Name: Siddharth Nilesh Joshi
ASU Id: 1217923356
Project 2: Machine Learning Mine vs Rock

Results:

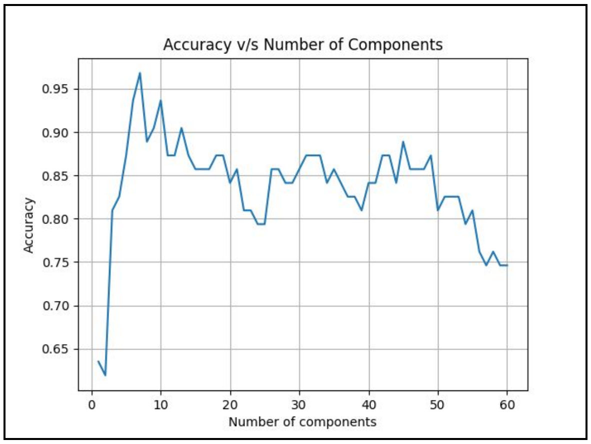
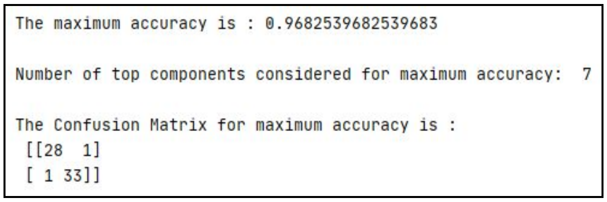


Fig 1: Plot of Accuracy v/s Number of Components



```
The maximum accuracy is : 0.9682539682539683

Number of top components considered for maximum accuracy:  7

The Confusion Matrix for maximum accuracy is :
 [[28  1]
 [ 1 33]]
```

Fig 2: Output Statistics

The plot and the statistics of the dataset analysis through *Principal Component Analysis (PCA)* and *Multi-level Perceptron* is as shown above.

Through *PCA* it was discovered that the 7 top components (features) given in the dataset were sufficient to train a model with a maximum accuracy of 0.9683 as evident from the figures above. This means that the top 7 features are sufficient to accurately differentiate between mines and rocks 96.83% of the time. However, still risky, it is interesting to know that the top 7 features are sufficient enough for accuracy this high. As observed from the plot, as the batch of features used to train the model increased a plunging trend was observed. This is because except the top 7 features all the other components are noise and considering them while training reduces the model accuracy.
The multi-level perceptron was designed with 100 hidden layers, using *logistic activation* and *adam solver* with 2000 iterations and an alpha of 0.0001.

The 2-by-2 confusion matrix for 2 classes is [[28 1]; [1 33]]. This means the model made 63 classifications out of which 28 and 33 were correct (true) classifications and the other 2 were incorrect (false) classifications. So accuracy = (28 + 33) / (28 + 33 + 2) = 0.9683. And the error rate of the model = 1 - accuracy = 1 - 0.9683 = 0.03175 (i.e. 3.175 %). Here true classifications mean that the mines and rocks were correctly classified as mines and rocks respectively; on the contrary false classifications predict rock for a mine and vice versa.

The parameters chosen for the multi-level perceptron were the result of Gris-search. This technique for parameter optimization can be implemented over models to iterate over each possible parameter combination. The only downside of this technique is that it is computationally expensive over complex models.