import pandas as pd

from sklearn import metrics

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LogisticRegression

import matplotlib.pyplot as plt

import seaborn as sns

%matplotlib inline

diabetes = pd.read\_csv('diabetes.csv')

diabetes.head()

|  | **Pregnancies** | **Glucose** | **BloodPressure** | **SkinThickness** | **Insulin** | **BMI** | **DiabetesPedigreeFunction** | **Age** | **Outcome** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | 6 | 148 | 72 | 35 | 0 | 33.6 | 0.627 | 50 | 1 |
| **1** | 1 | 85 | 66 | 29 | 0 | 26.6 | 0.351 | 31 | 0 |
| **2** | 8 | 183 | 64 | 0 | 0 | 23.3 | 0.672 | 32 | 1 |
| **3** | 1 | 89 | 66 | 23 | 94 | 28.1 | 0.167 | 21 | 0 |
| **4** | 0 | 137 | 40 | 35 | 168 | 43.1 | 2.288 | 33 | 1 |

#Agrupar variables, resultado en Y: outcome, no tomo en cuenta DiabetesPedigree

cols = ['Pregnancies', 'Insulin', 'BMI', 'Age','Glucose','BloodPressure']

x = diabetes[cols]

y = diabetes.Outcome

print(x)

Pregnancies Insulin BMI Age Glucose BloodPressure

0 6 0 33.6 50 148 72

1 1 0 26.6 31 85 66

2 8 0 23.3 32 183 64

3 1 94 28.1 21 89 66

4 0 168 43.1 33 137 40

logreg = LogisticRegression()

logreg.fit(X\_train, Y\_train)

y\_pred = logreg.predict(X\_test)

print(y\_pred)

[1 0 0 1 0 0 1 1 0 0 0 1 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0

0 0 1 0 0 0 1 1 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 1 1 0 1 1 1 1 0 0 0 0 0 0 1

1 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0

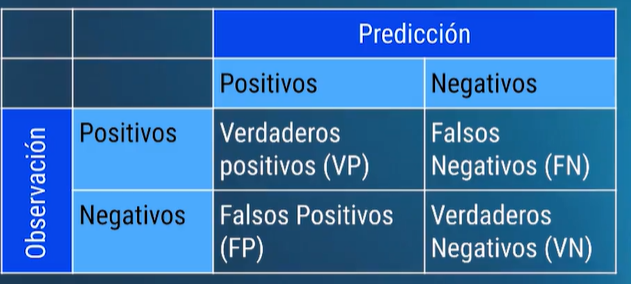
0 1 0 1 1 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 1 0 0 0 0 0 0

0 0 0 1 0 0 1 0 1 0 1 1 1 1 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 1 0

0 1 0 0 0 0 0]

(segun sus caracteristicas que estan en X (con todas las variables) se dice si tiene (1) o no diabetes (0)

MATRIZ DE CONFUSION



import pandas as pd

from sklearn import metrics

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LogisticRegression

import matplotlib.pyplot as plt

import seaborn as sns

%matplotlib inline

diabetes = pd.read\_csv('diabetes.csv')

diabetes.head()

#Agrupar variables, resultado en Y: outcome, no tomo en cuenta DiabetesPedigree

cols = ['Pregnancies', 'Insulin', 'BMI', 'Age','Glucose','BloodPressure']

x = diabetes[cols]

y = diabetes.Outcome

X\_train, X\_test, Y\_train, Y\_test = train\_test\_split(x,y, test\_size = 0.25, random\_state=0)

logreg = LogisticRegression()

logreg.fit(X\_train, Y\_train)

Y\_pred = logreg.predict(X\_test)

cnf\_matrix= metrics.confusion\_matrix(Y\_test, Y\_pred)

print("Exactitud ",metrics.accuracy\_score(Y\_test, Y\_pred))

import numpy as np

class\_names = [0,1]

fig, ax = plt.subplots()

tick\_marks=np.arange(len(class\_names))

plt.xticks(tick\_marks, class\_names)

plt.xticks(tick\_marks, class\_names)

sns.heatmap(pd.DataFrame(cnf\_matrix), annot=True, cmap='Blues\_r', fmt='g')

ax.xaxis.set\_label\_position('top')

plt.tight\_layout()

plt.title('Matriz de Confusion', y=1.1)

plt.ylabel('Actual')

plt.ylabel('Prediccion')

OUTPUT:

Exactitud 0.796875

|  |  |  |
| --- | --- | --- |
|  | 118 (0,0) No tiene diabetes, test dio (-)  12 (0,1) No tiene diabetes, test dio (+) ERROR  27 (1,1) Si tiene diabetes, test dio (-) ERROR  35 (1,0) Si tiene diabetes, test dio (+) |  |

Table

Description automatically generated