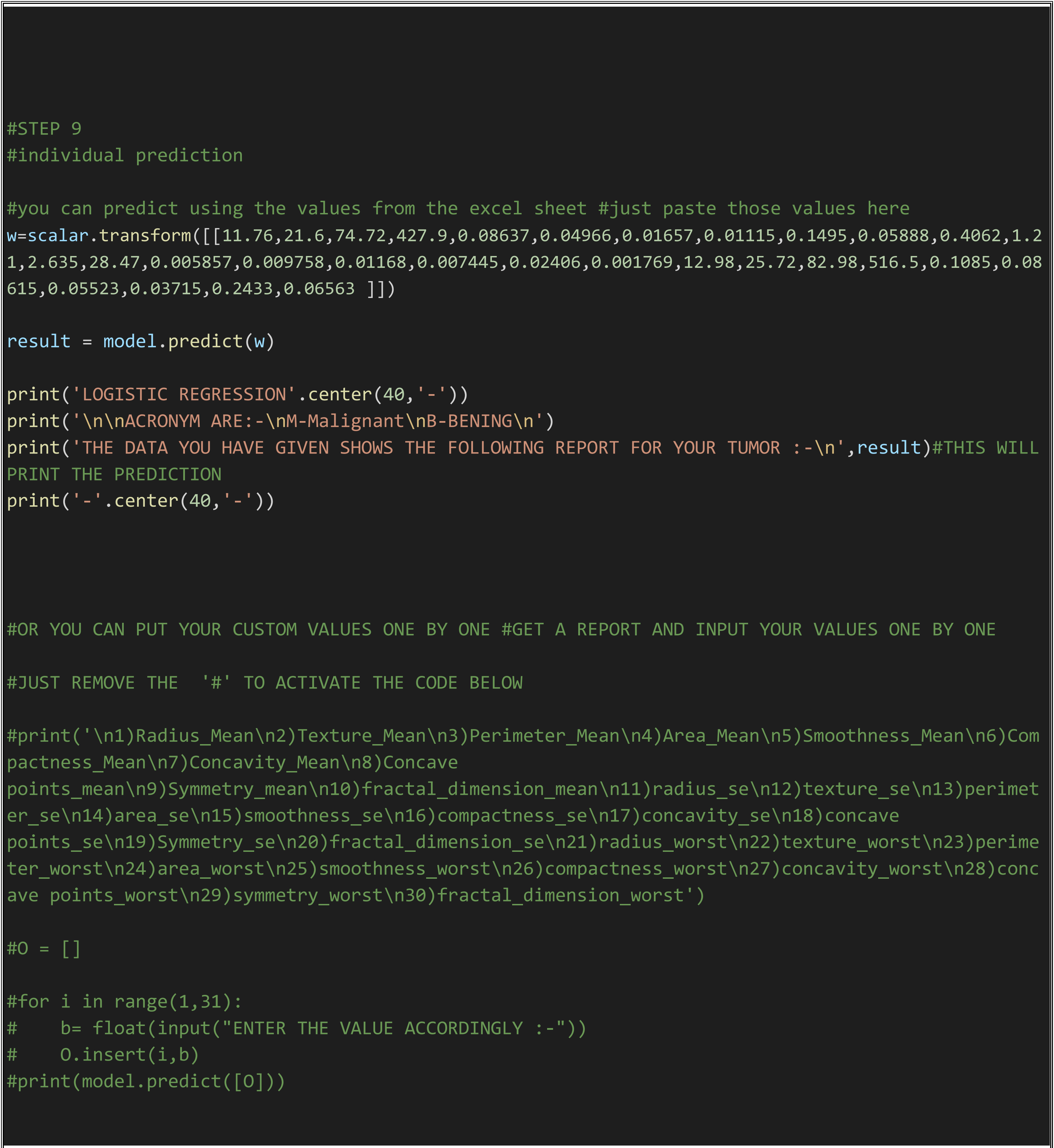
MAJOR PROJECT 1

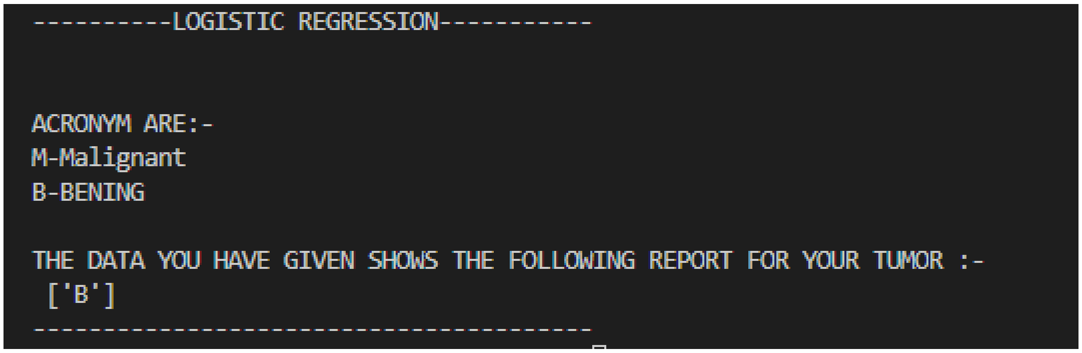
```
#MAJOR PROJECT 1

#NAME - AYUSH KUMAR SHARMA
#YEAR - 2ND  DEPT - ECE
#email - sharmaayushKv@gmail.com
#KALYANI GOVERNMENT ENGINEERING COLLEGE
#GITHUB - https://github.com/PHENOL47/TUMOR-LOGISTIC-REGRESSION-



#APPLYING LOGISTIC REGRESSION


#DATASET - (TUMOR DATASET) https://www.kaggle.com/code/headerstang/malignant-and-benign-cancer/data


#DESCRIPTION OF DATASET-

#Malignant - This type of tumors have cells that grow uncontrollably and spread locally and/or to distant sites. Malignant tumors are cancerous.

#BENING - This type of  tumors are those that stay in their primary location without invading other sites of the body. They do not spread and they are not cancerous.



#PROGRAM TO IDENTIFY FROM THE GIVEN DATA THAT THE TUMOR IS MALIGNANT OR BENING USING LOGISTIC REGRESSION.




#STEP 1
#TAKING THE DATAFRAME

import pandas as pd

df = pd.read_csv(r'C:\Users\PHENOL\Desktop\ml projects\TUMOR.csv')


#STEP 2

#taking inputs and the ouputs
#inputs
x = df.iloc[:,2:33].values #.values is to covert it into array
#output
y = df.iloc[:,1].values
```

```
#STEP 3
#Taking test train variables

from sklearn.model_selection import train_test_split

x_train,x_test,y_train,y_test = train_test_split(x,y,random_state=0)
#75% to train and 25% to test

#STEP 4
#NORMALIZING/SCALING THE VALUES

from sklearn.preprocessing import MinMaxScaler
scalar = MinMaxScaler()

X_test = scalar.fit_transform(x_test)
X_train = scalar.fit_transform(x_train)


#STEP 5
#APPLYING CLASSIFIER

from sklearn.linear_model import LogisticRegression

model = LogisticRegression()


#STEP 6
#FIT THE MODEL

model.fit(X_train,y_train)

#STEP 7
#Predicting the output

y_pred = model.predict(X_test)


#STEP 8
#checking the  accuracy


from sklearn.metrics import accuracy_score
m  = accuracy_score(y_pred,y_test)*100
```



```
#STEP 9
#individual prediction

#you can predict using the values from the excel sheet #just paste those values here
w=scalar.transform([[11.76,21.6,74.72,427.9,0.08637,0.04966,0.01657,0.01115,0.1495,0.05888,0.4062,1.21,2.635,28.47,0.005857,0.009758,0.01168,0.007445,0.02406,0.001769,12.98,25.72,82.98,516.5,0.1085,0.08615,0.05523,0.03715,0.2433,0.06563 ]])

result = model.predict(w)

print('LOGISTIC REGRESSION'.center(40,'-'))
print('\n\nACRONYM ARE:-\nM-Malignant\nB-BENING\n')
print('THE DATA YOU HAVE GIVEN SHOWS THE FOLLOWING REPORT FOR YOUR TUMOR :-\n',result)#THIS WILL PRINT THE PREDICTION
print('-'.center(40,'-'))




#OR YOU CAN PUT YOUR CUSTOM VALUES ONE BY ONE #GET A REPORT AND INPUT YOUR VALUES ONE BY ONE

#JUST REMOVE THE  '#' TO ACTIVATE THE CODE BELOW

#print('\n1)Radius_Mean\n2)Texture_Mean\n3)Perimeter_Mean\n4)Area_Mean\n5)Smoothness_Mean\n6)Compactness_Mean\n7)Concavity_Mean\n8)Concave points_mean\n9)Symmetry_mean\n10)fractal_dimension_mean\n11)radius_se\n12)texture_se\n13)perimeter_se\n14)area_se\n15)smoothness_se\n16)compactness_se\n17)concavity_se\n18)concave points_se\n19)Symmetry_se\n20)fractal_dimension_se\n21)radius_worst\n22)texture_worst\n23)perimeter_worst\n24)area_worst\n25)smoothness_worst\n26)compactness_worst\n27)concavity_worst\n28)concave points_worst\n29)symmetry_worst\n30)fractal_dimension_worst')

#O = []

#for i in range(1,31):
#    b= float(input("ENTER THE VALUE ACCORDINGLY :-"))
#    O.insert(i,b)
#print(model.predict([O]))
```

OUTPUT :-



```
----------LOGISTIC REGRESSION-----------


ACRONYM ARE:-
M-Malignant
B-BENING

THE DATA YOU HAVE GIVEN SHOWS THE FOLLOWING REPORT FOR YOUR TUMOR :-
 ['B']
----------------------------------------
```