Variable "parroquia_urbana" PRUEBA 1

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
from sklearn.compose 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')
#Dataset
# ------------------------------------------------------------------------------
df= pd.read_csv('AT2021_NBD.csv')
df.head()
```

```
Requirement already satisfied: plotly in /usr/local/lib/python3.10/dist-packa
Requirement already satisfied: tenacity>=6.2.0 in /usr/local/lib/python3.10/c
<frozen importlib._bootstrap>:914: ImportWarning: APICoreClientInfoImportHook
<frozen importlib._bootstrap>:914: ImportWarning: _PyDriveImportHook.find_spe
<frozen importlib._bootstrap>:914: ImportWarning: _OpenCVImportHook.find_spec
<frozen importlib._bootstrap>:914: ImportWarning: _BokehImportHook.find_spec(
<frozen importlib._bootstrap>:914: ImportWarning: _AltairImportHook.find_spec
<frozen importlib._bootstrap>:914: ImportWarning: APICoreClientInfoImportHook
<frozen importlib._bootstrap>:914: ImportWarning: _PyDriveImportHook.find_spe
<frozen importlib._bootstrap>:914: ImportWarning: _OpenCVImportHook.find_spec
<frozen importlib._bootstrap>:914: ImportWarning: _BokehImportHook.find_spec(
<frozen importlib._bootstrap>:914: ImportWarning: _AltairImportHook.find_spec
<frozen importlib._bootstrap>:914: ImportWarning: APICoreClientInfoImportHook
<frozen importlib._bootstrap>:914: ImportWarning: _PyDriveImportHook.find_spe
<frozen importlib._bootstrap>:914: ImportWarning: _OpenCVImportHook.find_spec
<frozen importlib._bootstrap>:914: ImportWarning: _BokehImportHook.find_spec(
<frozen importlib._bootstrap>:914: ImportWarning: _AltairImportHook.find_spec
```

	fecha	dia	hora	latitud	longitud	tipologia	direccion	zona
0	01-01	viernes	h11	-3.991993	-79.201155	estrellamiento	orillas del zamora y jose felix de valdivieso	urbana
1	01-05	martes	h17	-4.020370	-79.217962	choque lateral perpendicular	benjamin carrion y gustavo serrano	urbana
2	01-05	martes	h12	-3.987230	-79.202984	choque por alcance	nueva loja y guaranda	urbana
3	01-08	viernes	h10	-3.989410	-79.236506	atropello	angel felicisimo rojas	urbana
4	01-09	sabado	h06	-3.979784	-79.218689	choque lateral angular	isidro ayora y habana	urbana

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: Deprecatic
  and should_run_async(code)
<frozen importlib._bootstrap>:914: ImportWarning: APICoreClientInfoImportHook
<frozen importlib._bootstrap>:914: ImportWarning: _PyDriveImportHook.find_spe
<frozen importlib._bootstrap>:914: ImportWarning: _OpenCVImportHook.find_spec
<frozen importlib._bootstrap>:914: ImportWarning: _BokehImportHook.find_spec(
<frozen importlib._bootstrap>:914: ImportWarning: _AltairImportHook.find_spec
<frozen importlib._bootstrap>:914: ImportWarning: APICoreClientInfoImportHook
<frozen importlib._bootstrap>:914: ImportWarning: _PyDriveImportHook.find_spe
<frozen importlib._bootstrap>:914: ImportWarning: _OpenCVImportHook.find_spec
<frozen importlib._bootstrap>:914: ImportWarning: _BokehImportHook.find_spec(
<frozen importlib._bootstrap>:914: ImportWarning: _AltairImportHook.find_spec
<frozen importlib._bootstrap>:914: ImportWarning:

APICoreClientInfoImportHook.find_spec() not found; falling back to find_modul

<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()

<frozen importlib._bootstrap>:914: ImportWarning:

APICoreClientInfoImportHook.find_spec() not found; falling back to find_modul

<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()
```

```
#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
```

```
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
```

```
df.isnull().sum()
```

```
dia                0
hora               0
tipologia          0
parroquia_urbana   0
causas             0
nro_heridos        0
nro_fallecidos     0
dtype: int64
```

```
df.sample(10)
```

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

`should_run_async` will not call `transform_cell` automatically in the future
```

	dia	hora	tipologia	parroquia_urbana	causas	nro_heridos	r
112	sabado	h16	roce negativo	san sebastian	imprudencia del conductor	0	
314	domingo	h11	choque por alcance	el sagrario	conducir en estado de embriaguez	1	
356	martes	h19	choque lateral perpendicular	sucre	imprudencia del conductor	3	
14	lunes	h16	roce negativo	sucre	conducir en exceso de velocidad	5	
			choque		imprudencia		

```
df.head()
```

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

`should_run_async` will not call `transform_cell` automatically in the future
```

	dia	hora	tipologia	parroquia_urbana	causas	nro_heridos	nr
0	viernes	h11	estrellamiento	el valle	conducir en estado de embriaguez	1	
1	martes	h17	choque lateral perpendicular	punzara	conducir en estado de embriaguez	1	

```
ds=pd.DataFrame(df)
#Presenta el numero de filas
print("El numero de filas(observaciones) es: ",ds.shape[0])

#Presenta el numero de columnas
print("El numero de columnas(variables) es: ",len(ds.columns))
```

```
El numero de filas(observaciones) es:  370
El numero de columnas(variables) es:  7
/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
```

```
# tipos de la variables
ds.dtypes
```

```
dia                object
hora               object
tipologia          object
parroquia_urbana   object
causas             object
nro_heridos         int64
nro_fallecidos      int64
dtype: object
```

```
df.shape[0]
```

```
370
```

```
df['causas'].value_counts()
```

```
imprudencia del conductor                161
conducir en estado de embriaguez          88
conducir en exceso de velocidad           75
imprudencia del peaton                    15
no respetar las senales de transito       15
fallas mecanicas no previsibles            6
no mantener la distancia reglamentaria     2
condiciones climaticas desfavorables       2
no ceder el derecho de via                 2
impericia del conductor                    2
cruce de animales en la via                2
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
```

```
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]


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_fallecido
0	5	11	13	3	1	1	
1	2	17	9	5	1	1	
2	2	12	10	4	10	0	
3	5	10	3	4	9	1	

```
# datos aleatorios (muestra de 10 elementos)

df.sample(10)
```

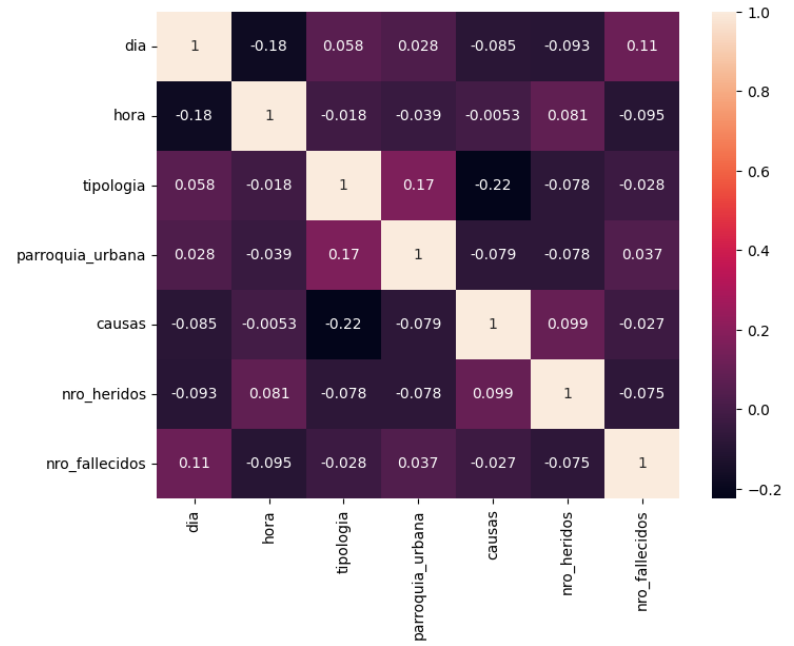
```
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: Deprecatic

`should_run_async` will not call `transform_cell` automatically in the future
```

Verificamos la correlaxion de los datos

```
60    5    15          9                1        2            0
```

```
#corelacion
corr_df = df.corr(method='pearson')

plt.figure(figsize=(8, 6))
sns.heatmap(corr_df, annot=True)
plt.show()
```

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

`should_run_async` will not call `transform_cell` automatically in the future

<frozen importlib._bootstrap>:914: ImportWarning:

APICoreClientInfoImportHook.find_spec() not found; falling back to find_modul

<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()
```




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

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

	dia	hora	tipologia	causas	nro_heridos	nro_fallecidos
0	5	11	13	1	1	0
1	2	17	9	1	1	0

```
feature_names = X.columns.tolist()

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

```
['dia', 'hora', 'tipologia', 'causas', 'nro_heridos', 'nro_fallecidos']
/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
```

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

	parroquia_urbana
0	3
1	5
2	4
3	4
4	4

Cargamos el modelo

```
from joblib import load
arbol_modelo = load('modelo_parroquia_urbana_python.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

/usr/local/lib/python3.10/dist-packages/sklearn/base.py:318: UserWarning:

Trying to unpickle estimator DecisionTreeClassifier from version 1.0.2 when using version 1.2.2. This might lead to breaking code c
https://scikit-learn.org/stable/model_persistence.html#security-maintainability-limitations
```

```
print(y_pred)
```

```
[4 1 4 4 4 4 4 4 4 1 4 4 4 4 5 4 4 1 4 1 1 4 5 4 4 1 1 1 1 4 1 1 1 4 4 4 4
 4 4 4 4 4 4 4 4 4 4 1 4 4 4 1 4 4 5 4 4 1 1 1 1 4 1 1 4 4 4 1 4 4 4 1 4 4
 4 4 4 1 4 4 1 4 4 4 1 4 4 1 4 4 4 4 4 4 4 4 1 4 1 4 4 4 4 4 4 4 4 4 1 1 4 4
 1 4 4 4 4 4 4 1 1 4 4 4 4 4 4 4 4 4 4 1 4 4 4 4 4 4 4 4 4 1 4 4 4 4 4 4 4 4 4
 4 4 4 4 1 1 4 4 4 4 4 4 4 4 1 4 4 4 4 1 1 4 4 4 4 1 4 4 4 4 4 1 4 4 4 4 4 4 4
 4 4 4 4 4 4 4 4 1 1 1 4 4 4 1 4 4 4 4 4 4 1 4 4 4 4 4 4 4 4 4 1 1 1 4 4 1 1 4
 4 1 1 4 1 4 4 1 4 4 4 4 4 4 4 4 4 4 4 1 1 4 4 1 4 4 4 4 4 1 4 4 4 4 4 4 4 1 4 4
 4 4 4 4 4 4 4 4 4 4 4 1 4 4 4 4 4 4 4 4 4 1 4 4 1 1 4 1 4 4 4 4 4 4 4 4 4 1 4
 1 4 5 4 4 4 4 4 4 4 4 4 4 4 1 4 1 4 4 4 4 4 1 1 4 1 5 1 4 4 4 4 4 4 4 4 4 4 1
 4 1 4 4 4 1 4 4 4 4 4 4 1 4 4 4 1 4 4 4 1 4 4 1 4 4 4 4 4 4 4 1 1 4 4 4 4 1 4]
/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_parroquia_urbana_2021.xlsx")
```

```
/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 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 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()
```

	precision	recall	f1-score	support
1	0.333333	0.444444	0.380952	63.000000
2	0.000000	0.000000	0.000000	55.000000
3	0.000000	0.000000	0.000000	33.000000
4	0.352313	0.860870	0.500000	115.000000
5	0.200000	0.016129	0.029851	62.000000
6	0.000000	0.000000	0.000000	42.000000
accuracy	0.345946	0.345946	0.345946	0.345946
macro avg	0.147608	0.220241	0.151801	370.000000
weighted avg	0.199773	0.345946	0.225272	370.000000

Pedicciones

```
#Predicciones
pred = pd.DataFrame(y_pred)
pred.head()
```

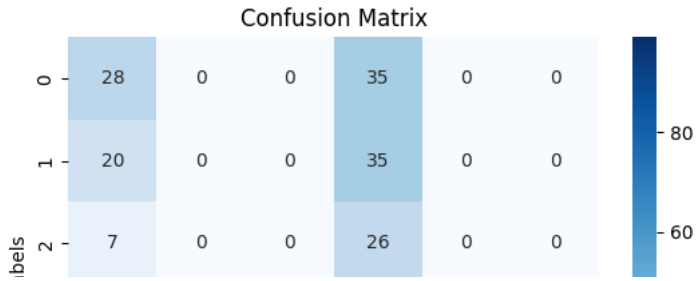
```
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: Deprecatic

`should_run_async` will not call `transform_cell` automatically in the future
```

	0
0	4
1	1
2	4
3	4
4	4

Generamos la matriz de confusión

```
#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');
```

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

`should_run_async` will not call `transform_cell` automatically in the future
```




```
#Para concatenar variables
import pandas as pd

prediccion = pd.DataFrame(y_pred,columns=['parroquia_urbana_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: Deprecatic

`should_run_async` will not call `transform_cell` automatically in the future
```

	parroquia_urbana_prediccion	parroquia_urbana
0	4	3
1	1	5
2	4	4
3	4	4
4	4	4
5	4	4
6	4	2
7	4	4
8	4	2

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:  34.5945945945946
Exhaustividad árboles de decisión:  34.5945945945946
Precision árboles de decisión:  34.5945945945946
/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
```

Transformamos las variables

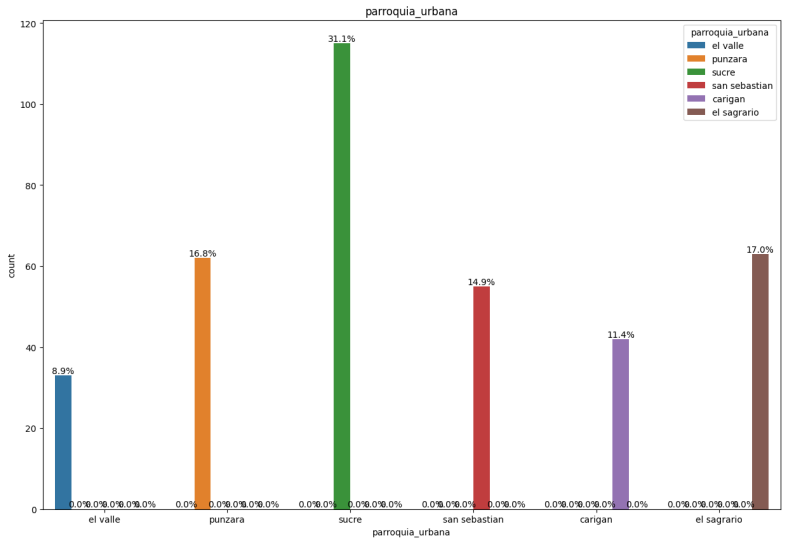
```
#transformar datos para visualización de variables predictoras y originales
df_combined['parroquia_urbana_prediccion']= df_combined['parroquia_urbana_prediccion'].apply(lambda x:
                                    'el sagrario' if x == 1 else
                                    'san sebastian' if x == 2 else
                                    'el valle' if x == 3 else
                                    'sucre' if x == 4 else
                                    'punzara' if x == 5 else
                                    'carigan')

df_combined['parroquia_urbana']= df_combined['parroquia_urbana'].apply(lambda x:
                                    'el sagrario' if x == 1 else
```

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

```
/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

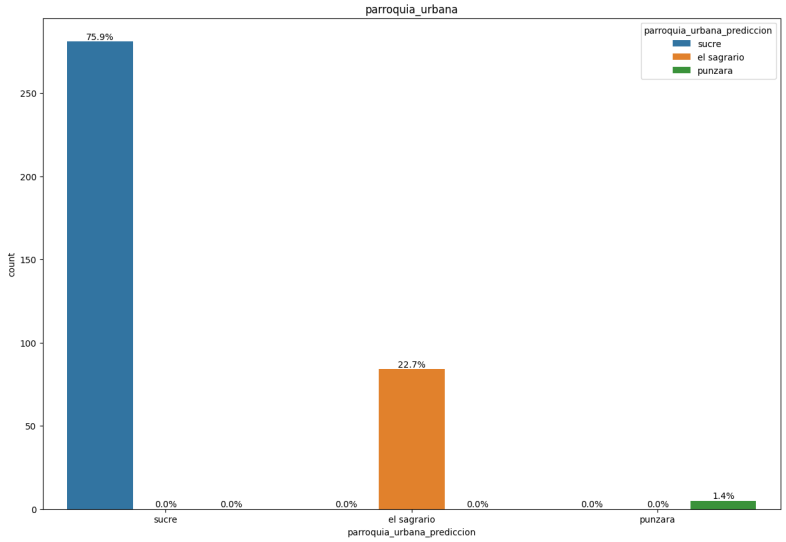
```
import seaborn as sns #permite generar fácilmente elegantes gráficos, proporciona una interfaz de alto nivel que es realmente sencilla d
ax = plt.subplots(figsize = (15,10))
ncount=len(df_combined)
sns.countplot(x='parroquia_urbana',hue = 'parroquia_urbana', data = df_combined, ax = ax[1]) #Muestre el conteo de observaciones en cada
ax[1].set_title('parroquia_urbana')



for p in ax[1].patches:
    x=p.get_bbox().get_points()[:,0]
    y=p.get_bbox().get_points()[1,1]
    ax[1].annotate('{:.1f}%'.format(100.*y/ncount), (x.mean(), y),
            ha='center', va='bottom') # set the alignment of the text

#Guardamos la figura de barras original
ax[0].savefig("barras_original_parroquia_urbana_2021.png")
```



Graficamos las estadisticas predecidas de los datos

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


for p in ax[1].patches:
    x=p.get_bbox().get_points()[:,0]
    y=p.get_bbox().get_points()[1,1]
    ax[1].annotate('{:.1f}%'.format(100.*y/ncount), (x.mean(), y),
            ha='center', va='bottom') # set the alignment of the text

#Guardamos la figura de barras de predicción
ax[0].savefig("barras_prediccion_parroquia_urbana_2021.png")
```

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

`should_run_async` will not call `transform_cell` automatically in the future
```



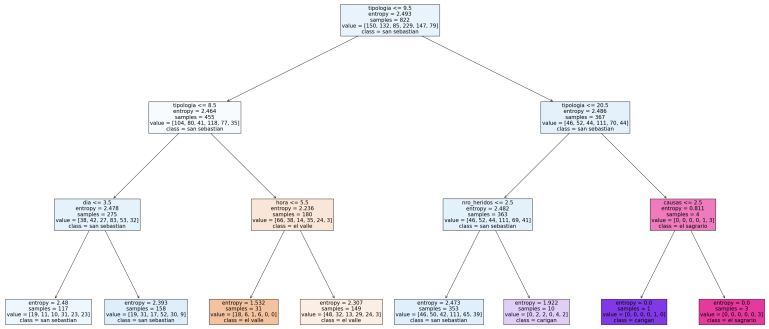
Crear el arbol de desición actual con el modelo con datos predichos del 2021

```
#target_names_str = [str(name) for name in target_names]
variable= df['parroquia_urbana'].apply(lambda x:
                                'el sagrario' if x == 1 else
                                'san sebastian' if x == 2 else
                                'el valle' if x == 3 else
                                'sucre' if x == 4 else
                                'punzara' if x == 5 else
                                'carigan')

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

```
/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
```

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()
```



```
figura_arbol.savefig("arbol_colab_parroquia_urbana_2021.png")
```

```
X.head()
```

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

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: [18.41 14.16 18.41 18.41 18.41 18.41 18.41 18.41 18.41 14.16 18.41 18.41
 18.41 18.41 11.05 18.41 18.41 14.16 18.41 14.16 14.16 18.41 11.05 18.41
 18.41 14.16 14.16 14.16 14.16 18.41 14.16 14.16 14.16 18.41 18.41 18.41
 18.41 18.41 18.41 18.41 18.41 18.41 18.41 18.41 18.41 18.41 18.41 14.16
 18.41 18.41 18.41 14.16 18.41 18.41 11.05 18.41 18.41 14.16 14.16 14.16
 14.16 18.41 14.16 14.16 18.41 18.41 18.41 14.16 18.41 18.41 18.41 14.16
 18.41 18.41 18.41 18.41 18.41 14.16 18.41 18.41 14.16 18.41 18.41 18.41
 14.16 18.41 18.41 14.16 18.41 18.41 18.41 18.41 18.41 18.41 18.41 14.16
 18.41 14.16 18.41 18.41 18.41 18.41 18.41 18.41 18.41 18.41 18.41 14.16
 14.16 18.41 18.41 14.16 18.41 18.41 18.41 18.41 18.41 14.16 14.16 18.41
 18.41 18.41 18.41 18.41 18.41 18.41 18.41 18.41 14.16 18.41 18.41 18.41
 18.41 18.41 18.41 18.41 18.41 14.16 18.41 18.41 18.41 18.41 18.41 18.41
 18.41 18.41 18.41 18.41 18.41 18.41 18.41 18.41 14.16 14.16 18.41 18.41
 18.41 18.41 18.41 18.41 18.41 14.16 18.41 18.41 18.41 14.16 14.16 18.41
 18.41 18.41 18.41 14.16 18.41 18.41 18.41 18.41 18.41 14.16 18.41 18.41
 18.41 14.16 14.16 14.16 18.41 18.41 18.41 14.16 18.41 18.41 18.41 18.41
 18.41 14.16 18.41 18.41 18.41 18.41 18.41 18.41 18.41 18.41 14.16 14.16
 14.16 18.41 18.41 14.16 14.16 18.41 18.41 14.16 14.16 18.41 14.16 18.41
 18.41 14.16 18.41 18.41 18.41 18.41 18.41 18.41 18.41 18.41 18.41 18.41
 14.16 14.16 18.41 18.41 14.16 18.41 18.41 18.41 18.41 14.16 18.41 18.41
 18.41 18.41 18.41 18.41 14.16 18.41 18.41 18.41 18.41 18.41 18.41 18.41
 18.41 18.41 18.41 18.41 18.41 14.16 18.41 18.41 18.41 18.41 18.41 18.41
 18.41 18.41 14.16 18.41 18.41 14.16 14.16 18.41 14.16 18.41 18.41 18.41
 18.41 18.41 18.41 18.41 18.41 18.41 18.41 18.41 14.16 18.41 11.05 18.41
 18.41 18.41 18.41 18.41 18.41 18.41 18.41 18.41 18.41 14.16 18.41 14.16
 18.41 18.41 18.41 18.41 14.16 14.16 18.41 14.16 11.05 14.16 18.41 18.41
 18.41 18.41 18.41 18.41 18.41 18.41 18.41 18.41 14.16 18.41 14.16 18.41
 18.41 18.41 14.16 18.41 18.41 18.41 18.41 18.41 18.41 14.16 18.41 18.41
 18.41 14.16 18.41 18.41 18.41 14.16 18.41 18.41 14.16 18.41 18.41 18.41
 18.41 18.41 18.41 14.16 14.16 18.41 18.41 18.41 14.16 18.41]%
/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
0           18.41
1           14.16
2           18.41
3           18.41
4           18.41
..            ...
365         18.41
366         18.41
367         18.41
368         14.16
369         18.41

[370 rows x 1 columns]
/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
```

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: Deprecatic

`should_run_async` will not call `transform_cell` automatically in the future
```

	dia	hora	tipologia	parroquia_urbana	causas	nro_heridos	nro_falleci
234	6	11	8	4	7	1	
73	1	21	13	6	2	1	
301	6	5	13	5	1	0	

```
probabilidades_2021_df['predicciones']= probabilidades_2021_df['predicciones'].apply(lambda x:
                                    'el sagrario' if x == 1 else
                                    '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['dia']= probabilidades_2021_df['dia'].apply(lambda x:
                                    '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
                                    '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
                                    'volcamiento longitudinal')

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')

probabilidades_2021_df.sample(10)
```

```
/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning:

ot call `transform_cell` automatically in the future. Please pass the result to `tran
```

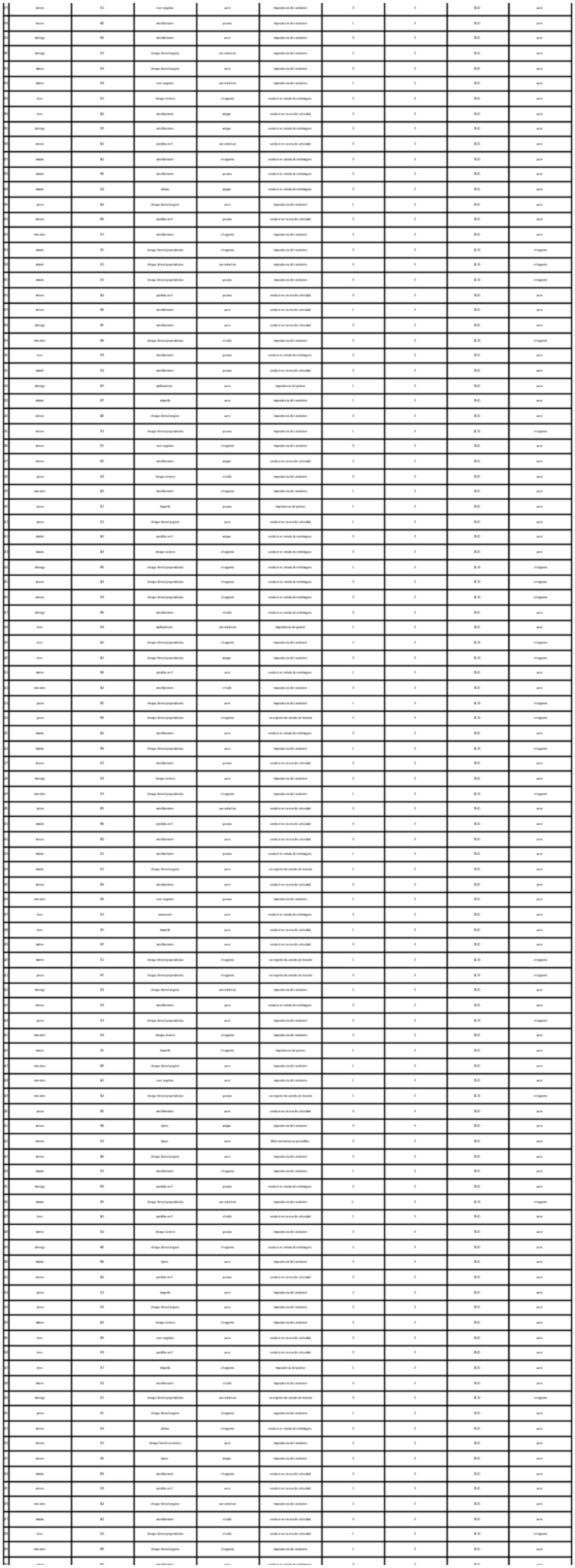
pologia	parroquia_urbana	causas	nro_heridos	nro_fallecidos	probabilidad	prec
llamiento	sucre	conducir en estado de embriaguez	0	0	18.41	
llamiento	sucre	conducir en exceso de velocidad	0	0	18.41	
llamiento	punzara	conducir en exceso de velocidad	1	0	18.41	
llamiento	punzara	conducir en exceso de velocidad	0	0	18.41	
choque lateral angular	el sagrario	imprudencia del conductor	2	0	18.41	

```
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_parroquia_urbana_2021.pdf') as pdf:
    pdf.savefig(fig, bbox_inches='tight')
```

```
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: Deprecatic
`should_run_async` will not call `transform_cell` automatically in the future
<frozen importlib._bootstrap>:914: ImportWarning:
APICoreClientInfoImportHook.find_spec() not found; falling back to find_modul
<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()
```

dc	hex	hpbigin	parroquia_reforma	motivo	no_lesiones	no_fallecidos	gravedad	predio.claseno.
jueves	01	atropellamiento	a huido	conducir en estado de embriaguez	1	0	ILES	acera
martes	07	choque lateral perpendicular	paseaba	conducir en estado de embriaguez	1	0	LE.SN	a la izquierda
martes	02	choque lateral	acera	no mantener la distancia de seguridad entre	0	0	ILES	acera
jueves	03	impacto	acera	imprudencia del peatón	1	0	ILES	acera
sábado	06	choque lateral angular	acera	conducir en estado de embriaguez	0	0	ILES	acera
domingo	03	choque lateral	acera	no mantener la distancia de seguridad entre	0	0	ILES	acera
domingo	01	atropellamiento	sam ambert san	conducir en exceso de velocidad	1	0	ILES	acera
domingo	02	atropellamiento	acera	conducir en exceso de velocidad	0	0	ILES	acera
lunes	02	atropellamiento	sam ambert san	conducir en exceso de velocidad	1	0	ILES	acera
martes	04	choque lateral perpendicular	sam ambert san	no respetar las señales de tránsito	0	0	LE.SN	a la izquierda
jueves	01	atropellamiento	acera	conducir en exceso de velocidad	1	0	ILES	acera
jueves	07	choque lateral angular	acera	condiciones climáticas desfavorables	0	0	ILES	acera
sábado	02	choque lateral angular	acera	conducir en estado de embriaguez	0	0	ILES	acera
domingo	02	choque lateral angular	antigua	no respetar las señales de tránsito	0	0	ILES	acera
lunes	01	vuelo rasgullas	acera	conducir en exceso de velocidad	1	0	ILES	paseaba
jueves	03	choque lateral angular	acera	no saber a dónde dirigirse	0	0	ILES	acera
jueves	01	atropellamiento	acera	conducir en exceso de velocidad	0	0	ILES	acera
sábado	06	choque lateral perpendicular	a la izquierda	imprudencia del conductor	0	0	LE.SN	a la izquierda
domingo	02	atropellamiento	antigua	imprudencia del conductor	1	0	ILES	acera
domingo	06	choque lateral perpendicular	paseaba	conducir en estado de embriaguez	2	0	LE.SN	a la izquierda
domingo	02	choque lateral perpendicular	a huido	no saber a dónde dirigirse	0	0	LE.SN	a la izquierda
domingo	03	choque lateral	paseaba	imprudencia del conductor	1	0	ILES	acera
miércoles	03	peatón cae sí	paseaba	conducir en estado de embriaguez	0	0	ILES	paseaba
sábado	01	atropellamiento	a huido	conducir en exceso de velocidad	0	0	ILES	acera
domingo	07	choque lateral angular	sam ambert san	imprudencia del conductor	1	0	ILES	acera
martes	06	choque lateral perpendicular	a la izquierda	conducir en exceso de velocidad	0	0	LE.SN	a la izquierda
miércoles	01	choque lateral perpendicular	a la izquierda	conducir en exceso de velocidad	1	0	LE.SN	a la izquierda
miércoles	01	choque lateral perpendicular	a la izquierda	imprudencia del conductor	1	0	LE.SN	a la izquierda
sábado	02	choque lateral perpendicular	a la izquierda	conducir en estado de embriaguez	0	0	LE.SN	a la izquierda
jueves	03	peatón pasito	acera	Mitos marcianos con posesión	2	0	ILES	acera
lunes	03	choque lateral perpendicular	sam ambert san	imprudencia del conductor	2	0	LE.SN	a la izquierda
lunes	02	choque lateral perpendicular	sam ambert san	conducir en estado de embriaguez	0	0	LE.SN	a la izquierda
martes	05	choque lateral perpendicular	sam ambert san	imprudencia del conductor	0	0	LE.SN	a la izquierda
martes	03	choque lateral angular	acera	imprudencia del conductor	0	0	ILES	acera
lunes	01	atropellamiento	paseaba	conducir en estado de embriaguez	1	0	ILES	acera
domingo	01	atropellamiento	paseaba	conducir en estado de embriaguez	2	0	ILES	acera
jueves	01	choque frontal sus mitos	antigua	imprudencia del conductor	0	1	ILES	acera
jueves	02	atropellamiento	a huido	conducir en exceso de velocidad	0	0	ILES	acera
domingo	02	atropellamiento	acera	conducir en exceso de velocidad	0	0	ILES	acera
sábado	01	atropellamiento	antigua	imprudencia del conductor	1	0	ILES	acera
jueves	06	choque lateral angular	acera	imprudencia del conductor	1	0	ILES	acera
miércoles	02	peatón cae sí	a huido	condiciones climáticas desfavorables	1	0	ILES	acera
miércoles	03	peatón pasito	acera	conducir en estado de embriaguez	0	0	ILES	acera
jueves	01	atropellamiento	a huido	imprudencia del conductor	0	0	ILES	acera
domingo	02	atropellamiento	acera	conducir en estado de embriaguez	1	0	ILES	acera
domingo	02	atropellamiento	a la izquierda	conducir en exceso de velocidad	0	0	ILES	acera
domingo	02	choque lateral angular	sam ambert san	conducir en estado de embriaguez	1	0	ILES	acera
jueves	01	choque lateral perpendicular	sam ambert san	no respetar las señales de tránsito	0	0	LE.SN	a la izquierda
sábado	07	peatón pasito	acera	conducir en exceso de velocidad	0	0	ILES	acera
domingo	01	vuelo rasgullas	paseaba	conducir en estado de embriaguez	0	0	ILES	acera
domingo	02	atropellamiento	acera	conducir en estado de embriaguez	0	0	ILES	acera
miércoles	02	choque lateral perpendicular	antigua	imprudencia del conductor	0	0	LE.SN	a la izquierda
miércoles	01	choque lateral angular	antigua	conducir en estado de embriaguez	0	0	ILES	acera
martes	01	atropellamiento	acera	imprudencia del conductor	0	0	ILES	acera
domingo	01	peatón cae sí	acera	Mitos marcianos con posesión	0	0	ILES	paseaba
lunes	07	choque lateral	antigua	imprudencia del conductor	0	0	ILES	acera
lunes	00	atropellamiento	antigua	conducir en exceso de velocidad	0	0	ILES	acera
martes	07	choque lateral perpendicular	sam ambert san	imprudencia del conductor	0	0	LE.SN	a la izquierda
domingo	03	choque lateral perpendicular	acera	no respetar las señales de tránsito	0	0	LE.SN	a la izquierda
sábado	02	choque lateral perpendicular	acera	conducir en exceso de velocidad	1	0	LE.SN	a la izquierda
jueves	01	choque lateral perpendicular	a la izquierda	imprudencia del conductor	0	0	LE.SN	a la izquierda
miércoles	02	impacto	a la izquierda	imprudencia del conductor	1	0	ILES	acera
sábado	03	choque lateral perpendicular	a la izquierda	conducir en estado de embriaguez	1	0	LE.SN	a la izquierda
sábado	02	choque lateral perpendicular	a la izquierda	imprudencia del conductor	1	0	LE.SN	a la izquierda
jueves	00	atropellamiento	acera	conducir en estado de embriaguez	1	0	ILES	acera
sábado	03	choque lateral angular	sam ambert san	conducir en exceso de velocidad	2	0	ILES	acera
jueves	01	atropellamiento	acera	conducir en exceso de velocidad	0	0	ILES	acera
lunes	05	choque lateral perpendicular	sam ambert san	imprudencia del conductor	2	0	LE.SN	a la izquierda
miércoles	00	atrition	acera	Mitos marcianos con posesión	0	0	ILES	acera
miércoles	01	choque lateral angular	antigua	imprudencia del conductor	0	0	ILES	acera
jueves	02	choque lateral angular	acera	conducir en estado de embriaguez	1	0	ILES	acera
jueves	02	choque lateral perpendicular	paseaba	imprudencia del conductor	1	0	LE.SN	a la izquierda
sábado	07	atrition	paseaba	conducir en estado de embriaguez	0	0	ILES	acera
lunes	01	atropellamiento	antigua	imprudencia del conductor	1	0	ILES	acera

	jueves	02	ninguna	ninguna	conduciendo normalmente	0	0	No	ninguna
	miercoles	03	choque lateral angular	sin anormalidad	conductor bajo el influjo de estupefacientes	0	0	No	ninguna
	sabado	05	peatonales en la calzada	sin anormalidad	conductor bajo el influjo de estupefacientes	0	0	No	ninguna
	jueves	02	choque lateral perpendicular	exceso de velocidad	imprudencia del conductor	2	0	Si	exceso de velocidad
	jueves	02	choque frontal accesorios	ninguna	imprudencia del conductor	0	0	No	ninguna
	lunes	06	choque lateral angular	exceso de velocidad	conducir en sentido contrario	0	0	No	ninguna
	martes	03	choque lateral perpendicular	exceso de velocidad	conducir en sentido contrario	0	0	Si	exceso de velocidad
	miercoles	04	atropellamiento	peatonales	conducir en sentido contrario	0	0	No	ninguna
	sabado	01	atropellamiento	peatonales	imprudencia del conductor	1	0	No	ninguna
	jueves	04	peatonales en la calzada	ninguna	conducir en sentido contrario	2	0	No	ninguna
	jueves	04	choque lateral perpendicular	sin anormalidad	no respetar las señales de tránsito	0	0	Si	exceso de velocidad
	jueves	04	choque lateral	sin anormalidad	imprudencia del conductor	0	0	No	ninguna
	jueves	04	objeto	exceso de velocidad	imprudencia del conductor	0	0	No	ninguna
	jueves	01	choque lateral perpendicular	peatonal	conducir en sentido contrario	0	0	Si	exceso de velocidad
	lunes	06	atropellamiento	peatonales	conducir en sentido contrario	2	0	No	ninguna
	miercoles	08	atropellamiento	exceso de velocidad	conducir en sentido contrario	0	0	No	ninguna
	miercoles	01	choque lateral angular	peatonal	conducir en sentido contrario	1	0	No	ninguna
	jueves	08	peatonales en la calzada	ninguna	conducir en sentido contrario	2	0	No	ninguna
	domingo	06	choque frontal accesorios	peatonales	imprudencia del conductor	0	0	No	ninguna
	lunes	04	atropellamiento	peatonal	conducir en sentido contrario	0	0	No	ninguna
	jueves	01	atropellamiento	sin anormalidad	exceso de velocidad	0	0	No	ninguna
	jueves	03	choque lateral perpendicular	exceso de velocidad	imprudencia del conductor	1	0	Si	exceso de velocidad
	jueves	02	atropellamiento	exceso de velocidad	conducir en sentido contrario	0	0	No	ninguna
	domingo	01	choque lateral perpendicular	exceso de velocidad	imprudencia del conductor	0	0	Si	exceso de velocidad
	jueves	04	atropellamiento	ninguna	imprudencia del conductor	0	0	No	ninguna
	miercoles	03	choque lateral angular	exceso de velocidad	imprudencia del conductor	2	0	No	ninguna
	martes	01	peatonales en la calzada	sin anormalidad	Mala maniobra con peatonales	1	0	No	ninguna
	martes	04	choque lateral angular	exceso de velocidad	conducir en sentido contrario	1	0	No	ninguna
	jueves	01	atropellamiento	ninguna	conducir en sentido contrario	0	0	No	ninguna
	jueves	06	atropellamiento	peatonal	conducir en sentido contrario	0	0	No	ninguna
	sabado	03	choque lateral angular	exceso de velocidad	imprudencia del conductor	1	0	No	ninguna
	sabado	03	objeto	peatonal	imprudencia del conductor	0	0	No	ninguna
	sabado	01	choque lateral	ninguna	imprudencia del conductor	1	0	No	ninguna
	lunes	03	choque lateral perpendicular	exceso de velocidad	imprudencia del conductor	0	0	Si	exceso de velocidad
	lunes	07	choque lateral perpendicular	ninguna	conducir en sentido contrario	0	0	Si	exceso de velocidad
	martes	03	choque lateral	ninguna	imprudencia del conductor	1	0	No	ninguna
	martes	04	atropellamiento	ninguna	conducir en sentido contrario	0	0	No	ninguna
	domingo	01	choque lateral perpendicular	peatonal	imprudencia del conductor	1	0	Si	exceso de velocidad
	sabado	05	vuelco ocupantes	sin anormalidad	imprudencia del conductor	0	0	No	ninguna
	jueves	02	antiflamencia	ninguna	conducir en sentido contrario	1	0	No	ninguna
	jueves	03	atropellamiento	peatonal	imprudencia del conductor	1	0	No	ninguna
	jueves	03	choque	peatonal	conducir en sentido contrario	0	0	No	ninguna
	jueves	01	ninguna	ninguna	conducir en sentido contrario	0	0	No	ninguna
	jueves	03	choque lateral perpendicular	sin anormalidad	no respetar las señales de tránsito	0	0	Si	exceso de velocidad
	jueves	03	choque lateral perpendicular	ninguna	imprudencia del conductor	0	0	Si	exceso de velocidad
	sabado	03	choque lateral angular	ninguna	conducir en sentido contrario	0	0	No	ninguna
	domingo	06	atropellamiento	ninguna	conducir en sentido contrario	0	0	No	ninguna
	martes	04	ninguna	ninguna	imprudencia del peatón	1	0	No	ninguna
	martes	03	atropellamiento	ninguna	imprudencia del conductor	0	0	No	ninguna
	lunes	05	peatonales en la calzada	ninguna	conducir en sentido contrario	1	0	No	ninguna
	lunes	07	atropellamiento	ninguna	conducir en sentido contrario	0	0	No	ninguna
	martes	06	atropellamiento	ninguna	conducir en sentido contrario	0	0	No	ninguna
	martes	02	choque lateral angular	exceso de velocidad	imprudencia del conductor	2	0	No	ninguna
	miercoles	01	choque lateral angular	exceso de velocidad	imprudencia del conductor	1	0	No	ninguna
	jueves	01	choque lateral perpendicular	exceso de velocidad	imprudencia del conductor	1	0	Si	exceso de velocidad
	jueves	01	ninguna	ninguna	imprudencia del peatón	1	0	No	ninguna
	jueves	06	atropellamiento	ninguna	imprudencia del conductor	0	0	No	ninguna
	jueves	04	choque	ninguna	conducir en sentido contrario	0	0	No	ninguna
	jueves	03	peatonales en la calzada	exceso de velocidad	conducir en sentido contrario	0	0	No	ninguna
	sabado	03	choque	peatonal	conducir en sentido contrario	0	0	No	ninguna
	jueves	04	atropellamiento	exceso de velocidad	conducir en sentido contrario	0	0	No	ninguna
	sabado	01	atropellamiento	peatonal	conducir en sentido contrario	1	0	No	ninguna
	domingo	04	choque	ninguna	conducir en sentido contrario	1	0	No	ninguna
	lunes	01	choque lateral perpendicular	sin anormalidad	no respetar las señales de tránsito	0	0	Si	exceso de velocidad
	miercoles	01	choque lateral angular	sin anormalidad	imprudencia del conductor	0	0	No	ninguna
	miercoles	07	vuelco ocupantes	exceso de velocidad	conducir en sentido contrario	1	0	No	ninguna
	sabado	03	atropellamiento	peatonal	imprudencia del conductor	0	0	No	ninguna
	miercoles	05	ninguna	peatonal	imprudencia del peatón	2	0	No	ninguna
	lunes	03	atropellamiento	ninguna	conducir en sentido contrario	0	0	No	ninguna
	jueves	04	choque	ninguna	conducir en sentido contrario	0	0	No	ninguna
	sabado	01	ninguna	ninguna	imprudencia del conductor	1	0	No	ninguna
	sabado	02	atropellamiento	ninguna	conducir en sentido contrario	0	0	No	ninguna
	domingo	00	atropellamiento	peatonal	conducir en sentido contrario	0	1	No	ninguna
	domingo	03	atropellamiento	sin anormalidad	imprudencia del conductor	1	0	No	ninguna
	lunes	03	ninguna	exceso de velocidad	imprudencia del peatón	1	0	No	ninguna
	sabado	01	choque frontal accesorios	sin anormalidad	imprudencia del conductor	1	0	No	ninguna
	jueves	01	vuelco ocupantes	sin anormalidad	imprudencia del conductor	0	0	No	ninguna
	lunes	04	objeto	peatonal	imprudencia del conductor	0	0	No	ninguna
	martes	03	choque lateral perpendicular	exceso de velocidad	imprudencia del conductor	0	0	Si	exceso de velocidad
	martes	02	choque lateral perpendicular	sin anormalidad	imprudencia del conductor	2	0	Si	exceso de velocidad
	jueves	07	choque lateral angular	exceso de velocidad	conducir en sentido contrario	1	0	No	ninguna
	jueves	03	objeto	ninguna	imprudencia del conductor	0	0	No	ninguna
	jueves	08	atropellamiento	sin anormalidad	imprudencia del conductor	1	0	No	ninguna
	jueves	03	ninguna	exceso de velocidad	imprudencia del peatón	1	0	No	ninguna
	domingo	05	atropellamiento	peatonal	conducir en sentido contrario	0	0	No	ninguna
	domingo	02	atropellamiento	peatonal	conducir en sentido contrario	1	0	No	ninguna
	domingo	03	choque lateral angular	peatonal	conducir en sentido contrario	0	0	No	ninguna
	miercoles	09	choque lateral perpendicular	sin anormalidad	imprudencia del conductor	0	0	Si	exceso de velocidad
	miercoles	07	choque lateral angular	exceso de velocidad	conducir en sentido contrario	0	0	No	ninguna
	martes	04	choque lateral angular	peatonal	imprudencia del conductor	0	0	No	ninguna
	jueves	03	vuelco ocupantes	ninguna	imprudencia del conductor	0	0	No	ninguna
	jueves	02	choque lateral perpendicular	ninguna	conducir en sentido contrario	0	0	Si	exceso de velocidad
	jueves	01	choque lateral perpendicular	exceso de velocidad	imprudencia del conductor	2	0	Si	exceso de velocidad
	sabado	01	ninguna	sin anormalidad	imprudencia del peatón	1	0	No	ninguna
	jueves	04	atropellamiento	peatonal	conducir en sentido contrario	0	0	No	ninguna
	domingo	03	atropellamiento	peatonal	imprudencia del conductor	1	0	No	ninguna
	domingo	08	choque frontal accesorios	peatonal	conducir en sentido contrario	1	0	No	ninguna
	domingo	02	choque lateral perpendicular	sin anormalidad	imprudencia del conductor	0	0	Si	exceso de velocidad
	domingo	01	atropellamiento	exceso de velocidad	conducir en sentido contrario	2	0	No	ninguna
	miercoles	01	ninguna	ninguna	imprudencia del peatón	1	0	No	ninguna
	jueves	03	choque lateral angular	ninguna	imprudencia del conductor	0	0	No	ninguna
	jueves	01	ninguna	ninguna	exceso de velocidad	1	0	No	ninguna
	jueves	01	choque lateral perpendicular	peatonal	imprudencia del conductor	1	0	Si	exceso de velocidad



Guardamos el modelo del año 2021

```
from joblib import dump
dump(arbol_modelo, 'modelo_parroquia_urbana2021_python.joblib')
```

```
/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

['modelo_parroquia_urbana2021_python.joblib']
```

Creamos un nuevo dato para verificar la eficiencia del modelo y la probabilidad del dato

```
#Nuevos datos
import pandas as pd
import numpy as np

x_test1 = pd.DataFrame(columns=('dia', 'hora', 'tipologia', 'parroquia_urbana', 'causas', 'nro_heridos', 'nro_fallecidos'))
print(x_test1)
x_test1.loc[0] = (1, 9, 9, 2, 10, 10, 10)
print(x_test1)
y_pred1 = arbol_modelo.predict(x_test1.drop(['parroquia_urbana'], axis=1))
print("Predicción: " + str(y_pred1))

y_proba1 = arbol_modelo.predict_proba(x_test1.drop(['parroquia_urbana'], axis=1))
probabilidad_acierto1 = np.round(y_proba1[0][y_pred1] * 100, 2)
print("Probabilidad de Acierto: " + str(probabilidad_acierto1) + "%")
```

```
Empty DataFrame
Columns: [dia, hora, tipologia, parroquia_urbana, causas, nro_heridos, nro_fallecidos]
Index: []
   dia  hora  tipologia  parroquia_urbana  causas  nro_heridos  nro_fallecidos
0    1     9          9                 2      10           10              10
Predicción: [1]
Probabilidad de Acierto: [21.48]%
```

✓ 0 s se ejecutó 09:20