Name:Carin Samy
ID:7481

Name:Hassan Tamer
ID:7405

Assignment one Report

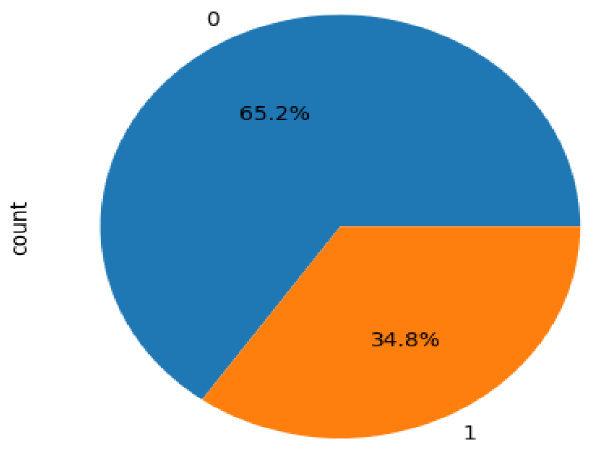
1) Data Preprocessing

Steps taken and problems solved

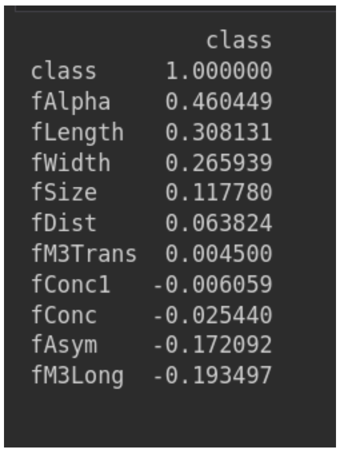
1) Input Header row in the following list

```
["fLength","fWidth","fSize","fConc","fConc1","fAsym","fM3Long","fM3Trans","fAlpha","fDist","class"]
```

2) Omit duplicates
3) Check for nulls or nan's and they were not found
4) Label encode categorical data (class)
5) Solving data imbalance between gamma and hadrons by under sampling
And under sampling was chosen because the dataset was big enough
The below Pie chart shows the imbalance

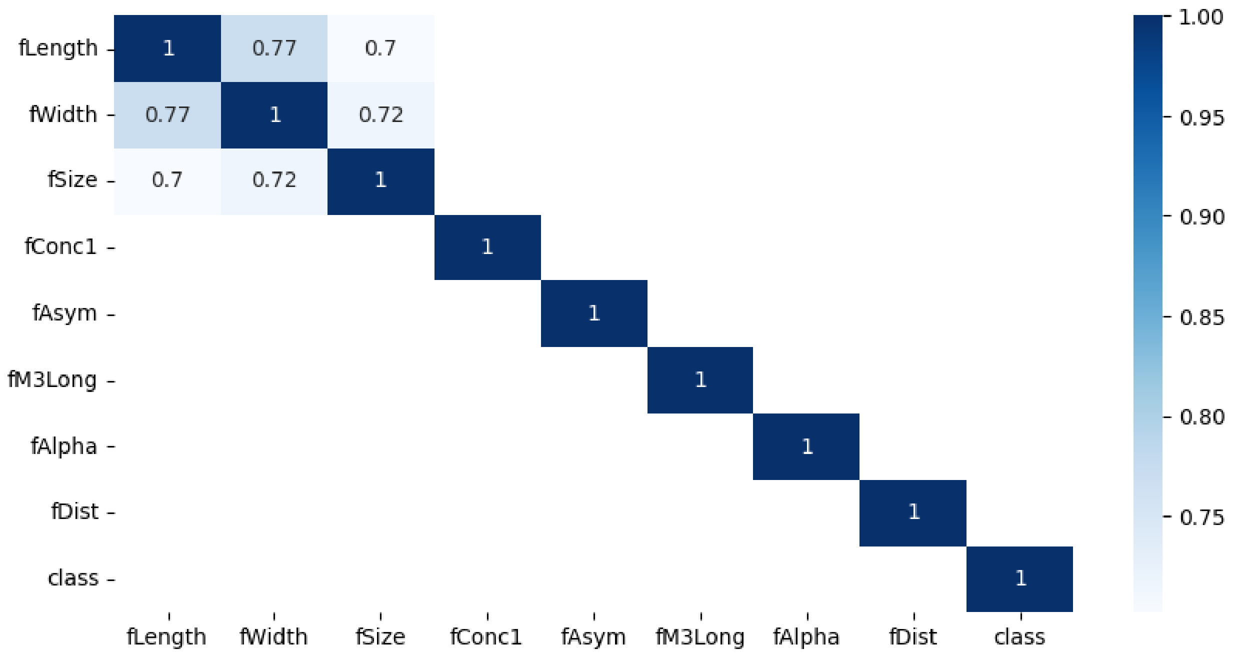


6) Visualizing the correlation between the features and the class column using R value
R value is between -1 and 1, where 1 indicates a sharp positive correlation and -1 is the opposite
And according to the below table ‘fConc’, ’fDist’, ’fConc1’ and ‘fM3trans’ were omitted



```
             class
class     1.000000
fAlpha    0.460449
fLength   0.308131
fWidth    0.265939
fSize     0.117780
fDist     0.063824
fM3Trans  0.004500
fConc1   -0.006059
fConc    -0.025440
fAsym    -0.172092
fM3Long  -0.193497
```

7) Visualizing correlations between features
So according to the below diagram width,length and size were combined into one feature called magnitude where magnitude = length*width*size

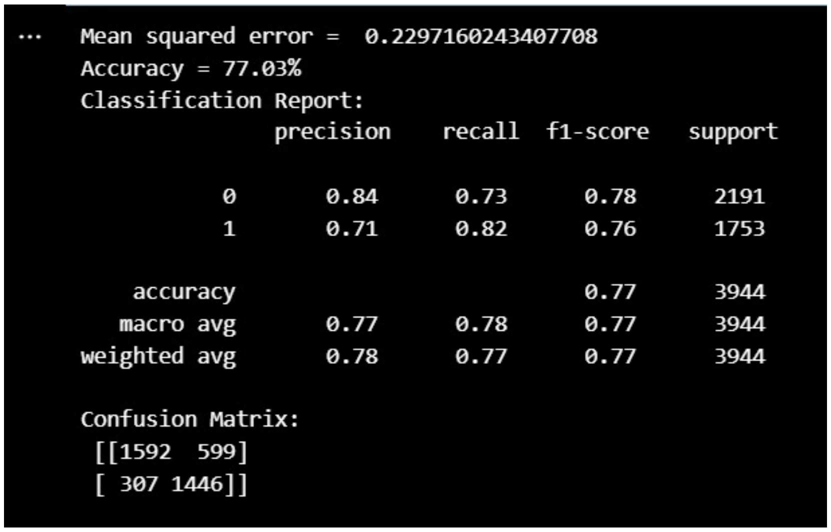


Training Logistic Regression

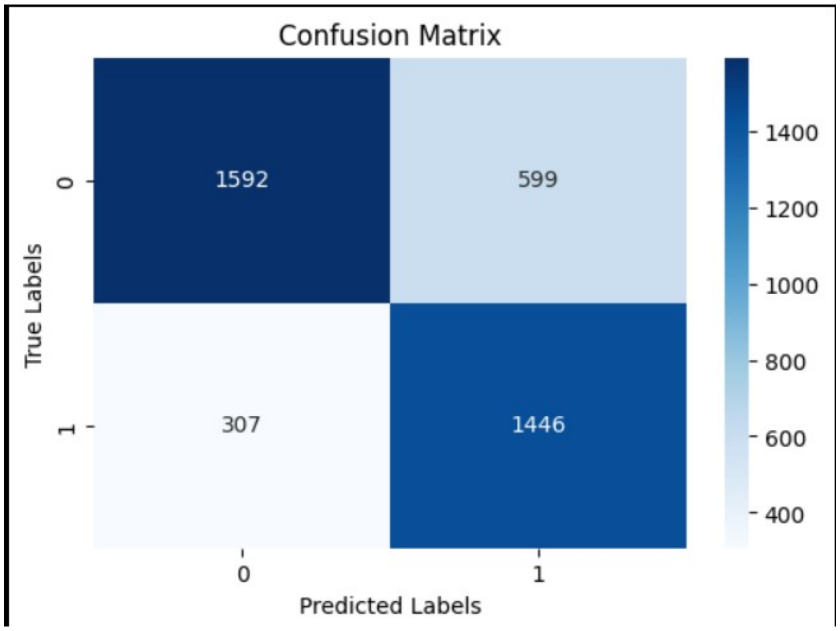
1) Split data to training and test
2) Normalize data using standard scalar in sklearn

It makes the mean = 0 and std = 1

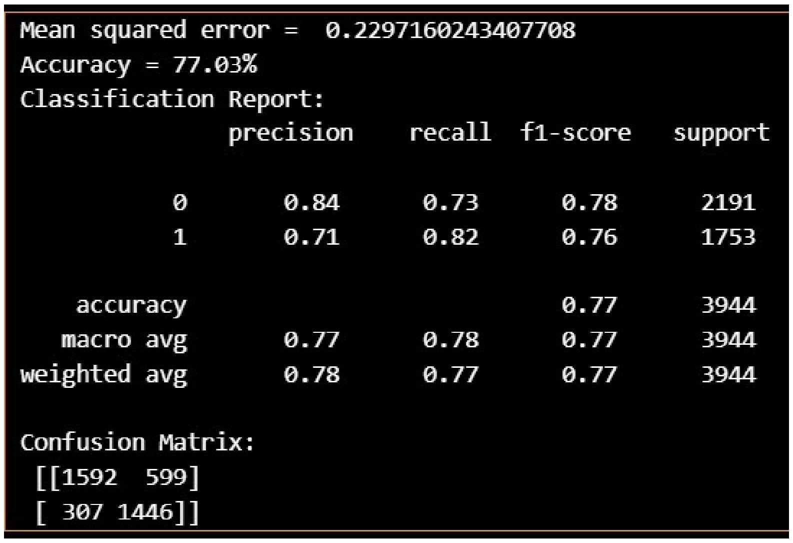
3) Training logistic regression using default c value = 1 lead to the following scores



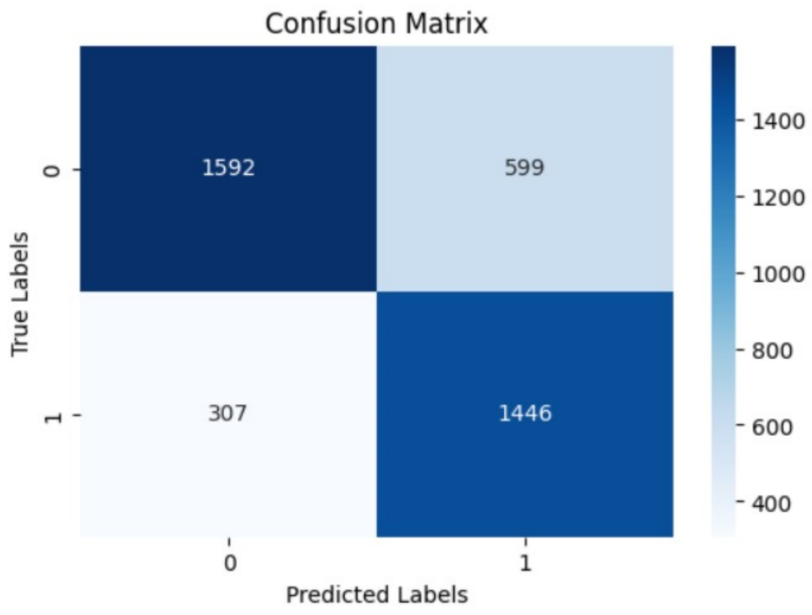
```
...  Mean squared error =  0.2297160243407708
     Accuracy = 77.03%
     Classification Report:
                   precision    recall  f1-score   support

                0       0.84      0.73      0.78      2191
                1       0.71      0.82      0.76      1753

         accuracy                           0.77      3944
        macro avg       0.77      0.78      0.77      3944
     weighted avg       0.78      0.77      0.77      3944

     Confusion Matrix:
      [[1592  599]
      [ 307 1446]]
```



4) After tuning c value we found the best c at 200 and best penalty to be l2 and got the following scores

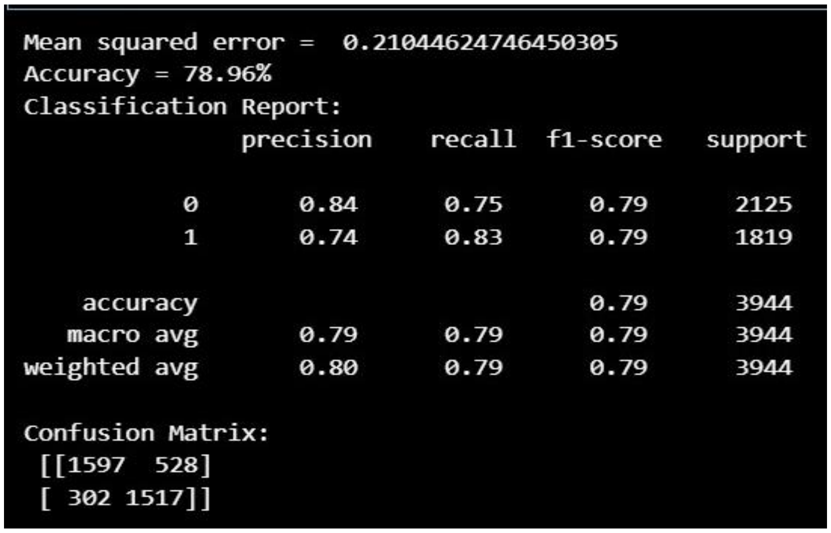


```
Mean squared error =  0.2297160243407708
Accuracy = 77.03%
Classification Report:
              precision    recall  f1-score   support

           0       0.84      0.73      0.78      2191
           1       0.71      0.82      0.76      1753

    accuracy                           0.77      3944
   macro avg       0.77      0.78      0.77      3944
weighted avg       0.78      0.77      0.77      3944

Confusion Matrix:
 [[1592  599]
 [ 307 1446]]
```

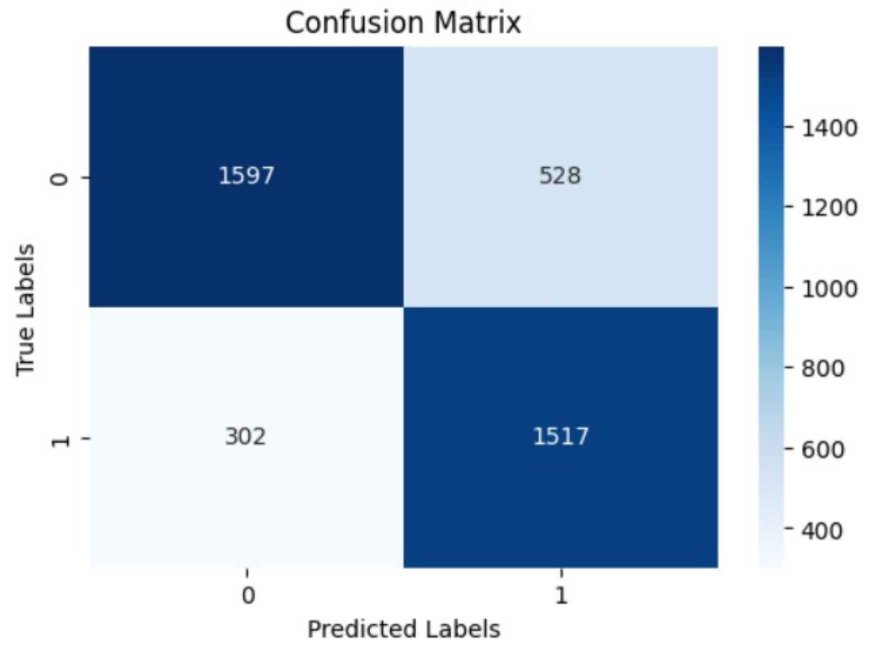


Training KNN

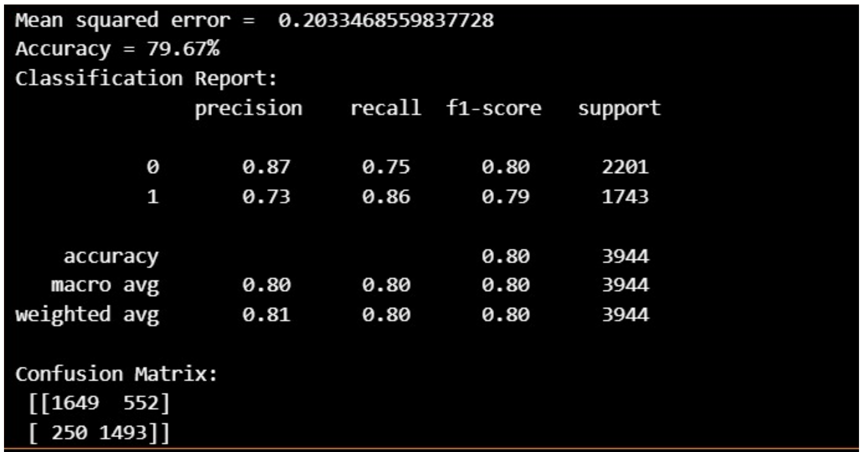
1) Training at default k = 5 we got the following results



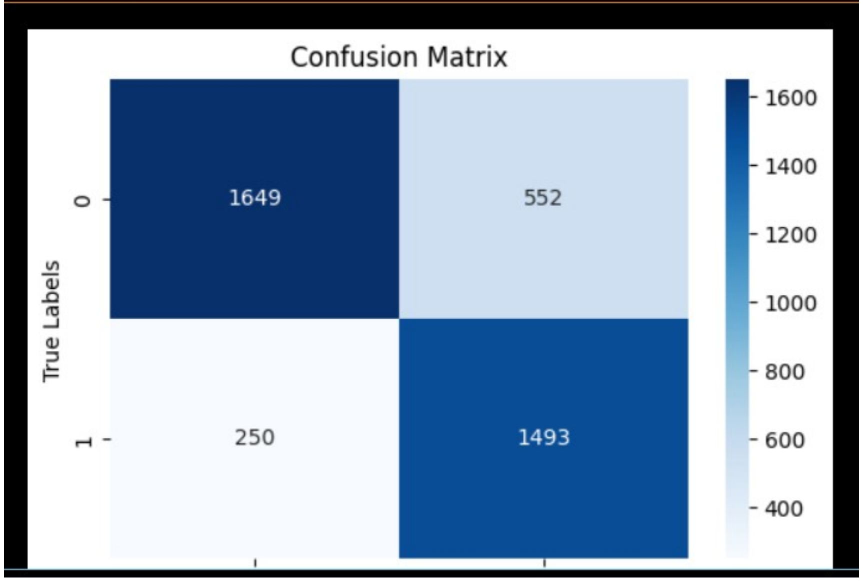
```
Mean squared error =  0.21044624746450305
Accuracy = 78.96%
Classification Report:
              precision    recall  f1-score   support

           0       0.84      0.75      0.79      2125
           1       0.74      0.83      0.79      1819

    accuracy                           0.79      3944
   macro avg       0.79      0.79      0.79      3944
weighted avg       0.80      0.79      0.79      3944

Confusion Matrix:
 [[1597  528]
 [ 302 1517]]
```



2) After tuning K we found the best K at k=11 which gave us the following improvement



```
Mean squared error =  0.2033468559837728
Accuracy = 79.67%
Classification Report:
              precision    recall  f1-score   support

           0       0.87      0.75      0.80      2201
           1       0.73      0.86      0.79      1743

    accuracy                           0.80      3944
   macro avg       0.80      0.80      0.80      3944
weighted avg       0.81      0.80      0.80      3944

Confusion Matrix:
 [[1649  552]
 [ 250 1493]]
```


Confusion Matrix
True Labels