Universidad Industrial de Santander

Observatorio de seguridad vial de Bucaramanga

Raysa Karina

Universidad Industrial de Santander

“Accidentes de tránsito, principal causa de muerte en niños de 5 a 14 años”

Federación de Aseguradores Colombianos fasecolda, 14 febrero 2019

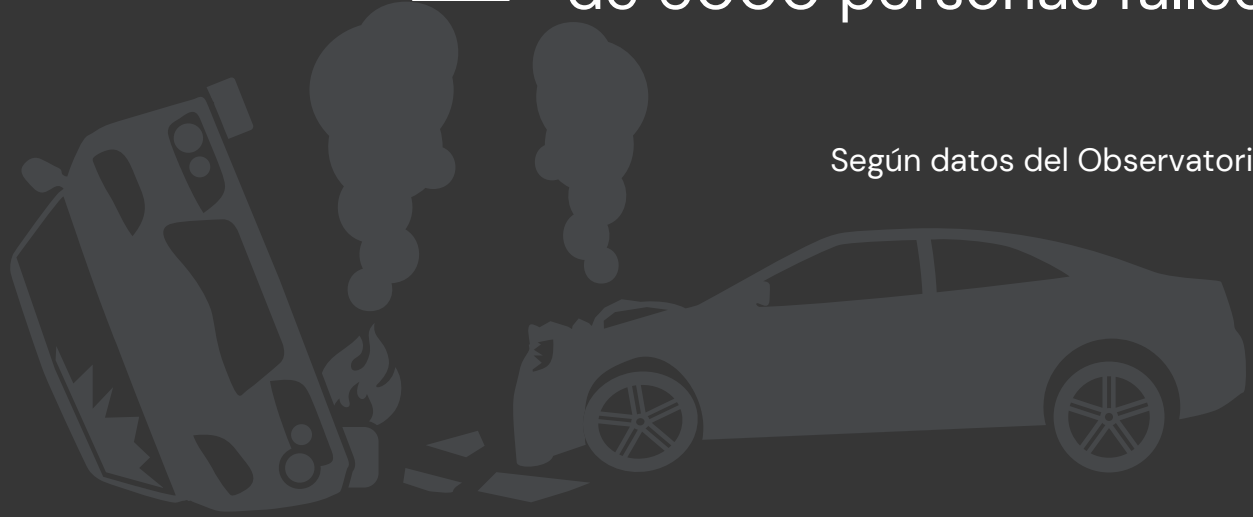
Introducción

1,35 millones de muertes en todo el mundo, según la Organización Mundial de la Salud.

Organización Mundial de la Salud

En Colombia el promedio anual gira alrededor de 6000 personas fallecidas en los últimos 10 años.

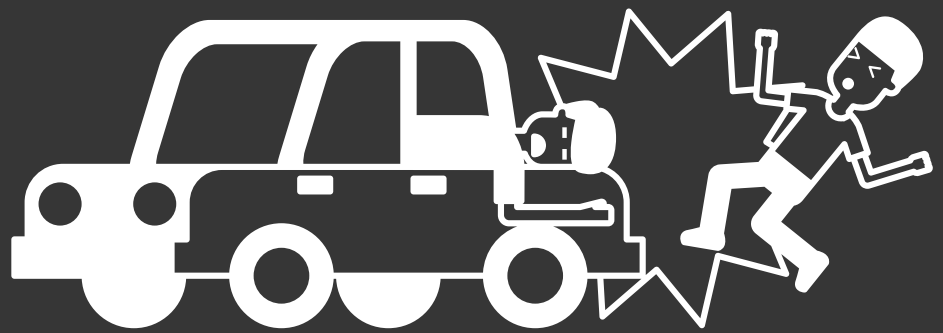
Según datos del Observatorio Nacional de Seguridad Vial en Colombia



El análisis e interpretación de datos de siniestralidad vial es:

Importante herramienta para intervenciones útiles en la mitigación de las condiciones inseguras, entender los comportamientos, reconocer los actores y sus vulnerabilidades y considerar la infraestructura,

Obtener esta información y apoyar el diseño, monitoreo, seguimiento y evaluación de las políticas públicas en Seguridad Vial.



Data Set

Datos libres

https://www.datos.gov.co/Transporte/03-ACCIDENTES-DE-TRANSITO-DESDE-ENERO-2012-A-AGOST/7cci-nqqb

Data Set

Universidad Industrial de Santander

Datos libres

03. ACCIDENTES DE TRANSITO DESDE ENERO 2012 A S...

Georreferenciación de los accidentes ocurridos en el municipio de Bucaramanga según

Más vistas | Filtrar | Visualizar

ORDEN	FECHA	AÑO	MES	DÍA	GRAVEDAD	PEAT...	AUTOMOVIL
1	01/01/2012	2012	01. Enero	07. Domingo	Con Heridos	0	1
2	01/01/2012	2012	01. Enero	07. Domingo	Solo Daños	0	1
3	01/01/2012	2012	01. Enero	07. Domingo	Solo Daños	0	0
4	01/01/2012	2012	01. Enero	07. Domingo	Solo Daños	0	1
5	01/01/2012	2012	01. Enero	07. Domingo	Con Heridos	1	0
6	01/01/2012	2012	01. Enero	07. Domingo	Solo Daños	0	1
7	02/01/2012	2012	01. Enero	01. Lunes	Con Heridos	1	0

1

Data Set

Universidad Industrial de Santander

Información dataset

Cantidad de registros:

```
listings = pd.read_csv('/content/gdrive/My Drive/Colab Notebooks/Proyecto IA 2023/03._ACCIDENTES_DE_TRANSITO_DESDE_ENERO_2012_A_25_MARZO_2023.csv', delimiter=',')
print(listings.shape)
#listings.head(3)
```

```
(38197, 26)
```

Visualización previa

```
pd.options.display.max_columns = 30
listings.head(100)
```

	ORDEN	FECHA	AÑO	MES	DÍA	GRAVEDAD	PEATON	AUTOMOVIL	CAMPAERO	CAMIONETA	MICRO	BUSETA	BUS	CAMION	VOLQUETA	MOTO	BICICLETA	OTRO	VIA 1
0	1	01/01/2012	2012	01. Enero	07. Domingo	Con Heridos	0	1	0	0	0	0	0	0	0	0	0	0	CALLE
1	2	01/01/2012	2012	01. Enero	07. Domingo	Solo Daños	0	1	0	1	0	0	0	0	0	0	0	0	VIA MATANZA
2	3	01/01/2012	2012	01. Enero	07. Domingo	Solo Daños	0	0	0	1	0	0	0	0	0	0	0	0	CARRERA
3	4	01/01/2012	2012	01. Enero	07. Domingo	Solo Daños	0	1	0	1	0	0	0	0	0	0	0	0	CARRERA

Menú presentación

Introducción al proyecto

Análisis de datos con IA

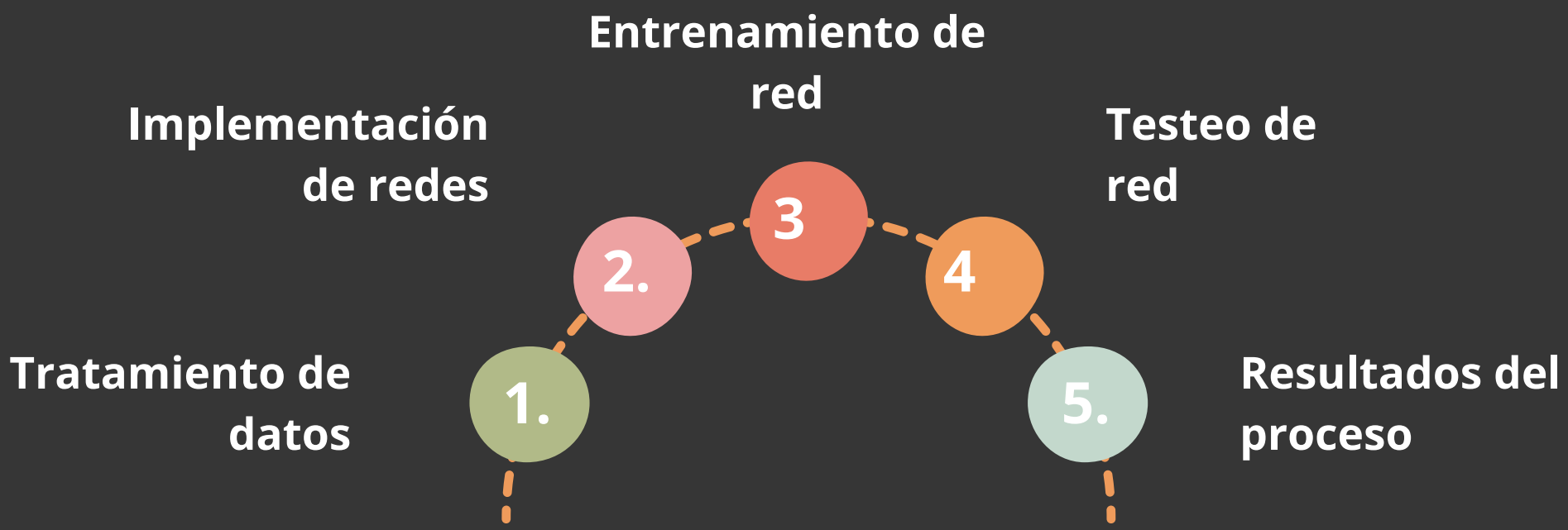
Reporte de resultados

Conclusiones

Introducción al proyecto

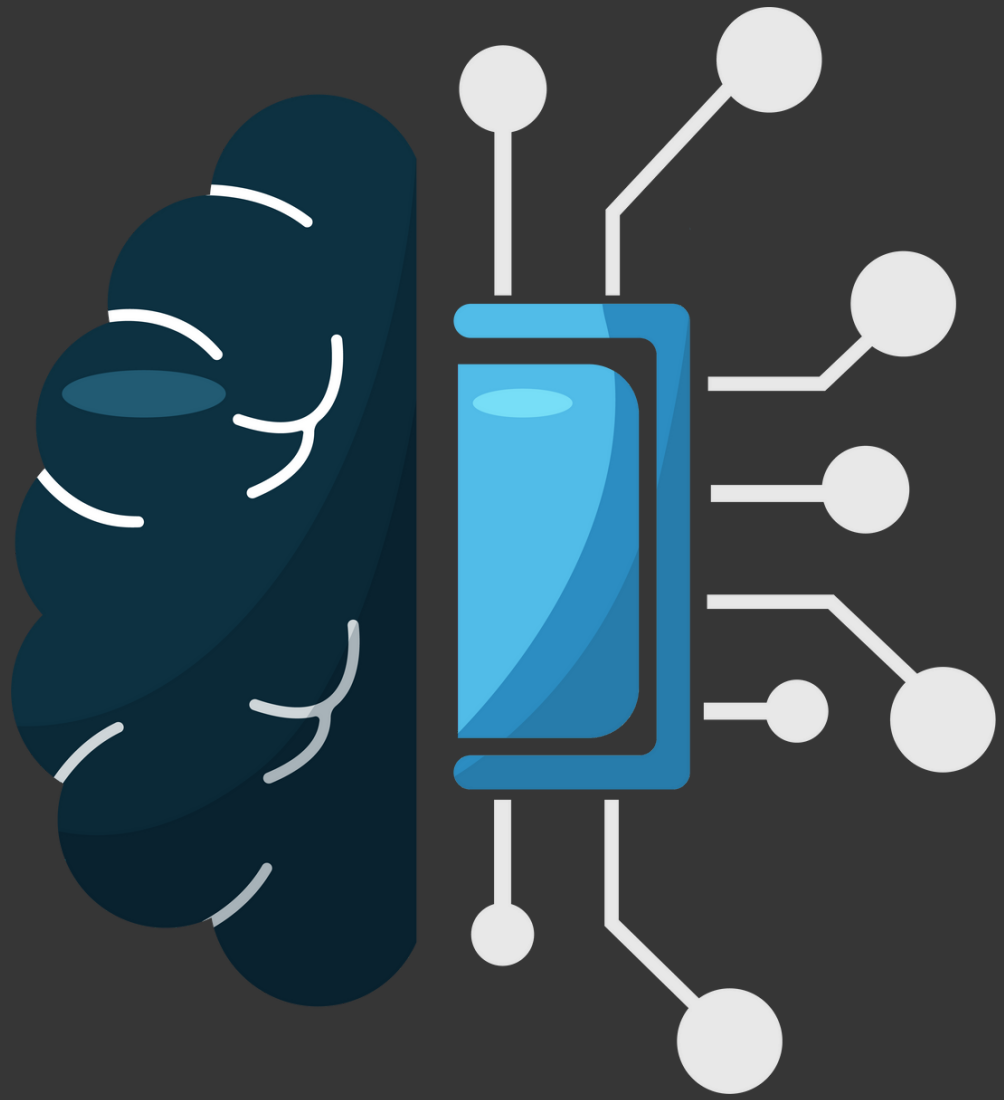
Se desea desarrollar y evaluar un modelo predictivo que de como resultado una estimación sobre el número de lesionados/heridos y fallecidos en siniestros viales a través del uso de algoritmos de inteligencia artificial.

Introducción al proyecto

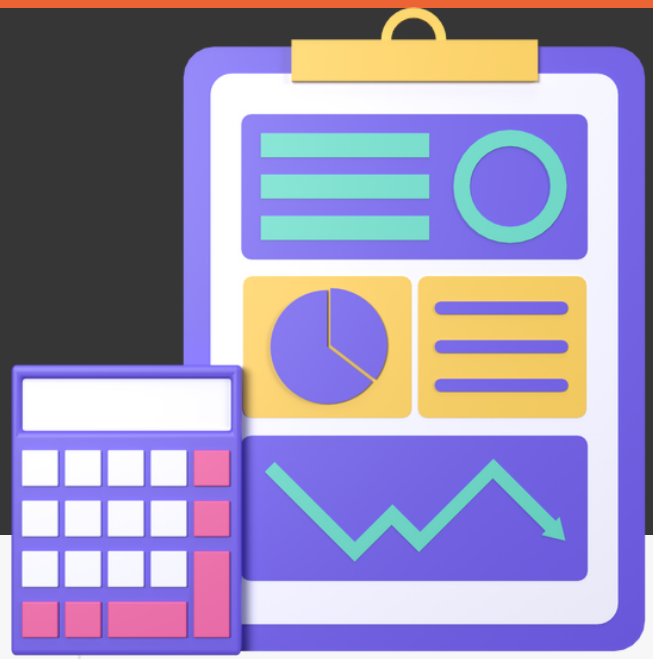


Universidad Industrial de Santander

Análisis de datos con IA



```
listings = listings.drop_duplicates()
print(listings.shape)
#no hay registro duplicados
listings.drop(['FECHA'], axis = 1)
```

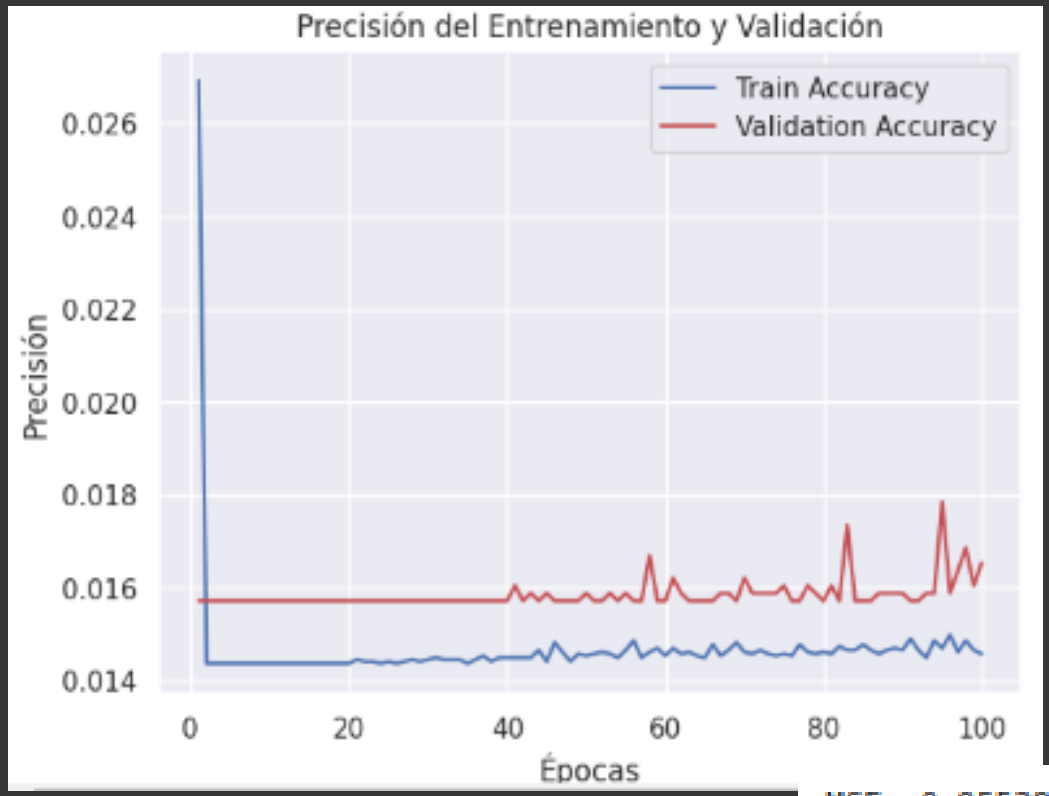


```
listings = listings.replace(['Heridos'], '2') #Heridos 2
listings = listings.replace(['Muertes'], '1') #Muertos 1
listings = listings.replace(['Daños'], '0') #Daños 0
listings = listings.replace(['Diurno'], '1') #dia
listings = listings.replace(['Nocturno'], '0') #
#Para este ejercicio no se usa otra heramienta, porque no fui capaz de hacerlas funcionar bien
#Use label_encoder = LabelEncoder() pero me daba errorValueError: Length of values (38197) does not match length of index (6)
#Cuando se usó el encoded_df = pd.get_dummies(df, columns=['feature1', 'feature2'])
#se crean demasiadas columnas... y hacer la codificacion de one-hot no aplica porque hay valores no vinarios dentro de las caracteristicas,
#Dos camionetas y una moto, o una cadena de vehiculos

#CAMBIO CATEGORIA
cat = listings[['nombrecomuna']]
ordinal_encoder = OrdinalEncoder(categories='auto')
encoded_data = ordinal_encoder.fit_transform(cat)
listings['comuna'] = encoded_data
listings.columns
```

Algoritmo 1

Algoritmo de optimización "adam" como el optimizador en la compilación del modelo (model.compile(loss='mean_squared_error', optimizer='adam'))

```
# Asignación de características
X = listings[['MES', 'DÍA', 'comuna', 'DIURNIO/NOCTURNO']].values
y = listings['GRAVEDAD'].values

# Conjunto de datos
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
y_train = y_train.astype(float)
y_test = y_test.astype(float)

# Normalización de datos
scaler = MinMaxScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

# Arquitectura con optimizador adam
model = Sequential()
model.add(Dense(64, activation='relu', input_shape=(4,)))
model.add(Dense(32, activation='relu'))
model.add(Dense(1, activation='linear'))

# Compilación y entrenamiento de la red neuronal
model.compile(loss='mean_squared_error', optimizer='adam', metrics=['accuracy'])
history = model.fit(X_train, y_train, epochs=100, batch_size=32, validation_split=0.2)
```

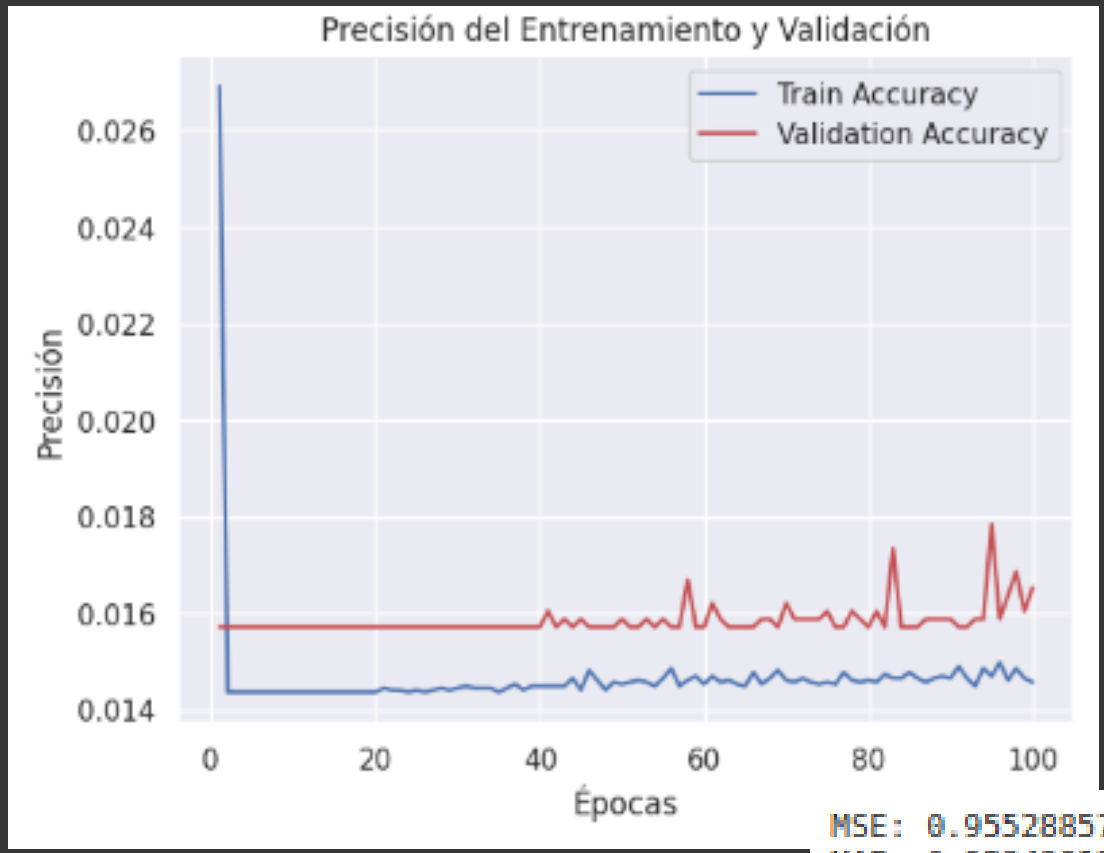


MSE: 0.9552885763681439
MAE: 0.9584991735781675

	precision	recall	f1-score	support
0.0	0.00	0.00	0.00	3842
1.0	0.02	1.00	0.03	121
2.0	0.00	0.00	0.00	3677
accuracy			0.02	7640
macro avg	0.01	0.33	0.01	7640
weighted avg	0.00	0.02	0.00	7640

Algoritmo 2

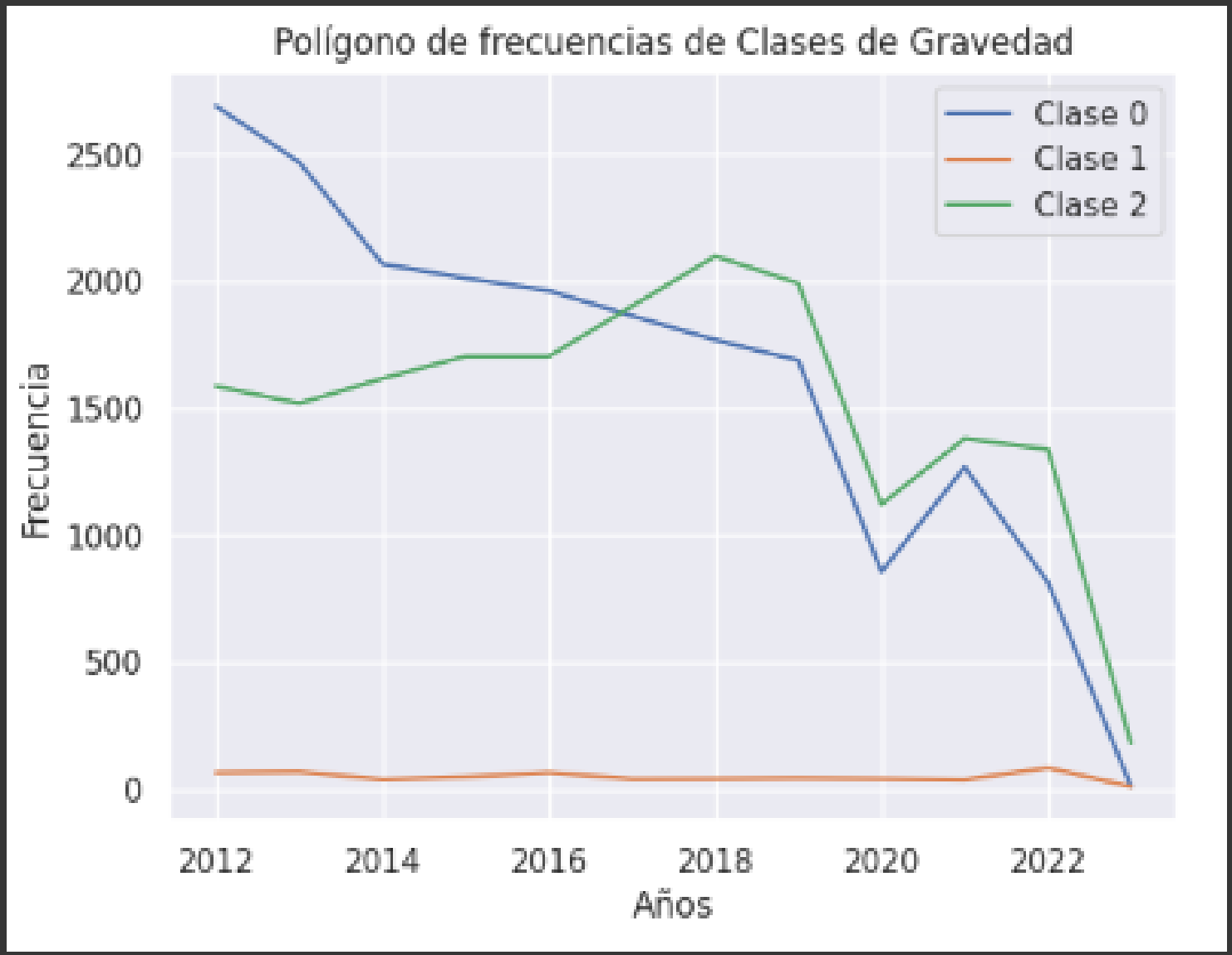
```
# Asignación de características
X = listings[['MES', 'DÍA', 'comuna', 'DIURNIO/NOCTURNO']].values
y = listings['GRAVEDAD'].values

# Conjunto de datos
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
y_train = y_train.astype(float)
y_test = y_test.astype(float)

# Normalización de datos
scaler = MinMaxScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

# Arquitectura con optimizador SGD
model = Sequential()
model.add(Dense(64, activation='relu', input_shape=(4,)))
model.add(Dense(32, activation='relu'))
model.add(Dense(1, activation='linear'))

# Compilación y entrenamiento de la red neuronal utilizando SGD
sgd = SGD(lr=0.01)
model.compile(loss='mean_squared_error', optimizer=sgd, metrics = ['accuracy'])
model.fit(X_train, y_train, epochs=100, batch_size=32, validation_split=0.2)

# Revisión del modelo
loss = model.evaluate(X_test, y_test)

# Predicciones
predictions = model.predict(X_test)
```



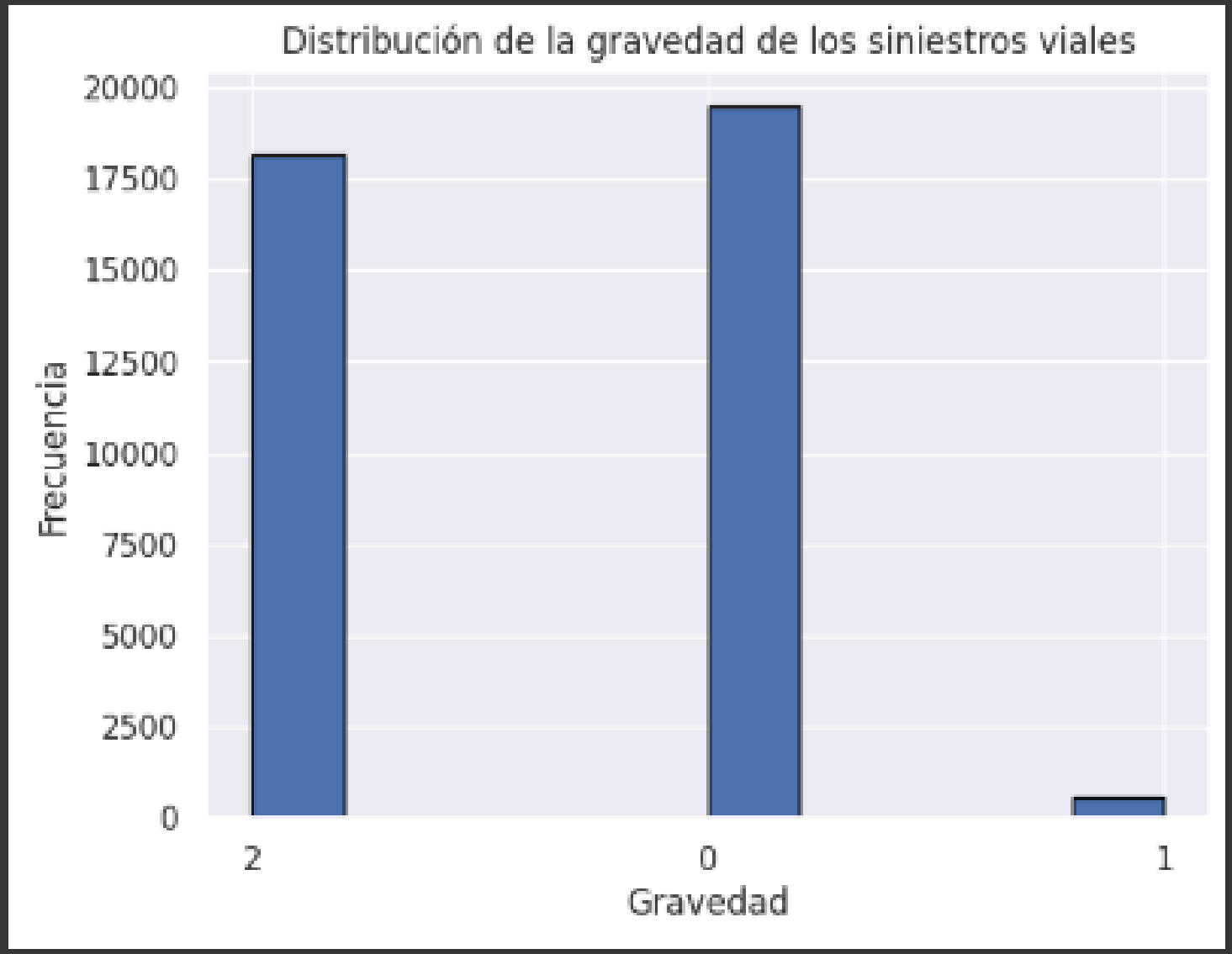
MSE: 0.9552885763681439
MAE: 0.9584991735781675

	precision	recall	f1-score	support
0.0	0.00	0.00	0.00	3842
1.0	0.02	1.00	0.03	121
2.0	0.00	0.00	0.00	3677
accuracy			0.02	7640
macro avg	0.01	0.33	0.01	7640
weighted avg	0.00	0.02	0.00	7640

Reportes


Polígono de frecuencias de Clases de Gravedad
Clase 0
Clase 1
Clase 2
Frecuencia
2500
2000
1500
1000
500
0
2012
2014
2016
2018
2020
2022
Años

Reportes


Distribución de la gravedad de los siniestros viales
20000
17500
15000
12500
10000
7500
5000
2500
0
Frecuencia
2
0
1
Gravedad

Conclusiones

El desbalance en el dataset, especialmente en la revisión y uso de las variables de salida, variables categoricas, afecta los resultados de los análisis. Esto hace que los métodos de Inteligencia Artificial, así como las gráficas y el preprocesamiento de datos, no muestren diferencias significativas en los resultados. Por lo tanto, no se logra una clasificación efectiva de los datos.

Conclusiones

Pese a que implementa una red neuronal de tres capas con dos diferentes algoritmos de optimización muestra resultados similares. Consecuencia de la falta de balance en los datos y a la dificultad del modelo para reconocer y clasificar las clases mayoritarias. Las métricas muestran un rendimiento deficiente, con altos niveles de error en ambos algoritmos, casi que mismos resultados.

Conclusiones

Se identifica la necesidad de replantear el modelo y considerar otros métodos de análisis debido a los resultados insatisfactorios. La falta de flexibilidad en el manejo de variables categóricas, como la imposibilidad de aplicar herramientas vistas en clase, codificación y la misma naturaleza de los datos, dificulta el análisis y la implementación de algoritmos.

Conclusiones

Se encuentra que se mencionaron herramientas como PSO (Optimización por Enjambre de Partículas) y ACO (Optimización por Colonias de Hormigas) como posibles opciones para el análisis, no fue posible implementarlos debido a la falta de disponibilidad de las librerías necesarias en el entorno utilizado. Se optó por utilizar algoritmos más sencillos y eficientes, adecuados para los datos proporcionados. PySwarms y Optunity no son compatibles de forma nativa en Google Colab.

Conclusiones

Es importante destacar que el dataset utilizado está incompleto en términos usarlo para hacer analisis de datos de siniestros viales. Ya que es necesario considerar los indicadores como: demográficos, panoramas de riesgos, datos de fallecidos, lesiones, entre otros, y es evidente la falta de desagregación de variables importantes, como grupo etario y género, limita la capacidad de análisis y la perspectiva de salud pública en relación a la seguridad vial.

Conclusiones

Estas conclusiones resaltan los desafíos y limitaciones encontrados y sugieren la necesidad de obtener un dataset más completo y adecuado para realizar análisis significativos y obtener conclusiones relevantes en el ámbito de la salud pública y la seguridad vial.

¡Gracias!