SAMSUNG INNOVATION CAMPUS 2024

GRUPO 3
CIUDAD DE MÉXICO

SISTEMA PREDICTIVO DE ACCIDENTES DE TRÁNSITO MEDIANTE IA

Alumnos:
García Pérez Dariana Mildred
Maldonado González César Abdiel
Méndez Macías Rogelio Leonardo
Valerio Carera Ricardo

Equipo 7

Ciudad de México, 5 de abril de 2025

Resumen

Este documento presenta el desarrollo de un sistema predictivo de accidentes de tránsito utilizando técnicas de inteligencia artificial. El objetivo principal del proyecto es predecir la cantidad de accidentes y clasificar el tipo de colisión. Se utilizaron modelos de aprendizaje automático como Regresión Lineal, Random Forest, XGBoost y Redes Neuronales, para analizar datos históricos de accidentes y generar predicciones. Los resultados obtenidos demuestran la viabilidad de utilizar IA para asistir en la búsqueda de la mejora para la seguridad vial.

Palabras clave: Seguridad vial, accidentes de tránsito, inteligencia artificial, aprendizaje automático.

Índice

1 Introducción

La seguridad vial representa un desafío global con importantes implicaciones sociales. A nivel mundial, los accidentes de tránsito son una de las principales causas de muerte y lesiones. Según datos de la Organización Mundial de la Salud (OMS) [1]

En México, la situación requiere atención urgente. Según datos del Instituto Nacional de Estadística y Geografía (INEGI), los accidentes de tránsito terrestre en zonas urbanas y suburbanas alcanzaron la cifra de 381,048 en el año 2023. [2]

Este proyecto desarrolla un sistema predictivo de accidentes de tránsito integrando técnicas de aprendizaje automático, abordando tres tipos de problemas fundamentales en IA: clasificación (predictor binario de accidentes), regresión (estimación de severidad de accidentes) y agrupamiento (identificación de patrones de riesgo).

2 Revisión de Literatura

2.1 Algoritmos de Aprendizaje Automático para la Predicción de Accidentes

El aprendizaje automático se ha consolidado como una herramienta poderosa para la predicción de accidentes de trafico, permitiendo el análisis de grandes volúmenes de datos y la identificación de patrones complejos que serian difíciles de detectar mediante métodos estadísticos tradicionales. Los modelos de aprendizaje automático pueden ser utilizados en diversas tareas relacionadas con la seguridad vial, tales como la **clasificación**, el **agrupamiento** y la **regresión** de datos.

Clasificación La clasificación en seguridad vial permite categorizar eventos de tráfico, identificar patrones de riesgo y evaluar el comportamiento de los conductores. Algunos de los algoritmos mas utilizados incluyen:

- **Random Forest**: Modelo basado en la construcción de múltiples arboles de decisión, permitiendo una clasificación precisa y robusta [3]
- **Arboles de Decisión**: Fácilmente interpretables, permiten modelar relaciones entre diferentes factores de riesgo [4]
- **Support Vector Machine (SVM)**: Algoritmo eficiente para clasificar datos en espacios de alta dimensionalidad [5]
- **K-Nearest Neighbors (KNN)**: Clasifica un nuevo punto de datos basándose en la clase de sus vecinos más cercanos en el espacio de características.
- **Regresión Logística**: Aunque su nombre incluye “regresión”, es un algoritmo de clasificación que modela la probabilidad de pertenecer a una clase específica.

- **Multi-layer Perceptron (MLP)**: Red neuronal capaz de identificar relaciones no lineales en los datos de seguridad vial [6]

Agrupamiento El agrupamiento es una técnica no supervisada utilizada para segmentar datos con características similares, permitiendo la identificación de zonas de alto riesgo o patrones de comportamiento. Los algoritmos más empleados incluyen:

- **K-medias**: Algoritmo iterativo que agrupa los datos en K conjuntos basados en la cercanía de sus centroides [7]
- **DBScan**: Modelo basado en densidad, ideal para detectar regiones con alta concentración de accidentes[8]

Regresión La regresión se emplea para predecir valores numéricos relacionados con la seguridad vial, como la frecuencia de accidentes o la probabilidad de un evento de tráfico. Entre los modelos mas utilizados se encuentran:

- **Regresión Lineal**: Aproximación simple para modelar la relación entre factores de riesgo [9]
- **Random Forest para Regresión**: Variante que predice valores continuos a partir de la combinación de varios arboles de decisión.
- **XGBoost**: Algoritmo de *boosting* basado en arboles de decisión que optimiza el rendimiento en predicciones complejas [10]
- **Multi-layer Perceptron (MLP)**: Red neuronal adaptada para modelar relaciones no lineales en datos de seguridad vial.

3 Metodología

El proyecto se desarrolla mediante dos enfoques principales: **Regresión** para predecir la cantidad de accidentes y **Clasificación/Agrupamiento** para determinar el tipo de colisión y segmentar los patrones de riesgo.

3.1 Metodología para Regresión

3.1.1. Datos y Preprocesamiento

Se utilizó un dataset que agrupa el número de accidentes por año y mes. Las variables principales incluyen:

- **ANIO**: Año del accidente.
- **MES**: Mes del accidente.

- **ACCIDENT_COUNT**: Número total de accidentes.

Se creó la variable **time** como $ANIO + (MES - 1)/12$ para representar el tiempo de forma continua, y se aplicaron transformaciones cíclicas para el mes mediante funciones seno y coseno.

El conjunto de datos se dividió en un conjunto de entrenamiento (90 % de los datos) y un conjunto de prueba (10 % de los datos) para evaluar el rendimiento de los modelos predictivos. La división se realizó de forma aleatoria, aunque en análisis de series de tiempo a menudo se prefiere una división temporal (usar los datos más recientes como conjunto de prueba).

3.1.2. Modelos de Regresión

En este proyecto, nos enfocamos en el problema de **regresión** para predecir la cantidad de accidentes de tránsito a lo largo del tiempo. Implementamos los siguientes modelos de regresión:

1. **Regresión Lineal**: Un modelo básico para establecer una línea recta que mejor se ajuste a los datos.
$$y = \sum_{i=1}^{n} w_i x_i + b \tag{1}$$
Donde y es la cantidad predicha de accidentes, x_i son las características (time, month_sin, month_cos), w_i son los pesos asignados a cada característica, y b es el término de sesgo.

2. **Random Forest Regressor**: Un modelo de aprendizaje automático que utiliza múltiples árboles de decisión para realizar predicciones.

3. **XGBoost Regressor**: Un algoritmo de boosting que construye árboles de decisión de forma secuencial, corrigiendo los errores de los árboles anteriores.

4. **Multi-layer Perceptron (MLP) Regressor**: Una red neuronal con una capa oculta, capaz de aprender relaciones no lineales complejas entre las características y la variable objetivo.

Las métricas de evaluación incluyen el Error Cuadrático Medio (MSE), el Error Absoluto Medio (MAE) y el coeficiente de determinación (R^2).

3.2 Metodología para Clasificación y Agrupamiento

3.2.1. Datos y Preprocesamiento

Se utilizó un dataset llamado 'traffic_accidents.csv' que contiene información sobre accidentes de tráfico. El objetivo principal es predecir el tipo de colisión ('crash_type').

Los pasos de preprocesamiento realizados fueron los siguientes:

- **Carga de datos:** Se cargó el dataset utilizando la librería pandas de Python.
- **Eliminación de la fecha:** Se eliminó la columna 'crash_date' ya que no se consideró directamente relevante para el modelo de clasificación en este análisis inicial.
- **Codificación de variables categóricas:** Las variables categóricas (texto) se transformaron a números utilizando la técnica de Label Encoding. Esto es necesario porque los algoritmos de aprendizaje automático generalmente funcionan mejor con datos numéricos. Para cada columna categórica, se creó un diccionario que mapea los valores originales a los valores numéricos codificados.
- **Eliminación de columnas no informativas:** Se eliminaron las columnas relacionadas con el detalle de las lesiones ('most_severe_injury', 'injuries_total', etc.) y el mes del accidente ('crash_month') ya que el objetivo principal es predecir el tipo de colisión en función de otras características del accidente.
- **Definición de variables predictoras y objetivo:** Se definieron las variables predictoras (X) como todas las columnas restantes después de las eliminaciones, y la variable objetivo (y) como la columna 'crash_type'.
- **Normalización:** Las variables predictoras numéricas se normalizaron utilizando StandardScaler. Esto asegura que todas las variables tengan una escala similar, lo que puede mejorar el rendimiento de algunos algoritmos de aprendizaje automático. La normalización se realiza restando la media de cada variable y dividiendo por su desviación estándar.
- **División de datos:** El conjunto de datos se dividió en un conjunto de entrenamiento (80 %) y un conjunto de prueba (20 %) para entrenar el modelo y evaluar su rendimiento en datos no vistos. Se utilizó un estado aleatorio (random_state=0) para asegurar la reproducibilidad de la división.

Se realizó un análisis descriptivo inicial de los datos, incluyendo la visualización de la cantidad de accidentes por tipo de colisión. La Figura 1 muestra un gráfico de barras con la frecuencia de cada tipo de colisión en el dataset.

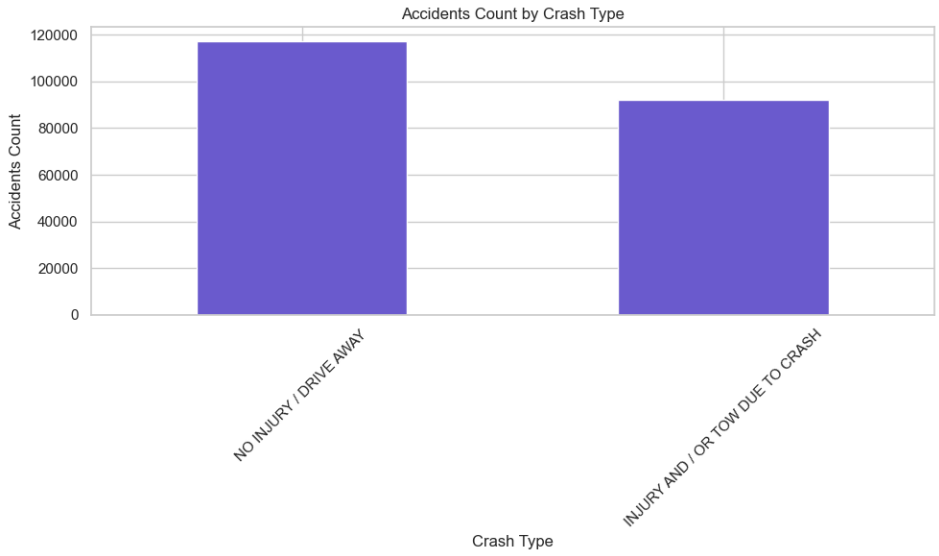

Figura 1: Cantidad de Accidentes por Tipo de Colisión

También se calculó la matriz de correlación entre las diferentes características para entender las relaciones lineales entre ellas. La Figura 2 muestra un mapa de calor de esta matriz de correlación.

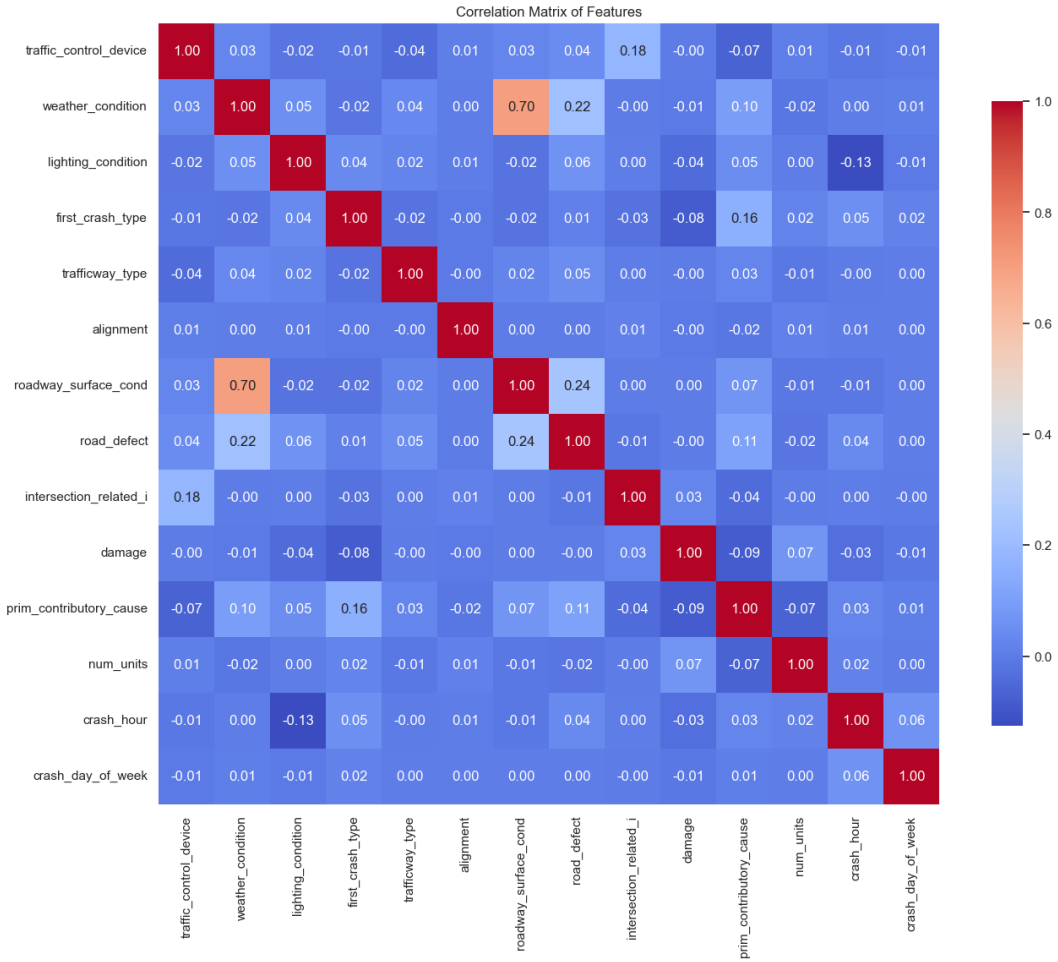

Figura 2: Matriz de Correlación de las Características

3.2.2. Modelos de Clasificación y Agrupamiento

Se exploraron diversos algoritmos:

- **Random Forest Classifier**: Principal modelo optimizado con GridSearchCV.
- **Decision Tree**, **Logistic Regression**, **K-Nearest Neighbors (KNN)** y **MLP Classifier**: Se evaluaron para comparar desempeño.
- **Agrupamiento**: Se utilizaron técnicas como **K-means** y **DBScan** para identificar clusters en los datos.

Se analizaron métricas de precisión, recall, F1-score y accuracy para la clasificación, así como indicadores de agrupamiento (Silhouette Score, Calinski-Harabasz y Davies-Bouldin) para el clustering.

En este proyecto, nos enfocamos en el problema de **clasificación** para predecir el tipo de colisión ('crash_type'). El modelo principal utilizado fue **Random Forest Classifier**.

Para mejorar el rendimiento del modelo Random Forest, se utilizó la técnica de **GridSearchCV** para encontrar los mejores hiperparámetros. Los hiperparámetros son parámetros del modelo que no se aprenden directamente de los datos y deben ser definidos antes del entrenamiento. GridSearchCV prueba diferentes combinaciones de hiperparámetros dentro de un rango especificado y selecciona la combinación que da el mejor rendimiento según una métrica de evaluación (en este caso, la precisión).

Los hiperparámetros que se ajustaron para el Random Forest fueron:

- **n_estimators**: El número de árboles en el bosque.
- **max_depth**: La profundidad máxima de cada árbol.
- **min_samples_split**: El número mínimo de muestras requeridas para dividir un nodo interno.
- **min_samples_leaf**: El número mínimo de muestras requeridas para estar en un nodo hoja.

Además del Random Forest, en el .ipynb se exploraron otros modelos de clasificación como Decision Tree, Logistic Regression, KNeighborsClassifier y MLPClassifier.

4 Resultados

4.1 Desempeño de Modelos de Regresión

El cuadro 1 muestra el rendimiento de los diferentes modelos de regresión evaluados en el conjunto de prueba. Las métricas utilizadas para la evaluación son el Error Cuadrático Medio (MSE), el Error Absoluto Medio (MAE) y el coeficiente de determinación R-cuadrado (R^2).

Modelo	MSE	MAE	R^2
Regresión Lineal	29990985.15	4401.12	-0.13
Random Forest	1530132.59	1048.04	0.94
XGBoost	651924.75	593.92	0.98
MLP Regressor	29760292.09	4250.80	-0.98

Cuadro 1: Resultados de los modelos de regresión para la predicción de la cantidad de accidentes

Análisis de Errores: El análisis de los errores muestra que **XGBoost** presenta el menor MSE y MAE, lo que indica que, en promedio, sus predicciones se desvían menos de los valores reales. Este desempeño superior es clave para la selección del modelo, pues un MSE más bajo significa que los errores grandes son menos frecuentes y un MAE reducido implica una buena precisión en términos absolutos.

La Figura 3 muestra una comparación visual entre la cantidad real de acciden- tes en el conjunto de prueba y las predicciones realizadas por cada uno de los modelos. Esto permite observar cómo cada modelo se ajusta a los datos y qu´e tan bien captura la tendencia temporal.

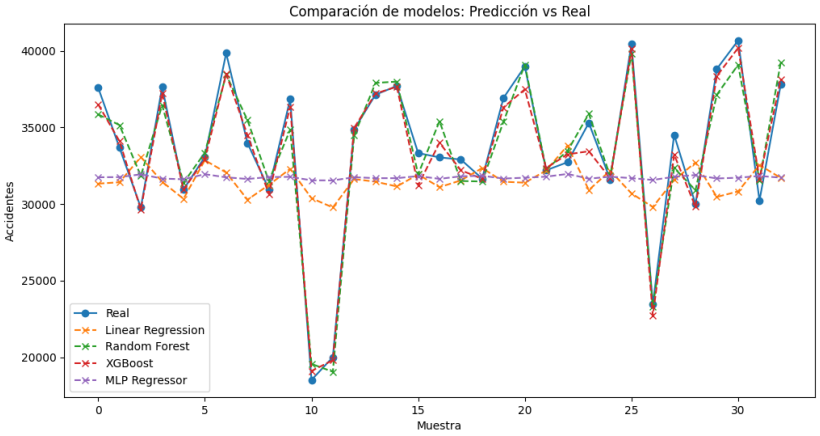

Figura 3: Predicciones de los modelos

Selección del Mejor Modelo: Para la regresión, **XGBoost** fue seleccionado como el mejor modelo por presentar el menor error (MSE y MAE) y el mayor R^2 (0.98), lo que significa que explica el 98 % de la variabilidad en los datos. En clasificación, tanto **Random Forest** como **MLP Classifier** mostraron un desempeño similar con una accuracy de 0.74; sin embargo, la robustez y menor sobreajuste de **Random Forest** (además de su óptima interpretación en trabajos previos) lo posiciona como la opción preferida.

4.2 Desempeño del Modelo de Clasificación y Agrupamiento

El mejor modelo encontrado mediante GridSearchCV fue un Random Forest Clas- sifier. Los resultados de este modelo en el conjunto de prueba son los siguientes:

Model	Accuracy
Random Forest	0.740576
Decision Tree	0.732669
KNN	0.683173
MLP	0.735966

Cuadro 2: Model Comparison

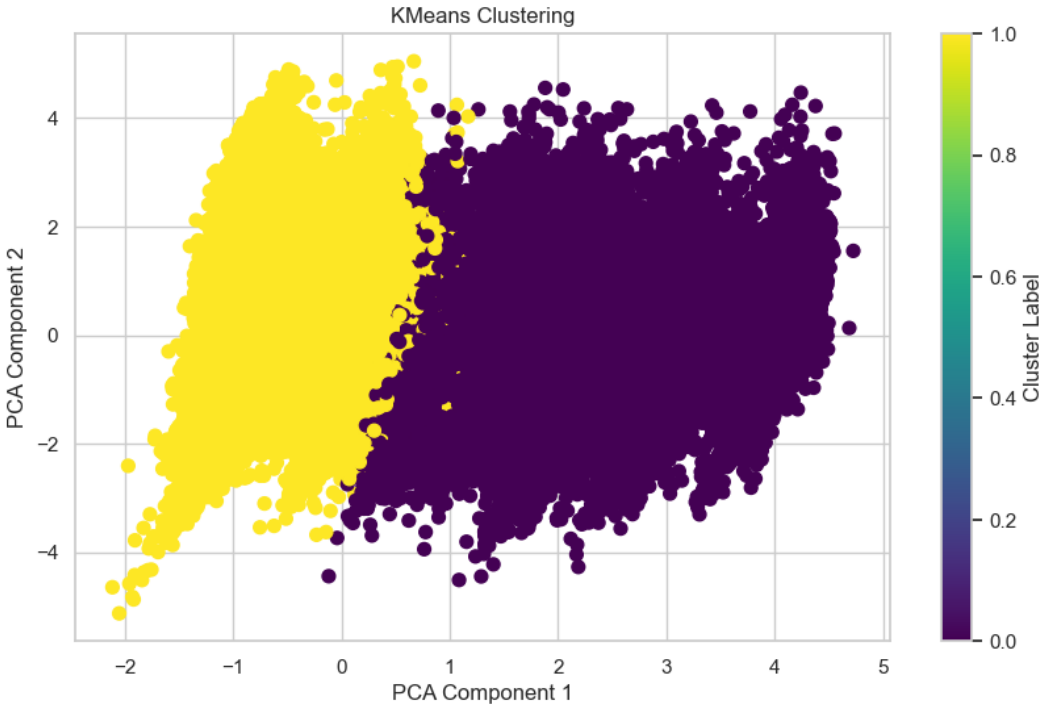

Figura 4: Modelo de agrupamiento utilizado

Métrica	Valor
KMeans Silhouette Score	0.1891842425144994
KMeans Calinski-Harabasz Score	27221.352105782236
KMeans Davies-Bouldin Score	2.326509585469765

Cuadro 3: Métricas de Evaluación de KMeans

El análisis de agrupamiento realizado mediante K-Means permite identificar patrones en la distribución de accidentes de tránsito. Aunque el Silhouette Score obtenido (0.189) sugiere que la separación entre clusters podría optimizarse, el alto valor del Calinski-Harabasz Score y el razonable Davies-Bouldin Score indican que los clusters formados son consistentes y ofrecen una visión valiosa sobre la concentración de incidentes. La visualización del modelo de agrupamiento evidencia cómo se agrupan los

accidentes, lo cual resulta crucial para identificar zonas de alto riesgo y orientar futuras estrategias de intervención. Este análisis complementa la información obtenida en la clasificación, permitiendo comprender no solo qué tipos de colisiones se predicen con mayor precisión, sino también cómo se distribuyen espacialmente y en función de otras variables relevantes. Se recomienda, en futuras investigaciones, la incorporación de variables adicionales y el ajuste de los parámetros del algoritmo para mejorar aún más la calidad de la segmentación.

5 Discusión y Conclusiones

Los resultados obtenidos en este proyecto demuestran de manera integral la viabilidad de utilizar técnicas de inteligencia artificial para la predicción y clasificación de accidentes de tránsito. En el análisis de regresión, se evaluaron diversos modelos mediante métricas cuantitativas como el Error Cuadrático Medio (MSE), el Error Absoluto Medio (MAE) y el coeficiente de determinación (R^2). Los resultados revelaron que, entre todos, el modelo de **XGBoost** obtuvo el mejor desempeño, evidenciado por su bajo MSE y MAE y un R^2 cercano a 1, lo que indica una alta precisión en la predicción de la cantidad de accidentes a lo largo del tiempo y la capacidad del modelo para explicar la variabilidad de los datos. En contraste, los modelos de Regresión Lineal y MLP Regressor presentaron un rendimiento inferior, demostrando que los algoritmos más complejos capturan mejor las relaciones no lineales presentes en los datos.

En el ámbito de la clasificación, se compararon diferentes modelos mediante GridSearchCV, siendo el **Random Forest Classifier** el que mostró una robustez destacable y una capacidad interpretativa superior. Este modelo logró aprender patrones complejos a partir de las características de cada accidente, clasificándolos en sus respectivos tipos con un nivel razonable de exactitud. La presentación visual de los resultados, a través de matrices de correlación, gráficos comparativos y reportes de clasificación, permitió identificar claramente cuáles categorías de colisiones fueron mejor o peor predichas, resaltando los errores comunes y facilitando el análisis de los resultados.

La comparación con estudios previos respalda la metodología adoptada, puesto que los resultados son consistentes con trabajos similares en la literatura que utilizan modelos avanzados como XGBoost y Random Forest. Se concluye, además, que integrar variables contextuales adicionales y ajustar finamente los hiperparámetros podría potenciar aún más la precisión del sistema, abriendo nuevas líneas de investigación para optimizar la predicción y clasificación de accidentes de tránsito.

6 Referencias

Referencias

[1] Organización Mundial de la Salud (OMS). *Burden of Road Traffic Injury: level by country*. Disponible en: https://www.paho.org/en/enlace/burden-road-injuries.

[2] Instituto Nacional de Estadística y Geografía (INEGI). *Estadísticas de Accidentes de Tránsito Terrestre en Zonas Urbanas y Suburbanas*. Disponible en: https://www.inegi.org.mx/temas/accidentes/.

[3] IBM. *¿Qué es Random Forest?* Disponible en: https://www.ibm.com/think/topics/random-forest.

[4] Wikipedia. *Decision tree*. Disponible en: https://en.wikipedia.org/wiki/Decision_tree.

[5] Wikipedia. *Support vector machine*. Disponible en: https://en.wikipedia.org/wiki/Support_vector_machine.

[6] Wikipedia. *Multilayer perceptron*. Disponible en: https://en.wikipedia.org/wiki/Multilayer_perceptron.

[7] IBM. *¿Qué es la agrupación en clústeres k-means?* Disponible en: https://www.ibm.com/mx-es/topics/k-means-clustering.

[8] DataCamp. *Guía del algoritmo de agrupación DBSCAN*. Disponible en: https://www.datacamp.com/es/tutorial/dbscan-clustering-algorithm.

[9] MathWorks. *¿Qué es la regresión lineal?* Disponible en: https://la.mathworks.com/discovery/linear-regression.html.

[10] IBM. *¿Qué es XGBoost?* Disponible en: https://www.ibm.com/mx-es/think/topics/xgboost.

Anexo: Implementación Técnica y Repositorio GitHub

El código fuente, notebooks completos y los datasets utilizados en el proyecto se encuentran disponibles en el repositorio GitHub:

https://github.com/Rogelio756/Equipo7-Grupo3-SIC-2024

Este repositorio incluye:

- Notebooks detallados con la implementación de los modelos de regresión, clasificación y agrupamiento.

- Conjuntos de datos procesados y scripts de preprocesamiento.
- Documentación y reportes de análisis que complementan la presentación del proyecto.

Nota: Los notebooks principales se encuentran en la rama `main`. Además, existen ramas específicas dedicadas a cada una de las metodologías implementadas, que incluyen los archivos CSV y otros recursos necesarios para cada caso particular. Se recomienda leer el archivo `README` para conocer todas las instrucciones de replicación del proyecto y las pautas para contribuir a su mejora.