**Lab Test 2**

**Open Source Software Lab**

Patil Amit Gurusidhappa

19104004

B11

Q1.a

import pandas as pd

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

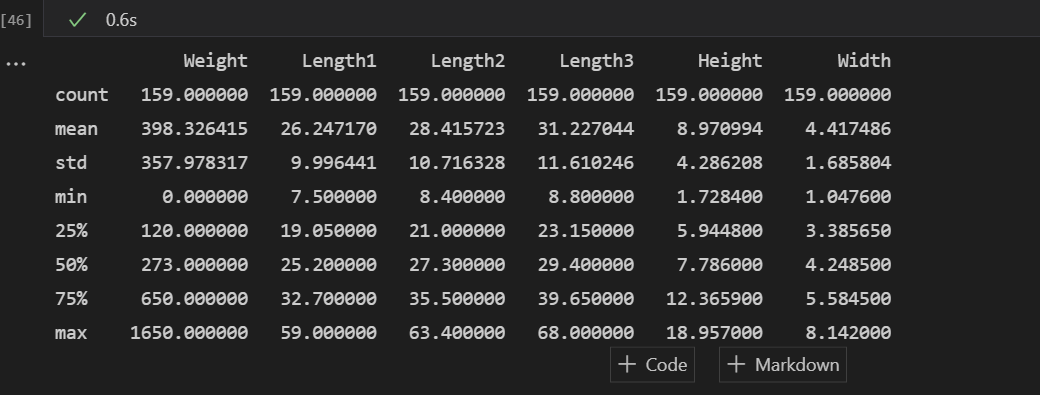
from sklearn import metrics

from sklearn.linear\_model import LogisticRegression

from sklearn.linear\_model import LinearRegression

fish = pd.read\_csv("Fish.csv")

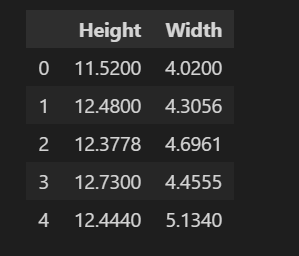
print(fish.describe())



b)

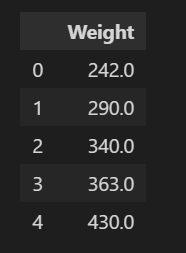
X = fish[['Height', 'Width']]  # Select columns using column name

X.head()



y = fish[['Weight']]

y.head(5)



X\_train, X\_test, y\_train, y\_test = train\_test\_split(

    X, y, test\_size=0.2, random\_state=42)

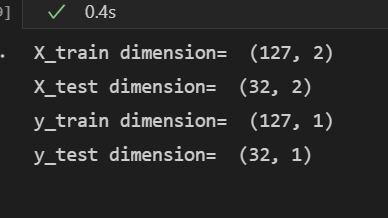
# Use paramter 'random\_state=1' if you want keep results same everytime you execute above code

print('X\_train dimension= ', X\_train.shape)

print('X\_test dimension= ', X\_test.shape)

print('y\_train dimension= ', y\_train.shape)

print('y\_test dimension= ', y\_test.shape)



Linear Regression

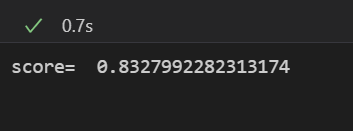
model = LinearRegression(

    copy\_X=True, fit\_intercept=True, n\_jobs=None).fit(X, y)

model.fit(X\_train, y\_train)

print('score= ', model.score(X\_test, y\_test))

Score of linear regression



For logistic Regression

model = LogisticRegression(

    copy\_X=True, fit\_intercept=True, n\_jobs=None).fit(X, y)

model.fit(X\_train, y\_train)

print('score= ', model.score(X\_test, y\_test))

Graph

df = pd.read\_csv("fish.csv")

df

df.columns = ['species', 'weight', 'vertical\_length',

              'diagnol\_length', 'cross\_length', 'height', 'width']

x = np.array(df['vertical\_length'])

y = np.array(df['weight'])

n = len(x)

sum\_x = np.sum(x)

sum\_y = np.sum(y)

sum\_xx = np.sum(x\*x)

sum\_xy = np.sum(x\*y)

mean\_x = np.mean(x)

mean\_y = np.mean(y)

numerator = sum\_xy - ((sum\_x\*sum\_y)/n)

denominator = sum\_xx - ((sum\_x\*sum\_x)/n)

m = numerator/denominator

b = mean\_y - m\*mean\_x

plt.scatter(x, y)

plt.plot(x, m\*x+b, color='red')

plt.title(f'Linear Regression Line , m = {m:.2f}, b = {b:.2f}')

plt.ylabel("Fish Weight")

plt.xlabel("Vertical Length")

plt.figure(figsize=(10, 5))

plt.show()