# ▼ Variable "Tipologia"

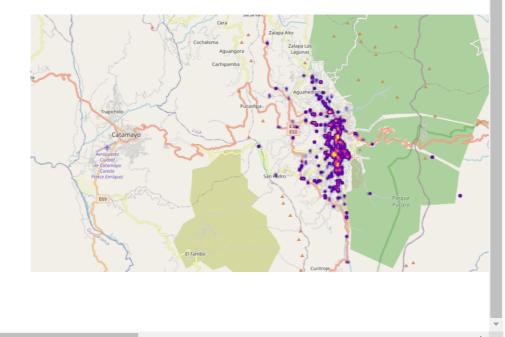
Cargamos las librerias necesarias para la elaboración y desarrollo de la minería de datos

```
!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')
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)
     <frozen importlib._bootstrap>:914: ImportWarning: APICoreClientInfoImportHook.find_spec() not found; fa
     <frozen importlib._bootstrap>:914: ImportWarning: _PyDriveImportHook.find_spec() not found; falling bac
     <frozen importlib._bootstrap>:914: ImportWarning: _OpenCVImportHook.find_spec() not found; falling back
     <frozen importlib._bootstrap>:914: ImportWarning: _BokehImportHook.find_spec() not found; falling back
     <frozen importlib._bootstrap>:914: ImportWarning:
                                                        _AltairImportHook.find_spec() not found; falling back
     <frozen importlib._bootstrap>:914: ImportWarning: APICoreClientInfoImportHook.find_spec() not found; f
     <frozen importlib._bootstrap>:914: ImportWarning: _PyDriveImportHook.find_spec() not found; falling bac
     <frozen importlib._bootstrap>:914: ImportWarning: _OpenCVImportHook.find_spec() not found; falling back
     <frozen importlib._bootstrap>:914: ImportWarning: _BokehImportHook.find_spec() not found; falling back
     <frozen importlib._bootstrap>:914: ImportWarning:
                                                        _AltairImportHook.find_spec() not found; falling back
     <frozen importlib._bootstrap>:914: ImportWarning: APICoreClientInfoImportHook.find_spec() not found; fa
     <frozen importlib._bootstrap>:914: ImportWarning:
                                                        _PyDriveImportHook.find_spec() not found; falling bad
     <frozen importlib._bootstrap>:914: ImportWarning: _OpenCVImportHook.find_spec() not found; falling back
     <frozen importlib._bootstrap>:914: ImportWarning:
                                                        _BokehImportHook.find_spec() not found; falling back
     <frozen importlib._bootstrap>:914: ImportWarning: _AltairImportHook.find_spec() not found; falling back
         fecha
                                         longitud
                                                      tipologia direccion
                                                                              zona parroquia urbana
                   dia hora
                               latitud
                                                                   orillas del
                                                                   zamora y
                                                                                                        conduci
                         h11 -3.991993 -79.201155 estrellamiento
         01-01 viernes
                                                                   iose felix
                                                                            urbana
                                                                                               el valle
                                                                                                         estado
                                                                        de
                                                                                                        embriac
                                                                  valdivieso
                                                                   benjamin
                                                         choque
                                                                                                        conduci
                                                                   carrion v
         01-05
                martes
                             -4.020370 -79.217962
                                                          lateral
                                                                            urbana
                                                                                              punzara
                                                                                                          estado
                         h17
                                                                    gustavo
                                                    perpendicular
                                                                                                        embriac
                                                                    serrano
                                                                  nueva loja
                                                                                                       no mante
                                                      choque por
         01-05
                martes
                             -3.987230 -79.202984
                                                                                                        la dista
                                                                                                sucre
                                                         alcance
                                                                  quaranda
                                                                                                       reglament
                                                                      angel
                                                                                                        imprude
        01-08
                             -3.989410 -79.236506
                                                        atropello
                viernes
                         h10
                                                                   felicisimo
                                                                            urbana
                                                                                                         del pea
                                                                      rojas
                                                         choque
                                                                      isidro
                                                                                                        conduci
         01-09
               sahado
                        h06 -3.979784 -79.218689
                                                          lateral
                                                                    ayora y
                                                                            urbana
                                                                                                sucre
                                                                                                          estado
                                                         angular
                                                                    habana
                                                                                                        embriag
            ıl.
```

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()
```

```
<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()
<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()
```



```
#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
dia
                           a
hora
                          0
latitud
                          1
longitud
                           1
tipologia
                           0
direccion
                           0
zona
                           0
parroquia_urbana
                           0
                           0
causas
gravedad
                           0
nro_heridos
                           0
nro_fallecidos
                           0
vehiculos_retenidos
                          0
senalizacion_existente
                          a
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

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: 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
299	miercoles	h04	estrellamiento	punzara	conducir en exceso de velocidad	0	0
212	sabado	h21	perdida de carril	carigan	conducir en estado de embriaguez	0	0
132	viernes	h18	perdida de carril	sucre	conducir en exceso de velocidad	0	0
293	sabado	h06	estrellamiento	punzara	imprudencia del conductor	1	0
121	martes	h14	atropello	sucre	imprudencia del peaton	1	0
11	viernes	h17	choque lateral angular	sucre	condiciones climaticas desfavorables	0	0
202	domingo	h07	arrollamiento	sucre	imprudencia del	1	0

df.head()

```
/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 1
```

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

```
0
                                                                                                           0
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
     no respetar las senales de transito
                                                 15
     fallas mecanicas no previsibles
                                                 6
     no mantener la distancia reglamentaria
                                                 2
     condiciones climaticas desfavorables
                                                 2
                                                 2
     no ceder el derecho de via
     impericia del conductor
                                                 2
     cruce de animales en la via
                                                  2
     Name: causas, dtype: int64
print(df['causas'].unique())# datos en texto
     /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
     ['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'
```

```
print(df['causas'].unique())# datos en texto
```

```
['conducir en estado de embriaguez'
```

'fallas mecanicas no previsibles' 'cruce de animales en la via']

<sup>&#</sup>x27;no mantener la distancia reglamentaria' 'imprudencia del peaton'

```
'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 if x == 'carigan' else
                                    7)
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
                                    5 if x == 'choque frontal excentrico' else
                                    5 if x == 'choque frontal longitudinal' else
                                    5 if x == 'choque lateral angular' else
                                    5 if x == 'choque lateral perpendicular' else
                                    5 if x == 'choque por alcance' else
                                    6 if x == 'colision' else
                                    7 if x == 'encunetamiento' else
                                    8 if x == 'estrellamiento' else
                                    9 if x == 'perdida de carril' else
                                    9 if x == 'perdida de pista' else
                                    10 if x == 'roce negativo' else
                                    10 if x == 'roce positivo' else
                                    10 if x == 'rozamiento' else
                                    11 if x == 'volcamiento' else
                                    11 if x == 'volcamiento lateral' else
                                    11)
print(df['tipologia'].unique()) # datos convertidos a numeros enteros
     [853109621114]
```

'conducir en exceso de velocidad' 'no respetar las senales de transito'

```
df['dia']= df['dia'].apply(lambda x:
                                    1 if x == 'lunes' else
                                    1 if x == 'martes' else
                                    1 if x == 'miercoles' else
                                    2 if x == 'jueves' else
                                    2 if x == 'viernes' else
                                    3 if x == 'sabado' else
                                    3)
print(df['dia'].unique()) # datos convertidos a numeros enteros
     [2 1 3]
df= df[df['parroquia_urbana'] <= 6]</pre>
df['hora']= df['hora'].apply(lambda x:
                                    0 if x == 'h00' else
                                    0 if x == 'h01' else
                                    0 if x == 'h02' else
                                    0 if x == 'h03' else
                                    0 if x == 'h04' else
                                    0 if x == 'h05' else
                                   1 if x == 'h06' else
                                   1 if x == 'h07' else
                                    1 if x == 'h08' else
                                    1 if x == 'h09' else
                                    1 if x == 'h10' else
                                    1 if x == 'h11' else
                                    2 if x == 'h12' else
                                    2 if x == 'h13' else
                                    2 if x == 'h14' else
                                    2 if x == 'h15' else
                                    2 if x == 'h16' else
                                    2 if x == 'h17' else
                                    3 if x == 'h18' else
                                    3 if x == 'h19' else
                                    3 if x == 'h20' else
                                    3 if x == 'h21' else
                                    3 if x == 'h22' else
                                    3)
```

df.head()

	dia	hora	tipologia	parroquia_urbana	causas	nro_heridos	nro_fallecidos	1	ıl.
0	2	1	8	3	1	1	0		
1	1	2	5	5	1	1	0		
2	1	2	5	4	10	0	0		
3	2	1	3	4	9	1	0		
4	3	1	5	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
Verificacion de correlacion de variables
#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: DeprecationWarning:
     `should_run_async` will not call `transform_cell` automatically in the future. Please pass the result 1
     <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()
```



Seleccionamos la columna que vamos a predecir los datos

<frozen importlib.\_bootstrap>:914: ImportWarning:

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

```
dia hora parroquia_urbana causas nro_heridos nro_fallecidos
           2
                2
      1
                                   5
                                           1
                                                        1
                                                                        0
      2
                 2
                                          10
                                                        0
                                                                        0
      3
          2
                                   4
                                           9
                                                                        0
feature_names = X.columns.tolist()
# Imprimir los nombres de las características
print(feature_names)
     ['dia', 'hora', 'parroquia_urbana', '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
```

Revisamos la columna a predecir seleccionada

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



Cargamos de modelo 2018-2020

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

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

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

←

Verificamos los datos a predecir

```
print(y_pred)
```

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

Configuramos la metrica de clasificación

```
# 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_tipologia_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/python 3.10/dist-packages/sklearn/metrics/\_classification.py: 1344: \ Undefined Metric Warning: 1.00 and 1.00 and 1.00 are also as a superior of the contract of the contract
         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
                                      0.000000 0.000000 0.000000
                                                                                                  3.000000
         2
                                      0.000000 0.000000 0.000000
                                                                                                10.000000
         3
                                      0.866667
                                                        0.650000
                                                                           0.742857
                                                                                                20.000000
         4
                                      0.000000
                                                        0.000000
                                                                           0.000000
                                                                                                 1.000000
         5
                                      0.552980
                                                        0.976608
                                                                           0.706131
                                                                                              171.000000
                                      0.000000 0.000000 0.000000
                                                                                               10.000000
         8
                                      0.660377
                                                        0.307018
                                                                           0.419162 114.000000
                                      0.000000 0.000000 0.000000
                                                                                               26.000000
                                      0.000000
                                                        0.000000 0.000000
                                                                                                14.000000
         10
                                      0.000000
                                                        0.000000 0.000000
                                                                                                 1.000000
         11
                                                        0.581081
                                                                           0.581081
                                                                                                  0.581081
         accuracy
                                      0.581081
         macro avg
                                      0.208002
                                                        0.193363 0.186815
                                                                                              370,000000
         weighted avg
                                     0.505881 0.581081 0.495649 370.000000
```

#### Predicciones de los datos obtenidas

```
pred = pd.DataFrame(y_pred)
pred.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



- 1 5
- 2 5
- **3** 3
- 4 5

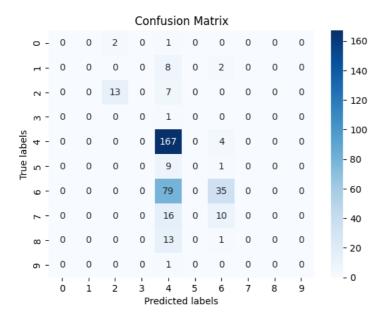
### Generamos la matriz de confusión

```
#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: DeprecationWarning:

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



#### Concatenamos los datos de las horas originales con los datos predichos

```
#Para concatenar variables
import pandas as pd

prediccion = pd.DataFrame(y_pred,columns=['tipologia_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

	tipologia_prediccion	tipologia	1	il
0	5	8		
1	5	5		
2	5	5		
3	3	3		
4	5	5		
5	5	5		
6	5	8		
7	8	8		
8	5	8		
9	5	5		

Precision árboles de decisión: 58.108108108108105

# 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: 58.108108108108105
Exhaustividad árboles de decisión: 58.108108108108108105
```

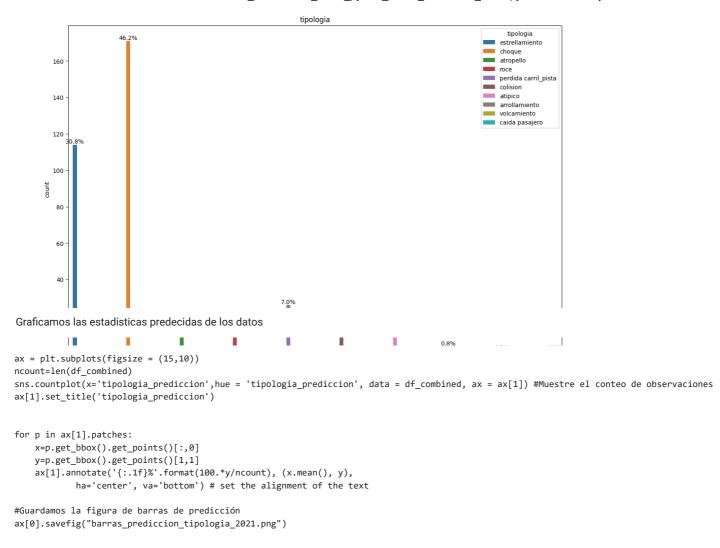
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning:

`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['tipologia'] = df_combined['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' if x == 5 else
                                     'colision' if x == 6 else
                                    'encunetamiento' if x == 7 else
                                     'estrellamiento' if x == 8 else
                                     'perdida carril_pista' if x == 9 else
                                     'roce' if x == 10 else
                                     'volcamiento')
df_combined['tipologia_prediccion'] = df_combined['tipologia_prediccion'].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' if x == 5 else
                                     'colision' if x == 6 else
                                    'encunetamiento' if x == 7 else
                                     'estrellamiento' if x == 8 else
                                     'perdida carril_pista' if x == 9 else
                                     'roce' if x == 10 else
                                     'volcamiento')
     /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



```
/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

```
tipologia_prediccion

-- ---
tipologia prediccion
```

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

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

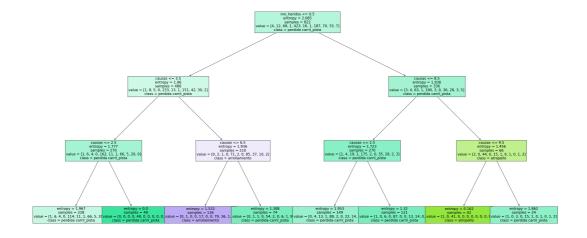
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning:

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

tipologia prediccion

# 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_tipologia_2021.png")
```

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)
print(y_proba.shape)
print(y_pred.shape)
probabilidad_acierto = np.round(y_proba[0][y_pred] * 100, 2)
print("Probabilidad de Acierto: " + str(probabilidad_acierto) + "%")
```

```
(370, 11)
   (370,)
   1.34 1.34 9.4 1.34 1.34 1.34 1.34 1.34 9.4 1.34 9.4 1.34 1.34
   1.34 1.34 1.34 9.4 1.34 1.34 9.4 1.34 1.34 1.34 1.34 1.34 1.34 1.34
   1.34 1.34 1.34 0.67 1.34 1.34 9.4 1.34 9.4 1.34 1.34 1.34 1.34 1.34
   1.34 0.67 1.34 1.34 1.34 9.4 9.4 1.34 0.67 1.34 1.34 1.34 1.34 1.34
   1.34 1.34 9.4 1.34 9.4 1.34 1.34 1.34 1.34 9.4 1.34 1.34 1.34 1.34
   9.4 1.34 9.4 1.34 1.34 9.4 0.67 1.34 1.34 1.34 9.4 1.34 1.34
   0.67 1.34 1.34 1.34 1.34 1.34 1.34 1.34 0.67 1.34 1.34 1.34 1.34 1.34
   1.34 1.34 1.34 9.4 1.34 1.34 9.4 9.4 9.4 1.34 1.34 9.4 1.34 1.34
   1.34 9.4 1.34 1.34 1.34 1.34 1.34 0.67 1.34 1.34 1.34 9.4 1.34
   9.4 1.34 1.34 1.34 1.34 1.34 1.34 1.34 9.4 1.34 1.34 9.4
   9.4 0.67 1.34 1.34 1.34 1.34 1.34 9.4 1.34 9.4 1.34 9.4 1.34 1.34
   1.34 1.34 1.34 9.4 1.34 9.4 1.34 1.34 9.4 1.34 1.34 1.34 1.34 1.34
   1.34 1.34 1.34 1.34 1.34 9.4 1.34 1.34 1.34 1.34 1.34 1.34 9.4
   1.34 1.34 1.34 1.34 1.34 1.34 1.34 0.67 1.34 1.34 1.34 1.34 1.34 1.34
   1.34 1.34 1.34 1.34 1.34 1.34]%
print(type(probabilidad_acierto))
columna_probabilidades = pd.DataFrame()
columna_probabilidades['probabilidad'] = pd.DataFrame(probabilidad_acierto)
print(columna_probabilidades)
   <class 'numpy.ndarray'>
      probabilidad
           1.34
   1
           1.34
   2
           1.34
   3
           0.67
   4
           1.34
   365
           1.34
   366
           1.34
   367
           1.34
   368
           1.34
   369
           1.34
   [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
  4
```

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 1
probabilidades_2021_df['predicciones']= probabilidades_2021_df['predicciones'].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' if x == 5 else
                                                                  'colision' if x == 6 else
                                                                  'encunetamiento' if x == 7 else
                                                                  'estrellamiento' if x == 8 else
                                                                   'perdida carril_pista' if x == 9 else
                                                                  'roce' if x == 10 else
                                                                  'volcamiento')
probabilidades_2021_df['dia'] = probabilidades_2021_df['dia'].apply(lambda x:
                                                                  'lunes_marte_miercoles' if x == 1 else
                                                                   'jueves_viernes' if x == 2 else
                                                                   'sabado_domingo')
probabilidades\_2021\_df['hora'] = probabilidades\_2021\_df['hora'].apply(lambda \ x: lambda) = lambda \ x: lambda \
                                                                  'h00-h05' if x == 0 else
                                                                  h06-h011' if x == 1 else
                                                                  'h012-h17' if x == 2 else
                                                                  'h18-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' if x == 5 else
                                                                  'colision' if x == 6 else
                                                                  'encunetamiento' if x == 7 else
                                                                  'estrellamiento' if x == 8 else
                                                                  'perdida carril pista' if x == 9 else
                                                                  'roce' if x == 10 else
                                                                  'volcamiento')
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)
```

/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_fallecido
356	lunes_marte_miercoles	h18- h23	choque	sucre	imprudencia del conductor	3	
97	sabado_domingo	h06- h011	choque	el sagrario	imprudencia del conductor	0	
274	sabado_domingo	h00- h05	estrellamiento	el sagrario	conducir en exceso de velocidad	0	
115	jueves_viernes	h06- h011	colision	punzara	conducir en estado de embriaguez	0	
90	lunes_marte_miercoles	h012- h17	choque	punzara	conducir en exceso de velocidad	1	
329	sabado_domingo	h00- h05	estrellamiento	el sagrario	imprudencia del conductor	0	

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\_tipologia\_2021.pdf') as pdf:
 pdf.savefig(fig, bbox\_inches='tight')

jerryless									
state, teriop	BLASS	atrifactoris atrifactoris	prose	sendorir ne salada de mediniquas Improducia dell'ambatan			IM IM	hape hape	
statication	BLOI	hope	poss	renderê se reladê de steletiques	,		114	hop	
jestjolene	80AH	map pages	servision	ingradenia del ambata			IM	h-p	
lon, nato, nimbr	BIGI	hope	pone	ingradiente del cardador			114	hap.	
jercjasa	BLIG.	anderen	***	under a ratali de matriagas		-	114	hap.	
$\overline{}$				_				_	
lans, made primedra	80A01	atrifactori	poss	contains remarks abouted	7		164	hope sinfaninis	
loss, marin prinsules									
state, terings	80.61	sinfanisis	****	contain se ratab de redelagues	,	,	ш	hap-	
staticalizings	BEACI	sinfanisis	poss	conductric exceptable de medicingua e	1	1	IM .	hops	
lon, maio primules	BLOT	alian .	*10*	Improductio dell'accidente	2	,	ш	hap-	
los, mate, minutes	BEACI	atrifactoris	chapute	conductive retails de medicingues	'	,	tM	hop	
jennysiene.	BLICE	peddored pin	pone	conductric reliabilità de medicingua e	L	'	114	hops	
lon,mate,minutes	#UAI1	atolianism		Improduction delicated action	'	,	tM	1	
less, marin, primeriles	BLICE	ati-faminis	pone	conductors ratiols de mobiliques	,	,	IM .	1	
slate/derings	81.61	atolianism	pone	contact or recovering orbidal			1	atorilaminos	
loss, mario, primeriro	86.60	atorilamino	serveletien	conductors extends deconductory	1		1M	hope	
jerrysiene	#UALT	hope	elvale	Improductio dell'assolution	1		114	1	
jerosylenos	86.101	ato Gardaria		Improductio del conductor			114	1	
jentylens	86.001	Impe	pone	Improduces delicandados		1	134	hope	
jennystenn.	BLIG	Impe	sereletin	contact or retail describingues			184	Impe	
state, terrings	96.H/C1	Impe		Improduces dell'assolution		1.	1M	hope	
state, terings	06.6001	hope	chapuis	conductive ratiols developing	1		1M	į	
lans, maio princoles	8UA21	repris	sentetie	improduce in this position	1	1	167	tops to	
state; famings	86.6001	Impe	riugusis	Improduce deli sentante	1	ı	114	hope	
less, mate, primedra	BUALI	hope	sereletin	conductors rated describility as			114	hops	
state, terrings	BLICE	atrolamino	sereletin	conductric retails de contribuyes	,	,	IM .	1	
lon,mate,minutes	8UA11	hope	rhapsis	Improduce del conductor	1	,	114	1	
lana, marke, primerales	enci	atrifactoris	miga.	Improduce delicandaries	,	,	134	1	
lan, mate, primedro	86.603	haper 1	sereletin	na respector las carneles de insculie	1.		114	1	
jercjima	MARI	ato Gardania	miga.	Improduction delicated action	ı		184	hap.	
state, terings	BLIG	at clarate	seredetien	conferio no ratindo de medicinguas			114	hope	
state, terings	80.00	and and a		Improduce del conductor			114	-	
jeretylene.	Blick	atrifactoris	chapute	Ingradienia del candador		,	134	hops	
state, terrings	BLIG	at-decises	pone	Improduces delicated and			114	-	
state, terings	BLACI	atrifactoris	sentelie	Ingestion in this contains		,	134	hops	
state, terings	#DAt1	atrifactoris	pone	Improduce delicandados		,	134	1-1-	
staticalism	BLASI	at decision	chapute	ingendenia del candador	-	-	134	hap.	
staticalism	BLOT	atriferiaris		Ingendenia del candada	-	1	IM	hope	
staticaring	MENTE	hope	pone	ingraderale del conductor	-	-	114	hap.	
$\overline{}$	BLME1		risk	ingradenia del ambato			IM IM		
lans, marks princedes	BLO	loge striketer					114	hape .	
lons, marks princedes			ariga.	conductric ratiosh de centrálegues				hope	
lon, maio primules	BLMEI	hope	pon	sa requeito de canados de tensales	1	,	114		
jerecjana.	#UAC1	hope	servicia	Improduce dell'ambation	'	'	IM .	hops	
state,teriop	Blich	atolianism	pone	conductric reliabilità de medicingua e	,	,	134	hap.	
state,turings	BLICE	atorilarizato	10.00	conductors extends de medicinguas			134	hope	
state,teriop	8UAI1	hope		Improduction delicated and an annual series	,	1	134	hope	
jentylene	BLAS	atorilarizato	elude	Improduction dell'association			134	hops	
jerecylene	86.101	Impe	chapus	Improduction dell'assolution	1		IM	hops	
lon, mate, pienoles	80321	atrifactoris		Improductio del conductor	1		1M	hope	
slate/serings	#UA21	hope		Improduces delicandador	1	,	1M	hop	
slate/derings	Blich	Inpe	***	Improduce del conductor	L.	1	114	hope	
lone, marke, pricessiles	BEACI	Inpe	sentelin	Improduce delicandados		,	1M	hop	
loss, maria prientales	96.000	Impe	riveporie	Improduce deli sendante	L	1	IM	hope	
loss, marke, pricessiles	BLICE	hope	servicin	ingradiente del cardador	1	1	1M	hope	
lone, marke, minmales	6UA21	repris	***	Improduces delicandador	1	,	114	hope	
state, teriop	86,600	politicalphi	elsale	contact or committee of cital	1	1	и	atolianinto	
loss, marks principles	86.MES	Impr	shapata	ingradiente del candante	2		1M	hops	
loss, maria, micrositro	BLICE	atolianism	pone	conductors recovered articided	1	,	и	atolianists	
jersystem	BENET			Improduce del conductor		,	134	1	
jercjina	80A0	mpris	chapunk	Improduce del pentire	ı		H1	topo la	
ion, mate, prientales	BUALT	hape		Improduce del conductor	,	,	134	hape .	
state, terings	BLOI	ation .	elude	Improduce del cardada	- 1		IM	hope	
jercjina	MARI	hope		Ingradenia del candador		,	114	hop	
lon, main, minutes	BLOI	hope		ingradenia del sentato			IM	hope	
lon, mate, minutes	Blick	hope	stude	conductron ratioals de medicingua e	ı	,	114	hop	
lon, nato primate	WUALT		elude	ingradienia del cardador	-		114	hope	
	#DAI1	spin-	sentelia	Ingradenia del candada	-	-	134	hap	
inn, mate missoles	Blick		rhapais	ingradienia del carolador	-	-	114	h	
lana, marke, priorecules			rhapais	under a retail de métriques			114		
less, marke prioreciles	88,600							_	
loo, mala primules statis, tanings	BEAUT BEAUT	lone .		conductive reliability and relations.			184	low.	
lon, nate, piendra slake, larings slake, larings	60A21	Inpe		conductor estado de embriagano			TAI	hape hape	
loo, mala primules statis, tanings		Impe Impe Impe	are are sentition	sando ir no salaulo de mediciaguas Improduccia dell'amelantes Improduccia dell'amelantes	1		EM EM	hape hape	