REPORT

Q1 : Fitting a line on the height - weight dataset using Linear regression.

Steps:

Q1- A,B(1,2):

1. Import the relevant libraries of python
    a. Matplotlib for plotting graphs
    b. Pandas for data set manipulation
    c. Numpy for vector algebra
2. Save the height column as X , and weight column as y which is the target.
3. Using *plt.scatter* plot visualize X vs y. We obtain a fairly linear relationship, hence we use Linear regression to train the model.
4. There is no need to do feature scaling in this example because we have only one feature.
5. Import *LinearRegression()* from sklearn, fit it, print the coefficient and slope of the line, plot the obtained line and the original data points.

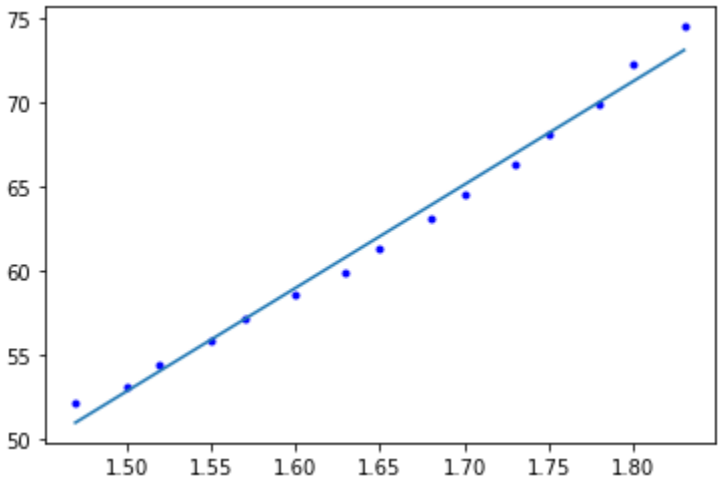


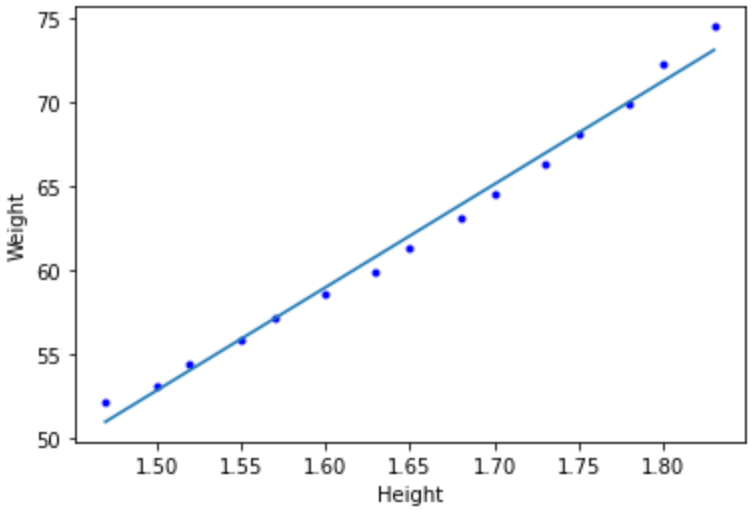
FIGURE 1

Q1 - B(3,4):

1. Add a column of 1's to 0th index of the feature matrix (X vector) to handle the bias parameter.
2. Using the normal equation solution, we can find the parameters theta_0 = coefficient and theta_1 = slope

$$\hat{\theta} = (X^T \cdot X)^{-1} \cdot X^T \cdot y$$

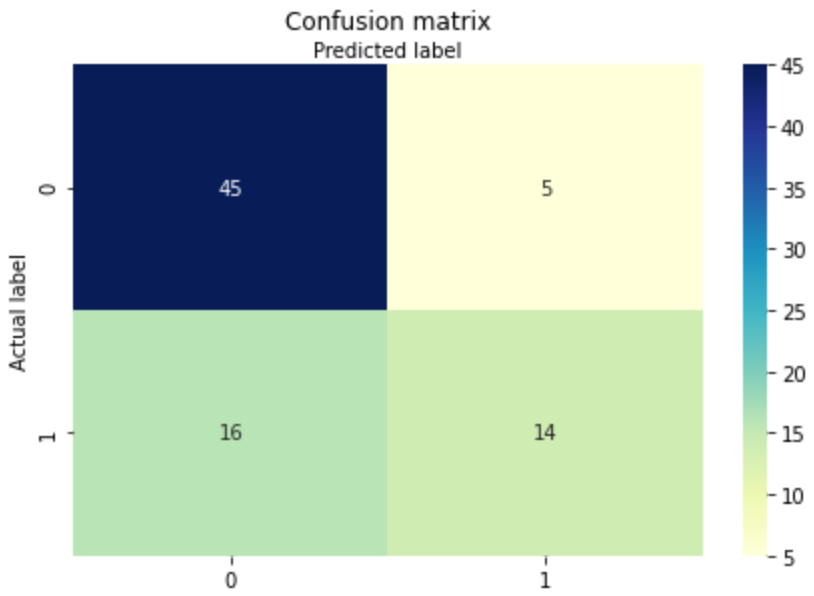
3. Again we plot this line and compare the result with Figure 1.
4. We see that both of them look very similar and the parameters returned by both are the same.



Q2 : Logistic Regression Task

Q2 :

1. Import the relevant libraries of python
    a. Matplotlib for plotting graphs
    b. Pandas for data set manipulation
    c. Numpy for vector algebra
2. 'Gender' column is a categorical variable however it doesn't have numerical values. We need to encode it so as to use it in training the model.
3. We have two choices for labelling - Label Encoding or OneHotEncoding.
4. Label encoding will say replace Male with 1, and female with 0. This gives an order of 1>0 which is not supposed to be present.
5. Hence we turn to OneHotEncoding. This we create two new columns 'Male', 'Female'. Where if the sample is a male, then 'Male' will have 1 and 'Female' 0.
6. We don't need the original 'Gender' column now so we drop it.
7. We can also drop the 'Male' column as not Female implies male.
8. Save the dataframe with 'Age' , 'EstimatedSalary', and 'Female' as X and 'Purchased' as y which is the target.
9. Now we scale all the features using *MinMaxScaler* imported from *sklearn.preprocessing*
10. We scale the features between [0,1]
11. We can now split the dataset in two parts : TRAIN and TEST with ratio 70:30
12. Now import LogisticRegression() from sklearn.
13. Fit the matrix X_train.
14. Predict the target variable using the model for the testing data.
15. Plotting a confusion matrix for *y_test* and *y_pred*.



We see that the confusion matrix shows how well our model has performed. 45+14 (entries along the left diagonal) are the correct predictions.