- **Asignatura:**
  Aprendizaje Inteligente

- **Catedrático:**
  FRANCISCO JAVIER LUNA ROSAS

- **Tema:**
  Regresión Lineal

- **Equipo:**

Michelle Stephanie Adame

Andrés Eloy Escobedo Esparza

- **Carrera:**
  I.C.I.

- **Semestre:**
  6º

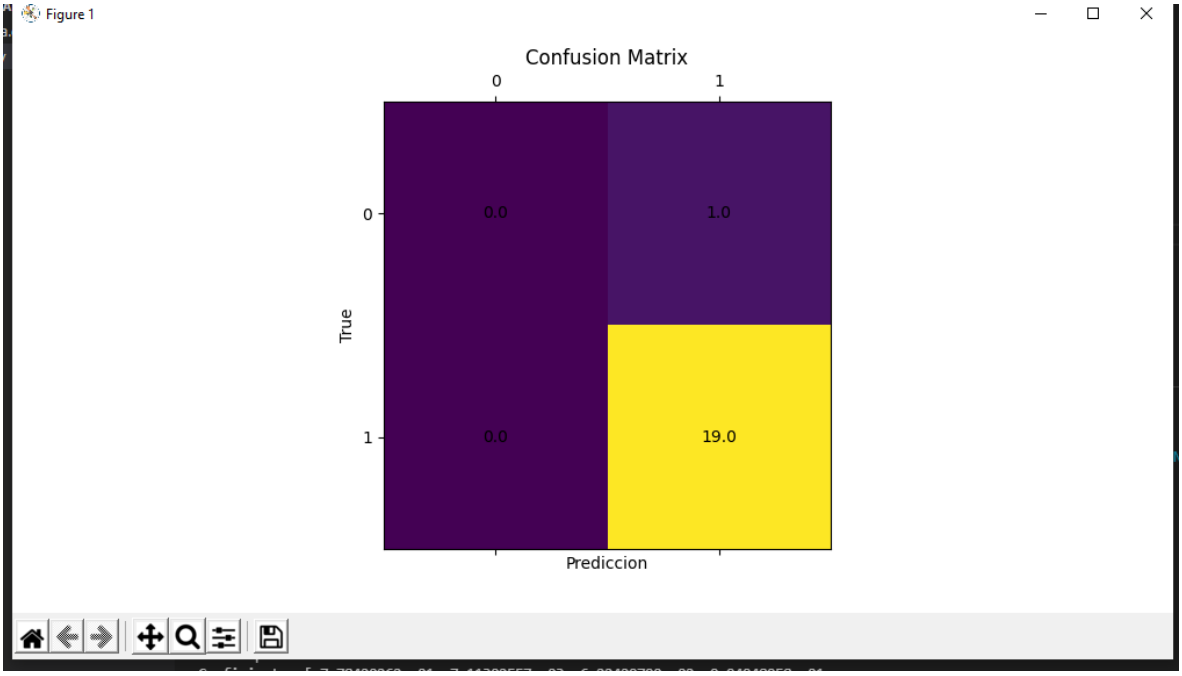
Evidencias de la práctica:

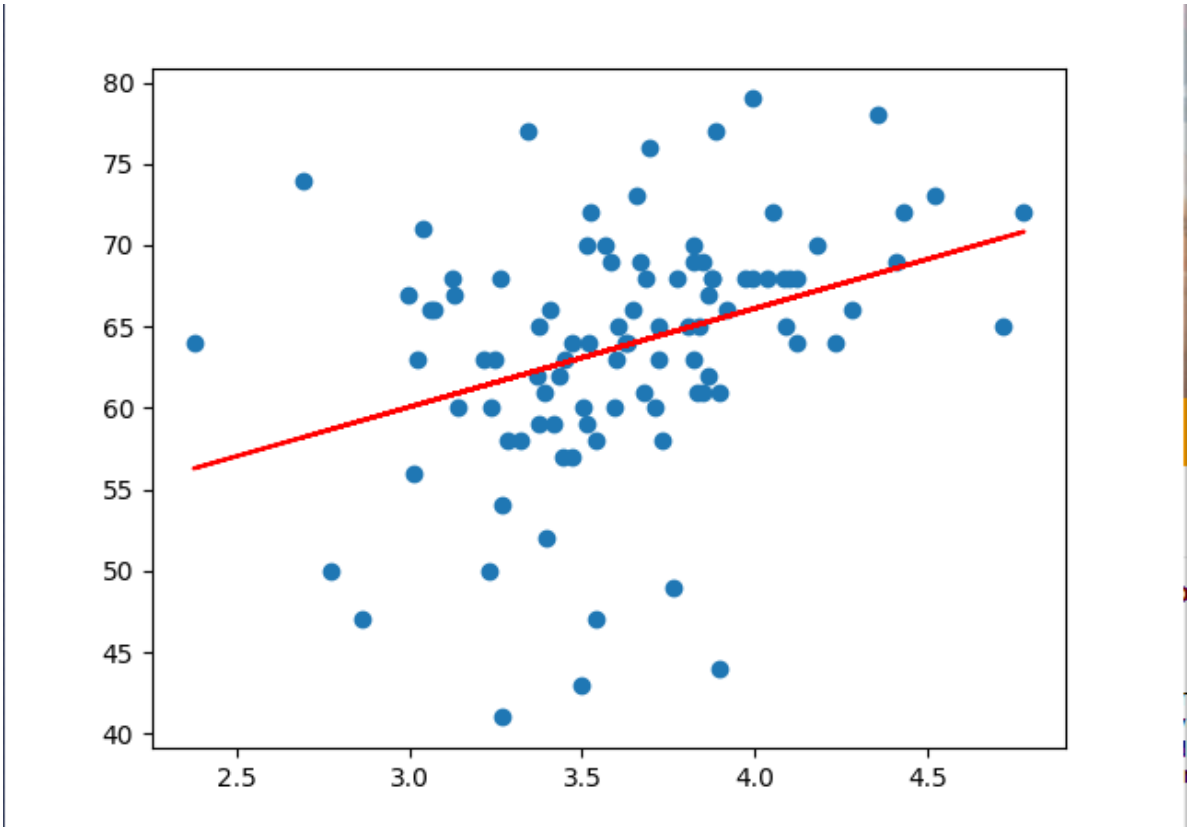
```
import numpy as np
import pandas as pd
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error
from sklearn.metrics import confusion_matrix, classification_report
import matplotlib.pyplot as plt

#Dado a que el csv esta separado por ; los encabezados y los decimales por ,
 es por eso que se pone esta linea para el csv
data = pd.read_csv("prostate_data.csv", sep=";",decimal=',', index_col=0)
#Separamos lpsa de los demas datos
X = np.array(data.drop(['lpsa'], 1))
y = np.array(data['lpsa'])
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
print(data.head())
print('Numero de datos para entrenar: {} ; Numero de datos para pruebas: {}'
.format(X_train.shape[0], X_test.shape[0]))
#Algoritmo de regresion lineal
algoritmo = LinearRegression()
algoritmo.fit(X_train, y_train)
Y_pred = algoritmo.predict(X_test)
print('Precisión Regresión Lineal: {}'.format(algoritmo.score(X_train, y_tra
in)))
# Modelo de regresion lineal
print("Intercept:", algoritmo.intercept_)
print("Coeficiente:", algoritmo.coef_)
print("R^2:", algoritmo.score(X, y)) #Coeficiente de determinación
rmse = mean_squared_error(
        y_true  =  y_test,
        y_pred  =  Y_pred,
        squared = False
       )
print("")
print(f"RMSE: {rmse}") #error del test
#Se realiza la matriz de confusion
cutoff = 0.7
y_pred_classes = np.zeros_like(Y_pred)
y_pred_classes[Y_pred > cutoff] = 1
y_test_classes = np.zeros_like(y_test)
y_test_classes[y_test > cutoff] = 1
cm = confusion_matrix(y_test_classes, y_pred_classes)
print(cm)
```

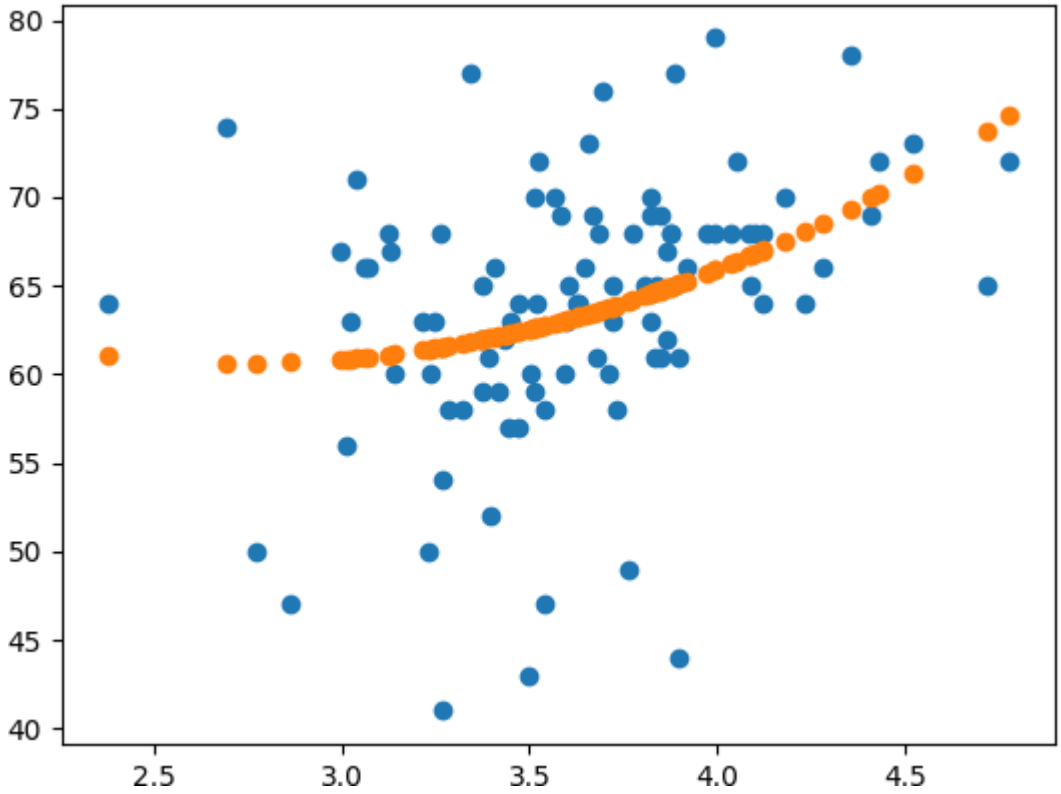
```
fig, ax = plt.subplots(figsize=(10,5))
ax.matshow(cm)
plt.title("Confusion Matrix")
plt.ylabel("True")
plt.xlabel("Prediccion")
for (i,j), z in np.ndenumerate(cm):
    ax.text(j, i, '{:0.1f}'.format(z), ha='center', va='center')
plt.show()


X = data.iloc[:, 0].values.reshape(-
1, 1) # values converts it into a numpy array
Y = data.iloc[:, 1].values.reshape(-1, 1) # -
1 means that calculate the dimension
linear_regressor = LinearRegression()
linear_regressor.fit(X, Y)
Y_pred = linear_regressor.predict(X)
plt.scatter(X, Y)
plt.plot(X, Y_pred, color='red')
plt.show()
from sklearn.preprocessing import PolynomialFeatures
pr = PolynomialFeatures(degree = 2)
X_poly = pr.fit_transform(X)
pr.fit(X_poly, Y)
lin_reg = LinearRegression()
lin_reg.fit(X_poly, Y)
plt.scatter(X, Y)
plt.scatter(X, lin_reg.predict(pr.fit_transform(X)))
plt.show()
```

Capturas de pantalla:

```
PS C:\Users\ANDY\Desktop\Regresion lineal> & C:/Users/ANDY/AppData/Local/Program
          lweight  age      lbph  svi       lcp  gleason  pgg45      lpsa
lcavol
-0.579818  2.769459   50 -1.386294    0 -1.386294        6      0 -0.430783
-0.994252  3.319626   58 -1.386294    0 -1.386294        6      0 -0.162519
-0.510826  2.691243   74 -1.386294    0 -1.386294        7     20 -0.162519
-1.203973  3.282789   58 -1.386294    0 -1.386294        6      0 -0.162519
 0.751416  3.432373   62 -1.386294    0 -1.386294        6      0  0.371564
Numero de datos para entrenar: 77 ; Numero de datos para pruebas: 20
Precisión Regresión Lineal: 0.45607124881511696
Intercept: -1.4841509126982895
Coeficiente: [ 7.78420262e-01 -7.11302557e-03  6.22429790e-02  9.94948958e-01
  1.38600426e-01  2.03519099e-01  1.54739994e-05]
R^2: 0.5017397872820388

RMSE: 0.6275619170155121
[[ 0  1]
 [ 0 19]]
```




80
75
70
65
60
55
50
45
40
2.5
3.0
3.5
4.0
4.5


80
75
70
65
60
55
50
45
40
2.5
3.0
3.5
4.0
4.5

Conclusión:

Las técnicas de regresión y correlación cuantifican la asociación estadística entre dos o más variables. La regresión lineal simple expresa la relación entre una variable dependiente Y y una variable independiente X, en términos de la pendiente y la intersección de la línea que mejor se ajuste a las variables.

La correlación simple expresa el grado o la cercanía de la relación entre las dos variables en términos de un coeficiente de correlación que proporciona una medida indirecta de la variabilidad de los puntos alrededor de la mejor línea de ajuste- Ni la regresión ni la correlación dan pruebas de relaciones causa – efecto.

Bibliografías:

Heras, J. M. (2020, 2 octubre). *Regresión Lineal: teoría y ejemplos en Python*.

IArtificial.net. https://www.iartificial.net/regresion-lineal-con-ejemplos-en-python/

Regresión lineal con python. (2018). Regresion Lineal.

https://www.cienciadedatos.net/documentos/py10-regresion-lineal-python.html