calculando el promedio de eventos ocurridos por jornada. En la Figura 6 se evidencia una concentración de accidentes y congestiones durante las horas punta. En contraste, los eventos de tipo peligro presentan una distribución más uniforme a lo largo del día.



(a) Accidentes



(b) Congestión

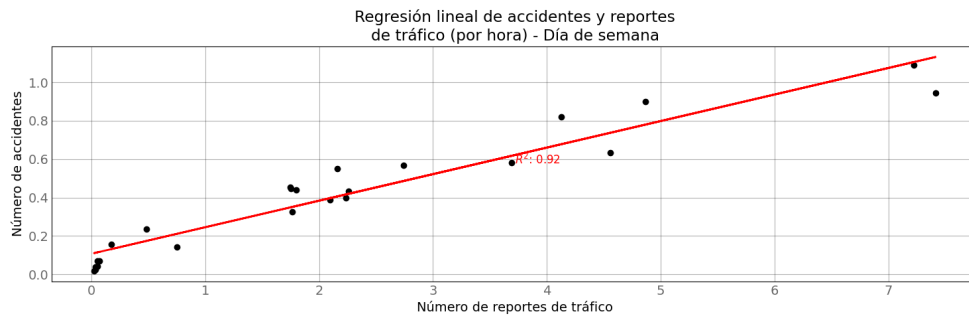


(c) Peligros

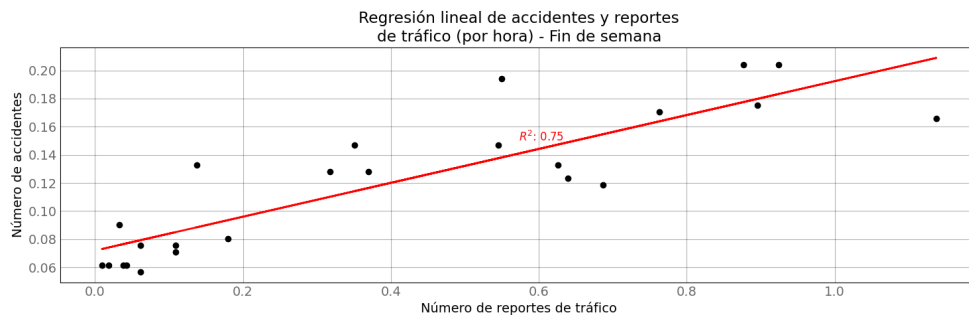
Figura 6: Eventos según hora del día en Antofagasta

4.3. Modelo de correlación lineal

Los resultados de la regresión lineal entre accidentes y reportes de congestión, presentados en la Figura 7, indican una fuerte correlación positiva durante los días de semana ($R^2 = 0.92$), lo que no implica necesariamente una causalidad. En cambio, durante los fines de semana la relación es más débil ($R^2 = 0.75$), aunque aún presente.



(a) Día de semana



(b) Fin de semana

Figura 7: Correlación lineal entre accidentes y congestión

4.4. Prueba de hipótesis de correlación lineal

Para evaluar la existencia de una relación estadísticamente significativa entre el número de reportes de congestión y la cantidad de accidentes por hora, se aplicó una prueba de correlación de Pearson.

La hipótesis nula (H_0) establece que no existe correlación lineal entre ambas variables, mientras que la hipótesis alternativa (H_1) plantea que sí existe tal correlación:

$$H_0 : \rho = 0 \quad (3)$$

$$H_1 : \rho \neq 0 \quad (4)$$

Donde ρ representa el coeficiente de correlación poblacional. El estadístico de prueba se calcula mediante:

$$t = \frac{r\sqrt{n-2}}{\sqrt{1-r^2}} \quad (5)$$

donde r es el coeficiente de correlación muestral y n el número de observaciones. El valor resultante se contrasta con la distribución t de Student con $n-2$ grados de libertad.

El análisis estadístico del modelo lineal entregó los siguientes resultados:

- Para días de semana:
 - El modelo ajustado fue $y = 0.1382x + 0.1072$
 - Se obtuvo un $R^2 = 0.92$
 - Coeficiente de Pearson: $r = 0.96$ (IC 95 %: [0.91, 0.98])
 - Significancia global: $F(1, 22) = 256.70, p < 0.0001$
 - Parámetros significativos: β_1 ($t = 16.02, p < 0.0001$); β_0 ($t = 4.08, p < 0.001$)
 - Estadístico t para la prueba de correlación: $t = 16.02, p < 0.0001$
- Para fines de semana:
 - El modelo ajustado fue $y = 0.1205x + 0.0719$

- Se obtuvo un $R^2 = 0.75$
- Coeficiente de Pearson: $r = 0.87$ (IC 95 %: [0.71, 0.94])
- Significancia global: $F(1, 22) = 66.23, p < 0.0001$
- Parámetros significativos: β_1 ($t = 8.14, p < 0.0001$); β_0 ($t = 9.24, p < 0.0001$)
- Estadístico t para la prueba de correlación: $t = 8.14, p < 0.0001$

Estos resultados permiten rechazar la hipótesis nula con un nivel de confianza del 99 % para los días de semana, confirmando la existencia de una relación lineal estadísticamente significativa entre el número de reportes de congestión y la cantidad de accidentes registrados por hora.

Los gráficos Q-Q (Figura 9) muestran una aproximación razonable a la distribución normal en ambos períodos. Para los días de semana, los puntos siguen la línea de referencia, con algunas desviaciones en los extremos, particularmente en los valores más altos. En el caso de fin de semana, la aproximación a la línea teórica es más consistente, con una distribución más cercana a la normal.

Los gráficos de residuos (Figura 8) revelan patrones distintos para cada período. En días de semana, aunque hay cierta dispersión alrededor de cero, se observa variabilidad considerable con algunos residuos positivos destacados (especialmente entre valores ajustados de 0.5 y 0.6) y residuos negativos pronunciados alrededor de 0.8. Para fines de semana, los residuos muestran menor magnitud global, pero un patrón menos aleatorio, sugiriendo posible estructura no capturada por el modelo lineal.

En resumen, el análisis confirma una relación lineal significativa entre congestión y accidentes, considerablemente más fuerte durante días laborables ($R^2 = 0.95$) que en fines de semana ($R^2 = 0.78$). Los patrones observados en los residuos, particularmente para fines de semana, sugieren la pertinencia de explorar modelos no lineales que

podrían capturar mejor las relaciones subyacentes en los datos.

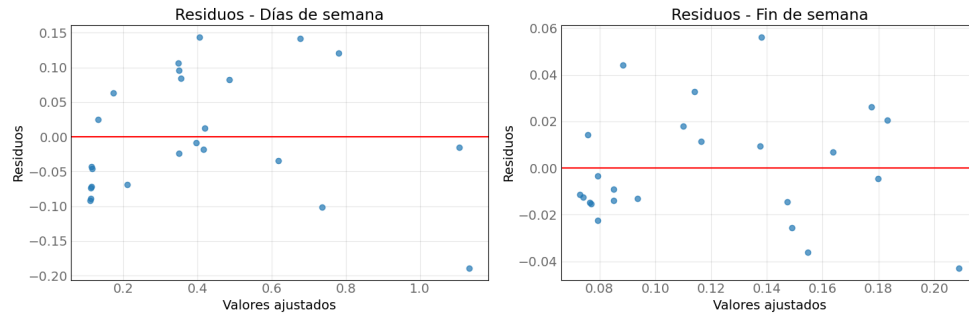


Figura 8: Residuos del modelo lineal para días de semana y fines de semana

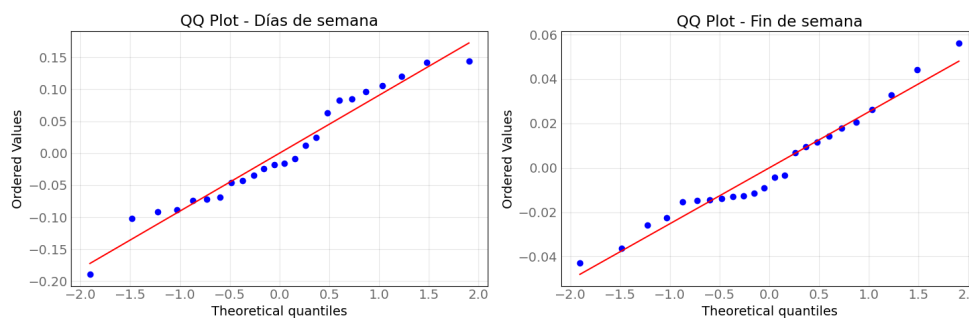


Figura 9: Gráficos Q-Q de los residuos del modelo lineal

A partir de la aplicación de la prueba de normalidad **D'Agostino–Pearson**, se obtuvo un valor de $p = 0.78$ para los días de semana y $p = 0.51$ para los fines de semana. En ambos casos, los valores de p resultan superiores al nivel de significancia ($\alpha = 0.05$), por lo que no se rechaza la hipótesis nula de normalidad. Esto sugiere que los residuos del modelo se distribuyen aproximadamente de forma normal, indicando que las desviaciones entre los valores observados y los estimados son esencialmente aleatorias. En consecuencia, puede inferirse que la relación modelada no se encuentra significativamente afectada por variables omitidas ni por errores sistemáticos.

4.5. Modelo de correlación exponencial inversa

Dado que la relación entre congestión y accidentes no necesariamente es lineal en todos los contextos —especialmente bajo alta densidad vehicular— se propuso un modelo exponencial inverso.

El modelo propuesto se ajusta a la siguiente forma funcional:

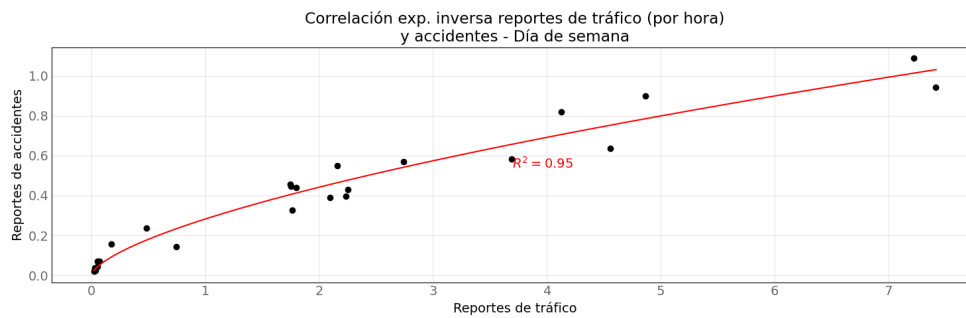
$$A(x) = \frac{\alpha}{x^\beta} + \gamma \quad (6)$$

donde:

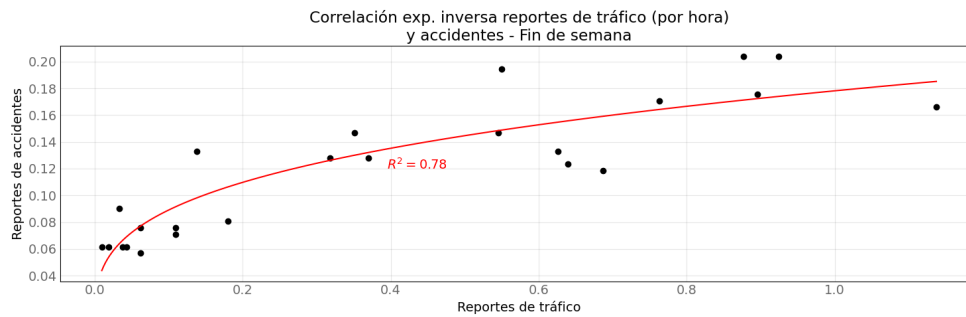
- $A(x)$ es el número estimado de accidentes por hora,
- x representa el número de reportes de congestión por hora,
- α , β y γ son parámetros ajustados mediante regresión no lineal.

Este modelo fue aplicado tanto a los días de semana como a los fines de semana. En ambos casos, se observó un ajuste visual considerable, respaldando la hipótesis de una relación decreciente no lineal entre ambas variables bajo ciertas condiciones de tráfico.

Los resultados del ajuste se presentan en la Figura 10, mostrando un ajuste significativo del modelo a los datos.



(a) Día de semana



(b) Fin de semana

Figura 10: Correlación exponencial inversa entre accidentes y congestión

El análisis estadístico del modelo exponencial inverso entregó resultados significativos tanto para días de semana como para fin de semana:

■ Para días de semana:

- Modelo ajustado: $y = \frac{0.2821}{x^{-0.6468}}$ (equivalente a $y = 0.2821 x^{0.6468}$)
- $R^2 = 0.9520$ ($R_{aj}^2 = 0.9499$)
- Significancia global: $F(2, 22) = 218.32, p < 0.0001$
- Parámetros: $a = 0.2821$ ($t = 13.80, p < 0.0001$, IC 95 %: $[0.2397, 0.3245]$), $b = -0.6468$ ($t = -13.50, p < 0.0001$, IC 95 %: $[-0.7461, -0.5475]$)

- Para fines de semana:

- Modelo ajustado: $y = \frac{0.1781}{x^{-0.3008}}$ (equivalente a $y = 0.1781 x^{0.3008}$)
- $R^2 = 0.7838$ ($R_{aj}^2 = 0.7740$)
- Significancia global: $F(2, 22) = 39.89, p < 0.0001$
- Parámetros: $a = 0.1781$ ($t = 20.55, p < 0.0001$, IC 95 %: $[0.1602, 0.1961]$), $b = -0.3008$ ($t = -7.48, p < 0.0001$, IC 95 %: $[-0.3841, -0.2174]$)

Este análisis revela diferencias importantes en la dinámica de la relación congestión-accidentes entre períodos. Durante los días de semana, el exponente negativo de mayor magnitud (-0.6319) indica una relación más sensible entre las variables, mientras que en fines de semana el exponente menor (-0.2776) sugiere una relación menos pronunciada.

Los gráficos de residuos (Figura 11) muestran una dispersión relativamente aleatoria en torno a cero, sin evidencias claras de heterocedasticidad ni patrones sistemáticos, lo que respalda la validez del ajuste no lineal propuesto. En particular, para los días de semana, los residuos muestran una distribución uniforme con algunos valores extremos (entre -0.1 y 0.2), mientras que para los fines de semana, la dispersión es menor y más equilibrada (aproximadamente entre -0.05 y 0.05).

Complementariamente, los gráficos Q-Q (Figura 12) revelan una aproximación a la distribución normal de los residuos en ambos casos, aunque con algunas desviaciones. En días de semana, se observan algunas desviaciones en los valores extremos, particularmente un punto notable en el extremo positivo. Para los fines de semana, la alineación con la línea teórica es excepcional, indicando una distribución muy cercana a la normal.

En conjunto, estos resultados indican que el modelo exponencial inverso no solo es estadísticamente significativo, sino que además presenta un mejor ajuste que el modelo

lineal, especialmente para días de semana donde alcanza un $R^2 = 0.94$ comparado con el $R^2 = 0.90$ del modelo lineal. La normalidad de los residuos y su distribución aleatoria fortalecen la hipótesis de que existe una relación no lineal entre la congestión y la probabilidad de accidentes en contextos urbanos, que puede ser modelada adecuadamente mediante una función exponencial inversa.

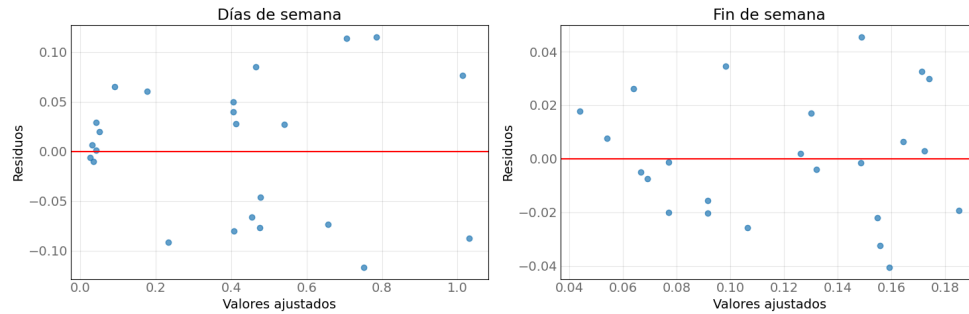


Figura 11: Residuos del modelo exponencial inverso para días de semana y fines de semana

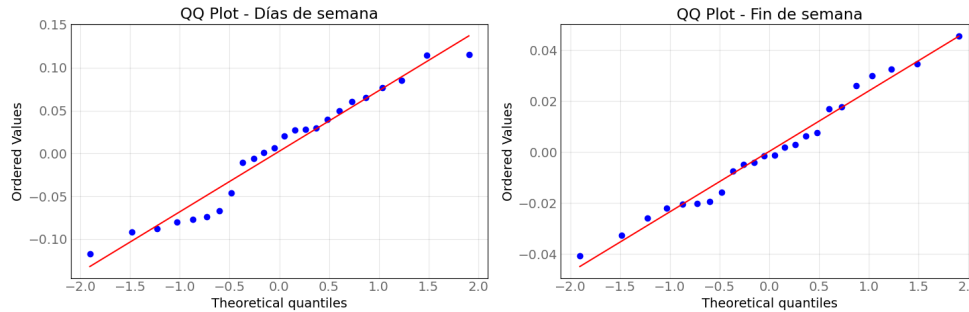


Figura 12: Gráficos Q-Q de los residuos del modelo exponencial inverso

En el caso del modelo exponencial, se aplicó igualmente la prueba de normalidad **D'Agostino–Pearson** sobre los residuos obtenidos. Para los días de semana se obtuvo un valor $p = 0.2579$, mientras que para los fines de semana el valor fue $p = 0.6500$. Dado que en ambos casos se cumple que $p > \alpha = 0.05$, no se rechaza la hipótesis

nula de normalidad. Esto indica que los residuos están distribuidos aproximadamente de manera normal, por lo que las variaciones no explicadas por el modelo pueden considerarse aleatorias y no evidencian la presencia de patrones sistemáticos o sesgos en la estimación.

4.6. Análisis de correlación no paramétrica de Spearman

Si bien los modelos lineal y exponencial inverso presentaron un ajuste adecuado y los residuos cumplieron con los supuestos de normalidad según la prueba de D'Agostino–Pearson, el análisis estadístico puede enriquecerse mediante la incorporación de un enfoque no paramétrico. Este tipo de métodos permite evaluar la relación entre variables sin requerir supuestos estrictos de normalidad, homocedasticidad ni linealidad, lo que otorga mayor robustez frente a posibles valores atípicos o distribuciones asimétricas.

En este contexto, se aplicó el **coeficiente de correlación de Spearman** (ρ_s), el cual mide la fuerza y dirección de la relación monótona entre dos variables a partir del orden de sus valores. Este enfoque es particularmente útil cuando la relación puede ser creciente o decreciente, pero no estrictamente lineal, como suele ocurrir en fenómenos de tráfico urbano donde el incremento de la congestión no siempre implica un aumento proporcional de accidentes.

El coeficiente de Spearman se define como la correlación de Pearson aplicada sobre los rangos de las observaciones, según la expresión:

$$\rho_s = 1 - \frac{6 \sum_{i=1}^n d_i^2}{n(n^2 - 1)} \quad (7)$$

donde d_i representa la diferencia entre los rangos de las variables x_i y y_i , y n corresponde al número total de observaciones. Los valores de ρ_s oscilan entre -1 y 1 , indicando respectivamente una correlación monótona negativa perfecta, nula o positiva perfecta.

Los resultados obtenidos para las correlaciones entre reportes de congestión y accidentes fueron los siguientes:

- Días de semana: $\rho_s = 0.95, p = 1.40 \times 10^{-12}$
- Fin de semana: $\rho_s = 0.85, p = 1.04 \times 10^{-7}$

En ambos casos, los valores p son significativamente inferiores al nivel de significancia adoptado ($\alpha = 0.01$), por lo que se rechaza la hipótesis nula de ausencia de correlación. Estos resultados confirman una relación monótona fuerte y positiva entre el número de reportes de congestión y la ocurrencia de accidentes, coherente con los hallazgos obtenidos en los modelos lineales y exponenciales.

La inclusión del método de Spearman aporta una **validación complementaria** desde una perspectiva no paramétrica, fortaleciendo la evidencia de asociación entre las variables analizadas. Dado que el análisis de residuos mostró normalidad y ausencia de patrones sistemáticos, la aplicación de ambos enfoques —paramétrico (Pearson y modelos lineales/exponenciales) y no paramétrico (Spearman)— permite asegurar una evaluación estadística rigurosa y robusta de la relación entre congestión y accidentes.

En síntesis, el uso combinado de modelos de regresión con análisis de residuos y correlaciones no paramétricas refuerza la consistencia de los resultados y la solidez de las conclusiones. Esta estrategia metodológica integra los beneficios de la inferencia clásica con la flexibilidad de los métodos robustos, asegurando que las conclusiones no dependan de supuestos restrictivos de distribución o forma funcional.

4.7. Frecuencia por zona

Para representar conjuntos estáticos de puntos espaciales, se utiliza la estructura Kd-tree que computa distancias relativas mediante spiral codes y las almacena usando Directly Addressable Codes (DACs), optimizando el uso de memoria sin sacrificar eficiencia en

consultas espaciales (Gutiérrez et al., 2023). Cada zona es llamada cuadrante o segmento.

En la Figura 13 se puede ver cómo se distribuyen los segmentos numéricamente, comenzando desde el segmento 1 en la esquina inferior izquierda. La opacidad del color rojo está relacionada con la cantidad de accidentes por segmento.

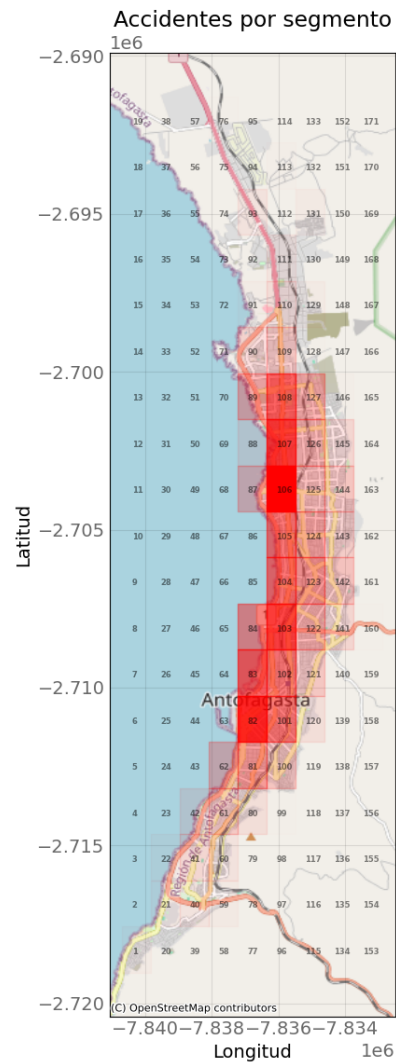


Figura 13: Identificación de cuadrantes en mapa de Antofagasta

Considerando la media de accidentes diarios, se agrupa por segmento, se define que las

zonas mayores a 0.5 accidentes por día son considerados críticos, se construye un mapa según la criticidad (14a), al tomar la suma de los accidentes, se obtiene la 14b que muestra la cantidad de accidentes reportados en el segmento en el periodo de estudio, que a su vez se clasifican como críticos si poseen más de 90 accidentes.

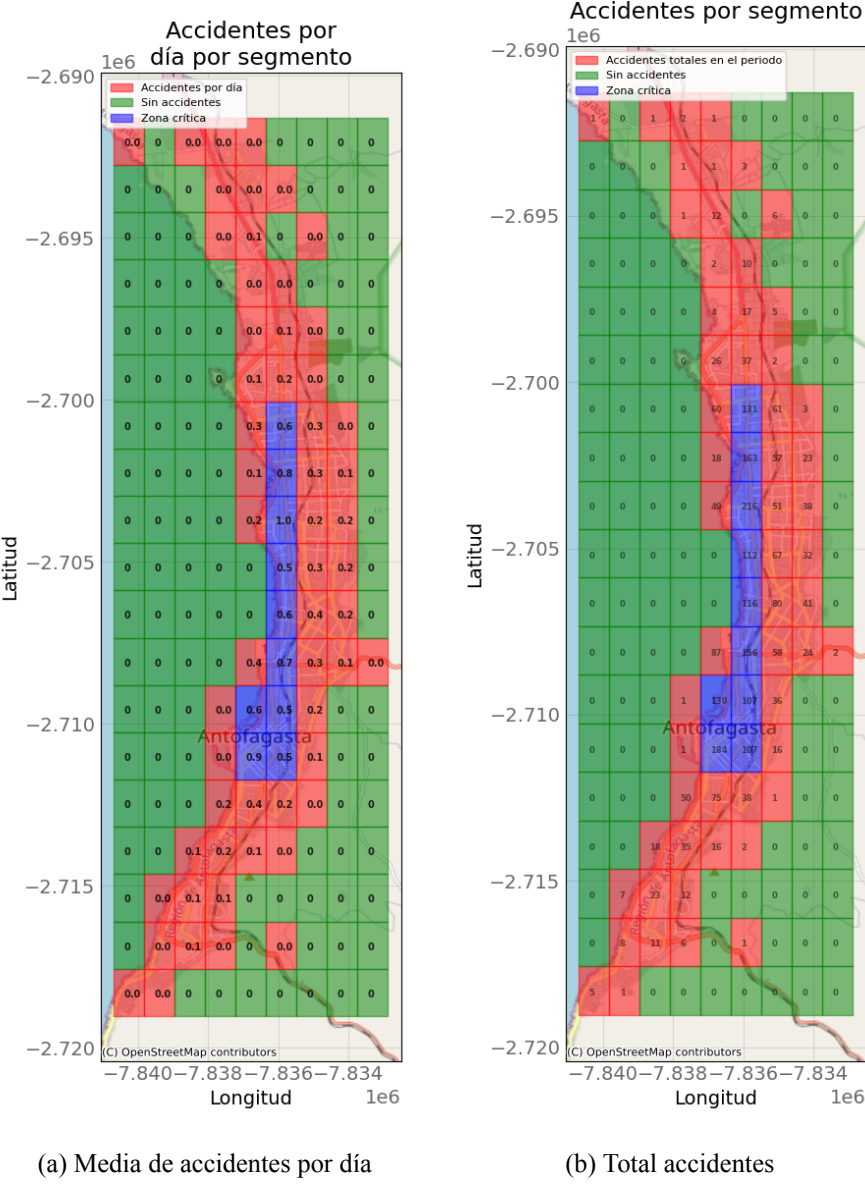


Figura 14: Accidentes por segmento

4.8. Selección del modelo de Machine Learning

Se evaluaron tres modelos de clasificación para seleccionar el con mejor rendimiento, para ello primero se seleccionaron los mejores hiperparámetros para cada uno basado en la curva ROC-AUC, realizando una búsqueda exhaustiva (*GridSearchCV*) sobre los tres modelos: **Random Forest**, **Logistic Regression** y **XGBoost**.

Random Forest. Se evaluaron los siguientes hiperparámetros:

- `n_estimators`: [200, 350, 500]
- `max_depth`: [None, 10, 20]
- `min_samples_leaf`: [1, 3]
- `class_weight`: [None, 'balanced']

Los mejores hiperparámetros obtenidos fueron:

- `n_estimators`: 500
- `max_depth`: 20
- `min_samples_leaf`: 3
- `class_weight`: None

Con estos valores, el modelo alcanzó una puntuación máxima de **0.8414** en validación cruzada (métrica ROC-AUC).

Regresión Logística. Se evaluaron los siguientes hiperparámetros:

- `clf__penalty`: [l2]
- `clf__solver`: ['lbfgs', 'liblinear']
- `clf__C`: [0.1, 1.0, 10.0]

- `clf__class_weight`: [None, 'balanced']

Los mejores hiperparámetros obtenidos fueron:

- `clf__penalty`: l2
- `clf__solver`: lbfgs
- `clf__C`: 0.1
- `clf__class_weight`: balanced

El modelo obtuvo una puntuación de **0.7566** en validación cruzada (ROC–AUC).

XGBoost. Se evaluaron los siguientes hiperparámetros:

- `colsample_bytree`: [0.8, 1.0]
- `gamma`: [0.8, 1.0]
- `learning_rate`: [0.1, 0.03]
- `max_depth`: [3, 6, 10, 20]
- `min_child_weight`: [1, 3]
- `n_estimators`: [80, 100]
- `subsample`: [0.8, 1.0]

Los mejores hiperparámetros obtenidos fueron:

- `colsample_bytree`: 0.8
- `gamma`: 1.0
- `learning_rate`: 0.1
- `max_depth`: 10

- min_child_weight: 1
- n_estimators: 80
- subsample: 0.8

Con estos valores, el modelo XGBoost alcanzó la mejor puntuación promedio en validación cruzada, con un **ROC–AUC de 0.8480**, confirmando su superior capacidad de generalización frente a los otros modelos.

En el Tabla 1 se puede observar la comparación de los modelos.

Modelo	Mejores Hiperparámetros	ROC AUC	PR AUC
Random Forest	class_weight: None max_depth: 10 max_features: 2 max_leaf_nodes: 8 min_samples_leaf: 2 n_estimators: 100	0.8394	0.8073
XGBoost	colsample_bytree: 0.7 gamma: 1 learning_rate: 0.1 max_depth: 30 n_estimators: 80	0.8433	0.8108
Regresión Logística	penalty: l2 solver: lbfgs max_iter: 1000	0.7463	0.6590

Cuadro 1: Comparativa de modelos ajustados con GridSearchCV según el área bajo las curvas ROC y Precision–Recall (PR AUC).

Los gráficos de la curva ROC están representados en la Figura 15 y de la curva Precision-Recall en la Figura 16.

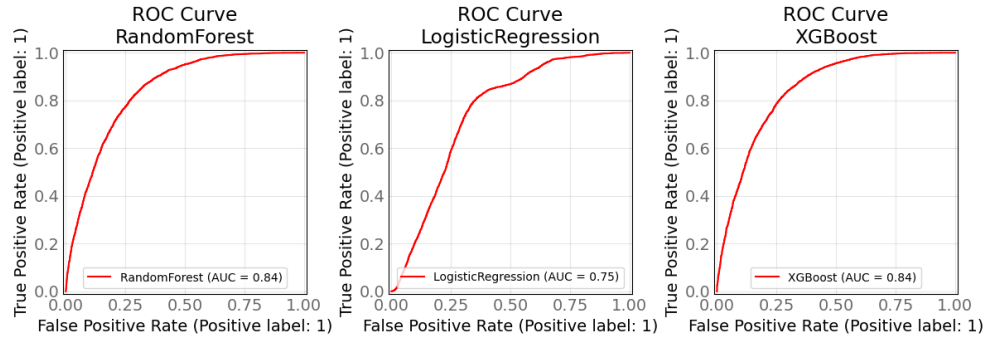


Figura 15: Curvas ROC

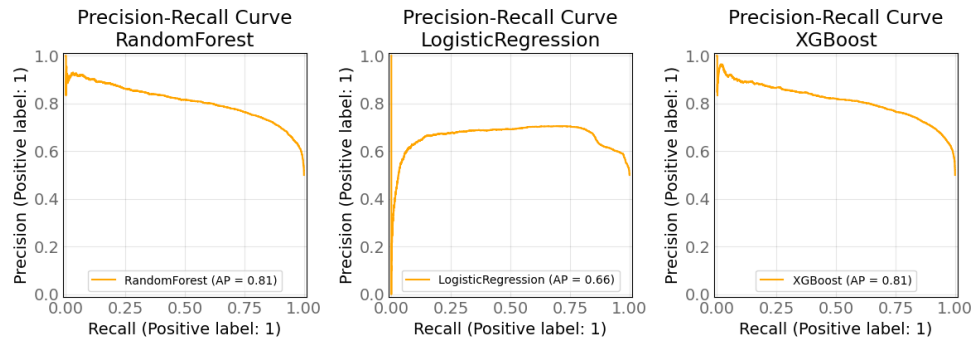


Figura 16: Curvas Precision-Recall

4.9. Generación de eventos negativos simulados

Dado que el conjunto original contiene únicamente eventos positivos (ocurridos), fue necesario generar instancias negativas (no ocurridos) para entrenar un modelo de clasificación binaria. Para ello, se implementó una estrategia inspirada en el enfoque de generación de eventos negativos artificiales propuesto por Goedertier et al. (Goedertier

et al., 2009), el cual permite representar problemas secuenciales como tareas de clasificación supervisada.

A diferencia del algoritmo original, en el caso del tráfico vehicular se adoptó una relación 1:1 entre eventos positivos y negativos —esto es, se genera un evento negativo por cada evento positivo—, priorizando la eficiencia mediante operaciones vectorizadas y una estructura de datos matricial. Además, debido a que cada evento es independiente de los demás, no se aplicó la noción de causalidad ni las restricciones de paralelismo introducidas por Goedertier, propias de los procesos secuenciales.

En este contexto, la generación de eventos negativos se basa en una grilla temporal con intervalos de 5 minutos, sobre la cual se combinan todas las posibles ubicaciones (segmento) y tipos de evento. Se identifican aquellas combinaciones que no se encuentran en los datos originales —donde no ocurrió un evento— y se etiquetan como eventos negativos (`happen = 0`). Estas instancias se muestrean aleatoriamente hasta igualar la cantidad de eventos positivos, obteniendo así un conjunto balanceado con un 50 % de instancias positivas y 50 % negativas.

Este enfoque permite preservar la estructura temporal y geoespacial de los datos reales, evitando introducir ruido artificial y facilitando una evaluación más robusta de métricas de desempeño como `recall` y `precision`.

El procedimiento se resume en el Algoritmo 2. El código completo se incluye en el Apéndice C.

Algoritmo 2 Generación de eventos negativos artificiales

Require: Conjunto de eventos positivos D , atributo geográfico g , tamaño de intervalo

$t = 5$ minutos

Ensure: Conjunto balanceado D' con eventos positivos y negativos

- 1: Convertir marcas de tiempo a milisegundos y definir intervalos temporales de tamaño t
 - 2: Agrupar eventos por combinación de $(intervalo, g, tipo)$
 - 3: Generar todas las combinaciones posibles de $(intervalo, g, tipo)$
 - 4: Identificar combinaciones ausentes en D como eventos negativos ($happen = 0$)
 - 5: Muestrear aleatoriamente los eventos negativos hasta igualar la cantidad de positivos
 - 6: Asignar ubicaciones aleatorias dentro del mismo grupo geográfico g
 - 7: Combinar eventos positivos y negativos en un único conjunto balanceado D'
 - 8: **return** D'
-

4.10. Resultados del modelo seleccionado

La versión final del modelo (v6) se entrenó utilizando los hiperparámetros obtenidos durante la etapa de validación cruzada. El modelo fue registrado y versionado mediante *MLflow*, permitiendo un seguimiento detallado de sus parámetros y métricas.

El modelo utiliza codificación *one-hot* y un esquema de clasificación binaria (*binary: logistic*), con una semilla aleatoria (*random_state = 42*) para garantizar la reproducibilidad (Géron, 2019)

En cuanto al rendimiento, se obtuvieron los siguientes resultados para una muestra de 52,672 eventos, con un 50 % de eventos ocurridos y un 50 % de no ocurridos. Esta proporción se logró mediante la simulación de eventos negativos sobre una grilla temporal de 5 minutos, generando combinaciones posibles de tiempo, localización y tipo de even-

to que no están presentes en los datos originales (Subsección 4.9). Esto permite evaluar el modelo en condiciones balanceadas, facilitando la interpretación de métricas como *precision*, *recall* y *F1-score*.

- **Accuracy:** 0.8795
- **Recall:** 0.9076
- **Precision:** 0.8583
- **F1-score:** 0.8823
- **Promedio validación cruzada (10 folds):** 0.7819

Respecto a la matriz de confusión, se registraron los siguientes valores:

- Verdaderos positivos: 5,539
- Verdaderos negativos: 5,843
- Falsos positivos: 965
- Falsos negativos: 595

Estas métricas evidencian un modelo con gran capacidad para identificar correctamente los casos positivos, sin sacrificar precisión ni equilibrio. La validación cruzada muestra estabilidad a lo largo de los distintos *fold*s, lo que refuerza la robustez del modelo final y confirma que no hubo sobreentrenamiento (Géron, 2019).

4.11. Curva de aprendizaje del modelo

La Figura 17 presenta la curva de aprendizaje del modelo *XGBoost* entrenado. Se observa un desempeño alto y estable tanto en el conjunto de entrenamiento como en el de validación (valores de exactitud entre 0.80 y 0.89), con una brecha reducida y constante entre ambas curvas.

Este comportamiento indica que el modelo ha alcanzado un nivel de generalización adecuado, sin evidencias de sobreajuste ni subajuste. Además, la forma aproximadamente horizontal de las curvas sugiere que el incremento del tamaño del conjunto de entrenamiento no produciría mejoras significativas en el rendimiento, lo que refleja una saturación en la capacidad de aprendizaje con las características actuales (C. Chen et al., 2015; Raschka & Mirjalili, 2018).

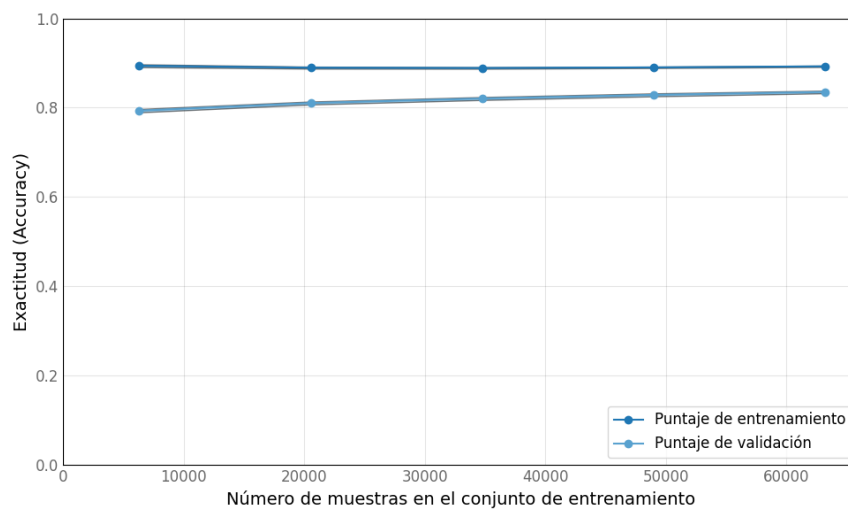


Figura 17: Curva de aprendizaje del modelo *XGBoost*: desempeño en entrenamiento con validación cruzada estratificada de 20 particiones.

El entrenamiento del modelo está programado para realizarse cada 30 días, utilizando los nuevos datos. Los datos del dashboard se actualizan cada 5 minutos.

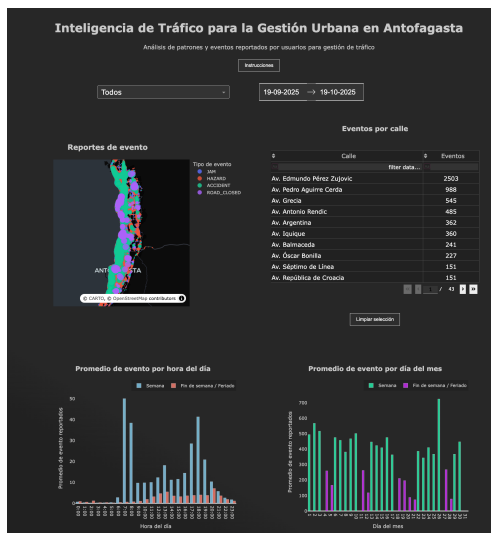
4.12. Despliegue del modelo y panel de visualización

Con el objetivo de facilitar la interpretación de los resultados y el uso práctico del modelo predictivo, se desarrolló un *dashboard* interactivo que integra tanto los análisis descriptivos como las predicciones generadas por el sistema. En la Figura 18 se presenta una vista global del panel; en particular, la sección de análisis descriptivo se ilustra

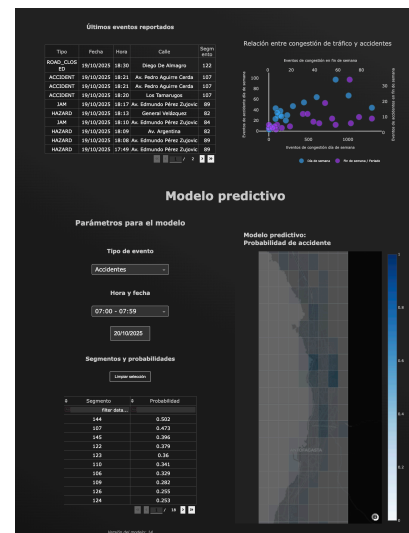
en la 18a y la sección dedicada al modelo predictivo en la 18b.

El *dashboard* fue implementado en *Python* utilizando el framework *Dash*, integrando directamente los resultados del modelo *XGBoost* con componentes visuales interactivos. La interfaz cuenta con filtros por calle y rango de fechas, posibilitando una exploración detallada de los eventos registrados. En la sección predictiva, el usuario puede ajustar los parámetros de entrada —como fecha, hora y tipo de evento (accidente, congestión u peligro)— para obtener proyecciones específicas del comportamiento del tráfico.

El sistema se encuentra desplegado en línea y disponible públicamente a través del sitio web <https://traficoantofagasta.com>. La estructura técnica y la arquitectura del sistema (cliente y *backend*) se describen de forma general en el Apéndice D.



(a) Sección de análisis descriptivo



(b) Sección del modelo predictivo

Figura 18: Vista general del *dashboard* desarrollado para el monitoreo y predicción del tráfico vehicular.

5. Discusiones

5.1. Patrones espaciales y temporales del tráfico

El análisis integrado de datos geoespaciales y temporales permitió identificar patrones críticos de siniestralidad en Antofagasta. Destacan como focos principales los segmentos del eje Pedro Aguirre Cerda, especialmente el cruce con Nicolás Tirado (segmentos 106–109), así como la intersección Salvador Allende–Edmundo Pérez Zujovic (segmentos 84 y 103), donde confluyen flujos significativos de vehículos desde distintos sectores de la ciudad. El centro urbano (segmento 82) también presenta una elevada concentración de incidentes.

El análisis temporal evidenció una correlación entre la frecuencia de accidentes y la congestión del tráfico, en línea con la literatura sobre tráfico urbano (Berhanu et al., 2024). Esto sugiere que podría existir una relación de causalidad entre ambas variables, lo cual debe ser evidenciado utilizando métodos adicionales para la investigación.

5.2. Desempeño y comparación de modelos predictivos

En la modelación estadística, se observó una correlación significativa entre la congestión reportada y la ocurrencia de accidentes, especialmente en días laborales. El modelo exponencial inverso logró un ajuste superior ($R^2 = 0.96$) respecto al modelo lineal ($R^2 = 0.92$).

El modelo XGBoost logró un desempeño destacado en la predicción de accidentes, con un **F1-score de 0.88** y un **Recall de 0.91** en el conjunto de validación. Si bien no se realizó una comparación directa y sistemática con todos los modelos clásicos de series temporales, los resultados obtenidos sugieren que los algoritmos de aprendizaje automático como XGBoost pueden capturar relaciones no lineales y complejas entre las variables predictoras.

El análisis temporal sugiere una mayor frecuencia de accidentes en días hábiles y horarios punta. Se debe contrastar esta relación utilizando nuevas fuentes de información como puede ser la visión por computadora o evaluaciones en terreno (Goodall & Lee, 2019).

5.3. Aporte del análisis geoespacial

El enfoque geoespacial permitió visualizar con claridad zonas críticas y patrones de concentración que no serían evidentes mediante análisis tabulares convencionales. Las visualizaciones generadas, como la 14b y 14a, ofrecen una herramienta valiosa para autoridades y planificadores, facilitando la priorización de intervenciones y el diseño de políticas viales focalizadas. Sin embargo, la calidad de estos análisis depende de la completitud y precisión de los datos de referencia, lo cual puede limitar su aplicabilidad en zonas con cartografía incompleta o baja densidad de reportes.

5.4. Implicancias prácticas y futuras aplicaciones

El sistema desarrollado evidencia el potencial de los datos colaborativos y el aprendizaje automático para mejorar la gestión del tráfico en ciudades de tamaño medio. Su arquitectura modular y su enfoque basado en datos abiertos facilitan la replicabilidad en otras urbes chilenas. Además, la actualización periódica del modelo y la visualización en tiempo real abren la puerta a su integración con sistemas de gestión vial, aplicaciones móviles o incluso plataformas de respuesta ante emergencias y eventos masivos.

Futuras aplicaciones podrían incluir la incorporación de datos meteorológicos, información de infraestructura o variables de comportamiento humano, así como la exploración de modelos más avanzados (por ejemplo, LSTM o transformers) que capturen mejor la naturaleza secuencial y dinámica del tráfico.

5.5. Limitaciones del estudio

El presente trabajo enfrentó diversas restricciones asociadas principalmente a la disponibilidad y calidad de los datos utilizados. Destaca la ausencia de fuentes independientes que permitieran contrastar y validar físicamente los reportes de incidentes, así como la falta de acceso a información relevante sobre la operación de semáforos y las características específicas de los accidentes (por ejemplo, gravedad, causa o participantes involucrados). Esta carencia de datos complementarios limita la posibilidad de validar exhaustivamente los resultados obtenidos. Adicionalmente, no se contó con evidencia en video ni registros visuales que permitieran corroborar la ocurrencia y naturaleza de los eventos reportados.

La generalización de los resultados debe abordarse con cautela, considerando las particularidades urbanas y culturales de cada ciudad, así como las limitaciones del enfoque aplicado. Finalmente, la adopción de modelos complejos puede suponer un desafío adicional en cuanto a la transparencia y comprensión de los resultados por parte de los usuarios finales.

6. Conclusiones

Este estudio demuestra la factibilidad y el impacto del uso de datos colaborativos para la gestión inteligente del tráfico urbano en Antofagasta. Los principales aportes incluyen:

- Identificación precisa de zonas críticas con alta concentración de accidentes.
- Confirmación de una correlación estadísticamente significativa entre congestión y accidentes, especialmente durante días laborales.
- Validación de modelos no lineales y de aprendizaje automático como herramientas para la predicción de incidentes en contextos urbanos densos.
- Desarrollo e implementación de un sistema funcional de monitoreo, visualización y actualización periódica del modelo, con potencial de escalabilidad.

El enfoque propuesto es viable y adaptable, constituyendo una base sólida para futuras iniciativas de movilidad urbana orientadas a la seguridad y eficiencia.

7. Recomendaciones

En base a los hallazgos, se sugiere:

- **Intervención focalizada:** Priorizar mejoras de infraestructura, señalización y fiscalización en los segmentos críticos identificados.
- **Monitoreo y actualización:** Mantener y ampliar la infraestructura de captura y análisis de datos, integrando nuevas fuentes (sensores, cámaras) y promoviendo la actualización continua del modelo.
- **Validación cruzada:** Complementar los análisis con estudios de campo y encuestas a usuarios para validar y enriquecer los hallazgos.
- **Escalabilidad:** Replicar y adaptar la metodología a otras ciudades chilenas con características similares, atendiendo a sus particularidades.
- **Investigación avanzada:** Incorporar nuevas variables (clima, infraestructura, comportamiento humano) y explorar modelos predictivos de mayor complejidad y capacidad explicativa.

Estas recomendaciones buscan fortalecer una gestión vial proactiva y basada en datos, orientada a la reducción de accidentes y la mejora sostenida de la movilidad urbana.

Referencias

- Anselin, L. (1995). Local Indicators of Spatial Association—LISA. *Geographical Analysis*, 27(2), 93-115. <https://doi.org/10.1111/j.1538-4632.1995.tb00338.x>
- Auld, J., & Mohammadian, A. (2009). Framework for modeling activity planning as a sequence of planned activities. *Transportation Research Part B: Methodological*, 43(6), 924-937. <https://doi.org/10.1016/j.trb.2009.01.001>
- Barceló, J., & Casas, J. (2005). Dynamic network simulation with AIMSUN. En R. Kitamura & M. Kuwahara (Eds.), *Simulation Approaches in Transportation Analysis: Recent Advances and Challenges* (pp. 57-98). Springer. https://doi.org/10.1007/0-387-24109-4_3
- Berhanu, Y., Schröder, D., Wodajo, B. T., & Alemayehu, E. (2024). Machine learning for predictions of road traffic accidents and spatial network analysis for safe routing on accident and congestion-prone road networks. *Results in Engineering*, 24, 102737. <https://doi.org/10.1016/j.rineng.2024.102737>
- Birant, D., & Kut, A. (2007). ST-DBSCAN: An algorithm for clustering spatial-temporal data. *Data & Knowledge Engineering*, 60(1), 208-221. <https://doi.org/10.1016/j.datak.2006.01.013>
- Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.
- Breiman, L. (2001). Random Forests. *Machine Learning*, 45(1), 5-32. <https://doi.org/10.1023/A:1010933404324>
- Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research*, 16, 321-357. <https://doi.org/10.1613/jair.953>
- Chen, C., Zhang, J., & Antoniou, C. (2015). Data-driven approaches to traffic prediction and management. *Transportation Research Procedia*, 10, 12-24. <https://doi.org/10.1016/j.trpro.2015.09.050>

- Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785-794. <https://doi.org/10.1145/2939672.2939785>
- CONASET. (2023). *Informe de accidentes de tránsito* (inf. téc.). Comisión Nacional de Seguridad de Tránsito. Antofagasta, Chile. <https://www.conaset.cl>
- Devore, J. L. (2011). *Probabilidad y estadística para ingeniería y ciencias* (8.^a ed.). Cengage Learning.
- Ferreira, N., Fink, C., & Wilson, A. (2017). Traffic Observations from Waze: Crowdsourced Traffic Data as an Alternative Data Source for Transportation Agencies. *Transportation Research Board 96th Annual Meeting Compendium of Papers*, (17-02468). <https://trid.trb.org/view/1439020>
- Field, A. (2018). *Discovering Statistics Using IBM SPSS Statistics* (5.^a ed.). SAGE Publications Ltd.
- Fitzpatrick, B. (2004). Distributed Caching with Memcached. *Linux Journal*, 2004(124), 5. <https://www.linuxjournal.com/article/7451>
- Friedman, J., Hastie, T., & Tibshirani, R. (2001). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer.
- Gama, J., Žliobaitė, I., Bifet, A., Pechenizkiy, M., & Bouchachia, A. (2014). A Survey on Concept Drift Adaptation. *ACM Computing Surveys*, 46(4), 44:1-44:37. <https://doi.org/10.1145/2523813>
- GeoPandas Development Team. (2024). *GeoPandas Documentation* (v0.14.3) (Ver. 0.14.3). <https://geopandas.org>
- Géron, A. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems* (2.^a ed.). O'Reilly Media.

- Getis, A., & Ord, J. K. (1992). The Analysis of Spatial Association by Use of Distance Statistics. *Geographical Analysis*, 24(3), 189-206. <https://doi.org/10.1111/j.1538-4632.1992.tb00261.x>
- Goedertier, S., Martens, D., Vanthienen, J., & Baesens, B. (2009). Robust Process Discovery with Artificial Negative Events. *Journal of Machine Learning Research*, 10, 1305-1340.
- Goodall, N., & Lee, E. (2019). Comparison of Waze crash and disabled vehicle records with video evidence. *Transportation Research Interdisciplinary Perspectives*, 1, 100019. <https://doi.org/10.1016/j.trip.2019.100019>
- Grönholm, A. (2025). *APScheduler: Advanced Python Scheduler* (Ver. 3.10.4). <https://pypi.org/project/APScheduler>
- Gutiérrez, G., Torres-Avilés, R., & Caniupán, M. (2023). cKd-tree: A Compact Kd-tree. *IEEE Access*, 12, 22641-22655. <https://doi.org/10.1109/ACCESS.2024.3365054>
- He, H., & Garcia, E. A. (2009). Learning from Imbalanced Data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9), 1263-1284. <https://doi.org/10.1109/TKDE.2008.239>
- Hosmer, D. W., Lemeshow, S., & Sturdivant, R. X. (2013). *Applied Logistic Regression* (3rd). John Wiley & Sons.
- INE. (2023). *Estadísticas parque automotriz* (inf. téc.). Instituto Nacional de Estadísticas. Santiago, Chile. <https://regiones.ine.gob.cl/antofagasta/estadisticas-regionales/economia/transporte-y-comunicaciones/parque-de-vehiculos>
- IOGP. (2024a). *EPSG:32719 – WGS 84 / UTM zone 19S*. International Association of Oil & Gas Producers. <https://epsg.io/32719>
- IOGP. (2024b). *EPSG:4326 – WGS 84*. International Association of Oil & Gas Producers. <https://epsg.io/4326>

- Lord, D., & Mannering, F. (2010). The statistical analysis of crash-frequency data: A review and assessment of methodological alternatives. *Accident Analysis & Prevention*, 44(1), 1-14. <https://doi.org/10.1016/j.aap.2010.02.015>
- Lundberg, S. M., & Lee, S.-I. (2017). A Unified Approach to Interpreting Model Predictions. En I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan & R. Garnett (Eds.), *Advances in Neural Information Processing Systems* 30 (pp. 4765-4774). Curran Associates, Inc. <https://proceedings.neurips.cc/paper/2017/file/8a20a8621978632d76c43dfd28b67767-Paper.pdf>
- Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press.
- McArthur, S. (2024). *Reqwest Documentation*. crates.io. <https://docs.rs/reqwest/latest/reqwest>
- Memcached Development Team. (2025). *Memcached Documentation*. Ver. 1.6.22. <https://memcached.org>
- MLflow Development Team. (2025). *MLflow: A Machine Learning Lifecycle Platform* (Ver. 2.10.0). <https://mlflow.org/docs/latest>
- Montgomery, D. C., & Runger, G. C. (2018). *Applied Statistics and Probability for Engineers* (7.^a ed.). John Wiley & Sons.
- Moran, P. A. P. (1950). Notes on Continuous Stochastic Phenomena. *Biometrika*, 37(1/2), 17-23. <https://doi.org/10.2307/2332142>
- Murphy, K. P. (2012). *Machine Learning: A Probabilistic Perspective*. MIT Press.
- NIMA. (2000). *Department of Defense World Geodetic System 1984: Its Definition and Relationships with Local Geodetic Systems* (inf. téc. N.º TR8350.2). National Imagery y Mapping Agency. Bethesda, MD. <https://earth-info.nga.mil/php/download.php?file=coord-wgs84>

- NIOSH. (2024). *Hierarchy of Controls* (inf. téc.) (Accedido el 9 de julio de 2025). National Institute for Occupational Safety y Health. Atlanta, GA. <https://www.cdc.gov/niosh/topics/hierarchy/default.html>
- OSGeo. (2023). *PROJ Coordinate Transformation Software Library*. Open Source Geospatial Foundation. <https://proj.org>
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12, 2825-2830. <https://www.jmlr.org/papers/volume12/pedregosa11a/pedregosa11a.pdf>
- Plotly Technologies Inc. (2025). *Dash: Python Web Framework for Building Analytical Web Applications* (Ver. 2.14.2). <https://plotly.com/dash>
- PostgreSQL Global Development Group. (2025). *PostgreSQL Documentation*. Ver. 16.1. <https://www.postgresql.org>
- Ramalho, L. (2015). *Fluent Python: Clear, Concise, and Effective Programming*. O'Reilly Media.
- Raschka, S., & Mirjalili, V. (2018). *Python Machine Learning* (2.^a ed.). Packt Publishing.
- Rust Team. (2024). *The Rust Programming Language*. Rust Foundation. <https://www.rust-lang.org>
- Saito, T., & Rehmsmeier, M. (2015). The Precision-Recall Plot Is More Informative than the ROC Plot When Evaluating Binary Classifiers on Imbalanced Datasets. *PLOS ONE*, 10(3), e0118432. <https://doi.org/10.1371/journal.pone.0118432>
- Snow, A., et al. (2024). *pyproj: Python interface to PROJ (v3.6.1)* (Ver. 3.6.1). <https://pyproj4.github.io/pyproj>
- Snyder, J. P. (1987). *Map Projections: A Working Manual*. U.S. Geological Survey. <https://pubs.er.usgs.gov/publication/pp1395>

- Van Lint, J. W. C., Hoogendoorn, S. P., & Van Zuylen, H. J. (2005). Accurate free-way travel time prediction with state-space neural networks under missing data. *Transportation Research Part C: Emerging Technologies*, 13(5-6), 347-369. <https://doi.org/10.1016/j.trc.2005.03.001>
- Waze. (2024a). *Waze API Documentation*. Google LLC. <https://support.google.com/waze/partners/answer/13458165>
- Waze. (2024b). Waze for Cities Case Studies. *Google LLC*. <https://www.waze.com/wazeforcities/casestudies>
- Xie, Z., & Yan, J. (2013). Kernel density estimation of traffic accidents in a network space. *Accident Analysis & Prevention*, 53, 1-7. <https://doi.org/10.1016/j.aap.2012.12.010>

A. Transformación de coordenadas UTM

La proyección UTM utilizada en este estudio (EPSG:32719) se basa en la proyección Transversa de Mercator, adaptada al hemisferio sur y centrada en el meridiano $\lambda_0 = -69^\circ$. Esta proyección es conforme y está diseñada para minimizar la distorsión métrica en una zona longitudinal específica (zona 19S), siendo especialmente adecuada para representar regiones como Antofagasta.

A continuación se presentan las fórmulas que permiten convertir coordenadas geográficas (ϕ, λ) —latitud y longitud en radianes— a coordenadas planas proyectadas (x, y) en metros, basadas en el elipsoide WGS84:

Parámetros del elipsoide WGS84

- Semieje mayor: $a = 6378137.0$ m
- Aplanamiento: $f = \frac{1}{298.257223563}$
- Excentricidad: $e^2 = 2f - f^2$
- Factor de escala: $k_0 = 0.9996$
- Falso este: $E_0 = 500\,000$ m
- Falso norte (hemisferio sur): $N_0 = 10\,000\,000$ m

Variables intermedias

$$e'^2 = \frac{e^2}{1 - e^2}$$

$$N = \frac{a}{\sqrt{1 - e^2 \sin^2 \phi}}$$

$$T = \tan^2 \phi$$

$$C = e'^2 \cos^2 \phi$$

$$A = (\lambda - \lambda_0) \cos \phi$$

Cálculo del arco meridiano

$$\begin{aligned} M = a \cdot & \left[\left(1 - \frac{e^2}{4} - \frac{3e^4}{64} - \frac{5e^6}{256} \right) \phi \right. \\ & - \left(\frac{3e^2}{8} + \frac{3e^4}{32} + \frac{45e^6}{1024} \right) \sin(2\phi) \\ & + \left(\frac{15e^4}{256} + \frac{45e^6}{1024} \right) \sin(4\phi) \\ & \left. - \left(\frac{35e^6}{3072} \right) \sin(6\phi) \right] \end{aligned}$$

Coordenadas proyectadas

$$\begin{aligned} y = N_0 + k_0 & \left[M + N \tan \phi \left(\frac{A^2}{2} \right. \right. \\ & \left. \left. + \frac{A^4}{24} (5 - T + 9C + 4C^2) + \frac{A^6}{720} (61 - 58T + T^2 + 600C - 330e'^2) \right) \right] \end{aligned} \quad (8)$$

Estas expresiones corresponden a un desarrollo en serie basado en una expansión de Taylor, que permite una transformación precisa desde la superficie curva del elipsoide a

un plano cartesiano. Estas fórmulas son utilizadas internamente por la biblioteca PROJ, y fueron documentadas formalmente por Snyder (Snyder, 1987).

Nota

La implementación computacional de estas fórmulas puede realizarse utilizando bibliotecas como Pyproj, que encapsulan estos desarrollos y gestionan automáticamente los parámetros de cada sistema de referencia (CRS).

B. Código fuente: Filtrado temporal y agrupación de eventos

```
def filter_by_group_time(self, timedelta_min: int, inplace: bool = False) ->
    ↪ gpd.GeoDataFrame | pd.DataFrame:
    """
    Filter and group data by time intervals.
    """
    if self.data is None or "pub_millis" not in self.data.columns:
        return pd.DataFrame()

    data = self.data.copy() if not inplace else self.data

    if not np.issubdtype(data["pub_millis"].dtype, np.integer):
        data["pub_millis"] = round(data["pub_millis"].astype(np.int64,
            ↪ errors="ignore") / 1_000_000).astype(np.int64)

    step = np.int64(60_000 * timedelta_min)
    data["interval_start"] = ((data["pub_millis"]).to_numpy() // step) * step

    data["interval_start"] = pd.to_datetime(data["interval_start"], unit="ms",
    ↪ utc=True)
    data["interval_start"] = data["interval_start"].dt.tz_convert(utils.TZ)

    data["group"] = data["group"].astype(np.int16)
    data["type"] = data["type"].astype(str)

    data.drop_duplicates(subset=["interval_start", "group", "type"],
    ↪ inplace=True)
    data.drop(columns=["interval_start"], inplace=True)

    if not pd.api.types.is_datetime64_any_dtype(data["pub_millis"]):
```

```
data["pub_millis"] = pd.to_datetime(data["pub_millis"], unit="ms",
    ↪ utc=True)
data["pub_millis"] = data["pub_millis"].dt.tz_convert(utils.TZ)

data.reset_index(drop=True, inplace=True)
return data
```

Código:

<https://github.com/richardhapb/antof-traffic-client/blob/72c1e2cea730670bffb58f0844cd71d1888481df/src/analytics/alerts.py#L100>

Nota: La función utiliza la zona horaria definida en el módulo `utils.TZ` y mantiene la consistencia tipológica de las columnas de entrada.

Referencias:

<https://github.com/richardhapb/antof-traffic-client/blob/72c1e2cea730670bffb58f0844cd71d1888481df/src/utils/utils.py#L27>

C. Código fuente: Generación de eventos negativos

```
def generate_neg_simulated_data(data: pd.DataFrame | GeoDataFrame, geodata: str
↳ = "group") -> GeoDataFrame:
    """
    Genera datos negativos simulando puntos sin evento.
    Basado en la adaptación del método de Goedertier et al. (2009).
    """
    # Conversión temporal y generación de intervalos de 5 minutos
    if not isinstance(data["pub_millis"].iloc[0], np.integer):
        data["pub_millis"] = data["pub_millis"].astype(np.int64,
↳ errors="ignore") / 1_000_000

    step = np.int64(60_000 * 5)
    min_tms = data["pub_millis"].to_numpy().min()
    max_tms = data["pub_millis"].to_numpy().max()
    intervals = np.arange(min_tms, max_tms + step, step)

    # Combinación de intervalos, grupos y tipos de evento
    data["interval"] = ((data["pub_millis"].to_numpy() - min_tms) // step) *
↳ step + min_tms
    allgroups = data[geodata].unique()
    alltypes = data["type"].unique()
    combinations = pd.MultiIndex.from_product(
        [intervals, allgroups, alltypes], names=["interval", geodata, "type"]
    ).to_frame(index=False)

    # Eventos originales
    event_combinations = data[["uuid", "interval", geodata, "type", "x", "y"]]
    event_combinations["happen"] = 1
    event_combinations["location"] = join_coords(event_combinations) # función
↳ auxiliar
```

```

# Unión y muestreo balanceado
merged = pd.merge(combinations, event_combinations, on=["interval",
→ geodata, "type"], how="left")
merged_pos = merged[merged["happen"] == 1]
merged_neg = merged[merged["happen"].isna()].sample(len(merged_pos),
→ random_state=42)
full_data = pd.concat([merged_pos, merged_neg])

# Agregación y etiquetado
full_data["happen"] = full_data["happen"].fillna(0).astype(int)
alerts = utils.generate_aggregate_data(full_data) # módulo auxiliar
return alerts.data

```

Código:

<https://github.com/richardhapb/antof-traffic-client/blob/72c1e2cea730670bffb58f0844cd71d1888481df/src/analytics/ml.py#L172>

Nota: Las funciones `join_coords()` y `generate_aggregate_data()` son funciones auxiliares definidas en el módulo `utils`, utilizadas para concatenar coordenadas geográficas y agregar estadísticas por grupo usadas en el modelo final, respectivamente.

Referencias:

<https://github.com/richardhapb/antof-traffic-client/blob/72c1e2cea730670bffb58f0844cd71d1888481df/src/utils/utils.py#L260>

<https://github.com/richardhapb/antof-traffic-client/blob/72c1e2cea730670bffb58f0844cd71d1888481df/src/utils/utils.py#L95>

D. Estructura general del sistema y despliegue

El sistema desarrollado se compone de dos componentes principales: un cliente, implementado en *Python*, y un servidor, desarrollado en *Rust*. Ambos módulos operan de manera conjunta para procesar los datos de tráfico, ejecutar los modelos de predicción y ofrecer un entorno de visualización accesible mediante un *dashboard* web.

Cliente

El cliente reúne los análisis exploratorios, la preparación de datos, el entrenamiento de los modelos y la interfaz visual. Su estructura principal se divide en los siguientes módulos:

- **Módulo analytics:** contiene las rutinas de análisis estadístico, correlaciones, agrupamiento y entrenamiento de los modelos predictivos.
- **Módulo dashboard:** implementa el panel de visualización construido con *Dash*, que integra filtros por calle, fecha y tipo de evento, junto con parámetros ajustables para generar predicciones personalizadas.
- **Módulo utils:** agrupa funciones auxiliares para la limpieza y exploración de los datos.
- **Módulo waze:** maneja la ingesta de reportes provenientes de la API de Waze, los cuales sirven como fuente principal del conjunto de datos.

La ejecución del cliente se gestiona mediante el archivo principal `main.py`, que coordina el flujo completo: desde la carga de datos hasta la visualización final en el *dashboard*. El sistema fue configurado para su despliegue con `gunicorn` y puede ejecutarse tanto en entorno local como remoto.

Servidor

El servidor fue desarrollado en *Rust* e implementa una arquitectura ligera basada en un servicio web que actúa como intermediario entre el modelo predictivo y la interfaz visual. Sus principales responsabilidades incluyen la administración de las solicitudes del *Cliente*, el acceso a la base de datos y la gestión de caché.

Despliegue y acceso

El sistema completo se encuentra desplegado en línea y disponible públicamente en <https://traficoantofagasta.com>, donde los usuarios pueden consultar tanto las métricas descriptivas como las predicciones generadas por el modelo. El diseño modular del proyecto permite su actualización y ampliación futura, integrando nuevas fuentes de datos o modelos de predicción más avanzados.