**Submitted by: Venkatachalapathy Othisamy**

**Models based on Accuracy:**

**Steps involved:**

**Feature Selection:**

The dataset contains 57 attributes, which is too many. Hence, I used Recursive Feature Elimination technique to remove the highly correlated features and the ones which are not correlated with the target variable at all.

I used different techniques for the RFE model such as

* LinearSVC,
* Extra Trees classifier,
* Support Vector Regression
* PCA

to get the most optimal number of features. I used the output of different RFE models to get the accuracy using k-NN model. In k-NN, Extra trees classifier gave the highest accuracy and hence I used Extra trees classifier output, which gave me 19 features.

**Normalization:**

I used min-max normalization technique to normalize all the predictors in the newly formed dataset, since many classifiers use distance based calculations.

**Balance classes:**

Since we are predicting accuracy, it is normal to balance classes. But balancing classes means we are losing out the data which can be used by model for learning. Hence I did not do balance classes even to calculate accuracy.

**Model: k-NN**

**Parameters tested:**

**Number of neighbors:**

In k-NN, the number of neighbors determine the performance of the model. Hence I ran a loop with different number of neighbors and calculated the accuracy. The best possible accuracy came for when the number of neighbors chosen is 15. Below is the graph depicting the accuracy vs number of neighbors.

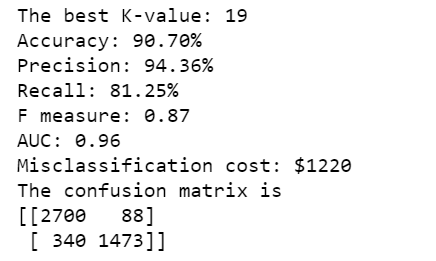
**Distance Metric:**

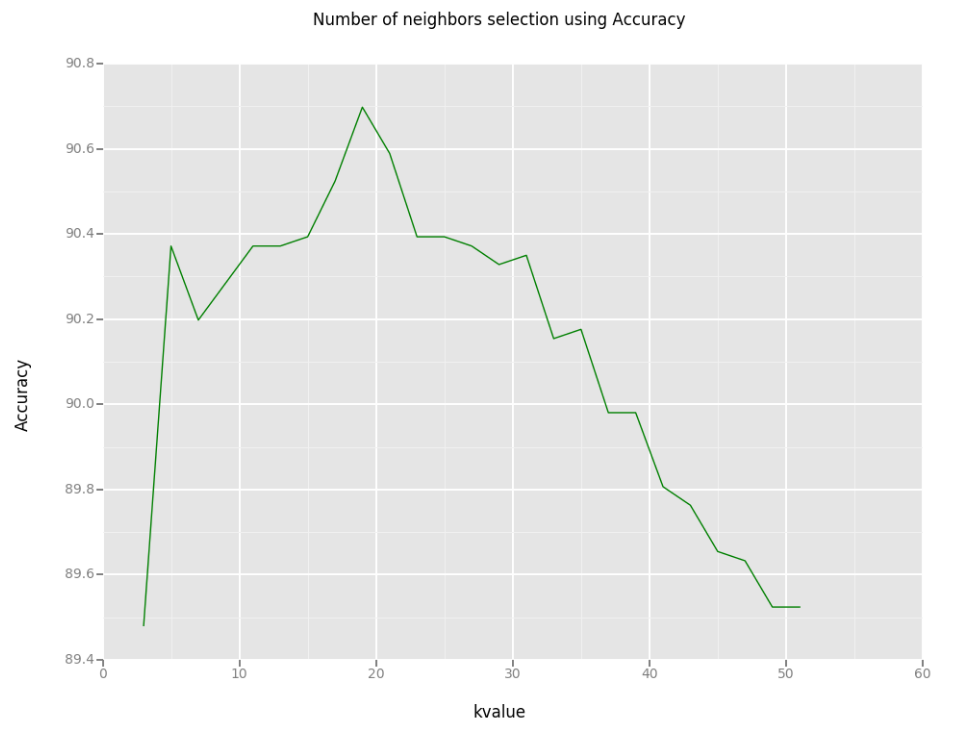
The Manhattan distance gave the highest accuracy. Euclidean distance metric gave slightly lower accuracy.

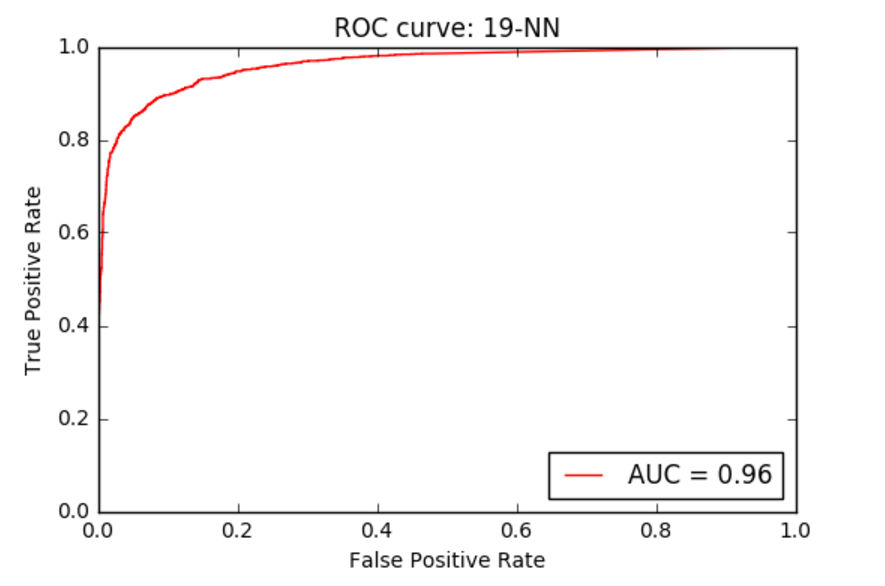
**Determination of nearest neighbors:**

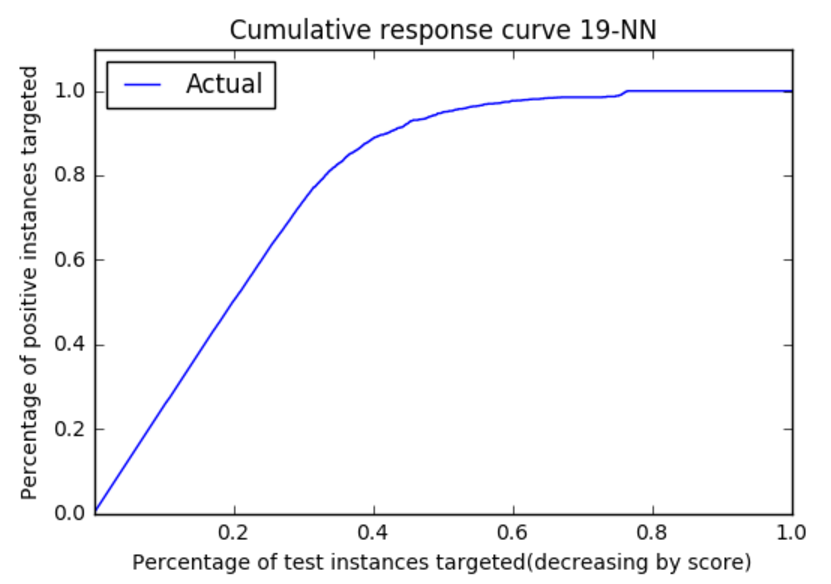
I used weighted k-NN to calculate the nearest neighbors. Majority voting gave slightly lower accuracy.

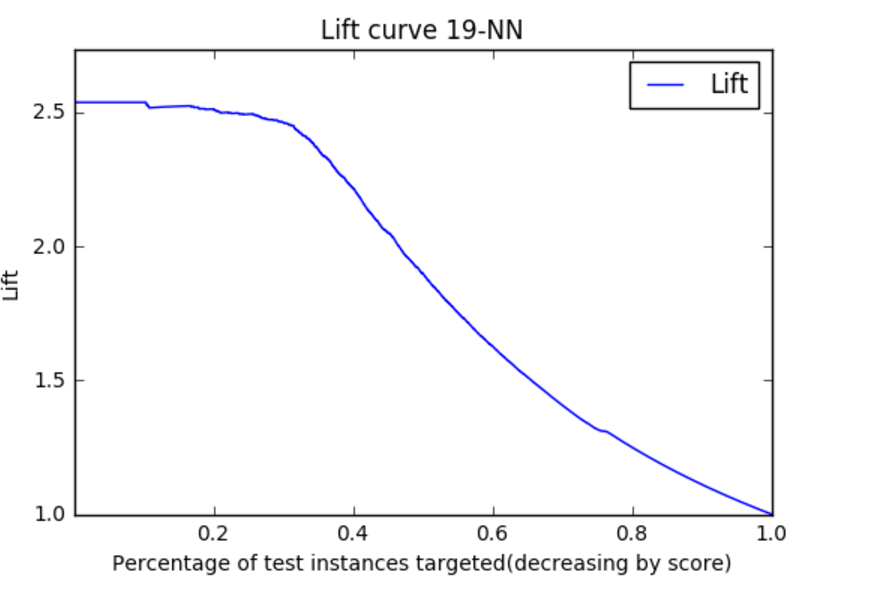
Below are the other performance metrics of the model.





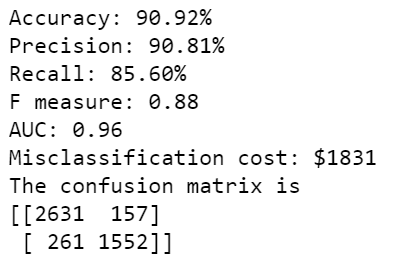


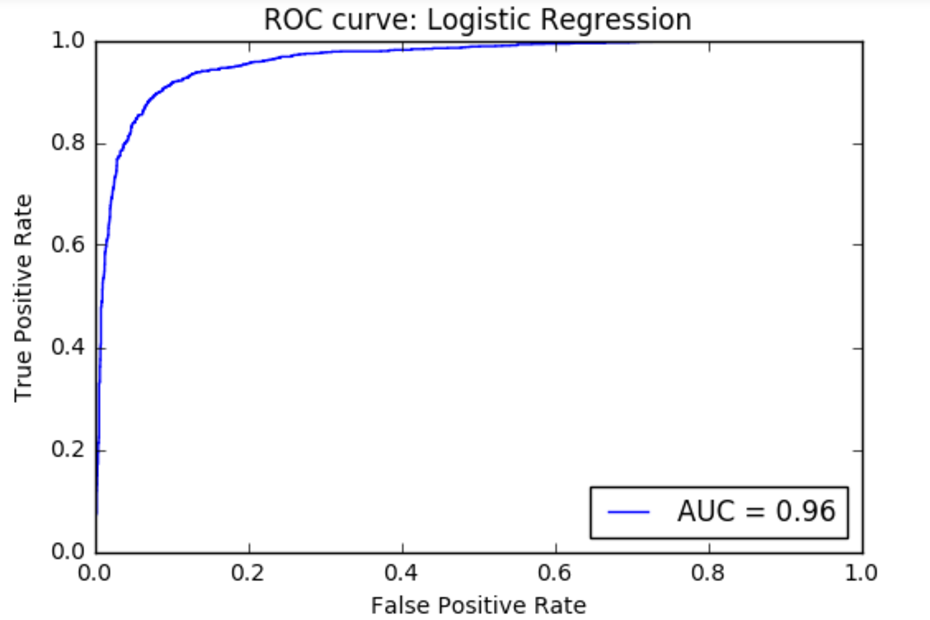


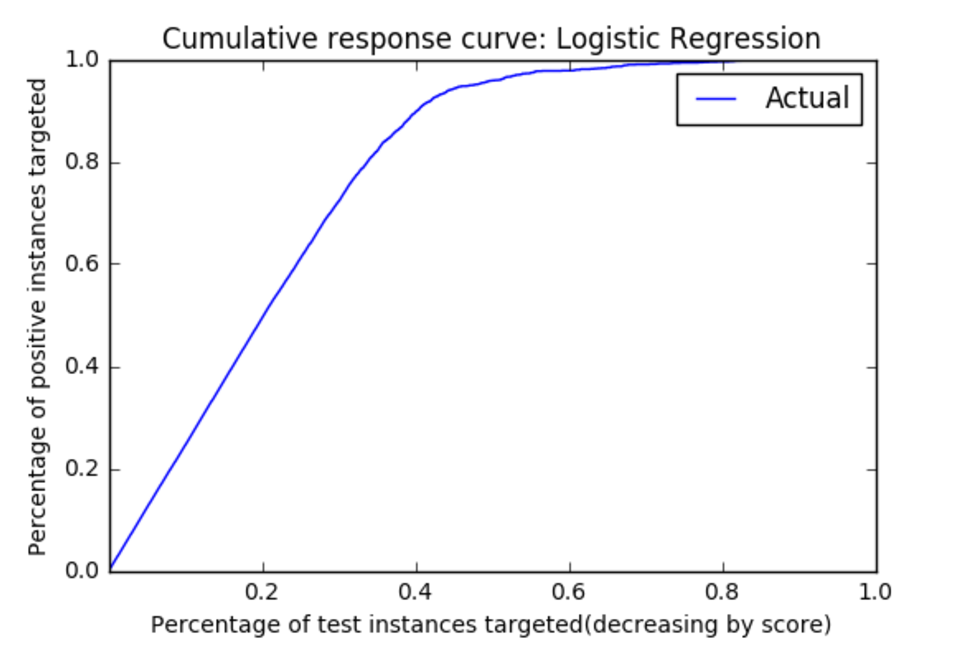


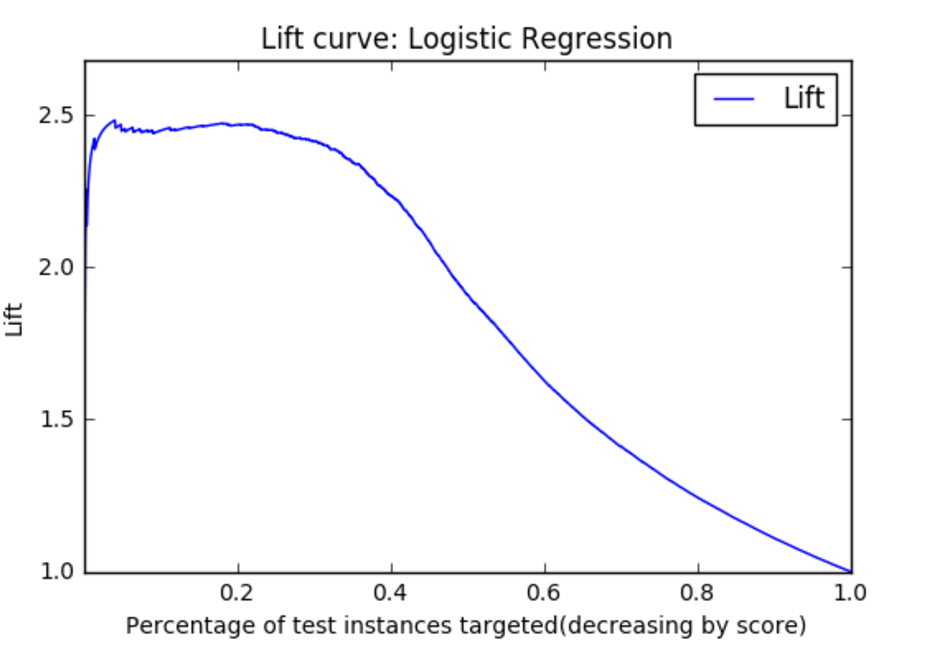
**Logistic Regression:**

Below is the performance metrics of Logistic regression.





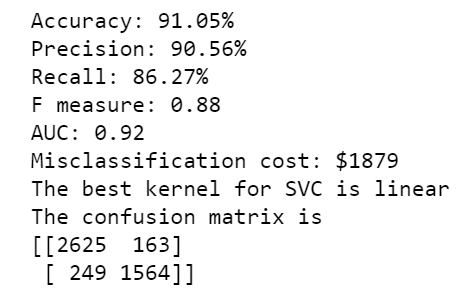


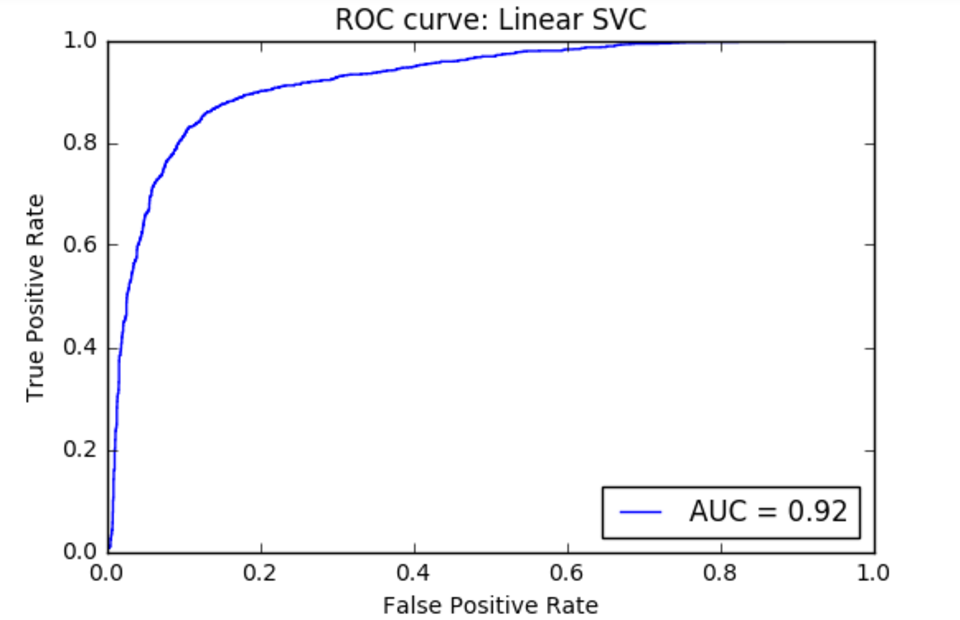


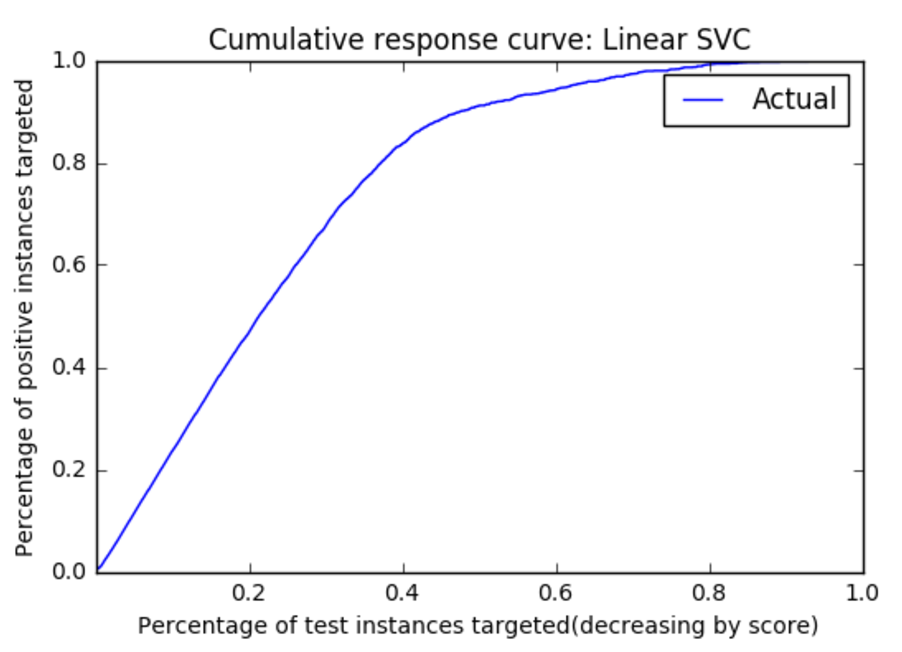
**SVC:**

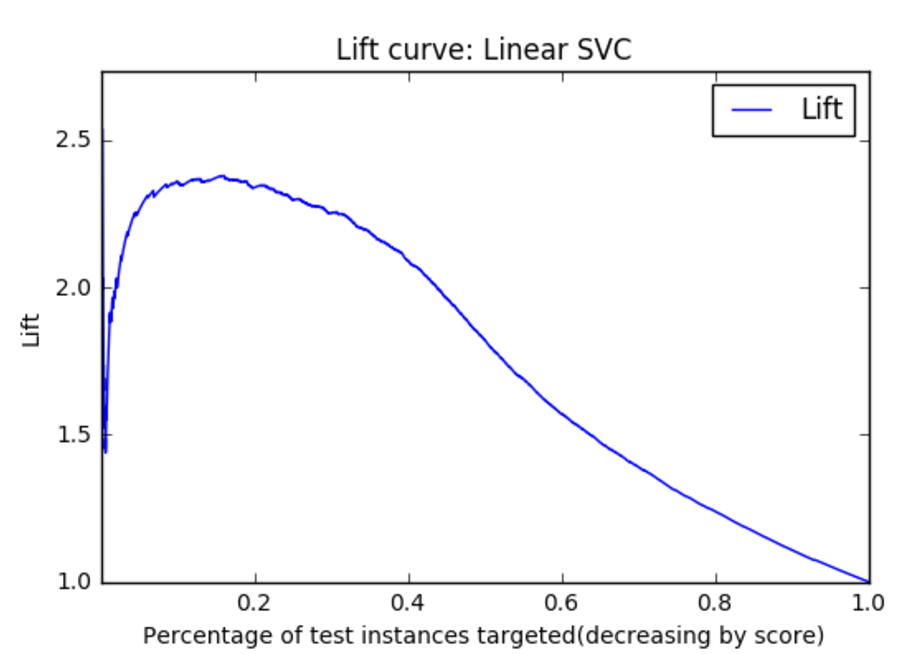
In support vector classifier, I changed the kernel to get the best possible accuracy.

Also, I changed the penalty parameter, to penalize the misclassifications more. Below is the performance of Support Vector Classifier.



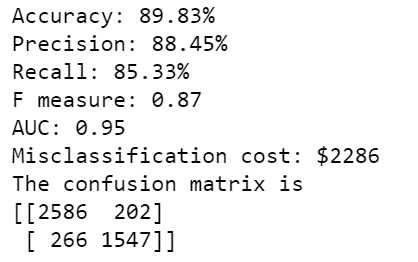


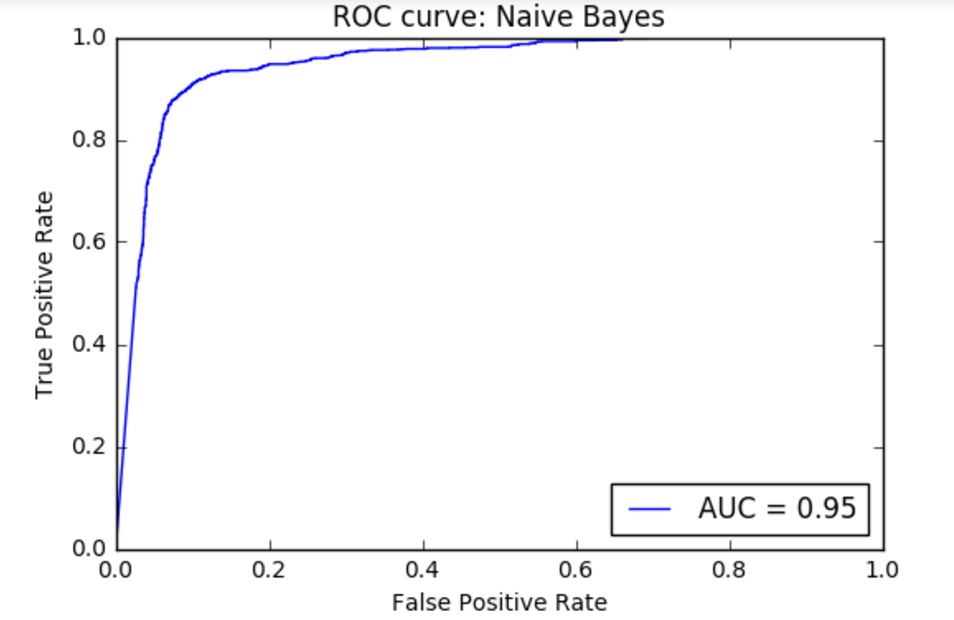


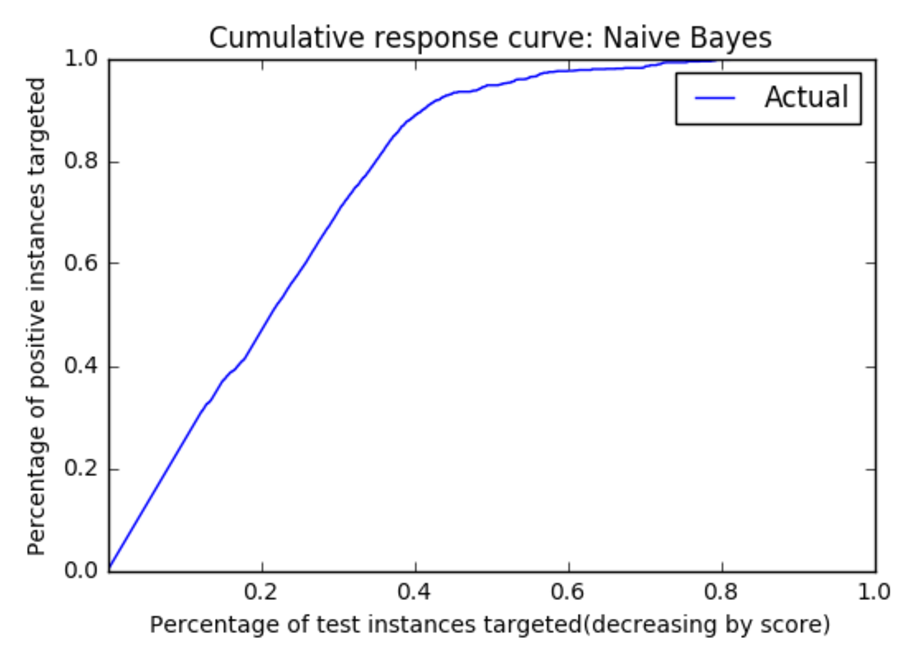


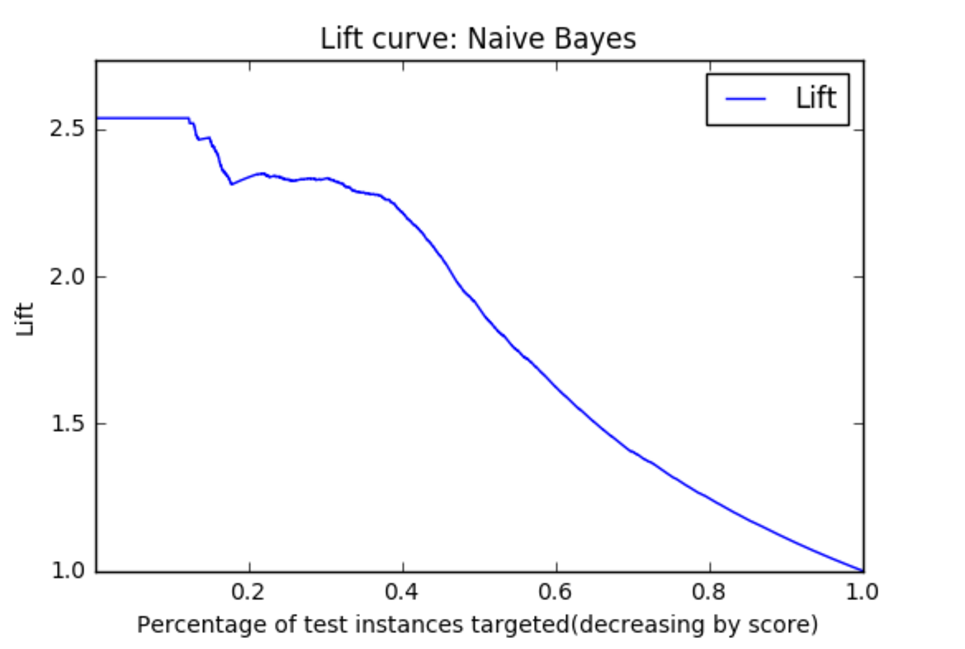
**Naïve Bayes:**

Naïve Bayes give the below performance. Of all the models, Naïve Bayes gave the lowest accuracy which is 89.83%









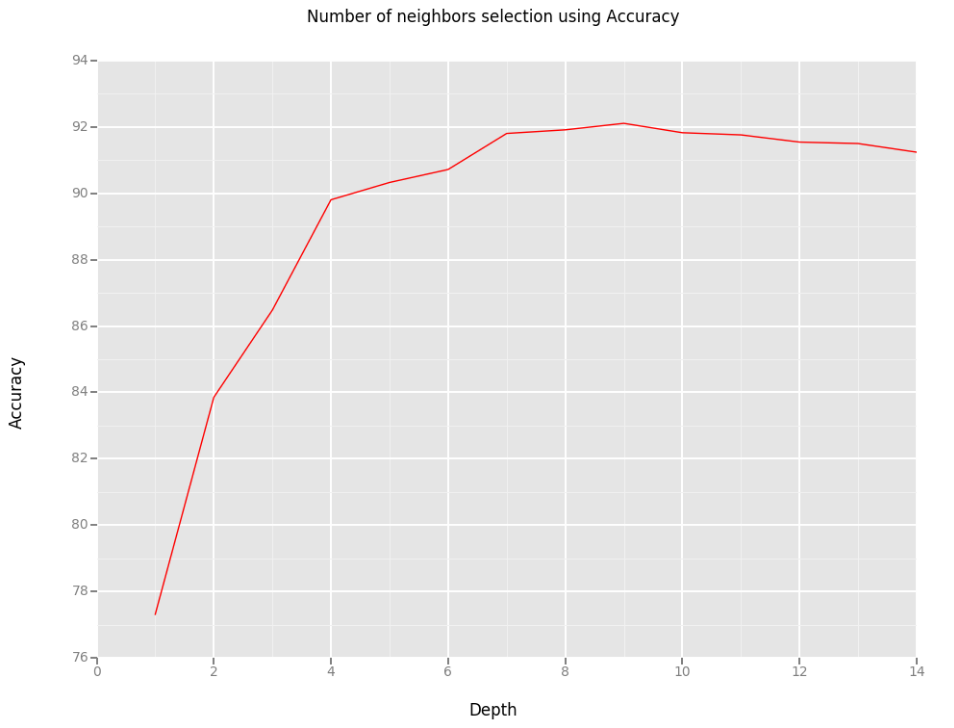
**Decision Tree:**

For decision trees, I have used different depth to get the possible accuracy. It is achieved for depth =9.

The criterion I have used is Entropy, which gave me the highest possible accuracy. I have changed the class weight option to ‘balanced’ to get the maximum accuracy.

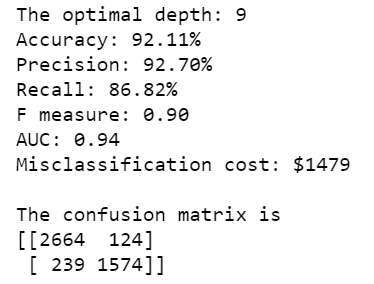
I have changed the minimum number of samples required to split the node etc. Also, I changed the minimum number of samples required to be at leaf node. But only the depth change influenced the accuracy.

Below is the graph that shows the relationship between the tree depth and the accuracy. We can clearly see that the best possible accuracy is achieved at depth =9.

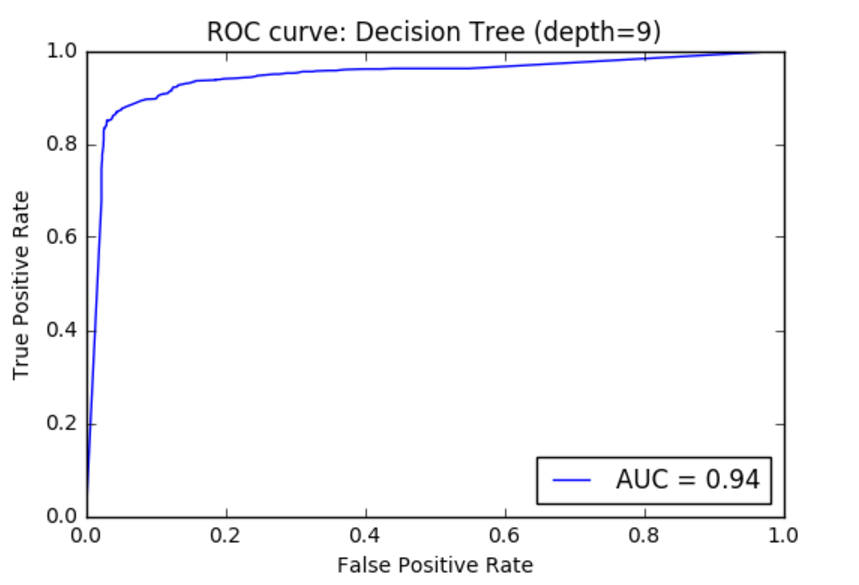


Below is the other performance metrics for the best decision tree model.

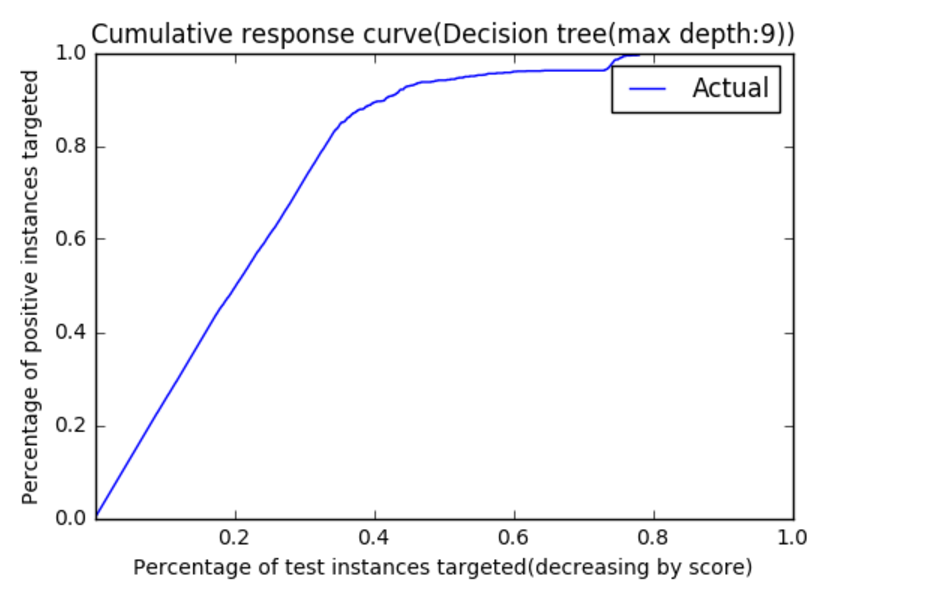
ROC curve:



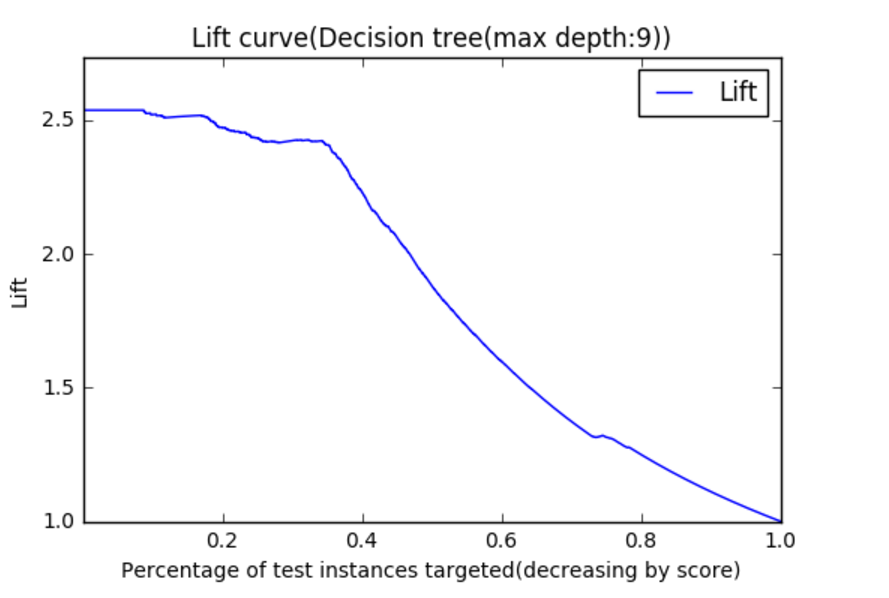
The area under the curve for decision tree model is 0.94, which is not the best which explains why Accuracy alone is not good to judge a model. Below is the visualization of the best model.



In cumulative response curve, we can see that almost 85% of the positive instances are targeted, with 40% of the test instances.



The below lift curve shows that the decision tree has very good initial lift of 2.6, which is better than Logistic Regression Naïve Bayes.



Hence, in terms of accuracy, Decision tree scores high even though, the AUC is not the best.

**Models based on Misclassification cost**

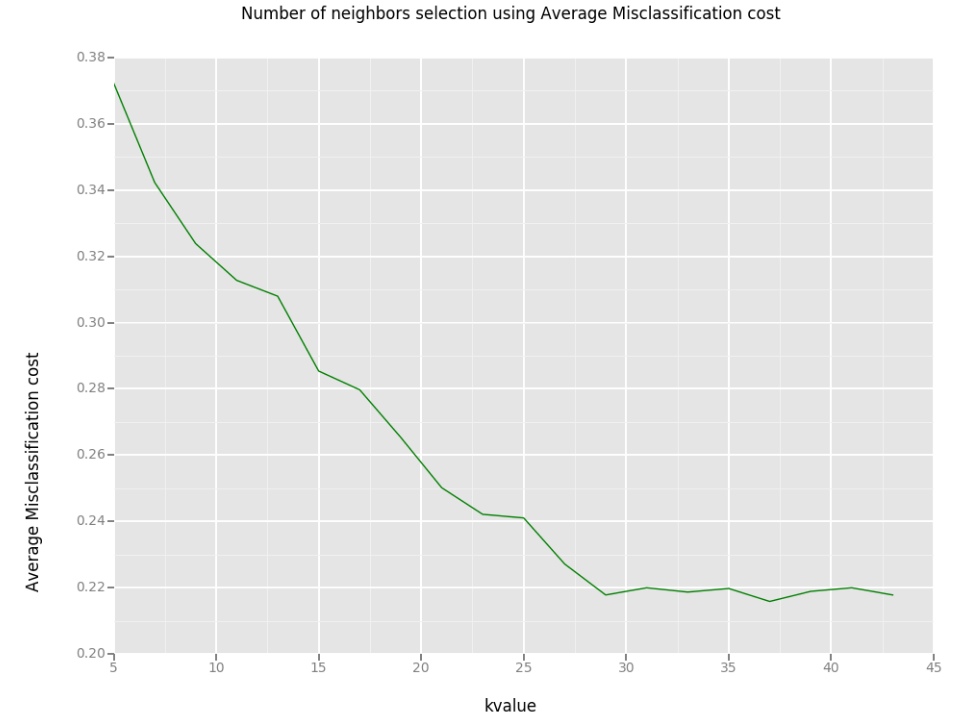
**Calculation of misclassification cost:**

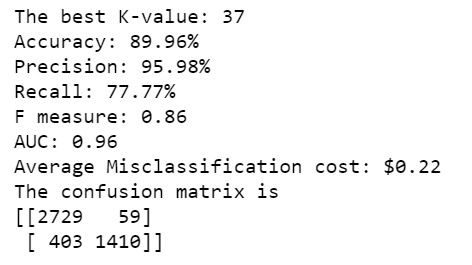
Here, the false positives are more penalized. Because, if a non-spam mail is detected ass spam, then the mail will go to spam folder and there is a high chance that the user will miss that mail. This is more dangerous than not detecting a spam mail. Below is the misclassification cost matrix.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | Actual | |
| Cost Matrix | | Spam | Not Spam |
| Predicted | Spam | 0 | 10 |
| Non-spam | 1 | 0 |

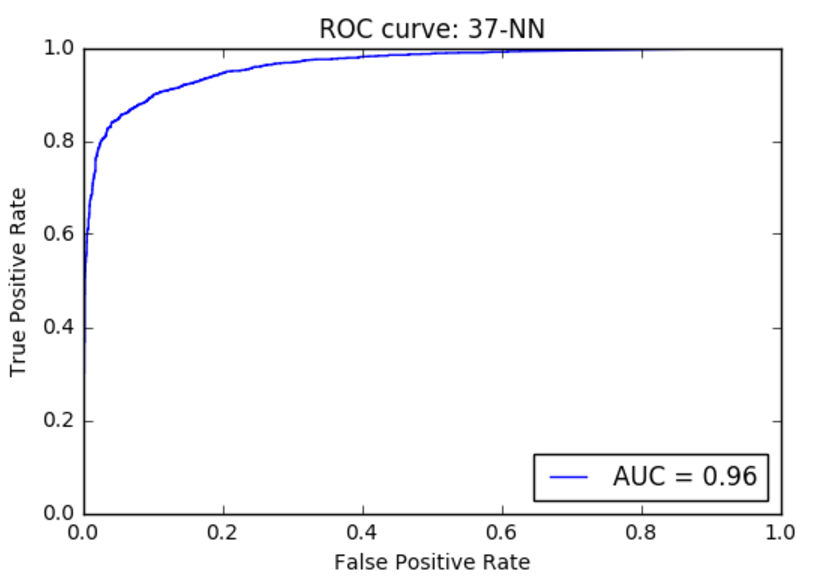
**Model: k-NN**

As in accuracy, the number of neighbors determine the misclassification cost the most. Below is the relation between the number of neighbors and the misclassification cost. The number of neighbors is 37 for the model, which has the least misclassification cost, which is 0.22.

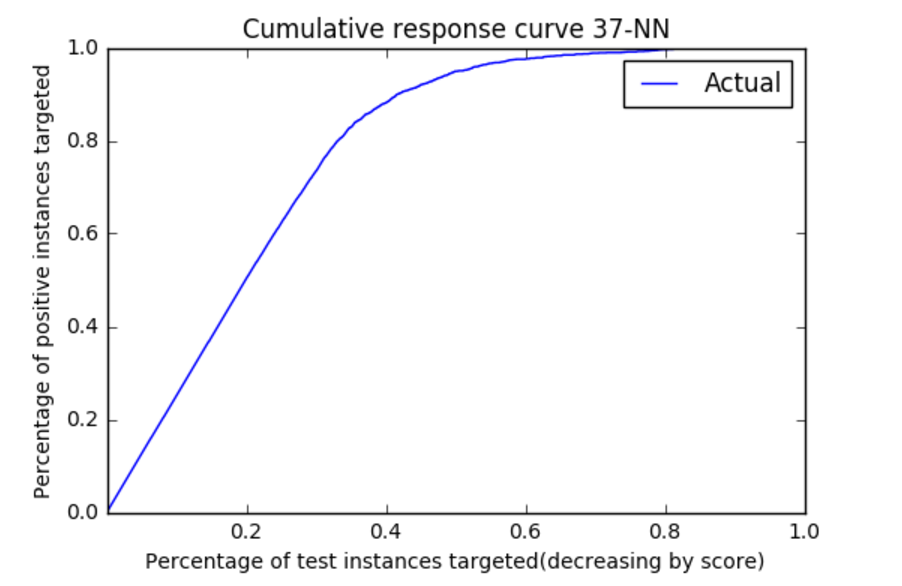




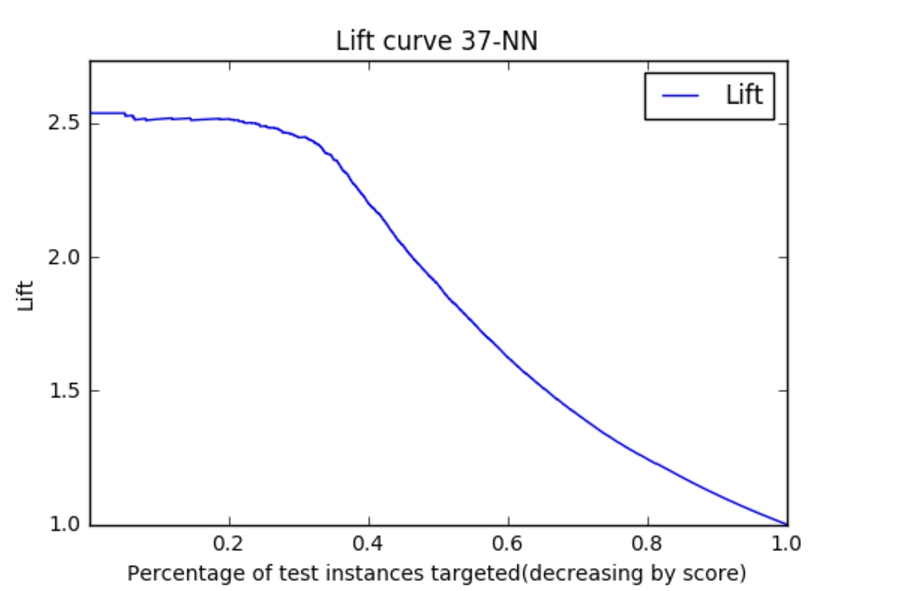
The ROC of the k-NN is 0.96 which is the best. This tells us that, if misclassification cost is known, then this can be a good indicator of model performance.



The cumulative response curve of the model shows that almost 85% of the positives are targeted when 40% of the test instances are targeted, which is very good.



k-NN has very good initial lift and sustains till 40% of the test instances.



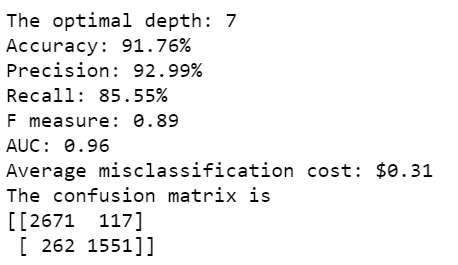
Decision Tree:

In decision tree, as in accuracy, the tree depth affected the misclassification cost the most. Below graph shows the relationship between, tree depth and misclassification cost.

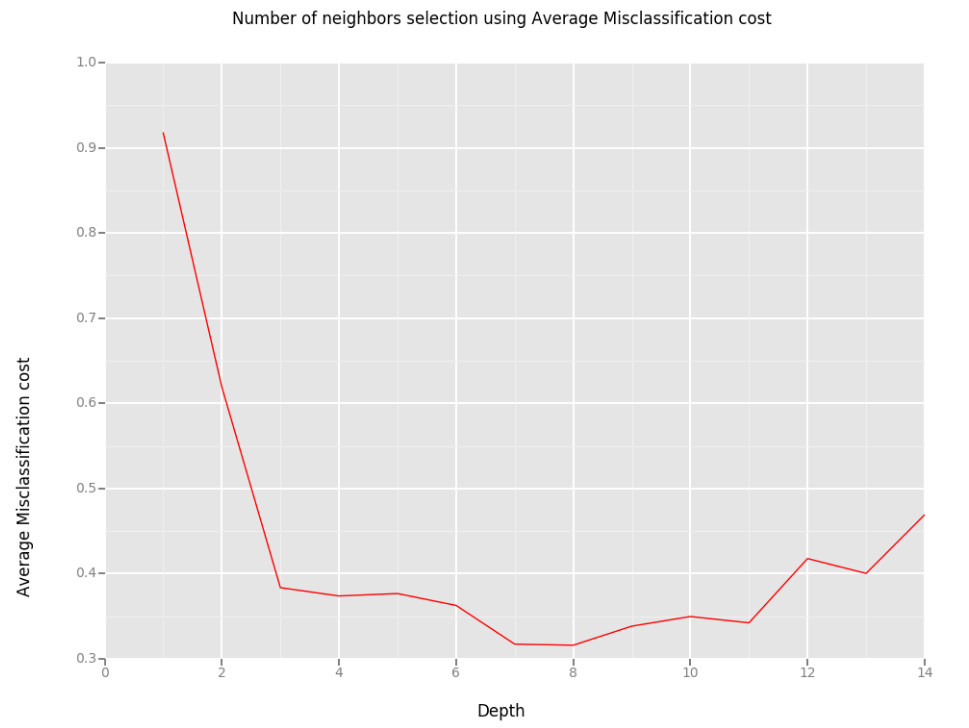
Other parameters tried:

* class weight option to ‘balanced’
* minimum number of samples required to split the node.
* minimum number of samples required to be at leaf node.

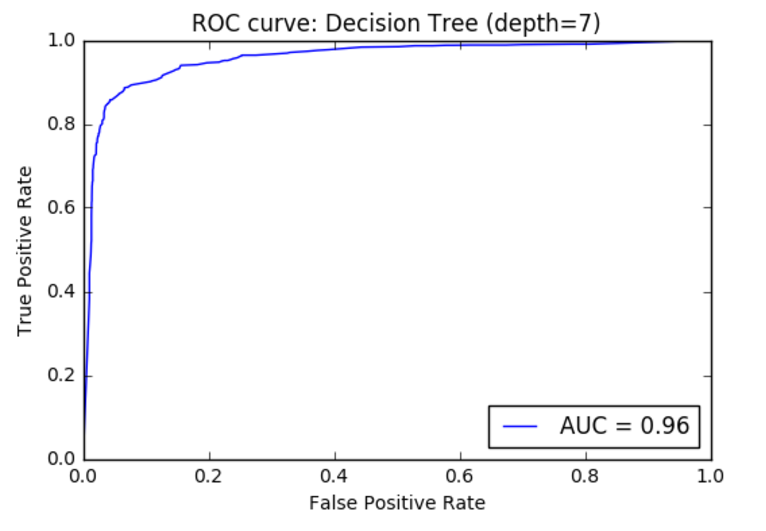
Here is the performance of decision tree.



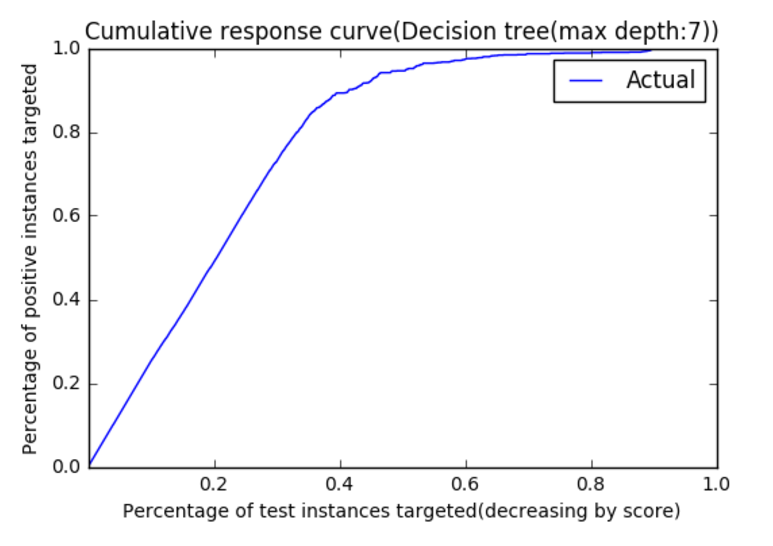
The average misclassification cost for decision tree is $0.31 which is higher than k-NN.

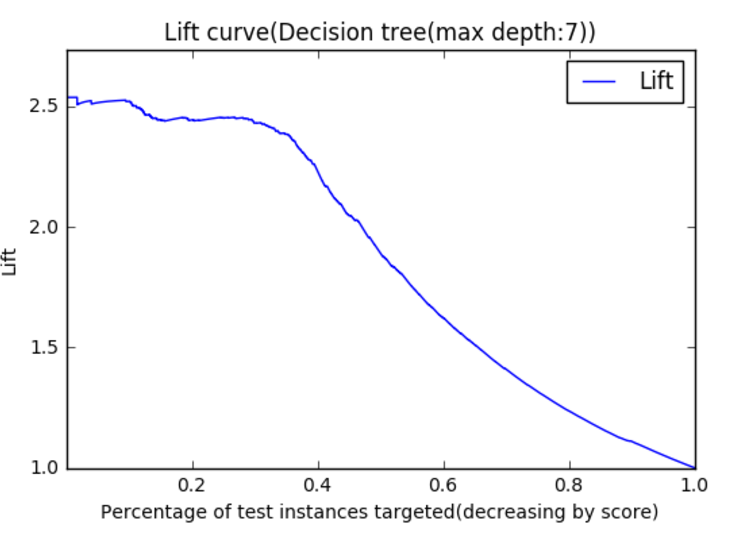


The ROC curve of k-NN is also 0.96, which means that this model performs equally well with k-NN in general.



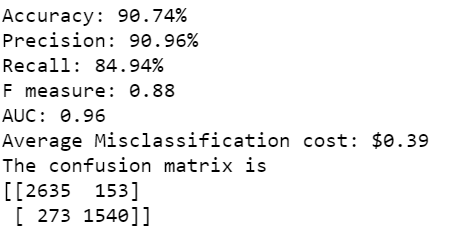
**Decision tree has** similar cumulative response and lift curves.

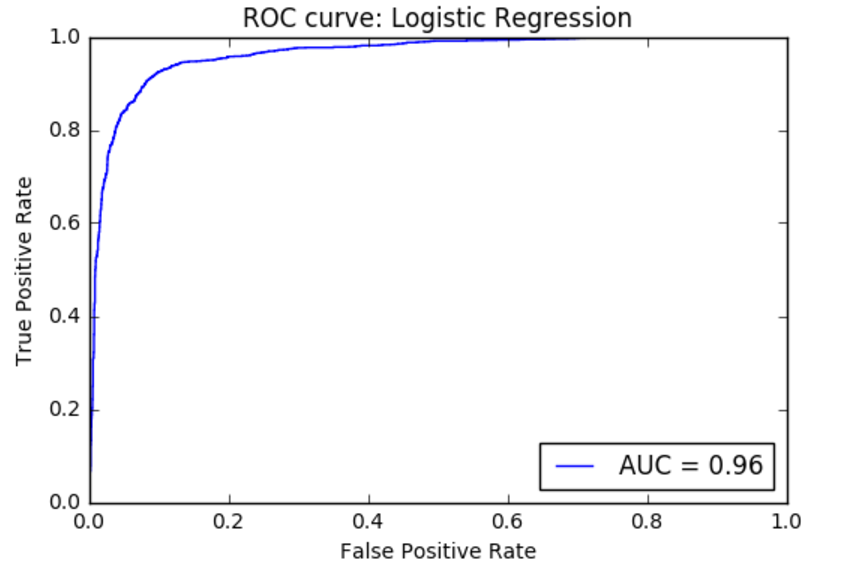




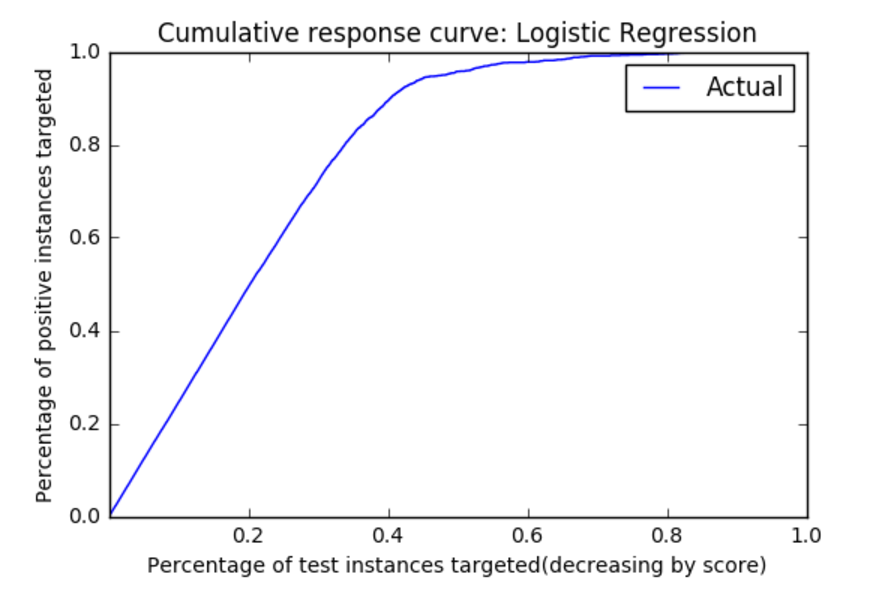
**Logistic regression:**

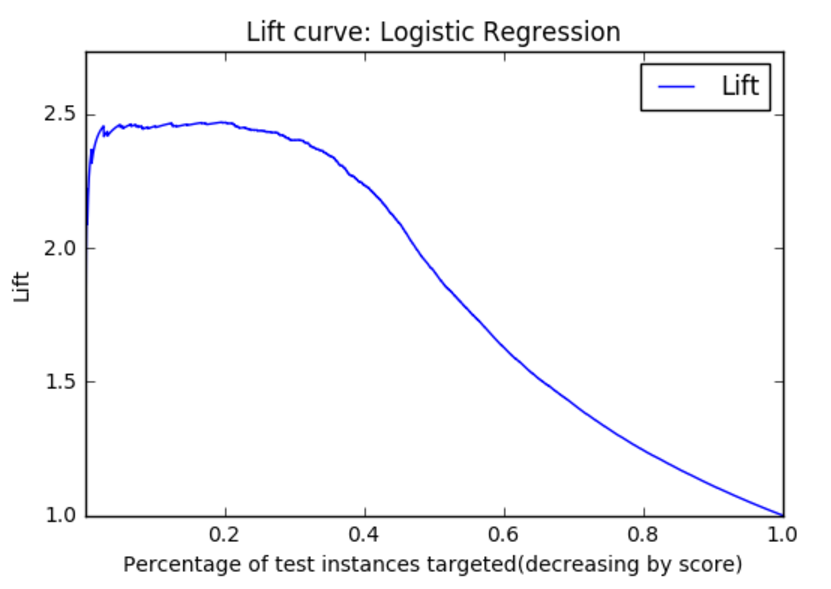
Below is the performance metrics of Logistic regression. As we can see that the average misclassification cost is quite higher than both Decision tree and k-NN.





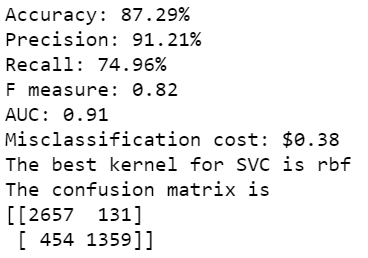
But the AUC of logistic regression is 0.96, which means that k-NN is making most number of the predictions correct, the only difference being the precision and recall measure. It has similar CRS and lift curves as that of other two models.



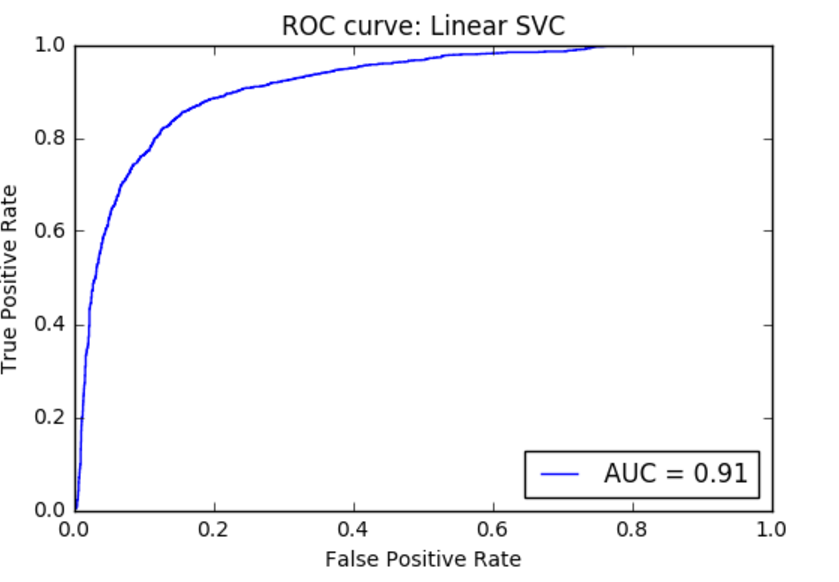


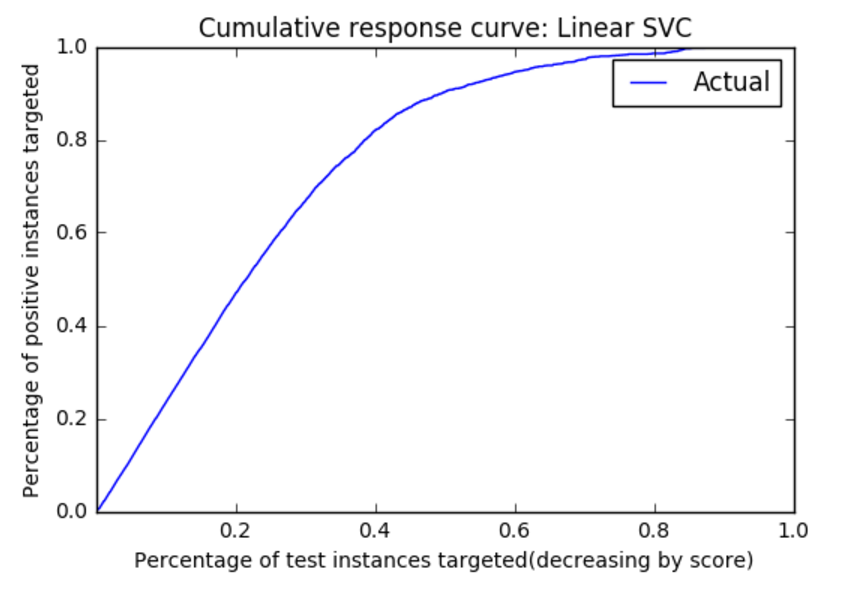
**SVM:**

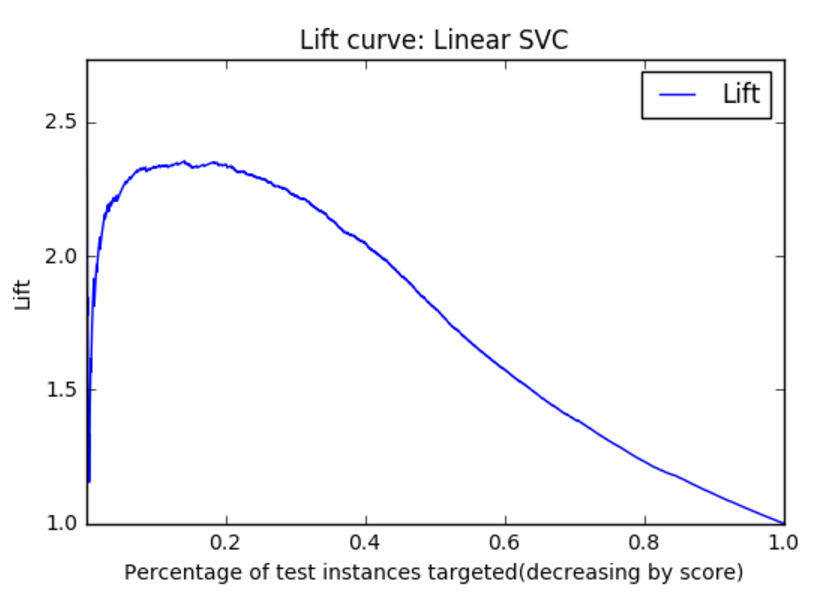
For SVC, I have tried with different classifiers and the best model came from Linear SVC. The parameters changed are penalty factor to penalize the misclassifications more.



The misclassification cost is in higher range.

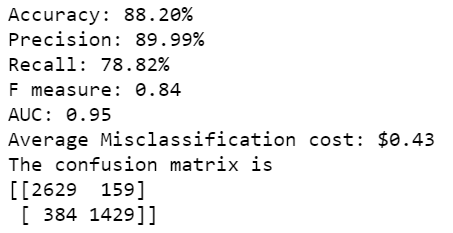


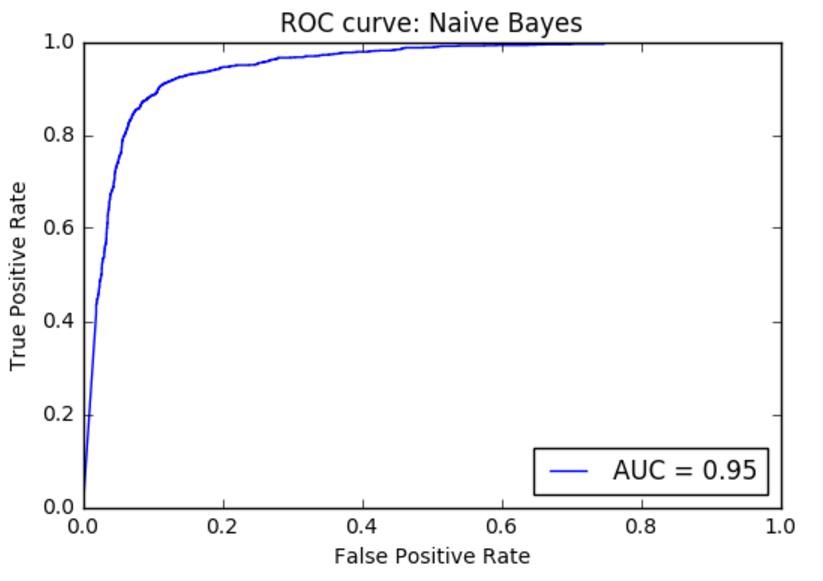


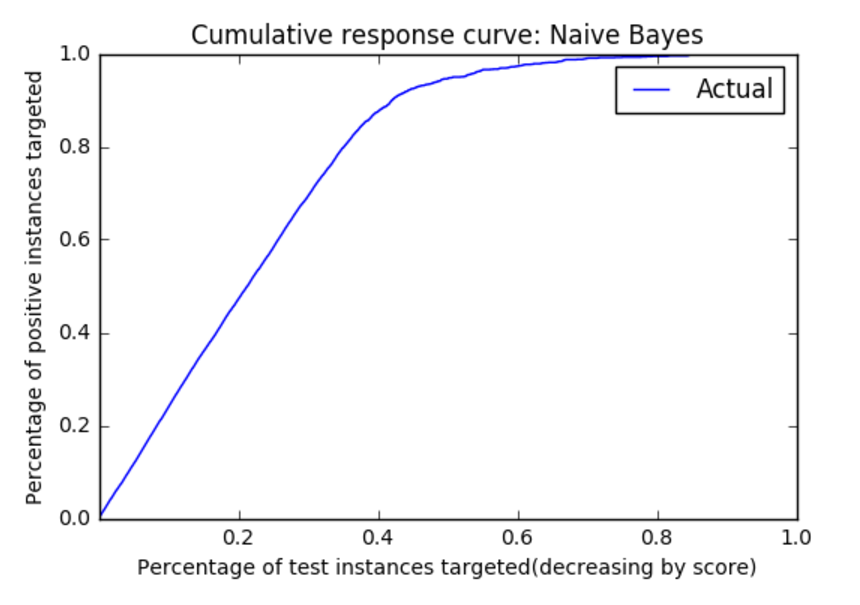


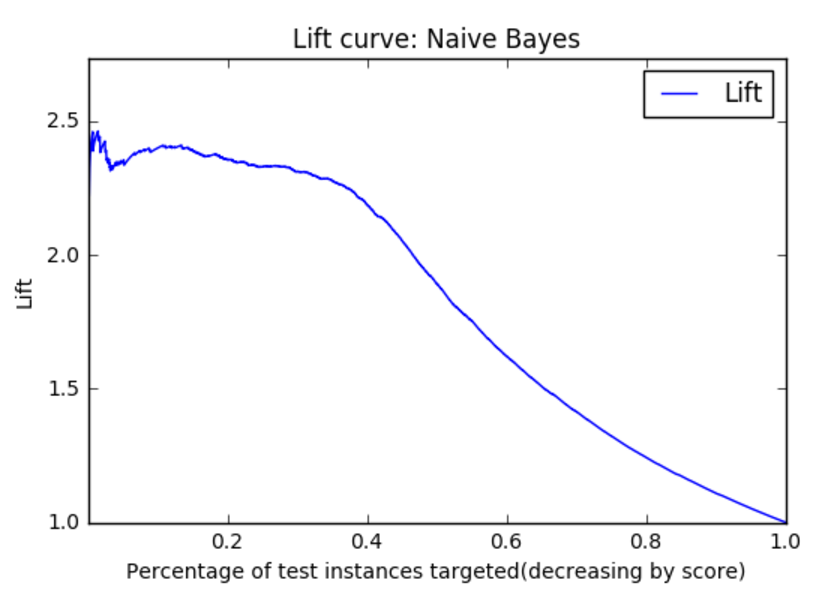
**Naïve Bayes**

Of all the models, Naïve Bayes has the highest average misclassification cost which is at $0.43. Below are the performance metrics of Naïve Bayes model.









**Conclusion:**

Hence, when it comes to average misclassification cost, k-NN performs better, whereas when it comes to accuracy decision tree performs better.