
EL OJO
DE BUCAR
Carlos Javier Aceros Blanco
Inteligencia artificial I - Proyecto final
Universidad Industrial de Santander

PROBLEMA A RESOLVER

- La inseguridad es un problema que en los últimos años ha afectado a Bucaramanga, la percepción de inseguridad llegó hasta el 79,2%, es decir, 8 de cada 10 ciudadanos se sienten inseguros en Bucaramanga y su área metropolitana, según el periódico Vanguardia

SOLUCIÓN PROPUESTA

- Implementar una herramienta web basada en inteligencia artificial que permita clasificar las zonas más peligrosas de la ciudad para poder implementar políticas públicas basadas en datos y lograr que la tranquilidad vuelva a la capital de Santander

METODOLOGÍA

- A través de una búsqueda en los datasets disponibles en Datos abiertos de Colombia(https://www.datos.gov.co/), se encontró un historial que consolida los delitos reportados desde Enero de 2010 hasta diciembre de 2021
- Se notó que los datos se actualizaron hasta el 10 octubre de 2023
- A través de técnicas aprendidas y guiadas por el docente Gustavo Garzón de análisis de datos se implementaron acciones de limpieza de datos, visualización y algoritmos de clasificación para lograr el objetivo planteado previamente

ALGUNOS ANÁLISIS

```
columns and shape Index(['ORDEN', 'ARMAS_MEDIOS', 'BARRIOS_HECHO', 'LATITUD', 'LONGITUD', 'ZONA',
       'NOM_COMUNA', 'ANO', 'MES', 'DIA', 'DIA_SEMANA', 'DESCRIPCION_CONDUCTA',
       'CONDUCTA', 'CLASIFICACIONES DELITO', 'EDAD', 'CURSO_DE_VIDA',
       'ESTADO_CIVIL_PERSONA', 'GENERO', 'MOVIL_AGRESOR', 'MOVIL_VICTIMA'],
      dtype='object') (135076, 20)
```

	ORDEN	ARMAS_MEDIOS	BARRIOS_HECHO	LATITUD	LONGITUD	ZONA	NOM_COMUNA	ANO	MES	DIA	DIA_SEMANA	DESCRIPCION_CONDUCTA	CONDUCTA	CLASIFICACIONES DELITO	EDAD	CURSO_DE_VIDA	ESTADO_CIVIL_PERSONA	GENERO	MOVIL_AGRESOR	MOVIL_VICTIMA
0	1	ARMA BLANCA / CORTOPUNZANTE	BUENOS AIRES	7.170557	-73.135108	URBANA	14. Morrorico	2010	01. Enero	1	05. Viernes	ARTÍCULO 111. LESIONES PERSONALES	LESIONES PERSONALES	Lesiones no fatales	30	05. Adultez	UNION LIBRE	MASCULINO	A PIE	A PIE
1	2	ARMA BLANCA / CORTOPUNZANTE	CAMPO HERMOSO	7.120645	-73.12605	URBANA	05. García Rovira	2010	01. Enero	1	05. Viernes	ARTÍCULO 111. LESIONES PERSONALES	LESIONES PERSONALES	Lesiones no fatales	21	04. Jovenes	SOLTERO	MASCULINO	A PIE	A PIE
2	3	ARMA BLANCA / CORTOPUNZANTE	CAMPO HERMOSO	7.120645	-73.12605	URBANA	05. García Rovira	2010	01. Enero	1	05. Viernes	ARTÍCULO 111. LESIONES PERSONALES	LESIONES PERSONALES	Lesiones no fatales	23	04. Jovenes	SOLTERO	MASCULINO	A PIE	A PIE
3	4	ARMA BLANCA / CORTOPUNZANTE	COMUNEROS	7.151359	-73.145705	URBANA	03. San Francisco	2010	01. Enero	1	05. Viernes	ARTÍCULO 111. LESIONES PERSONALES	LESIONES PERSONALES	Lesiones no fatales	36	05. Adultez	CASADO	MASCULINO	A PIE	A PIE
4	5	ARMA BLANCA / CORTOPUNZANTE	GIRARDOT	7.170557	-73.135108	URBANA	04. Occidental	2010	01. Enero	1	05. Viernes	ARTÍCULO 111. LESIONES PERSONALES	LESIONES PERSONALES	Lesiones no fatales	20	04. Jovenes	UNION LIBRE	MASCULINO	A PIE	A PIE
5	6	ARMA BLANCA / CORTOPUNZANTE	GIRARDOT	7.170557	-73.135108	URBANA	04. Occidental	2010	01. Enero	1	05. Viernes	ARTÍCULO 111. LESIONES PERSONALES	LESIONES PERSONALES	Lesiones no fatales	20	04. Jovenes	UNION LIBRE	MASCULINO	A PIE	A PIE
6	7	ARMA BLANCA /	LOS ANGELES	7.187455	-73.131727	URBANA	02. Nor	2010	01.	1	05. Viernes	ARTÍCULO 111. LESIONES	LESIONES	Lesiones no fatales	28	04. Jovenes	CASADO	MASCULINO	A PIE	A PIE

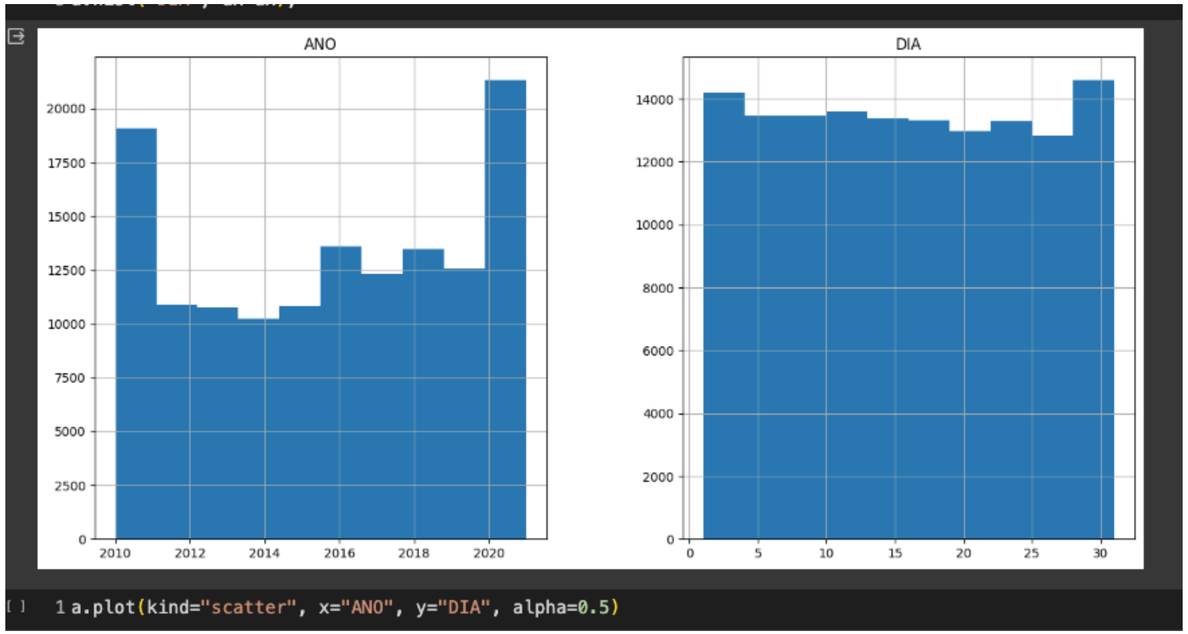
- Exploración del dataset: 20 columnas, 135.076 registros(delitos reportados)

ALGUNOS ANÁLISIS

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 135076 entries, 0 to 135075
Data columns (total 20 columns):
 #   Column                  Non-Null Count   Dtype 
---  ------                  --------------   ----- 
 0   ORDEN                   135076 non-null  int64 
 1   ARMAS_MEDIOS            135076 non-null  object
 2   BARRIOS_HECHO           135076 non-null  object
 3   LATITUD                 128713 non-null  object
 4   LONGITUD                128713 non-null  object
 5   ZONA                    135076 non-null  object
 6   NOM_COMUNA              135076 non-null  object
 7   ANO                     135076 non-null  int64 
 8   MES                     135076 non-null  object
 9   DIA                     135076 non-null  int64 
 10  DIA_SEMANA              135076 non-null  object
 11  DESCRIPCION_CONDUCTA    135076 non-null  object
 12  CONDUCTA                135076 non-null  object
 13  CLASIFICACIONES DELITO  135076 non-null  object
 14  EDAD                    135076 non-null  object
 15  CURSO_DE_VIDA           135076 non-null  object
 16  ESTADO_CIVIL_PERSONA    135076 non-null  object
 17  GENERO                  135076 non-null  object
 18  MOVIL_AGRESOR           135076 non-null  object
 19  MOVIL_VICTIMA           135076 non-null  object
dtypes: int64(3), object(17)
memory usage: 20.6+ MB
```

- La mayoría de tipos de datos categóricos, implicaba etiquetado para transformar a numéricos y poder realizar un análisis interesante

ALGUNOS ANÁLISIS

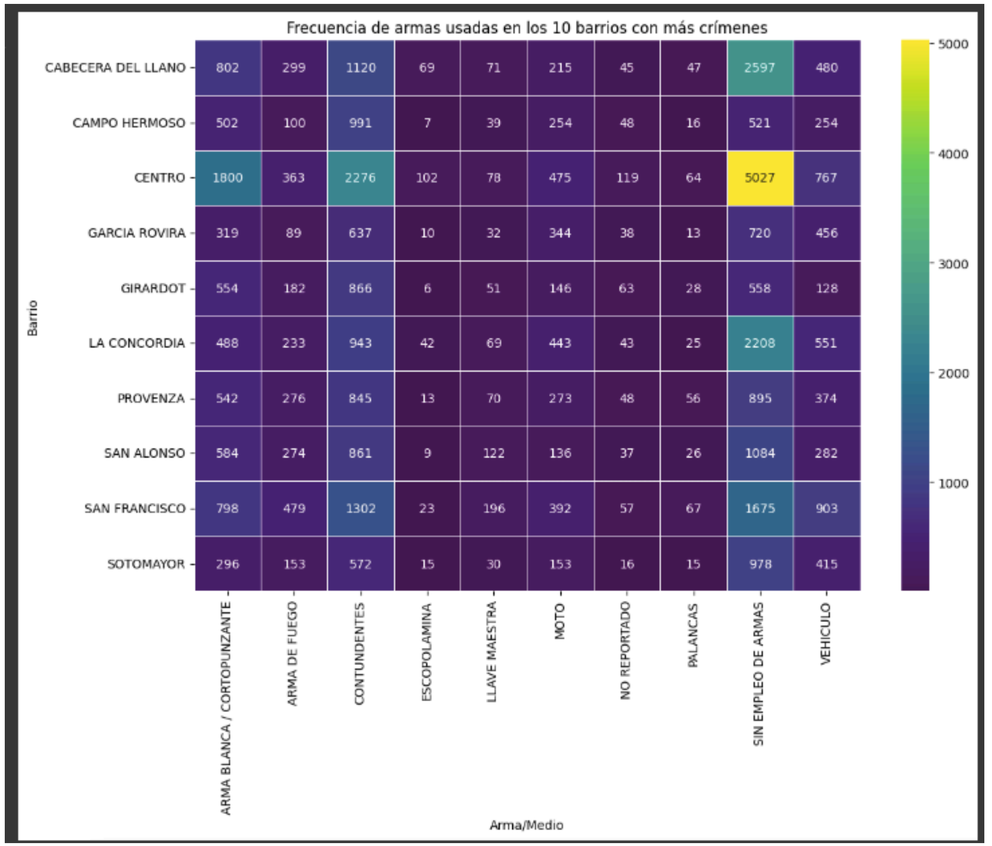


- Con los datos numéricos iniciales se lograron obtener algunas conclusiones interesantes: Los años en dónde más hubo delitos, los días del mes en donde más se cometen delitos(inicio y fin de mes) ¿Coincide con los días de pago?

ALGUNOS ANÁLISIS

```python
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Vamos a encontrar los 10 barrios más comunes y las 10 armas más comunes.
top_barrios = a['BARRIOS_HECHO'].value_counts().head(10).index
top_armas = a['ARMAS_MEDIOS'].value_counts().head(10).index

# Filtramos el DataFrame para incluir solo los datos más comunes.
filtered_data = a[a['BARRIOS_HECHO'].isin(top_barrios) & a['ARMAS_MEDIOS'].isin(top_armas)]

# Calcular la frecuencia de cada combinación de arma y barrio.
conteo_armas_barrios = filtered_data.groupby(['BARRIOS_HECHO', 'ARMAS_MEDIOS']).size().reset_index(name='conteo')

# Pivotear el DataFrame para preparar los datos para el heatmap.
pivot_table = conteo_armas_barrios.pivot(index='BARRIOS_HECHO', columns='ARMAS_MEDIOS', values='conteo').fillna(0).astype(int)

# Crear un heatmap para visualizar la frecuencia de las armas usadas en cada barrio.
plt.figure(figsize=(12, 8))  # Ajusta el tamaño si es necesario
sns.heatmap(pivot_table, annot=True, fmt="d", linewidths=.5, cmap='viridis')
plt.title('Frecuencia de armas usadas en los 10 barrios con más crímenes')
plt.ylabel('Barrio')
plt.xlabel('Arma/Medio')
plt.show()

```

Frecuencia de armas usadas en los 10 barrios con más crímenes



- Se determina con precisión el tipo de arma que más se usa en los delitos cometidos por zona

ALGUNOS ANÁLISIS

DELITOS MAS COMUNES POR BARRIO

```python
#@title DELITOS MAS COMUNES POR BARRIO

# Agrupamos por barrio y por conducta del delito y contamos las ocurrencias.
delitos_por_barrio = a.groupby(['BARRIOS_HECHO', 'CONDUCTA']).size().reset_index(name='conteo')

# Ordenamos los resultados para obtener los delitos más comunes en cada barrio.
delitos_mas_comunes_por_barrio = delitos_por_barrio.sort_values(by=['BARRIOS_HECHO', 'conteo'], ascending=[True, False])

# Ahora queremos tener solo el delito más común por barrio, por lo que agrupamos y tomamos el primero.
delito_mas_comun_por_barrio = delitos_mas_comunes_por_barrio.groupby('BARRIOS_HECHO').head(1)

# Finalmente, mostramos los resultados.
print(delito_mas_comun_por_barrio)
```

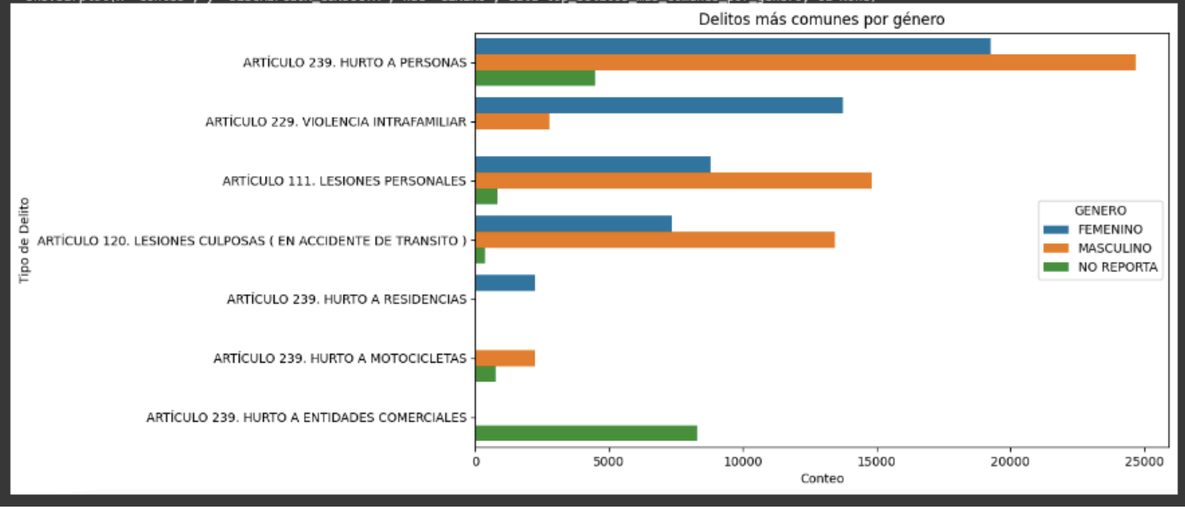
```
          BARRIOS_HECHO                 CONDUCTA  conteo
13        12 DE OCTUBRE  VIOLENCIA INTRAFAMILIAR      85
18          13 DE JUNIO          HURTO A PERSONAS      13
28          20 DE JULIO       LESIONES PERSONALES      13
41          23 DE JUNIO  VIOLENCIA INTRAFAMILIAR      30
46           5 DE ENERO       LESIONES PERSONALES       7
...                 ...                       ...     ...
4178  nuevo sotomayor          HURTO A PERSONAS       1
4179          pablo VI  VIOLENCIA INTRAFAMILIAR       2
4183          provenza       LESIONES PERSONALES      13
4185        san rafael  VIOLENCIA INTRAFAMILIAR       1
4187  sin informacion                 EXTORSIÓN      22

[494 rows x 3 columns]
```

- Se determinan los delitos más comunes por barrio

ALGUNOS ANÁLISIS

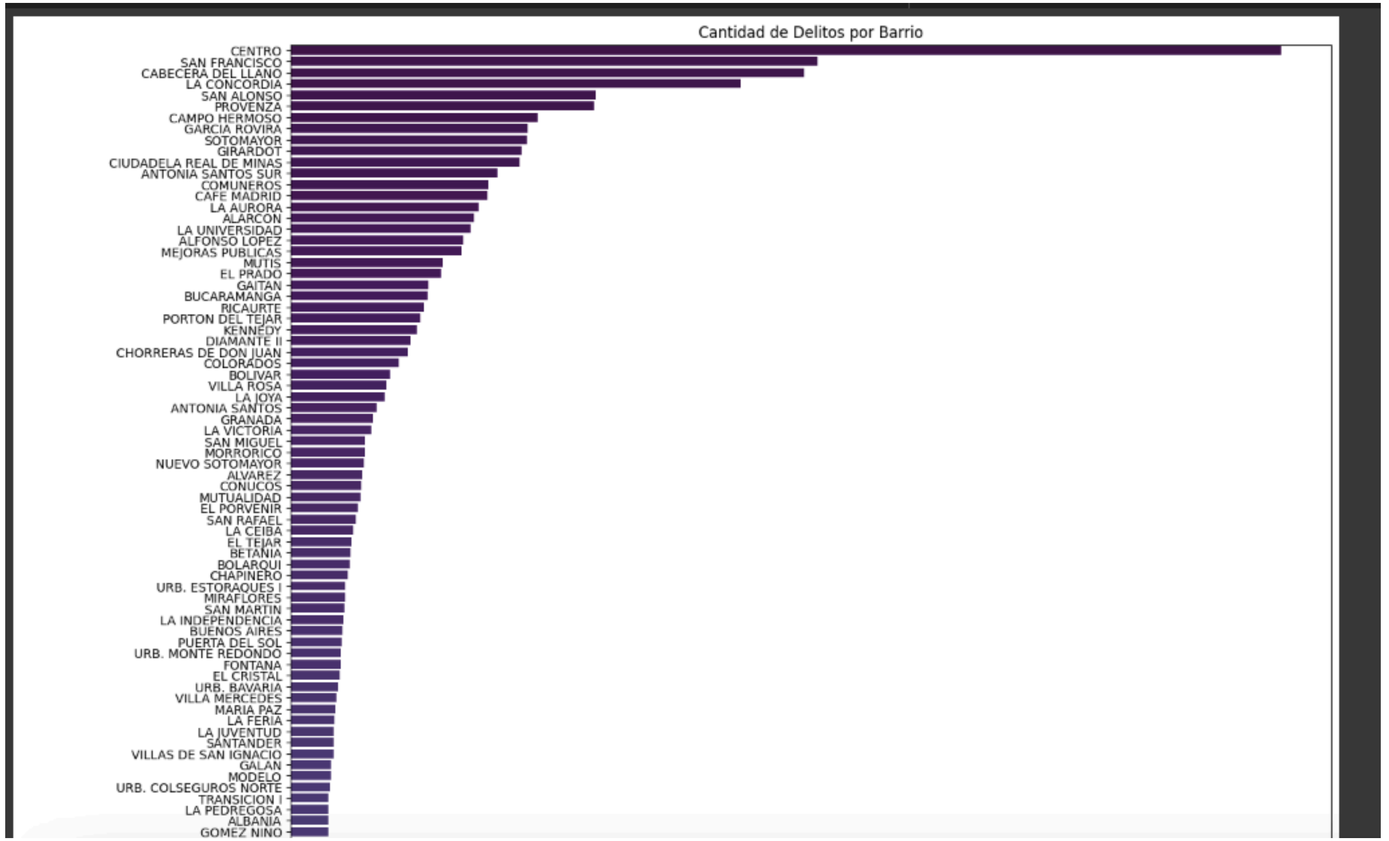
```python
#@title GRAFICA DE DELITOS MAS COMUNES POR GÉNERO


plt.figure(figsize=(10, 6))
sns.barplot(x='conteo', y='DESCRIPCION_CONDUCTA', hue='GENERO', data=top_delitos_mas_comunes_por_genero, ci=None)
plt.title('Delitos más comunes por género')
plt.xlabel('Conteo')
plt.ylabel('Tipo de Delito')
plt.show()

```

```
<ipython-input-37-23698ca5fd9b>:5: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.barplot(x='conteo', y='DESCRIPCION_CONDUCTA', hue='GENERO', data=top_delitos_mas_comunes_por_genero, ci=None)
```



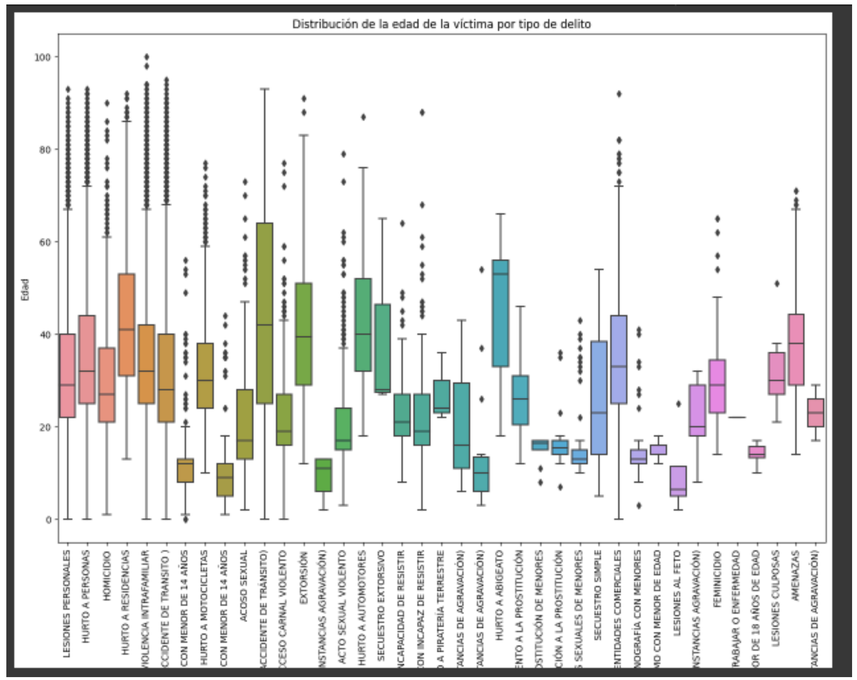
- Los delitos más comunes por género

ALGUNOS ANÁLISIS



- Se obtiene un primer acercamiento al objetivo deseado, determinar las zonas más peligrosas en Bucaramanga

ALGUNOS ANÁLISIS



- Se discriminan los delitos por edades de víctima usando boxplot

SE APLICA UN PRIMER MODELO DE CLASIFICACIÓN RANDOM FOREST

APLICANDO MODELOS DE CLASIFICACIÓN

```
#librerias
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report
from sklearn.model_selection import cross_val_score
from sklearn.preprocessing import LabelEncoder
```

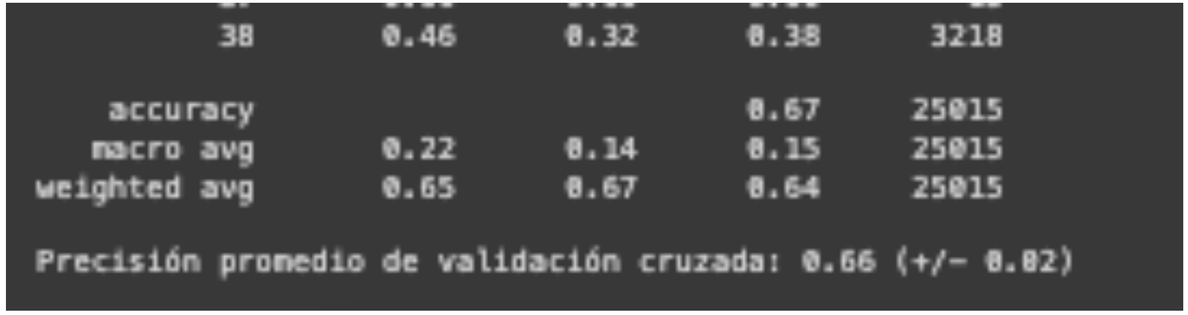
```
# Preprocesamiento: Convertir categorías a números
le_barrios = LabelEncoder()
a['BARRIOS_HECHO'] = le_barrios.fit_transform(a['BARRIOS_HECHO'])

le_armas = LabelEncoder()
a['ARMAS_MEDIOS'] = le_armas.fit_transform(a['ARMAS_MEDIOS'])

le_conducta = LabelEncoder()
a['CONDUCTA'] = le_conducta.fit_transform(a['CONDUCTA'])

# División del conjunto de datos
X = a[['BARRIOS_HECHO', 'ARMAS_MEDIOS']]  # Características
y = a['CONDUCTA']  # Variable objetivo

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=21)

# Construcción del modelo de clasificación
clf = RandomForestClassifier(n_estimators=100, random_state=21)
clf.fit(X_train, y_train)

# Predicciones
y_pred = clf.predict(X_test)

# Evaluación del modelo
print(classification_report(y_test, y_pred))

# Validación cruzada
scores = cross_val_score(clf, X, y, cv=5)
print("Precisión promedio de validación cruzada: %0.2f (+/- %0.2f)" % (scores.mean(), scores.std() * 2))
```

- Implementación de RF

warnings.warn(

	precision	recall	f1-score	support
0	0.15	0.02	0.04	189
1	0.00	0.00	0.00	3
2	0.07	0.02	0.03	45
3	0.00	0.00	0.00	24
4	0.11	0.01	0.02	87
5	0.00	0.00	0.00	1
6	0.04	0.02	0.03	43
7	0.00	0.00	0.00	80
8	0.00	0.00	0.00	1
9	0.29	0.17	0.21	312
10	0.00	0.00	0.00	2
11	0.25	0.02	0.03	55
12	0.00	0.00	0.00	2
13	0.00	0.00	0.00	5
14	0.00	0.00	0.00	1
15	0.91	0.83	0.87	93
16	0.00	0.00	0.00	7
17	0.35	0.02	0.05	241
18	0.00	0.00	0.00	128
19	0.00	0.00	0.00	3
20	0.00	0.00	0.00	33
21	0.50	0.02	0.04	215
22	0.86	0.61	0.71	641
23	0.71	0.84	0.77	9559
24	0.00	0.00	0.00	2
25	0.69	0.12	0.21	929
27	0.25	0.11	0.15	9
28	0.00	0.00	0.00	1
29	0.00	0.00	0.00	1
30	0.97	0.98	0.97	4198
31	0.48	0.61	0.54	4849
32	0.00	0.00	0.00	1
33	0.50	0.05	0.08	22
37	0.00	0.00	0.00	15
38	0.46	0.32	0.38	3218
accuracy			0.67	25015
macro avg	0.22	0.14	0.15	25015
weighted avg	0.65	0.67	0.64	25015

Precisión promedio de validación cruzada: 0.66 (+/- 0.02)



SE APLICA UN SEGUNDO MODELO DE CLASIFICACIÓN SUPPORT VECTOR MACHINE(SVM)

```
ggg                                                    # División del conjunto de datos
from sklearn.svm import SVC
from sklearn.preprocessing import StandardScaler
from sklearn.pipeline import make_pipeline
X = a[['BARRIOS_HECHO', 'ARMAS_MEDIOS']]  # Características
y = a['CONDUCTA']  # Variable objetivo

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Para cada kernel, crea un pipeline con un StandardScaler y un SVC con el kernel correspondiente.
# Entrena y evalúa el modelo.
kernels = ['linear', 'poly', 'rbf']
for kernel in kernels:
    clf = make_pipeline(StandardScaler(), SVC(kernel=kernel))
    clf.fit(X_train, y_train)
    y_pred = clf.predict(X_test)
    print(f"Resultados para kernel {kernel}:")
    print(classification_report(y_test, y_pred))

```

LINEAR

Resultados para kernel linear:

	precision	recall	f1-score	support
0	0.00	0.00	0.00	174
1	0.00	0.00	0.00	1
2	0.00	0.00	0.00	43
3	0.00	0.00	0.00	15
4	0.00	0.00	0.00	92
6	0.00	0.00	0.00	53
7	0.00	0.00	0.00	82
8	0.00	0.00	0.00	2
9	0.00	0.00	0.00	278
10	0.00	0.00	0.00	3
11	0.00	0.00	0.00	48
12	0.00	0.00	0.00	2
13	0.00	0.00	0.00	1
14	0.00	0.00	0.00	2
15	0.00	0.00	0.00	79
16	0.00	0.00	0.00	8
17	0.00	0.00	0.00	257
18	0.00	0.00	0.00	127
19	0.00	0.00	0.00	3
20	0.00	0.00	0.00	32
21	0.00	0.00	0.00	226
22	0.00	0.00	0.00	628
23	0.39	1.00	0.56	9701
25	0.00	0.00	0.00	926
27	0.00	0.00	0.00	5
28	0.00	0.00	0.00	2
29	0.00	0.00	0.00	1
30	0.00	0.00	0.00	4205
31	0.00	0.00	0.00	4736
32	0.00	0.00	0.00	1
33	0.00	0.00	0.00	12
34	0.00	0.00	0.00	1
35	0.00	0.00	0.00	1
37	0.00	0.00	0.00	7
38	0.00	0.00	0.00	3261
accuracy			0.39	25015
macro avg	0.01	0.03	0.02	25015
weighted avg	0.15	0.39	0.22	25015

POLY

Resultados para kernel poly:

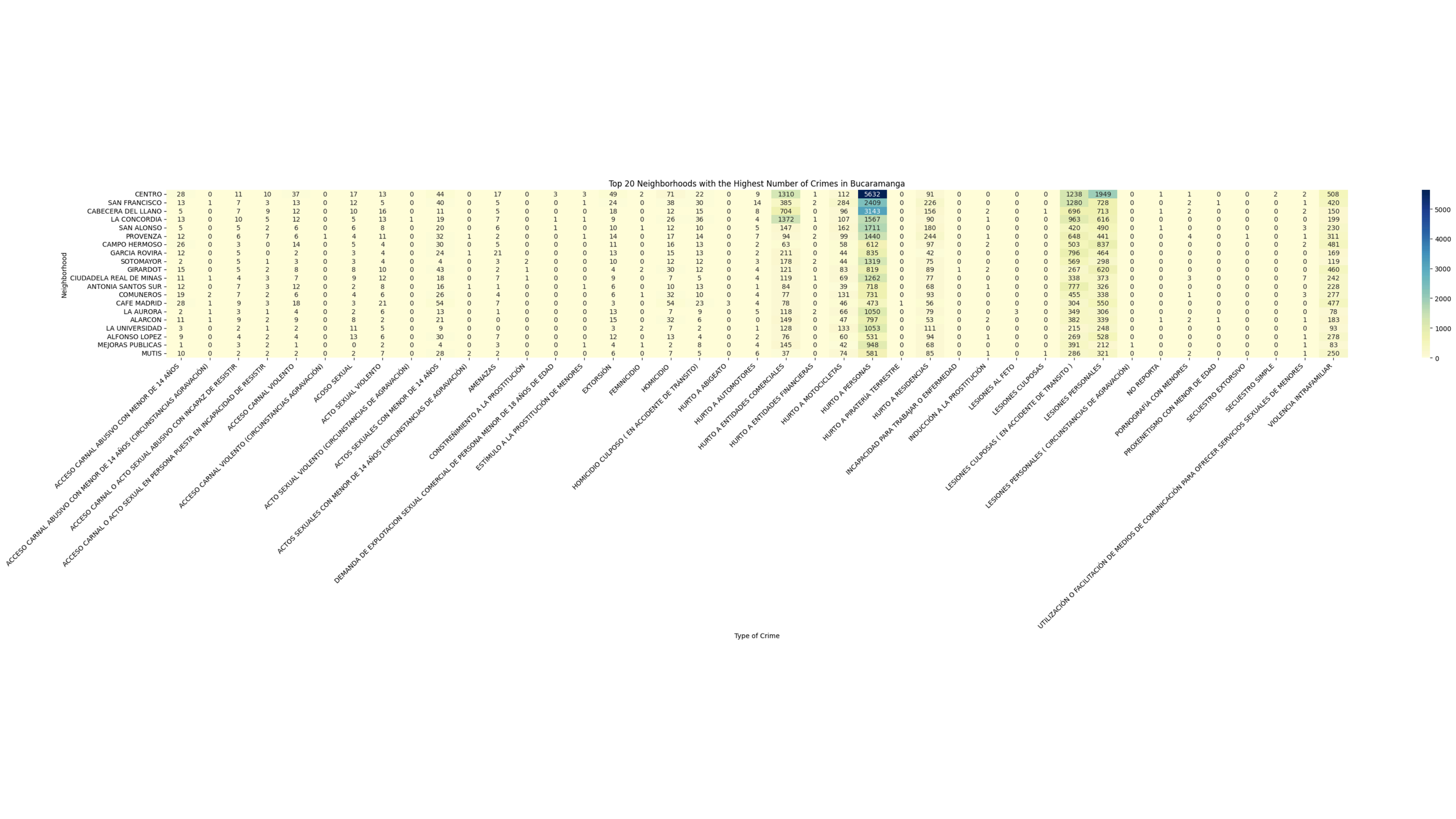
	precision	recall	f1-score	support
0	0.00	0.00	0.00	174
1	0.00	0.00	0.00	1
2	0.00	0.00	0.00	43
3	0.00	0.00	0.00	15
4	0.00	0.00	0.00	92
6	0.00	0.00	0.00	53
7	0.00	0.00	0.00	82
8	0.00	0.00	0.00	2
9	0.00	0.00	0.00	278
10	0.00	0.00	0.00	3
11	0.00	0.00	0.00	48
12	0.00	0.00	0.00	2
13	0.00	0.00	0.00	1
14	0.00	0.00	0.00	2
15	0.00	0.00	0.00	79
16	0.00	0.00	0.00	8
17	0.00	0.00	0.00	257
18	0.00	0.00	0.00	127
19	0.00	0.00	0.00	3
20	0.00	0.00	0.00	32
21	0.00	0.00	0.00	226
22	0.00	0.00	0.00	628
23	0.39	1.00	0.56	9701
25	0.00	0.00	0.00	926
27	0.00	0.00	0.00	5
28	0.00	0.00	0.00	2
29	0.00	0.00	0.00	1
30	0.00	0.00	0.00	4205
31	0.00	0.00	0.00	4736
32	0.00	0.00	0.00	1
33	0.00	0.00	0.00	12
34	0.00	0.00	0.00	1
35	0.00	0.00	0.00	1
37	0.00	0.00	0.00	7
38	0.00	0.00	0.00	3261
accuracy			0.39	25015
macro avg	0.01	0.03	0.02	25015
weighted avg	0.15	0.39	0.22	25015

RBF

Resultados para kernel rbf:

	precision	recall	f1-score	support
0	0.00	0.00	0.00	174
1	0.00	0.00	0.00	1
2	0.00	0.00	0.00	43
3	0.00	0.00	0.00	15
4	0.00	0.00	0.00	92
6	0.00	0.00	0.00	53
7	0.00	0.00	0.00	82
8	0.00	0.00	0.00	2
9	0.00	0.00	0.00	278
10	0.00	0.00	0.00	3
11	0.00	0.00	0.00	48
12	0.00	0.00	0.00	2
13	0.00	0.00	0.00	1
14	0.00	0.00	0.00	2
15	0.05	0.08	0.06	79
16	0.00	0.00	0.00	8
17	0.00	0.00	0.00	257
18	0.00	0.00	0.00	127
19	0.00	0.00	0.00	3
20	0.00	0.00	0.00	32
21	0.00	0.00	0.00	226
22	0.79	0.64	0.71	628
23	0.69	0.88	0.77	9701
25	0.58	0.13	0.21	926
27	0.00	0.00	0.00	5
28	0.00	0.00	0.00	2
29	0.00	0.00	0.00	1
30	0.89	0.98	0.93	4205
31	0.44	0.66	0.53	4736
32	0.00	0.00	0.00	1
33	0.00	0.00	0.00	12
34	0.00	0.00	0.00	1
35	0.00	0.00	0.00	1
37	0.00	0.00	0.00	7
38	0.00	0.00	0.00	3261
accuracy			0.65	25015
macro avg	0.10	0.10	0.09	25015
weighted avg	0.54	0.65	0.58	25015

VISUALIZANDO LOS 20 BARRIOS MÁS PELIGROSOS Y LOS DELITOS QUE MÁS SE COMETEN


Top 20 Neighborhoods with the Highest Number of Crimes in Bucaramanga
Neighborhood
Type of Crime
CENTRO 28 0 11 10 37 0 17 13 0 44 0 17 0 3 3 49 2 71 22 0 9 1310 1 112 5632 0 91 0 0 0 0 1238 1949 0 1 1 0 0 2 2 508
SAN FRANCISCO 13 1 7 3 13 0 12 5 0 40 0 5 0 0 1 24 0 38 30 0 14 385 2 284 2409 0 226 0 0 0 0 1280 728 0 0 2 1 0 0 1 420
CABECERA DEL LLANO 5 0 7 9 12 0 10 16 0 11 0 5 0 0 0 18 0 12 15 0 8 704 0 96 3143 0 156 0 2 0 1 696 713 0 1 2 0 0 0 2 150
LA CONCORDIA 13 0 10 5 12 0 5 13 1 19 0 7 0 1 1 9 0 26 36 0 4 1372 1 107 1567 0 90 0 1 0 0 963 616 0 0 0 0 0 0 0 199
SAN ALONSO 5 0 5 2 6 0 6 8 0 20 0 6 0 1 0 10 1 12 10 0 5 147 0 162 1711 0 180 0 0 0 0 420 490 0 1 0 0 0 0 3 230
PROVENZA 12 0 6 7 6 1 4 11 0 32 1 2 0 0 1 14 0 17 14 0 7 94 2 99 1440 0 244 0 1 0 0 648 441 0 0 4 0 1 0 1 311
CAMPO HERMOSO 26 0 3 0 14 0 5 4 0 30 0 5 0 0 0 11 0 16 13 0 2 63 0 58 612 0 97 0 2 0 0 503 837 0 0 0 0 0 0 2 481
GARCIA ROVIRA 12 0 5 0 2 0 3 4 0 24 1 21 0 0 0 13 0 15 13 0 2 211 0 44 835 0 42 0 0 0 0 796 464 0 0 0 0 0 0 0 169
SOTOMAYOR 2 0 5 1 3 0 3 4 0 4 0 3 2 0 0 10 0 12 12 0 3 178 2 44 1319 0 75 0 0 0 0 569 298 0 0 0 0 0 0 0 119
GIRARDOT 15 0 5 2 8 0 8 10 0 43 0 2 1 0 0 4 2 30 12 0 4 121 0 83 819 0 89 1 2 0 0 267 620 0 0 0 0 0 0 0 460
CIUDADELA REAL DE MINAS 11 1 4 3 7 0 9 12 0 18 0 7 1 0 0 9 0 7 5 0 4 119 1 69 1262 0 77 0 0 0 0 338 373 0 0 3 0 0 0 7 242
ANTONIA SANTOS SUR 12 0 7 3 12 0 2 8 0 16 1 1 0 0 1 6 0 10 13 0 1 84 0 39 718 0 68 0 1 0 0 777 326 0 0 0 0 0 0 0 228
COMUNEROS 19 2 7 2 6 0 4 6 0 26 0 4 0 0 0 6 1 32 10 0 4 77 0 131 731 0 93 0 0 0 0 455 338 0 0 1 0 0 0 3 277
CAFE MADRID 28 1 9 3 18 0 3 21 0 54 0 7 0 0 0 3 0 54 23 3 4 78 0 46 473 1 56 0 0 0 0 304 550 0 0 0 0 0 0 0 477
LA AURORA 2 1 3 1 4 0 2 6 0 13 0 1 0 0 0 13 0 7 9 0 5 118 2 66 1050 0 79 0 0 3 0 349 306 0 0 0 0 0 0 0 78
ALARCON 11 1 9 2 9 0 8 2 0 21 0 0 0 0 0 15 0 32 6 0 0 149 0 47 797 0 53 0 2 0 0 382 339 0 1 2 1 0 0 1 183
LA UNIVERSIDAD 3 0 2 1 2 0 11 5 0 9 0 0 0 0 0 3 2 7 2 0 1 128 0 133 1053 0 111 0 0 0 0 215 248 0 0 0 0 0 0 0 93
ALFONSO LOPEZ 9 0 4 2 4 0 13 6 0 30 0 7 0 0 0 12 0 13 4 0 2 76 0 60 531 0 94 0 1 0 0 269 528 0 0 0 0 0 0 1 278
MEJORAS PUBLICAS 1 0 3 2 1 0 0 2 0 4 0 3 0 0 1 4 1 2 8 0 4 145 0 42 948 0 68 0 0 0 0 391 212 1 0 0 0 0 0 1 83
MUTIS 10 0 2 2 2 0 2 7 0 28 2 2 0 0 0 6 0 7 5 0 6 37 0 74 581 0 85 0 1 0 1 286 321 0 0 2 0 0 0 1 250
ACCESO CARNAL ABUSIVO CON MENOR DE 14 AÑOS
ACCESO CARNAL ABUSIVO CON MENOR DE 14 AÑOS (CIRCUNSTANCIAS AGRAVACIÓN)
ACCESO CARNAL O ACTO SEXUAL ABUSIVO CON INCAPAZ DE RESISTIR
ACCESO CARNAL O ACTO SEXUAL EN PERSONA PUESTA EN INCAPACIDAD DE RESISTIR
ACCESO CARNAL VIOLENTO
ACCESO CARNAL VIOLENTO (CIRCUNSTANCIAS AGRAVACIÓN)
ACOSO SEXUAL
ACTO SEXUAL VIOLENTO
ACTO SEXUAL VIOLENTO (CIRCUNSTANCIAS DE AGRAVACIÓN)
ACTOS SEXUALES CON MENOR DE 14 AÑOS
ACTOS SEXUALES CON MENOR DE 14 AÑOS (CIRCUNSTANCIAS DE AGRAVACIÓN)
AMENAZAS
CONSTREÑIMIENTO A LA PROSTITUCIÓN
DEMANDA DE EXPLOTACION SEXUAL COMERCIAL DE PERSONA MENOR DE 18 AÑOS DE EDAD
ESTÍMULO A LA PROSTITUCIÓN DE MENORES
EXTORSIÓN
FEMINICIDIO
HOMICIDIO
HOMICIDIO CULPOSO (EN ACCIDENTE DE TRANSITO)
HURTO A ABIGEATO
HURTO A AUTOMOTORES
HURTO A ENTIDADES COMERCIALES
HURTO A ENTIDADES FINANCIERAS
HURTO A MOTOCICLETAS
HURTO A PERSONAS
HURTO A PIRATERÍA TERRESTRE
HURTO A RESIDENCIAS
INCAPACIDAD PARA TRABAJAR O ENFERMEDAD
INDUCCIÓN A LA PROSTITUCIÓN
LESIONES AL FETO
LESIONES CULPOSAS
LESIONES CULPOSAS (EN ACCIDENTE DE TRANSITO)
LESIONES PERSONALES
LESIONES PERSONALES (CIRCUNSTANCIAS DE AGRAVACIÓN)
NO REPORTA
PORNOGRAFÍA CON MENORES
PROXENETISMO CON MENOR DE EDAD
SECUESTRO EXTORSIVO
SECUESTRO SIMPLE
SECUESTRO SEXUALES DE MENORES
UTILIZACIÓN O FACILITACIÓN DE MEDIOS DE COMUNICACIÓN PARA OFRECER SERVICIOS SEXUALES DE MENORES
VIOLENCIA INTRAFAMILIAR
5000
4000
3000
2000
1000
0

PROBANDO MODELOS, COMPARÁNDOLOS Y HACIENDO PREDICCIONES

```
from sklearn.compose import ColumnTransformer

# Preparando los datos para la clasificación

# Selecciono las columnas que necesito y elimino filas vacías
classification_data = clean_data[['BARRIOS_HECHO', 'DIA_SEMANA', 'GENERO', 'CONDUCTA']].dropna()

# Transformando datos categóricos a numéricos
label_encoder = LabelEncoder()
onehot_encoder = ColumnTransformer(transformers=[('onehot', OneHotEncoder(), ['BARRIOS_HECHO', 'DIA_SEMANA', 'GENERO'])],
                                   remainder='passthrough')

X = onehot_encoder.fit_transform(classification_data[['BARRIOS_HECHO', 'DIA_SEMANA', 'GENERO']])
y = label_encoder.fit_transform(classification_data['CONDUCTA'])

# Dividiendo datos de entrenamiento y prueba
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

# Se inicializan modelos
dt_classifier = DecisionTreeClassifier(random_state=42)
rf_classifier = RandomForestClassifier(random_state=42)
svm_classifier = SVC(random_state=42)
mlp_classifier = MLPClassifier(random_state=42)

# Entrenando modelos
dt_classifier.fit(X_train, y_train)
rf_classifier.fit(X_train, y_train)
svm_classifier.fit(X_train, y_train)
mlp_classifier.fit(X_train, y_train)

# Haciendo predicciones
dt_pred = dt_classifier.predict(X_test)
rf_pred = rf_classifier.predict(X_test)
svm_pred = svm_classifier.predict(X_test)
mlp_pred = mlp_classifier.predict(X_test)

# Evaluando los modelos
models = {'Decision Tree': dt_pred, 'Random Forest': rf_pred, 'SVM': svm_pred, 'Neural Network': mlp_pred}
performance = {model: accuracy_score(y_test, prediction) for model, prediction in models.items()}

performance


```

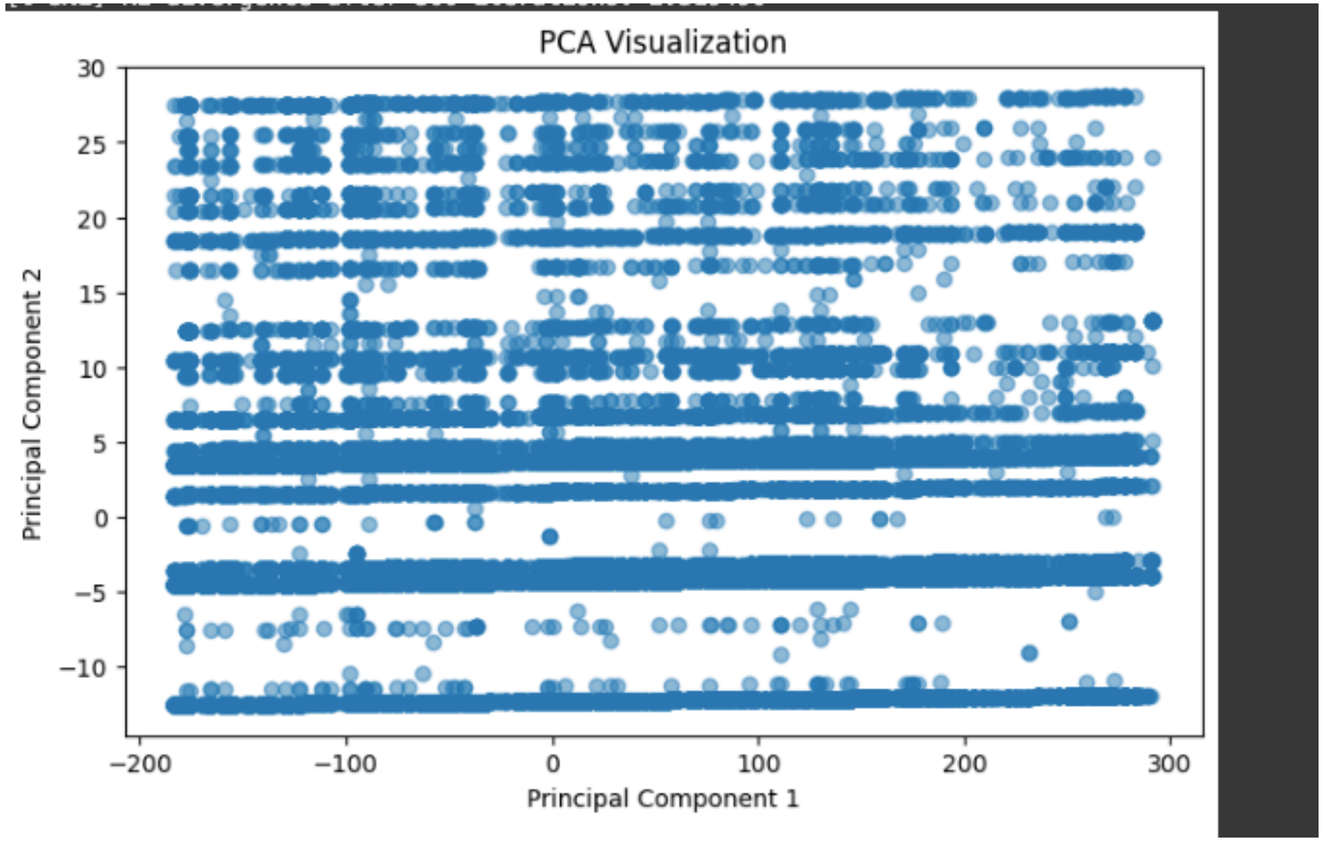
```
from sklearn.compose import ColumnTransformer

# Preparando los datos para la clasificación

# Selecciono las columnas que necesito y elimino filas vacías
classification_data = clean_data[['BARRIOS_HECHO', 'DIA_SEMANA', 'GENERO', 'CONDUCTA']].dropna()

# Transformando datos categóricos a numéricos
label_encoder = LabelEncoder()
onehot_encoder = ColumnTransformer(transformers=[('onehot', OneHotEncoder(), ['BARRIOS_HECHO', 'DIA_SEMANA', 'GENERO'])],
                                   remainder='passthrough')

X = onehot_encoder.fit_transform(classification_data[['BARRIOS_HECHO', 'DIA_SEMANA', 'GENERO']])
y = label_encoder.fit_transform(classification_data['CONDUCTA'])

# Dividiendo datos de entrenamiento y prueba
X_train, X_test, y_train,

# Se inicializan modelos
dt_classifier = DecisionTr
rf_classifier = RandomFore
svm_classifier = SVC(rando
mlp_classifier = MLPClassi

# Entrenando modelos
dt_classifier.fit(X_train, y_train)
rf_classifier.fit(X_train, y_train)
svm_classifier.fit(X_train, y_train)
mlp_classifier.fit(X_train, y_train)

# Haciendo predicciones
dt_pred = dt_classifier.predict(X_test)
rf_pred = rf_classifier.predict(X_test)
svm_pred = svm_classifier.predict(X_test)
mlp_pred = mlp_classifier.predict(X_test)

# Evaluando los modelos
models = {'Decision Tree': dt_pred, 'Random Forest': rf_pred, 'SVM': svm_pred, 'Neural Network': mlp_pred}
performance = {model: accuracy_score(y_test, prediction) for model, prediction in models.items()}

performance

```

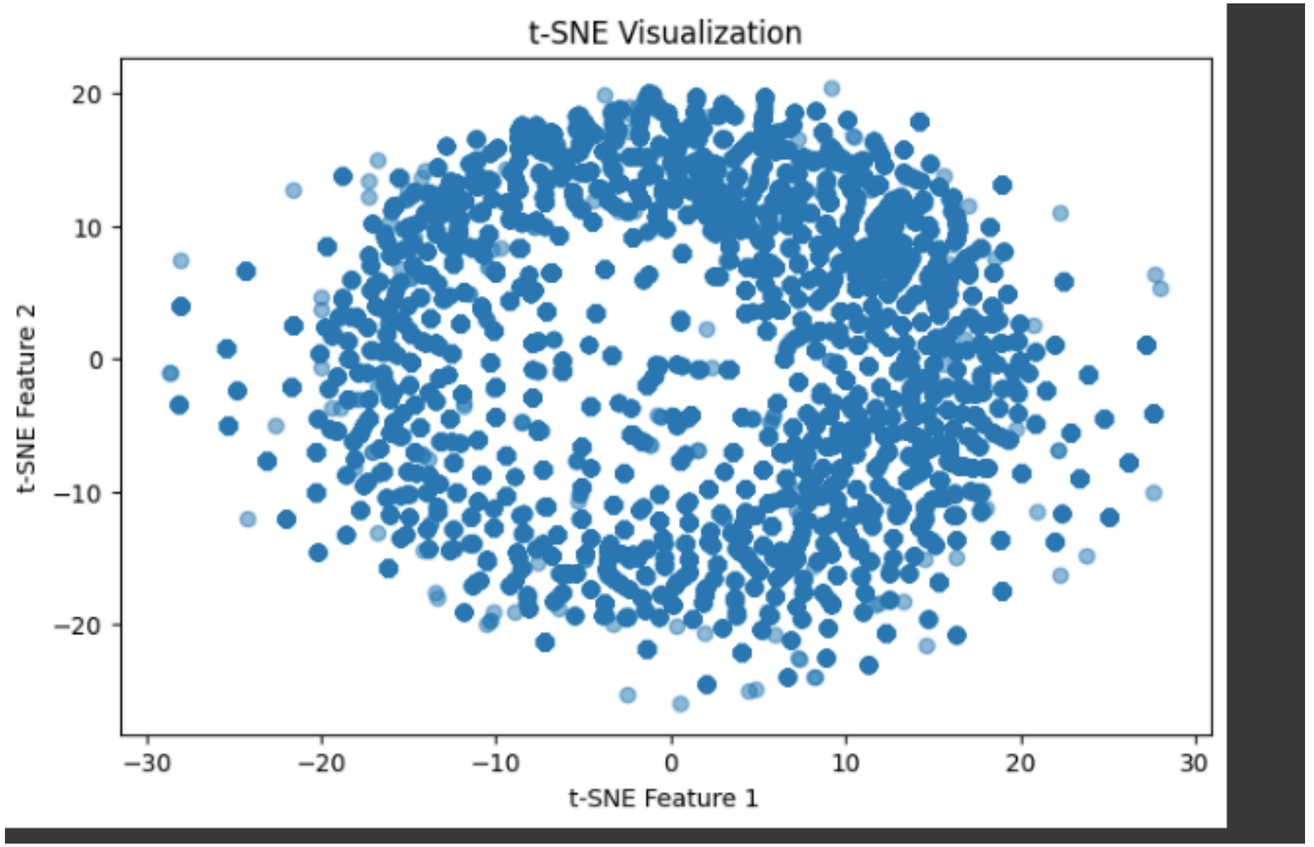
```
{'Decision Tree': 0.4416096193194737,
 'Random Forest': 0.4441357011140021,
 'SVM': 0.4528759441230707,
 'Neural Network': 0.4445146133831813}
```

APLICANDO KMEANS,PCA Y T-SNE

```python
def get_similar_records(row, model, train_data, le_conducta, le_barrios):
    # Predecir el cluster para el registro dado
    conducta_encoded = le_conducta.transform([row['CONDUCTA']])[0]
    barrios_hecho_encoded = le_barrios.transform([row['BARRIOS_HECHO']])[0]
    cluster = model.predict([[conducta_encoded, barrios_hecho_encoded]])[0]

    # Obtener registros del mismo cluster
    same_cluster_data = train_data[train_data_kmeans.apply(lambda x: model.predict([x])[0], axis=1) == cluster]

    # Seleccionar tres registros aleatorios del mismo cluster
    if len(same_cluster_data) > 3:
        return same_cluster_data.sample(3)
    else:
        return same_cluster_data

# Eliminar registros con valores desconocidos en 'BARRIOS_HECHO'
known_barrios = set(le_barrios.classes_)
test_data = test_data[test_data['BARRIOS_HECHO'].isin(known_barrios)]

# Aplicar LabelEncoder al conjunto de prueba corregido
test_data['BARRIOS_HECHO_encoded'] = le_barrios.transform(test_data['BARRIOS_HECHO'])

# Probar la función con un registro de prueba
example = test_data.iloc[0]
similar_records = get_similar_records(example, kmeans_with_categorical, train_data, le_conducta, le_barrios)
```

- Inicialmente apliqué k means para los valores numéricos solamente, pero no eran muy significativos para un análisis interesante(orden, año y dia)
- Decidí encontrar relaciones entre tipo de delito y ubicación del delito para eso tomé las columnas correspondientes y apliqué todo el procesamiento en estas columnas categóricas convirtiéndolas en numéricas con label encoding.
- Finalmente creo que se pudo encontrar casos similares o representativos dentro de un conjunto de datos grande. Esto podrá ser útil en aplicaciones como análisis criminal donde encontrar incidentes similares puede ayudar a entender mejor un caso específico.(de acuerdo a análisis previos coincide con los delitos más comunes y las ubicaciones más peligrosas)


PCA Visualization
Principal Component 2
Principal Component 1
30
25
20
15
10
5
0
−5
−10
−200
−100
0
100
200
300


t-SNE Visualization
t-SNE Feature 2
t-SNE Feature 1
20
10
0
−10
−20
−30
−20
−10
0
10
20
30

CONCLUSIONES

- Se logró extraer información relevante para la toma de decisiones futura
- La visualización de datos aporta en gran medida a la comprensión del dataset y el contexto de la situación
- La limpieza es fundamental para aumentar la confiabilidad de los análisis y resultados
- Se logró identificar las zonas acorde al objetivo del proyecto. Sin embargo se desea implementar un mapa interactivo a nivel web a futuro.

GRACIAS


EL OJ
DE BUCAR
Carlos Javier Aceros Blanco
Inteligencia artificial I - Proyecto final
Universidad Industrial de Santander