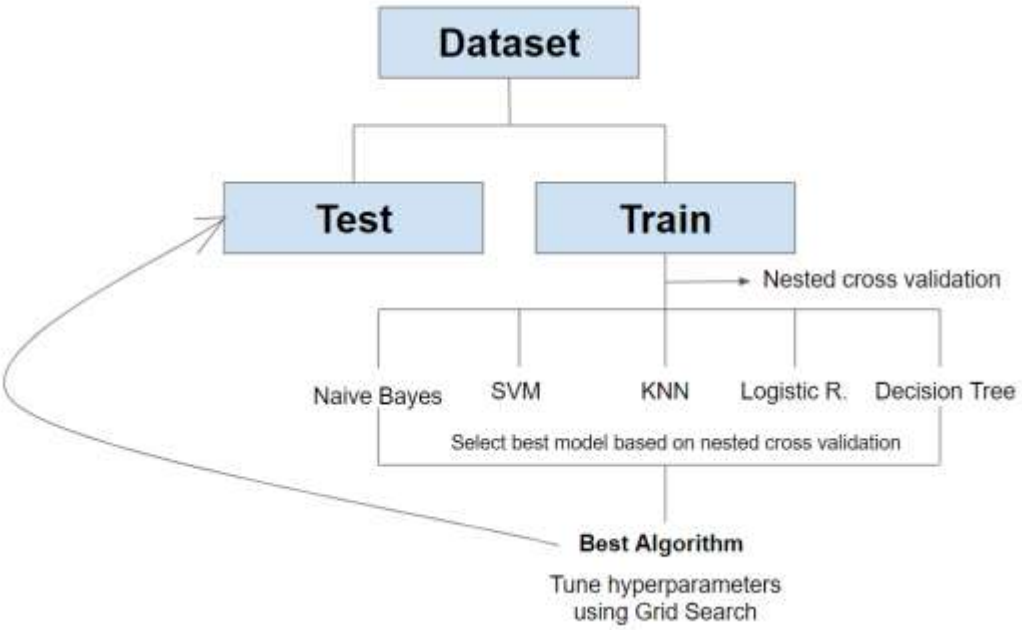
Perform a predictive modeling analysis on this same dataset (Problem 5 of HW1) using the decision tree, k-NN techniques, logistic regression and SVM (explore how well the model performs for several different hyper-parameter values). Present a brief overview of your predictive modeling process, explorations, and discuss your results. Make sure you present information about the model "goodness" (possible things to think about: confusion matrix, predictive accuracy, precision, recall, f-measure). Briefly discuss ROC and lift curves.

Process Flow



Data Transformation

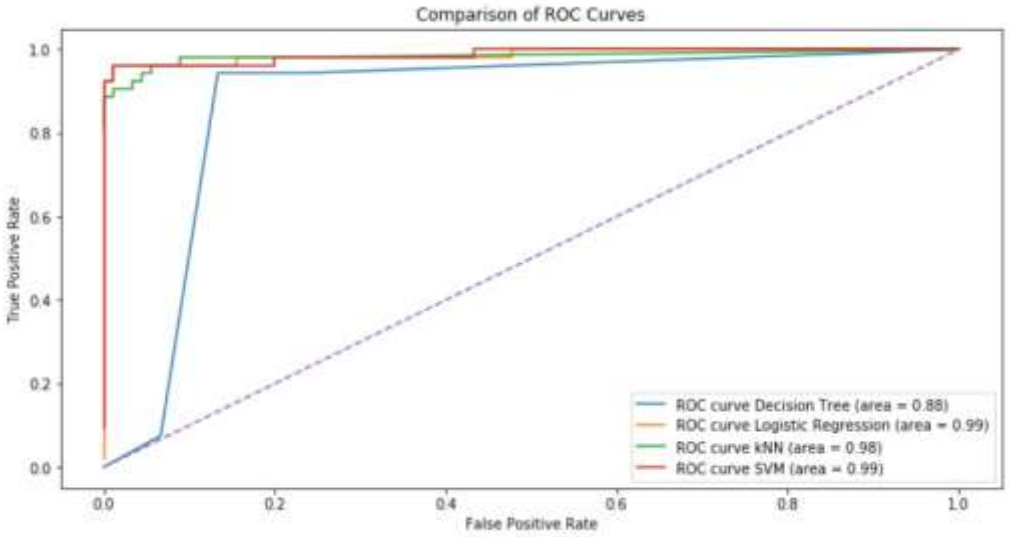
- Split the dataset into train and test split
- Normalise the train data and use the same transform function to normalise test set

Model Comparison

Nested Cross Validation Results : Mean/Standard deviation of Recall

Algorithm	Mean/Standard Deviation - Recall
KNN	91.91% / 0.6%
Decision Tree	85.73% / 2.38%
SVM	95.86% / 0.8%
Logistic Regression	96.33% / 0.49%

Based on the recall matric, Logistic Regression is the most robust algorithm here.

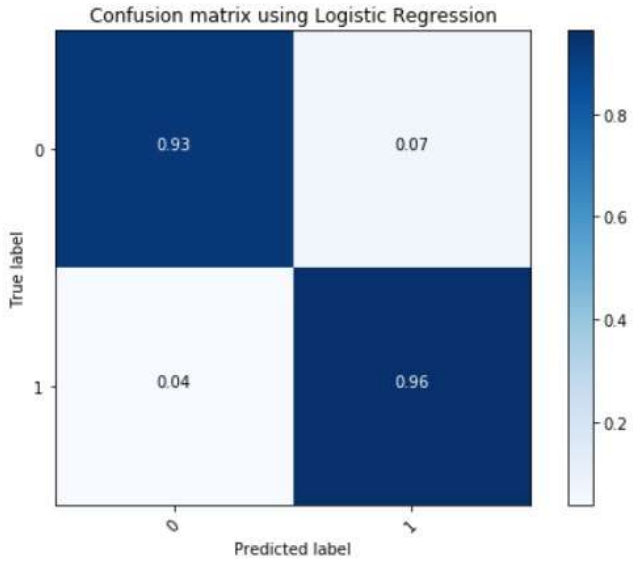


ROC curve also confirms that Logistic regression is the right algorithm here as area under the curve is 0.99

Grid Search for hyperparameters tuning in Logistic Regression :

Penalty : L1
C : 0.1

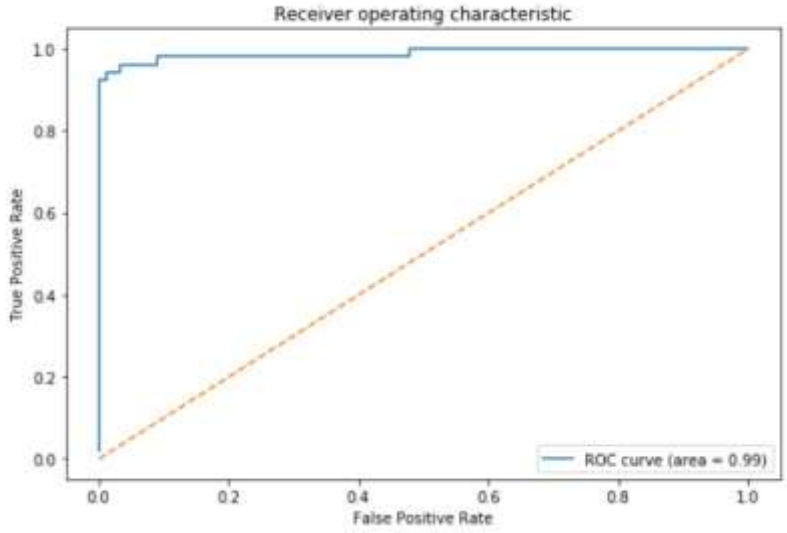
Predicting on Test Set :
Confusion Matrix :



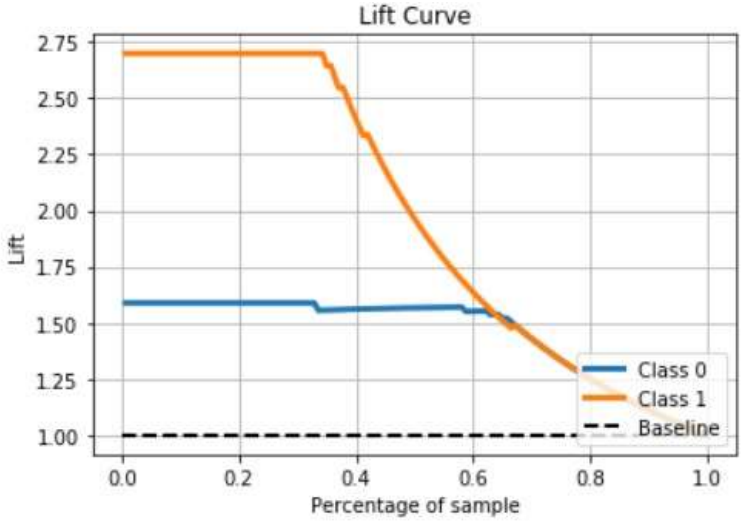
* Normalized confusion matrix in terms of percentages

Recall for 0 : 93%
Recall for 1 : 96%

Lift and ROC Curves for Logistic Regression



Area under ROC curve is 0.99 . This implies that the model is robust. Model is able to clearly distinguish between benign and malignant cases.



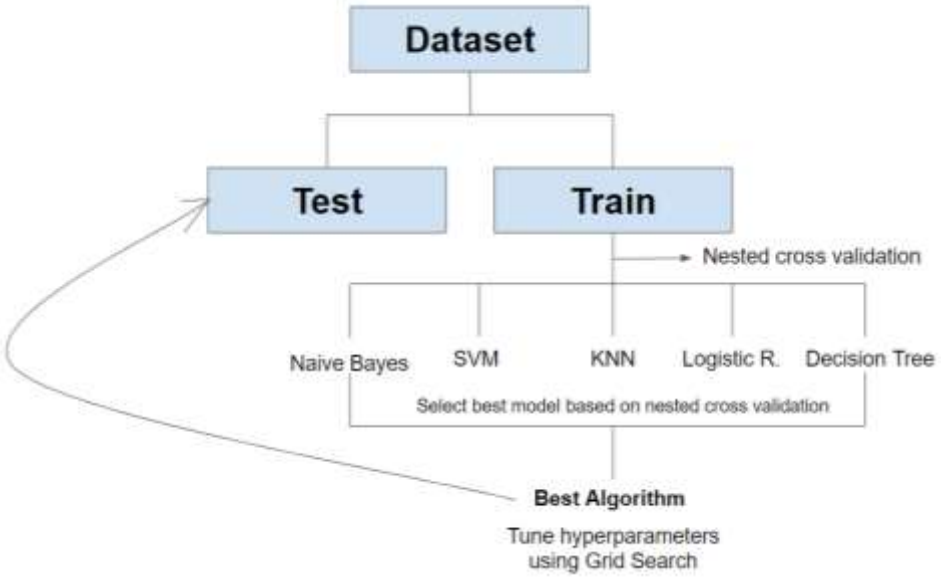
Orange line represents no of times of cancer cases which model would have predicted by chance. Lift is high and model performs well.

Among the basic classification techniques that you are familiar with (i.e., decision tree, k-NN, logistic regression, NB, SVM) use all that would be applicable to this dataset to predict the evaluation of the cars based on their characteristics. Explore how well these techniques perform for several different parameter values. Present a brief overview of your predictive modeling process, explorations, and discuss your results. Present your final model (i.e., the best predictive model that you were able to come up with), and discuss its performance in a comprehensive manner (overall accuracy; per-class performance, i.e., whether this model predicts all classes equally well, or if there some classes for which it does much better than others; etc.).

Data Transformation Steps

- Run models separately for categorical and numerical features separately
    - For categorical case, do one hot encoding by creating dummy variables
    - For numerical cases, encode each column to numeric value i.e. 1,2,3,4 etc.
- Split data in train and test set
- For numerical cases normalise the train set and use the same transform function to normalise test data

Process Flow



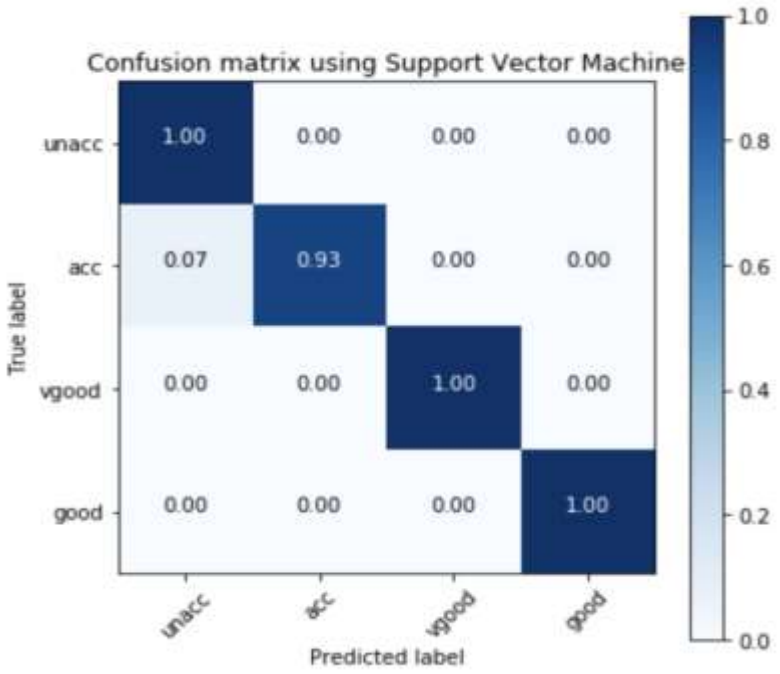
Nested Cross Validation Results : Mean/Standard deviation of accuracy

Algorithm	Categorical	Numeric
KNN	90.40% / 0.70%	95.94% / 0.50%
Decision Tree	96.76% / 0.50%	97.51% / 0.50%
Naive Bayes	85.72% / 0.52%	76.02% / 0.31%
Logistic Regression	93.77% / 0.40%	84.78% / 0.40%
SVM	99.34% / 0.32%	91.70% / 0.28%

Model Selected: SVM with treating the data as Categorical gives the best accuracy and thus choosing this model to finally train our model and testing it on the test data

Hyper parameters: 'C': 1000, 'degree': 3, 'kernel': 'poly'

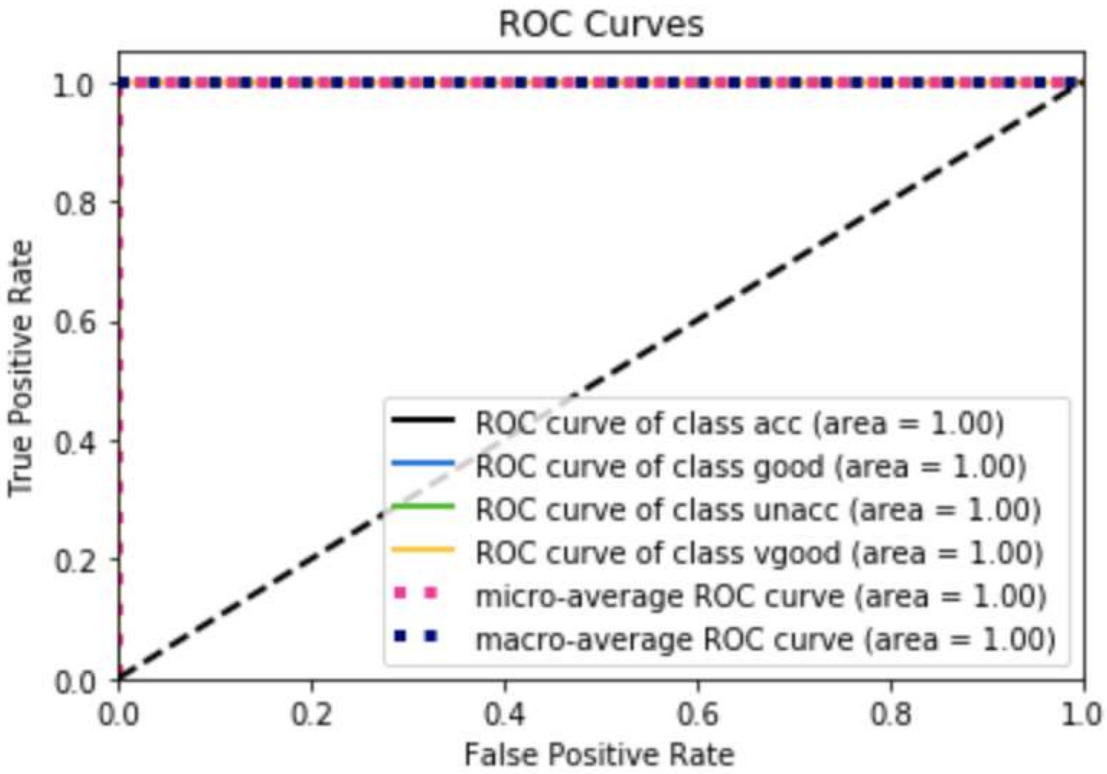
Confusion matrix:



From the below summary of model, we can see that the model performs equally well on both the testing and training data.

```
Best score is 99.64%
Best parameters are {'C': 1000, 'degree': 3, 'kernel': 'poly'}

Prediction Accuracy: 99.71%

             precision    recall  f1-score   support

        acc       0.99      1.00      0.99        77
       good       1.00      0.93      0.96        14
      unacc       1.00      1.00      1.00       242
      vgood       1.00      1.00      1.00        13

avg / total       1.00      1.00      1.00       346
```

- The SVM model with categorical data gives an overall accuracy of 99.71%
- We can see from the confusion matrix, that the model is predicting 3 out of 4 classes with 100% accuracy and one class with 93% accuracy. Thus overall accuracy comes out to be 99.71%
- The area under ROC curve is 1 showing that the model is robust predicting all classes perfectly



Comparison between Categorical and Numerical:

- SVM Categorical model shows 99.71% accuracy whereas SVM continuous is around 91.7%
- SVM continuous is computationally heavier than SVM categorical

- While labeling attributes as numbers, numerical distances between the labels is assumed to be the same whereas when we treat them as categorical, we don’t assume any distances between the categories
- While treating as categorical data, we lose the ordinal nature of our variables