

```
In [51]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from matplotlib import cm
```

Pregunta:

¿Que tanto influyeron factores de la población como, educación, violencia y participación en las elecciones del senado del año 2019 considerando únicamente al partido Centro Democrático?

Partido con mayor votación e índice de abstención por municipio

Datos obtenidos de Misión de observación electoral (MOE)

<https://www.datoselectorales.org/datos-y-resultados-electorales/>

```
In [128... Elecciones = pd.read_excel('MOE-ResultSen2018.Partido-municipio_Datos.xlsx', header=1)
Elecciones = Elecciones.drop('Code', 1)

Abstención_Municipal = Elecciones[['Municipio', '%Abstención', 'Ganador Circunscripción N
Abstención_Municipal = Abstención_Municipal.rename(columns={'Ganador Circunscripción Na
print('Porcentaje de abstención y Ganadores de Elecciones al senado por municipio')
print('Número de datos: ' + str(len(Abstención_Municipal)))
Abstención_Municipal.head()
```

Porcentaje de abstención y Ganadores de Elecciones al senado por municipio
Número de datos: 1124

```
Out[128... Municipio %Abstención Ganador
```

	Municipio	%Abstención	Ganador
1	Abejorral	0.716650	Centro Democrático
780	Abrego	0.422288	Cambio Radical
2	Abriaquí	0.480441	Partido Conservador
687	Acacías	0.394339	Centro Democrático
575	Acandí	0.561571	Partido de la U

Tasa de homicidios por cada 100 mil habitantes en cada municipio

Datos obtenidos de base de datos de la página de Medicina Legal

[https://www.medicinalegal.gov.co/documents/20143/388157/1-](https://www.medicinalegal.gov.co/documents/20143/388157/1-Homicidios.+Colombia%2C+2018.xlsx/722560a4-2e77-b155-cc08-2ca513dfc7c0)

[Homicidios.+Colombia%2C+2018.xlsx/722560a4-2e77-b155-cc08-2ca513dfc7c0](https://www.medicinalegal.gov.co/documents/20143/388157/1-Homicidios.+Colombia%2C+2018.xlsx/722560a4-2e77-b155-cc08-2ca513dfc7c0)

```
In [127... Homicidios = pd.read_excel('Homicidios-Municipios.xlsx', header=1)#, index_col=1)
Homicidios = Homicidios.iloc[:-9,[0,1,-6]]
```

```
Homicidios = Homicidios.rename(columns={'CÓDIGOS': 'Codigo',
                                         'SEXO, NÚMERO DE CASOS Y TASAS POR 100.000 HABI
                                         'Tasas Totales x 100.000 Hab.': 'Homicidios'})

Homicidios_Municipal = Homicidios[Homicidios['Codigo']>1000]
Homicidios_Municipal = Homicidios_Municipal.sort_values(by=['Municipio'])[['Municipio',
print('Datos de violencia por municipio')
print('Número de datos: ' + str(len(Homicidios_Municipal)))
Homicidios_Municipal.head()
```

Datos de violencia por municipio
Número de datos: 765

Out[127]...

	Municipio	Homicidios
4	Abejorral	21.062609
536	Acacías	16.295492
338	Acandí	64.274237
466	Acevedo	25.635913
146	Achí	4.122181

Puntaje promedio por municipio obtenido en las pruebas Saber11 en el año 2019

Datos obtenidos de base de datos del gobierno

<https://www.datos.gov.co/Educaci-n/PUNTAJE-ICFES-POR-DEPARTAMENTOS/x9vi-iv8c>

In [129]...

```
Saber11 = pd.read_csv('Saber_11__2019-2.csv')
Saber11 = Saber11[['ESTU_MCPIO_RESIDE', 'PUNT_GLOBAL']]
Saber11 = Saber11.rename(columns={'ESTU_MCPIO_RESIDE': 'Municipio', 'PUNT_GLOBAL': 'Punt

Saber11_Municipal = Saber11.groupby(by=["Municipio"]).mean()
Saber11_Municipal = Saber11_Municipal.reset_index().sort_values(by=['Municipio'])
def capital(palabra):
    return palabra.capitalize()
Saber11_Municipal['Municipio'] = Saber11_Municipal['Municipio'].apply(capital)
print('Puntaje promedio pruebas Saber11 años 2019 por municipio')
print('Número de datos: ' + str(len(Saber11_Municipal)))
Saber11_Municipal.head()
```

C:\Users\Jonathan Diaz\anaconda3\lib\site-packages\IPython\core\interactiveshell.py:316
5: DtypeWarning: Columns (76,78) have mixed types.Specify dtype option on import or set
low_memory=False.

has_raised = await self.run_ast_nodes(code_ast.body, cell_name,
Puntaje promedio pruebas Saber11 años 2019 por municipio
Número de datos: 1031

Out[129]...

	Municipio	Puntaje
0	Abejorral	234.222798
1	Abriaquí	219.088235
2	Acacías	255.321721
3	Acandí	196.428571

	Municipio	Puntaje
4	Acevedo	239.872093

Entrenamiento del modelo usando Regresión Logística

In [105...

```
Datos = Abstención_Municipal.join(Homicidios_Municipal.set_index('Municipio'), on='Muni
Datos = Datos.join(Saber11_Municipal.set_index('Municipio'), on='Municipio')
Datos = Datos.dropna()
Datos = Datos.set_index('Municipio')
Datos['Ganador'] = Datos['Ganador'].apply(lambda x: 1 if x == 'Centro Democrático' else
print('Número de datos: ' + str(len(Datos)))
Datos.head()
```

Número de datos: 632

Out[105...

	%Abstención	Ganador	Homicidios	Puntaje
Municipio				
Abejorral	0.716650	1	21.062609	234.222798
Acacías	0.394339	1	16.295492	255.321721
Acandí	0.561571	0	64.274237	196.428571
Acevedo	0.509628	0	25.635913	239.872093
Achí	0.361792	0	4.122181	207.709163

In [130...

```
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split

X = Datos[['%Abstención', 'Homicidios', 'Puntaje']]
y = Datos['Ganador']

X_train, X_test, y_train, y_test = train_test_split(X, y, random_state = 0)
modelo = LogisticRegression().fit(X_train, y_train)
print('Número de datos de entrenamiento: ' + str(len(X_train)))
print('Número de datos de testeo: ' + str(len(X_test)))
```

Número de datos de entrenamiento: 474

Número de datos de testeo: 158

In [142...

```
print('Clases: ' + str(modelo.classes_[0]) + ' y ' + str(modelo.classes_[1]))
print('Coeficientes: ' + str(round(modelo.coef_[0,0], 3)) + ', ' + str(round(modelo.coef_[0,1], 3)) + ', ' + str(round(modelo.coef_[0,2], 3)))
print('Intercepto: ' + str(round(modelo.intercept_[0], 3)))
```

Clases: 0 y 1

Coeficientes: 1.205, -0.003, 0.044

Intercepto: -12.388

Resultados

```
In [158... import sklearn.metrics
from sklearn.metrics import confusion_matrix, precision_score, recall_score, f1_score

y_predict = modelo.predict(X_test)

print('Si gana el Centro Democrático')
print('Precision: ', round(precision_score(y_test, y_predict, pos_label=1),3))
print('Recall: ', round(recall_score(y_test, y_predict, pos_label=1),3))
print('F1: ', round(f1_score(y_test, y_predict, pos_label=1),3))
print('\n')

print('No gana el Centro Democrático')
print('Precision: ', round(precision_score(y_test, y_predict, pos_label=0),3))
print('Recall: ', round(recall_score(y_test, y_predict, pos_label=0),3))
print('F1: ', round(f1_score(y_test, y_predict, pos_label=0),3))
```

Si gana el Centro Democrático
Precision: 0.25
Recall: 0.033
F1: 0.059

No gana el Centro Democrático
Precision: 0.812
Recall: 0.977
F1: 0.887