# ▼ Variable "parroquia\_urbana"

Cargamos las librerias necesarias para la elaboracion y desarrollo de la minería de datos

```
!pip install plotly
import numpy as np #Operaciones matemáticas rápidas sobre matrices
import pandas as pd #biblioteca de análisis y manipulación de datos para Python
import plotly.express as px
import matplotlib.pyplot as plt #Proporciona una forma de trazado similar a MATLAB. pyplot está diseñado principalmente para gráficos in
import seaborn as sns #permite generar fácilmente elegantes gráficos, proporciona una interfaz de alto nivel que es realmente sencilla d
import statsmodels.api as sm
# Preprocesado y modelado
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.tree import plot_tree
from sklearn.tree import export graphviz
from sklearn.tree import export_text
from sklearn.model_selection import GridSearchCV
{\it from sklearn.} compose {\it import ColumnTransformer}
from sklearn.preprocessing import OneHotEncoder
from sklearn.metrics import accuracy score
from sklearn.metrics import confusion_matrix
# Configuración warnings
import warnings
warnings.filterwarnings('once')
df= pd.read_csv('AT2021_NBD.csv')
```

Requirement already satisfied: plotly in /usr/local/lib/python3.10/dist-packages (5.13.1)
Requirement already satisfied: tenacity>=6.2.0 in /usr/local/lib/python3.10/dist-packages (from plotly)

	fecha	dia	hora	latitud	longitud	tipologia	direccion	zona	parroquia_urbana	cau
0	01-01	viernes	h11	-3.991993	-79.201155	estrellamiento	orillas del zamora y jose felix de valdivieso	urbana	el valle	conduci estad embriaç
1	01-05	martes	h17	-4.020370	-79.217962	choque lateral perpendicular	benjamin carrion y gustavo serrano	urbana	punzara	conduci estadi embriaç
2	01-05	martes	h12	-3.987230	-79.202984	choque por alcance	nueva loja y guaranda	urbana	sucre	no mante la dista reglament
3	01-08	viernes	h10	-3.989410	-79.236506	atropello	angel felicisimo rojas	urbana	sucre	imprude del pea
4	01-09	sabado	h06	-3.979784	-79.218689	choque lateral angular	isidro ayora y habana	urbana	sucre	conduci estadı embriaç



Graficamos el mapa de calor de accidentabilidad dentro del cantón Loja

fig = px.density\_mapbox(df,lat='latitud', lon='longitud',radius=3,center=dict(lat=-3.99313,lon=-79.20422),zoom=10.5,mapbox\_style="open-s
fig.show()

```
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning:
```

`should\_run\_async` will not call `transform\_cell` automatically in the future. Please pass the result 1

```
#verifico datos nulos
df.isnull().sum()
     /usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning:
     `should_run_async` will not call `transform_cell` automatically in the future. Please pass the result to `transformed_cell` argumer
     fecha
                               0
     dia
                               0
     hora
                               0
     latitud
                               1
     longitud
                               1
     tipologia
                               0
     direccion
                               0
     zona
                               0
     parroquia_urbana
                               0
     causas
                               0
     gravedad
                               0
     nro_heridos
                               0
     nro_fallecidos
                               0
     vehiculos retenidos
                               0
     senalizacion existente
                               0
     condicion_calzada
                               0
     condicion_atmosferica
                               0
     dtype: int64
     4
df=df.loc[:,df.columns!="fecha"]
df=df.loc[:,df.columns!="zona"]
df=df.loc[:,df.columns!="latitud"]
df=df.loc[:,df.columns!="longitud"]
df=df.loc[:,df.columns!="direccion"]
df=df.loc[:,df.columns!="gravedad"]
df=df.loc[:,df.columns!="vehiculos_retenidos"]
df=df.loc[:,df.columns!="senalizacion_existente"]
df=df.loc[:,df.columns!="condicion_calzada"]
df=df.loc[:,df.columns!="condicion_atmosferica"]
     /usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning:
     `should_run_async` will not call `transform_cell` automatically in the future. Please pass the result to `transformed_cell` argumer
    4
df.isnull().sum()
     dia
                         0
     hora
                         0
     tipologia
                         0
     parroquia_urbana
                         0
     causas
                         0
     nro_heridos
                         0
     nro_fallecidos
     dtype: int64
df.sample(10)
```

`should\_run\_async` will not call `transform\_cell` automatically in the future. Please pass the result t

	dia	hora	tipologia	parroquia_urbana	causas	nro_heridos	nro_fallecidos
35	domingo	h16	estrellamiento	punzara	conducir en estado de embriaguez	2	0
158	domingo	h05	estrellamiento	punzara	conducir en exceso de velocidad	0	0
212	sabado	h21	perdida de carril	carigan	conducir en estado de embriaguez	0	0
106	sabado	h15	choque por alcance	sucre	imprudencia del conductor	1	0
23	sabado	h21	estrellamiento	el valle	conducir en exceso de velocidad	0	0
274	sabado	h04	estrellamiento	el sagrario	conducir en exceso de velocidad	0	0
259	domingo	h20	choque lateral	el sagrario	conducir en estado	2	0
df.head()							

/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning:

`should\_run\_async` will not call `transform\_cell` automatically in the future. Please pass the result 1

	dia	hora	tipologia	parroquia_urbana	causas	nro_heridos	nro_fallecidos
0	viernes	h11	estrellamiento	el valle	conducir en estado de embriaguez	1	0
1	martes	h17	choque lateral perpendicular	punzara	conducir en estado de embriaguez	1	0
2	martes	h12	choque por alcance	sucre	no mantener la distancia reglamentaria	0	0
3	viernes	h10	atropello	sucre	imprudencia del peaton	1	0

`should\_run\_async` will not call `transform\_cell` automatically in the future. Please pass the result to `transformed\_cell` argumer

# tipos de la variables

```
ds.dtypes
```

dia object
hora object
tipologia object
parroquia\_urbana object
causas object
nro\_heridos int64
nro\_fallecidos dtype: object

df.shape[0]

370

df['causas'].value\_counts()

```
imprudencia del conductor
conducir en estado de embriaguez
conducir en exceso de velocidad
formudencia del peaton
no respetar las senales de transito
fallas mecanicas no previsibles
formunate del senales de transito
fallas mecanicas no previsibles
formunate del senales de transito
fallas mecanicas no previsibles
formunate del senales de transito
fallas mecanicas no previsibles
formunate del senales de transito
fallas mecanicas no previsibles
formunate del senales d
```

```
condiciones climaticas desfavorables
     no ceder el derecho de via
                                                    2
     impericia del conductor
                                                    2
     cruce de animales en la via
     Name: causas, dtype: int64
print(df['causas'].unique())# datos en texto
     ['conducir en estado de embriaguez'
       'no mantener la distancia reglamentaria' 'imprudencia del peaton'
      'conducir en exceso de velocidad' 'no respetar las senales de transito'
      'condiciones climaticas desfavorables' 'no ceder el derecho de via'
      'impericia del conductor' 'imprudencia del conductor'
      'fallas mecanicas no previsibles' 'cruce de animales en la via']
     /usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning:
     `should_run_async` will not call `transform_cell` automatically in the future. Please pass the result to `transformed_cell` argumer
    4
print(df['causas'].unique())# datos en texto
     ['conducir en estado de embriaguez'
       'no mantener la distancia reglamentaria' 'imprudencia del peaton'
      'conducir en exceso de velocidad' 'no respetar las senales de transito'
      'condiciones climaticas desfavorables' 'no ceder el derecho de via'
      'impericia del conductor' 'imprudencia del conductor'
      'fallas mecanicas no previsibles' 'cruce de animales en la via']
Transformamos los datos a numéricos
df['causas']= df['causas'].apply(lambda x:
                                      1 if x == 'conducir en estado de embriaguez' else
                                      2 if x == 'imprudencia del conductor' else
                                      3 if x == 'no ceder el derecho de via' else
                                      4 if x == 'conducir en exceso de velocidad' else
                                      5 if x == 'cambio brusco e indebido de carril' else
                                      6 if x == 'fallas mecanicas no previsibles' else
                                      7 if x == 'no respetar las senales de transito' else
                                      8 if x == 'invadir carril de circulacion' else
                                      9 if x == 'imprudencia del peaton' else
                                      10 if x == 'no mantener la distancia reglamentaria' else
                                      11 if x == 'razones desconocidas' else
                                     12 if x == 'condiciones climaticas desfavorables' else
                                      13 if x == 'negligencia del conductor' else
                                      14 if x == 'no respetar las ordenes del agente de transito' else
                                      15 if x == 'impericia del conductor' else
                                      16)
print(df['parroquia_urbana'].unique()) # datos convertidos a numeros enteros
     ['el valle' 'punzara' 'sucre' 'san sebastian' 'carigan' 'el sagrario']
df['parroquia_urbana']= df['parroquia_urbana'].apply(lambda x:
                                     1 if x == 'el sagrario' else
                                      2 if x == 'san sebastian' else
                                      3 if x == 'el valle' else
                                      4 if x == 'sucre' else
                                      5 if x == 'punzara' else
                                      6)
print(df['parroquia_urbana'].unique()) # datos convertidos a numeros enteros
     [3 5 4 2 6 1]
print(df['tipologia'].unique()) # datos convertidos a numeros enteros
     ['estrellamiento' 'choque lateral perpendicular' 'choque por alcance' 'atropello' 'choque lateral angular' 'roce negativo' 'perdida de carril'
      'perdida de pista' 'choque frontal excentrico' 'colision' 'atipico' 'arrollamiento' 'volcamiento' 'roce positivo' 'rozamiento'
      'caida de pasajero']
df['tipologia']= df['tipologia'].apply(lambda x:
                                     1 if x == 'arrollamiento' else
                                      2 if x == 'atipico' else
                                      3 if x == 'atropello' else
```

```
4 if x == 'caida de pasajero' else
                                   5 if x == 'choque frontal' else
                                    6 if x == 'choque frontal excentrico' else
                                    7 if x == 'choque frontal longitudinal' else
                                    8 if x == 'choque lateral angular' else
                                    9 if x == 'choque lateral perpendicular' else
                                    10 if x == 'choque por alcance' else
                                    11 if x == 'colision' else
                                   12 if x == 'encunetamiento' else
                                   13 if x == 'estrellamiento' else
                                   14 if x == 'perdida de carril' else
                                   15 if x == 'perdida de pista' else
                                   16 if x == 'roce negativo' else
                                    17 if x == 'roce positivo' else
                                   18 if x == 'rozamiento' else
                                    19 if x == 'volcamiento' else
                                    20 if x == 'volcamiento lateral' else
                                    21)
print(df['tipologia'].unique()) # datos convertidos a numeros enteros
     [13 9 10 3 8 16 14 15 6 11 2 1 19 17 18 4]
df['dia']= df['dia'].apply(lambda x:
                                   1 if x == 'lunes' else
                                    2 if x == 'martes' else
                                    3 if x == 'miercoles' else
                                    4 if x == 'jueves' else
                                    5 if x == 'viernes' else
                                    6 if x == 'sabado' else
                                    7)
print(df['dia'].unique()) # datos convertidos a numeros enteros
     [5 2 6 7 1 4 3]
df= df[df['parroquia_urbana'] <= 6]</pre>
df['hora']= df['hora'].apply(lambda x:
                                   0 if x == 'h00' else
                                    1 if x == 'h01' else
                                   2 if x == 'h02' else
                                   3 if x == 'h03' else
                                   4 if x == 'h04' else
                                    5 if x == 'h05' else
                                   6 if x == 'h06' else
                                    7 if x == 'h07' else
                                   8 if x == 'h08' else
                                   9 if x == 'h09' else
                                   10 if x == 'h10' else
                                    11 if x == 'h11' else
                                   12 if x == 'h12' else
                                   13 if x == 'h13' else
                                   14 if x == 'h14' else
                                   15 if x == 'h15' else
                                   16 if x == 'h16' else
                                   17 if x == 'h17' else
                                   18 if x == 'h18' else
                                   19 if x == 'h19' else
                                    20 if x == 'h20' else
                                    21 if x == 'h21' else
                                    22 if x == 'h22' else
                                    23)
df.head()
```

	dia	hora	tipologia	parroquia_urbana	causas	nro_heridos	nro_fallecidos	1	th
0	5	11	13	3	1	1	0		
1	2	17	9	5	1	1	0		
2	2	12	10	4	10	0	0		
3	5	10	3	4	9	1	0		
4	6	6	8	4	1	0	0		

# datos aleatorios (muestra de 10 elementos)

df.sample(10)

/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning:

`should\_run\_async` will not call `transform\_cell` automatically in the future. Please pass the result 1

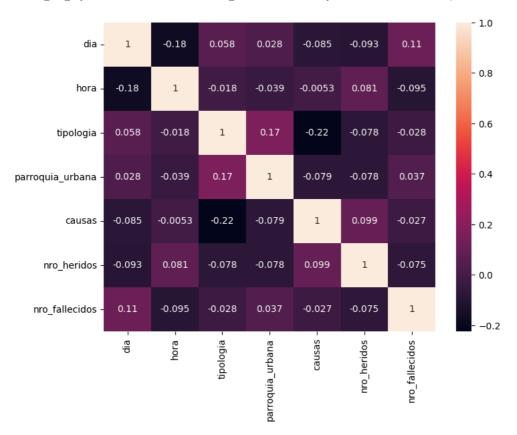
	dia	hora	tipologia	parroquia_urbana	causas	nro_heridos	nro_fallecidos	<b>/</b>
346	3	20	8	2	2	0	0	
64	4	20	13	4	1	1	0	
330	6	20	13	4	2	0	1	
213	6	23	10	1	1	0	0	
5	7	10	10	4	10	0	0	
125	2	0	13	6	4	0	0	
94	4	11	13	2	16	0	0	
241	4	7	9	1	7	0	0	
10	4	23	13	4	4	1	0	
209	3	23	13	1	2	1	0	

#### Verificamos la correlaxion de los datos

```
#corelacion
corr_df = df.corr(method='pearson')
plt.figure(figsize=(8, 6))
sns.heatmap(corr_df, annot=True)
plt.show()
```

 $/usr/local/lib/python 3.10/dist-packages/ipykernel/ipkernel.py: 283: \ Deprecation Warning: 1.00/dist-packages/ipykernel/ipkernel.py: 283: \ Deprecation Warning: 1.00/dist-packages/ipykernel/ipkernel$ 

`should\_run\_async` will not call `transform\_cell` automatically in the future. Please pass the result 1



X = df.iloc[:, [0,1,2,3,4,6]] # atributos de entrada seran las primeras columnas <math>Y = df.iloc[:,[5]] # atributos de destino

#presentacion de los atributos de entrada X.head()

```
dia hora tipologia parroquia_urbana causas nro_fallecidos
0
    5
          11
                     13
                                         3
                                                 1
                                                                  0
    2
                      9
                                         5
1
          17
                                                 1
                                                                  0
2
    2
          12
                     10
                                         4
                                                10
                                                                  0
                                         4
3
          10
                      3
                                                 9
                                                                  0
    5
    6
           6
                                                                  0
```

feature names = X.columns.tolist()

```
# Imprimir los nombres de las características
print(feature_names)
```

```
['dia', 'hora', 'tipologia', 'parroquia_urbana', 'causas', 'nro_fallecidos']
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning:
```

#presentacion de los atributos de destino Y.head()

nro_	_heridos	1	ıl.
0	1		
1	1		
2	0		
3	1		
4	0		

### Cargamos el modelo

```
from joblib import load
arbol_modelo = load('modelo_nro_heridos.joblib')
y_pred = arbol_modelo.predict(X)
```

/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning:

`should\_run\_async` will not call `transform\_cell` automatically in the future. Please pass the result to `transformed\_cell` argumer

4

### print(y\_pred)

/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning:

`should run async` will not call `transform cell` automatically in the future. Please pass the result to `transformed cell` argumer

```
# Resumen de las predicciones hechas por el clasificador
from sklearn import metrics
reporte = metrics.classification_report(Y, y_pred,output_dict=True)
pre = pd.DataFrame(reporte).transpose()
print(pre)
pre.to_excel("resumen_precision_nro_heridos_2021.xlsx")
                                recall f1-score
                   precision
                                                     support
```

```
0
              0.619883 0.972477 0.757143 218.000000
              0.714286 0.172414 0.277778 116.000000
```

<sup>`</sup>should\_run\_async` will not call `transform\_cell` automatically in the future. Please pass the result to `transformed\_cell` argumer

```
0.000000 0.000000
                                  0.000000
                                             26.000000
3
               0.000000 0.000000
                                  0.000000
                                              6.000000
               0.000000
                        0.000000
                                  0.000000
                                              1.000000
4
               0.000000
                        0.000000 0.000000
                                              1.000000
6
               0.000000
                        0.000000
                                  0.000000
                                              1.000000
               0.000000 0.000000 0.000000
                                              1,000000
                        0.627027
                                  0.627027
accuracy
               0.627027
                                              0.627027
macro avg
               0.166771
                        0.143111
                                  0.129365
                                            370.000000
weighted avg
              0.589167 0.627027 0.533187 370.000000
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning:
Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to cc
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning:
Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero division` parameter to co
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning:
Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to cc
<frozen importlib._bootstrap>:914: ImportWarning:
APICoreClientInfoImportHook.find_spec() not found; falling back to find_module()
<frozen importlib. bootstrap>:914: ImportWarning:
_PyDriveImportHook.find_spec() not found; falling back to find_module()
<frozen importlib._bootstrap>:914: ImportWarning:
_OpenCVImportHook.find_spec() not found; falling back to find_module()
<frozen importlib._bootstrap>:914: ImportWarning:
BokehImportHook.find spec() not found; falling back to find module()
<frozen importlib. bootstrap>:914: ImportWarning:
_AltairImportHook.find_spec() not found; falling back to find_module()
```

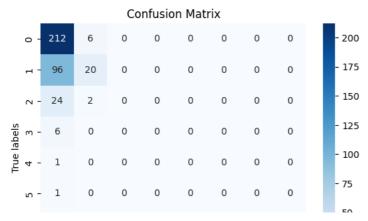
#### Pedicciones

### Generamos la matriz de confusión

4 0

```
#Matriz de confusion
matriz=confusion_matrix(Y, y_pred)
ax= plt.subplot()
sns.heatmap(matriz, annot=True, cmap="Blues",fmt='g');
ax.set_xlabel('Predicted labels');ax.set_ylabel('True labels');
ax.set_title('Confusion Matrix');
```

`should\_run\_async` will not call `transform\_cell` automatically in the future. Please pass the result 1



#Para concatenar variables
import pandas as pd

```
prediccion = pd.DataFrame(y_pred,columns=['nro_heridos_prediccion'])
original = Y
original.reset_index(drop=True, inplace=True)
df_combined = pd.concat([prediccion,original], axis=1)
df combined.head(10)
```

/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning:

`should\_run\_async` will not call `transform\_cell` automatically in the future. Please pass the result 1

	nro_heridos_prediccion	nro_heridos	1
0	0	1	
1	0	1	
2	0	0	
3	1	1	
4	0	0	
5	0	0	
6	0	1	
7	0	0	
8	0	1	
9	0	0	

#### Visualizamos la exactitud del modelo

```
# Precisión
```

```
from sklearn.metrics import accuracy_score, recall_score, precision_score
print('Exactitud árboles de decisión: ',accuracy_score(pred,Y)*100)
print('Exhaustividad árboles de decisión: ', recall_score(pred,Y,average='micro')*100)
print('Precision árboles de decisión: ',accuracy_score(pred,Y)*100)

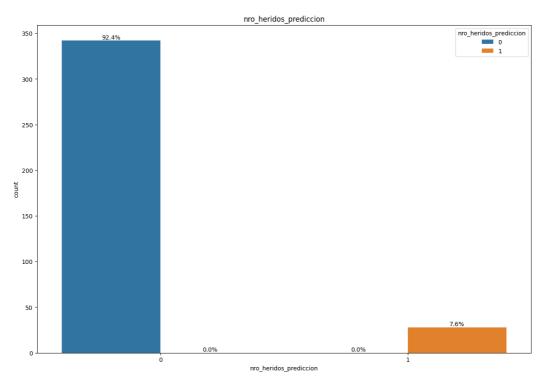
Exactitud árboles de decisión: 62.70270270270271
Exhaustividad árboles de decisión: 62.70270270270271
Precision árboles de decisión: 62.70270270270271
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning:
```

`should\_run\_async` will not call `transform\_cell` automatically in the future. Please pass the result to `transformed\_cell` argumer

# Graficamos las estadisticas originales de los datos

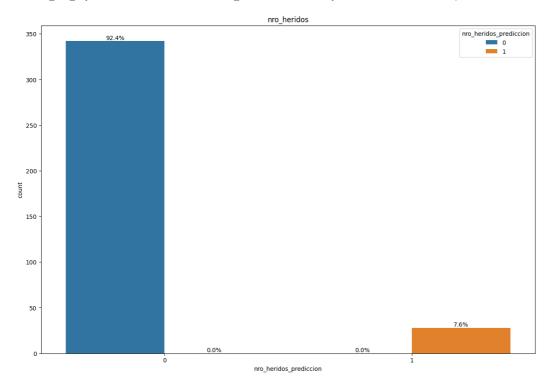
```
ax = plt.subplots(figsize = (15,10))
ncount=len(df_combined)
sns.countplot(x='nro_heridos_prediccion', hue = 'nro_heridos_prediccion', data = df_combined, ax = ax[1]) #Muestre el conteo de observaci
ax[1].set_title('nro_heridos_prediccion')
```

`should\_run\_async` will not call `transform\_cell` automatically in the future. Please pass the result t



# Graficamos las estadisticas predecidas de los datos

`should\_run\_async` will not call `transform\_cell` automatically in the future. Please pass the result 1



Crear el arbol de desición actual con el modelo con datos predichos del 2021

```
#target_names_str = [str(name) for name in target_names]
variable= df['nro_heridos']

claseVar = variable.unique().tolist()
target_names_str = [str(name) for name in claseVar]
```

 $/usr/local/lib/python 3.10/dist-packages/ipykernel/ipkernel.py: 283: \ Deprecation Warning: \\$ 

`should\_run\_async` will not call `transform\_cell` automatically in the future. Please pass the result to `transformed\_cell` argumer

# Creamos la figura del arbol

figura\_arbol = plt.figure(figsize=(40,20)) # Le indicamos las dimensiones que queremos que tenga
plot\_tree(arbol\_modelo,feature\_names=feature\_names,filled=True, class\_names=target\_names\_str, fontsize=15)
plt.show()

3

4

365

14.04

77.08

77.08



Verificamos la probabilidad de los accidentes de tránsito a través de la predicción del modelo

```
#Verificamos la probabilidad de todas las variables
y_proba = arbol_modelo.predict_proba(X)
probabilidad_acierto = np.round(y_proba[0][y_pred] * 100, 2)
print("Probabilidad de Acierto: " + str(probabilidad acierto) + "%")
     Probabilidad de Acierto: [77.08 77.08 77.08 14.04 77.08 77.08 77.08 77.08 77.08 77.08 77.08 77.08 77.08
      77.08 77.08 77.08 77.08 77.08 77.08 77.08 77.08 77.08 77.08 77.08
      77.08 77.08 77.08 77.08 77.08 77.08 77.08 77.08 77.08 77.08 77.08 77.08 77.08
      77.08 77.08 77.08 77.08 77.08 77.08 77.08 77.08 77.08 77.08 77.08 77.08
      77.08 77.08 77.08 77.08 77.08 77.08 77.08 77.08 77.08 77.08 77.08 77.08 77.08
      77.08 14.04 77.08 77.08 77.08 77.08 77.08 77.08 77.08 77.08 77.08
      77.08 77.08 77.08 77.08 77.08 77.08 77.08 77.08 77.08 77.08 77.08
      77.08 77.08 14.04 77.08 77.08 77.08 77.08 77.08 77.08 77.08 77.08 77.08
      77.08 77.08 77.08 77.08 77.08 77.08 77.08 77.08 77.08 77.08 77.08
      77.08 77.08 77.08 77.08 77.08 77.08 77.08 77.08 77.08 77.08 77.08
      77.08 14.04 77.08 77.08 77.08 77.08 77.08 77.08 77.08 14.04 77.08 77.08
      77.08 77.08 77.08 77.08 77.08 77.08 77.08 77.08 77.08 14.04 77.08 77.08
      14.04 77.08 14.04 77.08 14.04 77.08 77.08 14.04 77.08 77.08 77.08 77.08
      77.08 14.04 77.08 77.08 77.08 77.08 77.08 77.08 77.08 77.08 77.08 14.04
      77.08 77.08 77.08 77.08 77.08 14.04 77.08 14.04 77.08 77.08 77.08 77.08
      77.08 77.08 77.08 77.08 77.08 77.08 77.08 77.08 77.08 77.08 77.08
      77.08 77.08 77.08 77.08 77.08 77.08 77.08 77.08 77.08 77.08 77.08 14.04
      77.08 77.08 77.08 77.08 77.08 77.08 14.04 77.08 77.08 77.08 77.08 77.08
      77.08 77.08 14.04 77.08 77.08 77.08 77.08 77.08 77.08 77.08 77.08 77.08 77.08
      77.08 77.08 77.08 77.08 77.08 77.08 77.08 77.08 77.08 77.08 14.04 77.08
      77.08 77.08 77.08 77.08 77.08 77.08 14.04 77.08 77.08 77.08 77.08 77.08
      77.08 77.08 77.08 77.08 77.08 77.08 77.08 77.08 77.08 77.08 14.04 77.08
      77.08 77.08 77.08 14.04 77.08 77.08 77.08 77.08 77.08 77.08 77.08 77.08
      77.08 77.08 77.08 77.08 77.08 77.08 77.08 14.04 77.08 77.08 77.08 77.08
      77.08 77.08 77.08 77.08 77.08 77.08 77.08 14.04 77.08 77.08 77.08 77.08
      77.08 77.08 77.08 77.08 77.08 77.08 77.08 77.08 77.08 77.08 77.08
      77.08 77.08 77.08 14.04 77.08 77.08 77.08 77.08 77.08 77.08 77.08
      77.08 77.08 77.08 77.08 77.08 77.08 14.04 77.08 77.08 77.08 77.08
      77.08 77.08 77.08 77.08 77.08 77.08 77.08 77.08 77.08 77.08 77.08 14.04
      77.08 77.08 77.08 77.08 14.04 77.08 77.08 77.08 77.08 77.08 14.04
      77.08 77.08 77.08 77.08 77.08 77.08 77.08 77.08 77.08 77.08]%
     /usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning:
     `should_run_async` will not call `transform_cell` automatically in the future. Please pass the result to `transformed_cell` argumer
print(type(probabilidad acierto))
columna probabilidades = pd.DataFrame()
columna_probabilidades['probabilidad'] = pd.DataFrame(probabilidad_acierto)
print(columna_probabilidades)
     <class 'numpy.ndarray'>
          probabilidad
                 77.08
     1
                 77.08
     2
                 77.08
```

```
      366
      77.08

      367
      77.08

      368
      77.08

      369
      77.08
```

[370 rows x 1 columns]

/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning:

### Presentamos la probabilidad de los datos

```
y_prediccion = pd.DataFrame()
y_prediccion['predicciones'] = pd.DataFrame(y_pred)
probabilidades_2021_df = pd.concat([df,columna_probabilidades, y_prediccion], axis=1)
probabilidades_2021_df.sample(10)
```

/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning:

`should\_run\_async` will not call `transform\_cell` automatically in the future. Please pass the result  ${\tt t}$ 

	dia	hora	tipologia	parroquia_urbana	causas	nro_heridos	nro_fallecidos	probabilidad	predicc
177	5	11	16	4	2	0	0	77.08	
0	5	11	13	3	1	1	0	77.08	
319	2	14	9	1	2	1	0	77.08	
73	1	21	13	6	2	1	0	77.08	
72	6	17	11	5	1	0	0	77.08	
254	6	13	13	1	2	1	0	77.08	
142	1	19	13	6	1	0	0	77.08	
334	3	8	9	5	7	1	0	77.08	
331	6	6	8	5	2	0	0	77.08	
251	5	6	2	6	2	0	0	77.08	

```
** 11.
```

```
probabilidades\_2021\_df['dia'] = probabilidades\_2021\_df['dia'].apply(lambda \ x: lambda) = probabilidades\_2021\_df['dia'] = probabilidades\_2021\_df['dia'].apply(lambda) = probabilidades\_2021\_df['dia'] = probabilidades\_2021\_df['dia'].apply(lambda) = prob
                                                                                                                                  'lunes' if x == 1 else
                                                                                                                                  'martes' if x == 2 else
                                                                                                                                  'miercoles' if x == 3 else
                                                                                                                                  'jueves' if x == 4 else
                                                                                                                                  'viernes' if x == 5 else
                                                                                                                                  'sabado' if x == 6 else
                                                                                                                                  'domingo')
probabilidades_2021_df['hora'] = probabilidades_2021_df['hora'].apply(lambda x:
                                                                                                                                  'h00' if x == 0 else
                                                                                                                                  'h01' if x == 1 else
                                                                                                                                  'h02' if x == 2 else
                                                                                                                                  'h03' if x == 3 else
                                                                                                                                  'h04' if x == 4 else
                                                                                                                                  'h05' if x == 5 else
                                                                                                                                  'h06' if x == 6 else
                                                                                                                                  'h07' if x == 7 else
                                                                                                                                  'h08' if x == 8 else
                                                                                                                                  'h09' if x == 9 else
                                                                                                                                  'h10' if x == 10 else
                                                                                                                                 'h11' if x == 11 else
                                                                                                                                  'h12' if x == 12 else
                                                                                                                                  'h13' if x == 13 else
                                                                                                                                 'h14' if x == 14 else
                                                                                                                                  'h15' if x == 15 else
                                                                                                                                  'h16' if x == 16 else
                                                                                                                                  'h17' if x == 17 else
                                                                                                                                  'h18' if x == 18 else
                                                                                                                                  'h19' if x == 19 else
                                                                                                                                  'h20' if x == 20 else
                                                                                                                                  'h21' if x == 21 else
                                                                                                                                  'h22' if x == 22 else
                                                                                                                                  'h23')
```

 $probabilidades\_2021\_df['parroquia\_urbana'] = probabilidades\_2021\_df['parroquia\_urbana']. apply(lambda x: 'el sagrario' if x == 1 else$ 

<sup>`</sup>should\_run\_async` will not call `transform\_cell` automatically in the future. Please pass the result to `transformed\_cell` argumer

```
'san sebastian' if x == 2 else
'el valle' if x == 3 else
'sucre' if x == 4 else
'punzara' if x == 5 else
'carigan')
```

probabilidades\_2021\_df['tipologia'] = probabilidades\_2021\_df['tipologia'].apply(lambda x: 'arrollamiento' if x == 1 else 'atipico' if x == 2 else 'atropello' if x == 3 else 'caida pasajero' if x == 4 else 'choque frontal' if x == 5 else 'choque frontal excentrico' if x == 6 else 'choque frontal longitudinal' if x == 7 else 'choque lateral angular' if x == 8 else 'choque lateral perpendicular' if x == 9 else 'choque alcance' if x == 10 else 'colision' if x == 11 else 'encunetamiento' if x == 12 else 'estrellamiento' if x == 13 else 'perdida carril' if x == 14 else 'perdida pista' if x == 15 else 'roce negativo' if x == 16 else 'roce posositivo' if x == 17 else 'rozamiento' if x == 18 else 'volcamiento' if x == 19 else 'volcamiento lateral' if x == 20 else

probabilidades\_2021\_df['causas'] = probabilidades\_2021\_df['causas'].apply(lambda x: 'conducir en estado de embriaguez' if x == 1 else 'imprudencia del conductor' if x == 2 else 'no ceder el derecho de via' if x == 3 else 'conducir en exceso de velocidad' if x == 4 else 'cambio brusco e indebido de carril' if x == 5 else 'fallas mecanicas no previsibles' if x == 6 else 'no respetar las senales de transito' if x == 7 else 'invadir carril de circulacion' if x == 8 else 'imprudencia del peaton' if x == 9 else 'no mantener la distancia reglamentaria' if x == 10 else 'razones desconocidas' if x == 11 else 'condiciones climaticas desfavorables' if x == 12 else 'negligencia del conductor' if x == 13 else 'no respetar las ordenes del agente de transito' if x == 14 else 'impericia del conductor' if x == 15 else 'cruce de animales en la via')

'volcamiento longitudinal')

probabilidades\_2021\_df.sample(10)

/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning:

dia hana dinalanda nannanda nuhana

`should\_run\_async` will not call `transform\_cell` automatically in the future. Please pass the result 1

------ mus bouddes mus Collegides musbabilid

	dia	hora	tipologia	parroquia_urbana	causas	nro_heridos	nro_fallecidos	probabilid
191	viernes	h03	perdida carril	punzara	conducir en exceso de velocidad	0	0	77.
362	domingo	h13	choque lateral angular	sucre	conducir en estado de embriaguez	0	0	77.
210	jueves	h11	atropello	punzara	imprudencia del peaton	1	0	14.
181	martes	h16	choque lateral angular	sucre	imprudencia del conductor	0	0	77.
43	viernes	h01	estrellamiento	el valle	imprudencia del conductor	0	0	77.
147	domingo	h19	estrellamiento	el sagrario	imprudencia del conductor	1	0	77.
240	martes	h11	choque lateral perpendicular	el sagrario	no respetar las senales de transito	1	0	77.

import pandas as pd
import matplotlib.pyplot as plt
from matplotlib.backends.backend\_pdf import PdfPages

```
# Crear una figura y un eje
fig, ax = plt.subplots(figsize=(8, 6))

# Eliminar marcas del eje
ax.axis('off')

# Crear la tabla a partir del DataFrame
tabla = pd.plotting.table(ax, probabilidades_2021_df, loc='center', cellLoc='center', fontsize=14)

# Guardar la tabla en un archivo PDF
with PdfPages('tabla_probabilidades_nro_heridos_2021.pdf') as pdf:
    pdf.savefig(fig, bbox_inches='tight')
```

				D_NRO_H				_	
Merchin.		stronger latered geoperalization	rhapsis	contact or recovering artistical	1	L .	710	-	
nie de	6	drope bland paperalisates	chaputa	Improduceia del candanter	1		711		
whete	61	drope bland poperalisates	shapois	conductric restricts de contribuyous			7.00		
jeren	81	problepida		Managaran pedalin	,		7.0		
inn.	61	drope bland paperalisable	servetetien	Improduce dell'ambata	2		210		
less.	**	drospe Mend propositionine	servicion	construir en extrata de embriagas e			7.0		
meters		drope bland papersitedor	sereletin	Improduceia dell'ambata			70	1	
min	61	strope birmslangster		Improduced delicated and a			210		
ine.		atolaninis	pone	sando é se salada de sentráspas e	1		zu		
derings	81	strikerisets	pom	conductors extends de medicinguas	2		7.0		
		almagar hantad narantsis s	migar.	ingendenia del candado			710		
jenn		and and	stude	restore recent obtain			7.0		
derings		andaran.		contains recounty attacked			7.0		
nints	81	al claries	migra.	improduce dell'escalaries		-	711	1	
-		drope Mendangler		improducia del candador		-	70	,	
simin.		political	shale	restricted direction designation		<del>                                     </del>	70		
		_			1		_	_	
electric.	81	perditopida		conductive ratiosh describingues			7.0		
-		atrofamients	elude	Improduce dell'ambate			7.0		
derings	*	sintensis		rando ir ar ratuda de ambringum	1		710	,	
derings		atrolaminos	shapon	conductrus recovered actividad	1	- 1	7.0	1	
derings		drope bireal angular	sereletin	conductric ratiols de moletages	1		20		
jeren	61	drope bland poperatedor	servicio	sa corgoniar las camules de insculte			70		
ninis	81	perditopida		conductive recovering articipal			7.0		
derings		no septe	pone	contact or reliab describing as			70		
derings		atolianists		conductors extends the conductory and	-	-	710	1	
nimp.		_		_	-	-	7.0		
		drospo literal geoperalizados	****	Improduceia dell'assolution					
elecular.		drope bireal angles	****	rando ir se ratude de medelagues		1	20	1	
min		alcolarisate.		Improduce delicandaries			70		
derings		politicant		Marania mpolitic	,	,	7.0		
inn.	81	drope slame	miga-	improduccia dell'ambaton			7.0	1	
ins.		atolaminis	migra	contact or consists of cital			710	1	
mates	81	shape binni popositralo	sereletin	improduccia dell'ambaton	- 1	-	210	1	
derings	61	drope bland poperationly		sa corporisor des cantales de installa			710	1	
niels	61	drope bland papersinder		conductors recovered a charided	1		70		
-		drospe birmi proprisinalne	rhapura	Improduce delicandador			7.0		
riendo.	61	rep-h	chapmin	ingradienia di Cambalan	1		10		
ninis		drope bireal propositivalse	rhapsis	randoù er raindoù de medelaguez	1	-	7.0		
elet-		drope bland populatedo	rhapsis	ingradienia di Cambador	1	-	70	-	
		_		_				_	
jen		alcelanianis		conductron ratiosh de medelagues		-	710	1	
whete	61	drope bireal angles	sereletin	contact or recorder articidad	,		7.0	1	
		atolaniais		conductors recovered automited			710	,	
ine.		drosper Messel perpendicular	sereletin	Ingerulansia deli sentante	,		20		
electric.		-		Mu nemiorogenidies			70		
minusion.	61	drope Mendangler	nips	Improduce dell'assolution	1	1	20		
jenn		shope blood angles		randoù er ratudi de redelaguez	1		7.0		
-	61	drospe bissed propositioning	pone	improduccia del cardador	1		210		
state	81	-	pone	rando ir ne ratiola de medelaguas			7.0		
ine.		alcilariatio	wigo.	Improduce delicandador	1		710		
-	61			rando ir ar ratula dir architegas			7.0		
riendo.	81	drope bireal angular	sereletin	Improduces delicandados			7.0		
nint:		politicant	servicio	contact or relate de medicingues	-	-	711	1	
		drope bireal poperationise	rhapois	Ingradienta del candador	,	-	7.0		
	61			Improducia del andador	,	<del>                                     </del>	_	<u> </u>	
jem		shape bestel assessing		_			- 11		
less.	*	drope bireal angular	risale	conductive recovering articidad			7.0		
min.	61	drope bland paperalisate	chapute	contain a record of civiled			711		
elecular.		sintenioris	miga-	conductors recovered a charided			70		
ninis	81	atolianism	wiger	ingradenia dili sedante	1		210		
-		politicard		conductors recovered a charided	2	- 1	70		
-		drope bland poperations	serveletien	na corporise de camales de transite		-	zu		
i		drope states	sereletin	ingradenia di Cambatan			210		
jeres		100-1	chapute	ingendersia del sentante			111	1.	
jenn	81	drope bland paperalisable	pone	renderir ne recreativ articidad			7.0		
less.		alcelaniania.	***	contains records obtained	,		710		
nimin.		alcilariate	stude	confesion relation de modelagues	-	-	710		
minds.		drope Mendanglar	poss	restor to record or coloidel		-	70		
				_				_	
jeres		petitions		conductric retirals de medicinguas	1	-	710	1	
-	*	shaque hantal succession	***	Improduceia del cambados			710		
less.		sintenioris	pom	rando ir ne ratuda de mediniagum			710		
jeren	61	sintenan	serveletien	arus de eximales en la de		- 1	7.0		
jeren	81	drope Mend population	rivale	Improduceia dell'ambator	1	- 1	70		
		alodanists	elude	conductors recovered articidad		- 1	zu		
-							70		
derings	61	drope bland populations	chapute	Improduce delicandador					
	61	drospor latered properalization annullaterisation	chapuis ave	ingeralenia del candador ingeralenia del candador	-	-	7.0	,	