Traffic Violations Analysis

* **ALGORITHMS**

The traffic violations data sample consists of categorical data. Therefore, for processing and analysis purposes, classification algorithms have been considered.

Following are the five algorithms implemented to study and predict patterns:

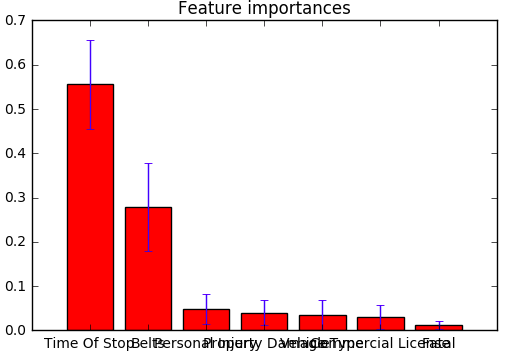
1. Multinomial Logistic Regression
2. Decision Tree
3. Random Forest
4. Extra Tree Classification
5. Support Vector Machine

* **Data Sample**

The data sample used for this checkpoint consists of 28,959 records and 18 attributes.

Based on p- value and near- zero- variance, the following features have been taken into consideration for all the above algorithms:

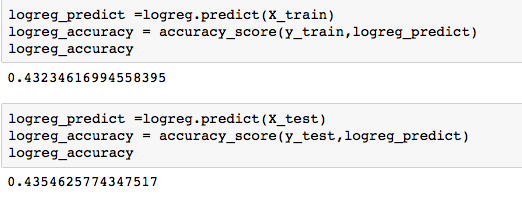
* Time of Stop
* Belts
* Personal Injury
* Property Damage
* Vehicle Type
* Commercial License
* Fatal



The training set has 19,112 records and test set has 9,847 records.

* **Multinomial Logistic Regression**

Multinomial Logistic Regression (MLR) is considered as modelling algorithm as our dependent variable is nominal with three levels. It is used to describe data and to explain the relationship between dependent nominal variable and sets of independent variables. Here response variable Damage Class values – Class 1, Class 2, Class 3, are set as reference point.

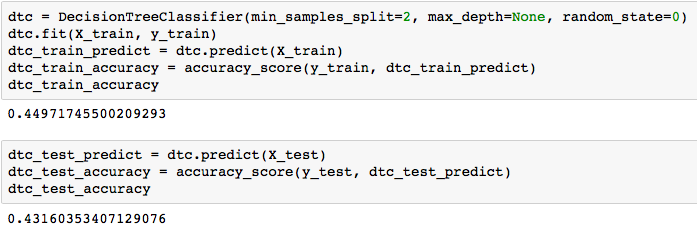


Analysis:

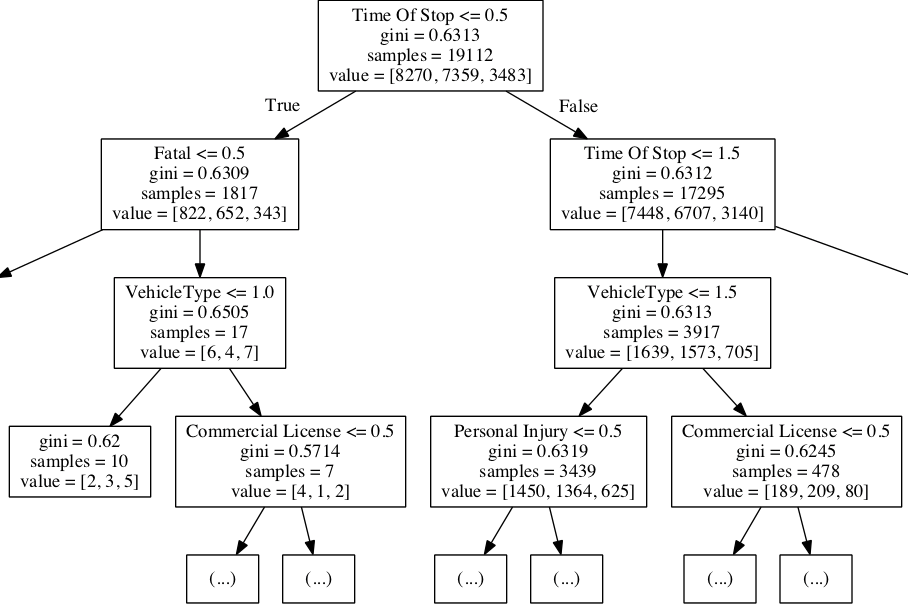
The accuracy for training data is 43.2% whereas for test data it is 43.5%. There is a marginal difference between the two accuracies, and both the values are significantly low.

* **Decision Tree**

It a non-parametric supervised learning method used for classification. The goal is to create a model that predicts the value of a damage class by learning simple decision rules inferred from the data features.



Tree Diagram for Decision Tree:

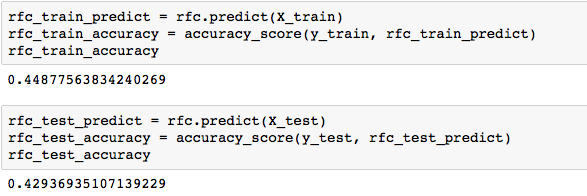


Analysis:

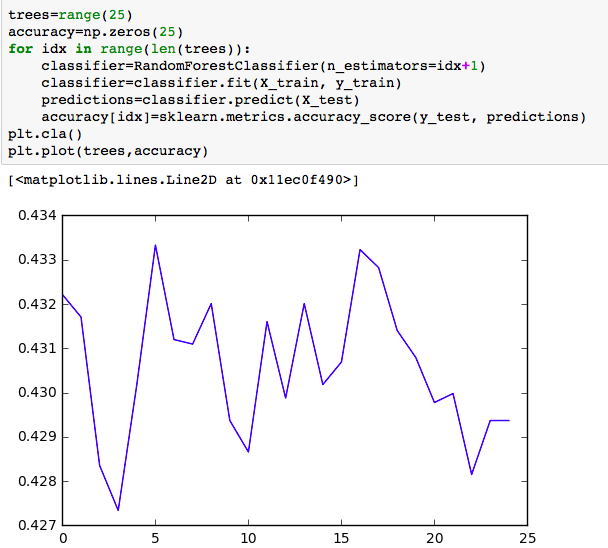
The accuracy for training data is 44.9% whereas for test data it is 43.1%. The conditional Inference tree is based on p-values and gini. It also shows that Time of Stop is the most significant attribute, followed by Fatal and Vehicle Type.

* **Random Forest**

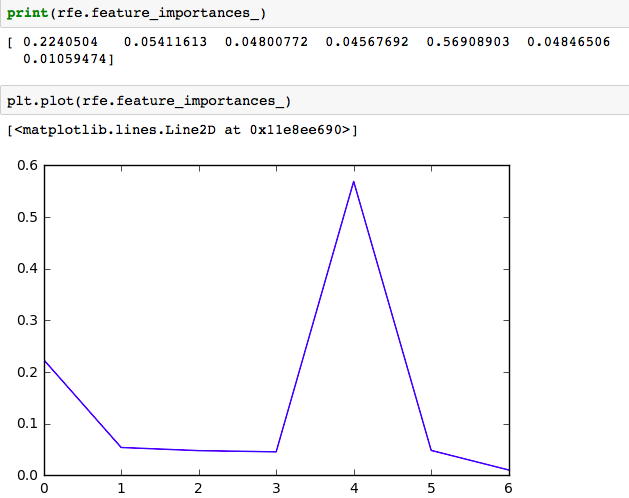
Random forests are an ensemble learning method that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification). Random decision forests correct for decision trees' habit of [overfitting](https://en.wikipedia.org/wiki/Overfitting) to their training set.



Plot between random forest trees and their accuracies:



Plot to see the importance of each feature in random forest:

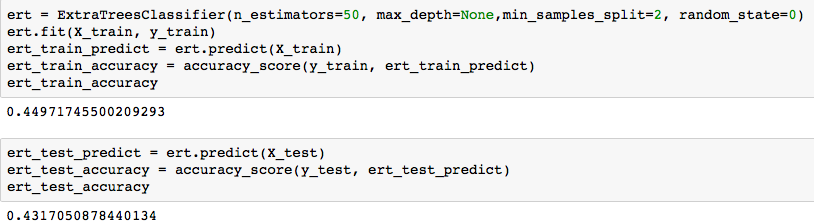


Analysis:

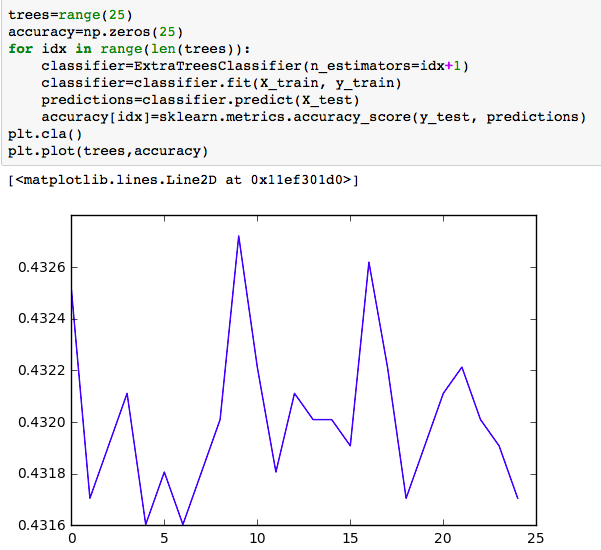
The accuracy for training data is 44.8% whereas for test data it is 42.9%. Also, if 25 trees are taken into consideration, maximum accuracy can be achieved when the number of trees is around 5 or 16 with Time of Stop contributing the most.

* **Extra Tree Classification**

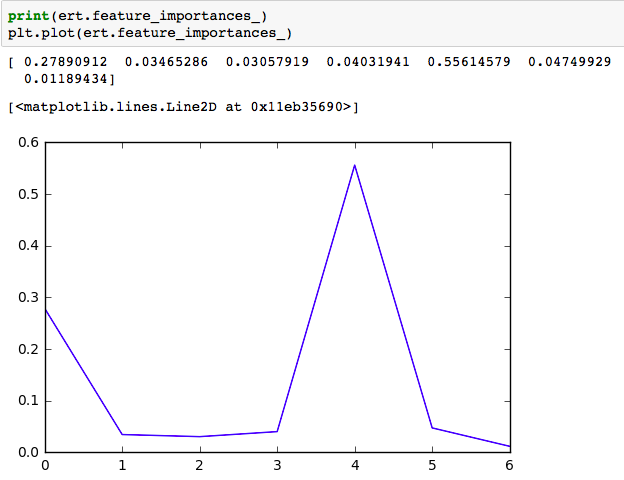
It is a variant of random forest. Unlike a random forest, at each step the entire sample is used and decision boundaries are picked at random, rather than the best one.



Plot between extra trees and their accuracies:



Plot to see the importance of each feature in extra tree:

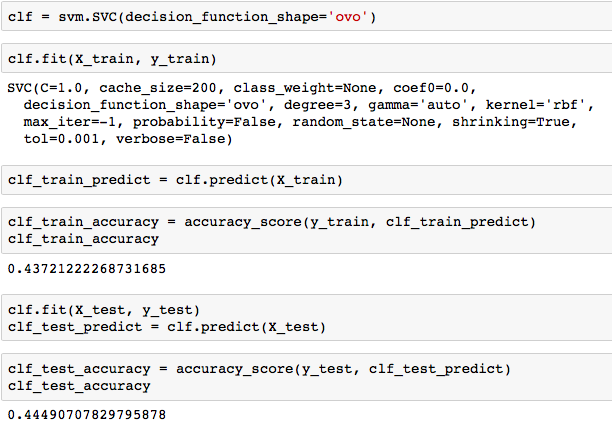


Analysis:

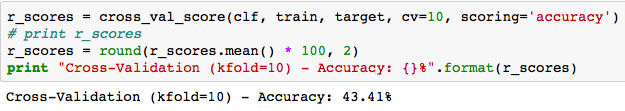
The accuracy for training data is 44.9% whereas for test data it is 43.1%. Also, if 25 trees are taken into consideration, maximum accuracy can be achieved when the number of trees is around 8 with Time of Stop contributing the most.

* **Support Vector Machine – Core Algorithm**

SVM is a discriminative classifier formally defined by a separating hyperplane. Given labeled training data, the algorithm outputs an optimal hyperplane which categorizes new examples.



**SVM with Cross- Validation:**



Analysis:

The accuracy for training data is 43.7% whereas for test data it is 44.4%. Although the accuracy for test data is much better in case of SVM compared to other models, we cannot consider this accuracy to be accurate. To confirm the results of SVM, we apply cross validation in order to thoroughly verify the output accuracy. Cross validation assures an accuracy of 43.4%.

Based on the results obtained, we will be considering **SVM as our core algorithm** for this project.

* **FINE TUNING OF SVM and its PERFORMANCE**

It is observed that SVM gives best results for Traffic Violations dataset. However, the result into consideration is based on default values for all SVM parameter.

The accuracy can be improved by fine tuning these parameters. To implement this tuning, a Grid Search can be applied.

**Grid Search** runs the algorithm with all the specified values of the parameters taken into consideration one by one. This provides us with a set of combinations of parameters that can be considered to get maximum accuracy.

**Fine Tuning of SVM over Training and Test Datasets**

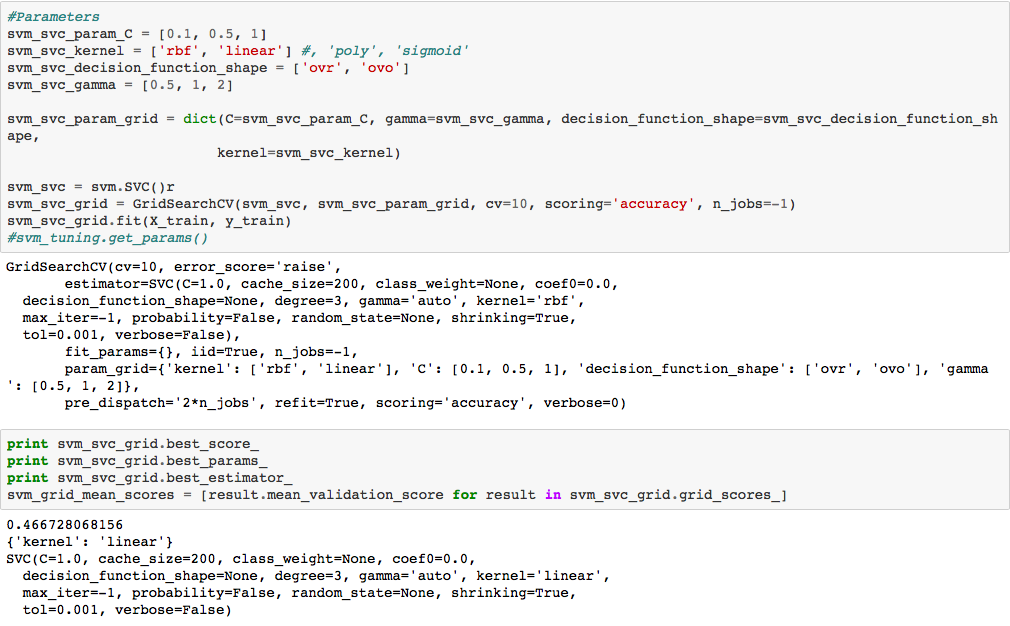
In this case, the Grid Search considers following values:

Cost, C : [0.1, 0.5, 1]

Kernel : [‘rbf’, ‘linear’]

Decision Function : [‘ovo’, ‘ovr’]

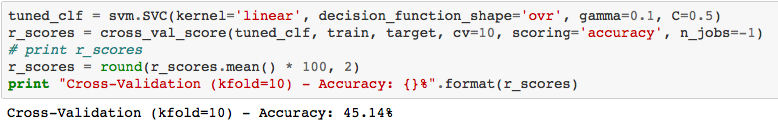
Gamma : [0.5, 1, 2]



Analysis:

By implementing Grid Search for the above mentioned parameters on training and test sets, we can increase the accuracy for SVM model up to 46.6% from 43.4%. There is an increase of 3.2%. It is a significant increase compared to the other models.

**Fine tuning of SVM over Complete Dataset**



Analysis:

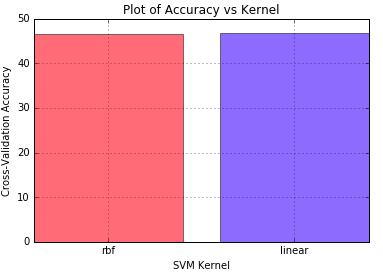
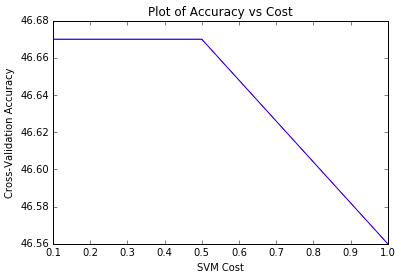
If the finely tuned SMV is used to process the whole dataset (with linear kernel, decision function as One vs Rest, gamma=0.1 and cost=0.5) against Damage Classes, the accuracy still increases by 1.74%.

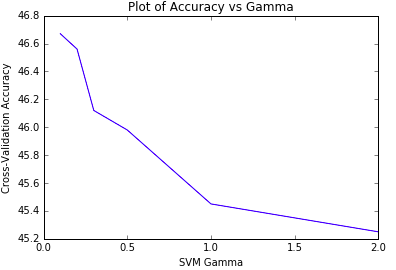
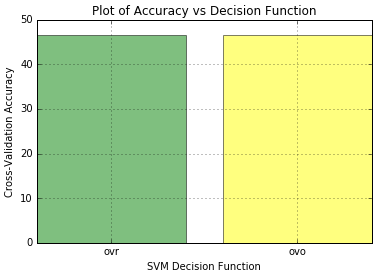
**Tuning of Parameters:**

Mainly four SVM parameters have been fine tuned to improve the performance.

* Cost
* Kernel
* Decision Function
* Gamma

Their impact on Accuracy can be seen in the graphs below.



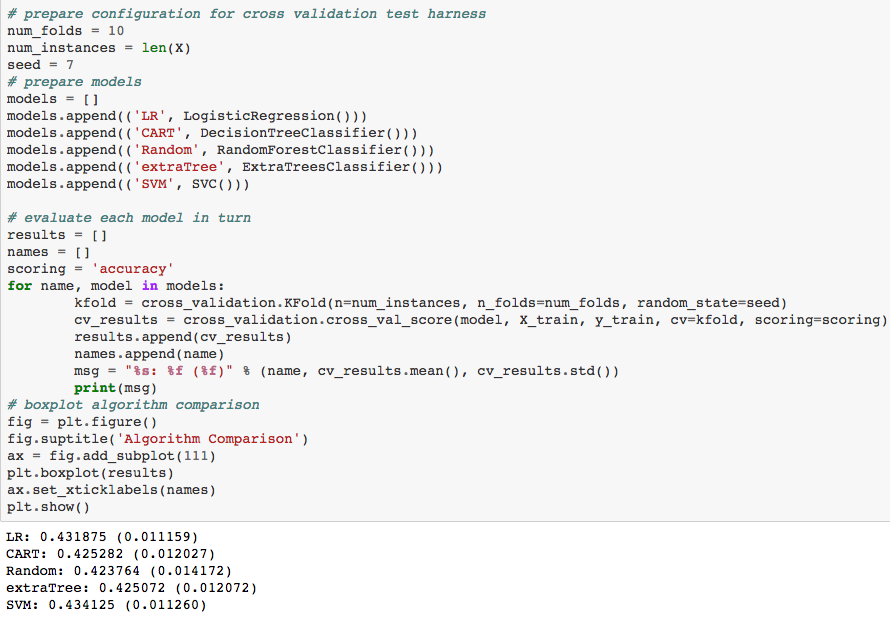


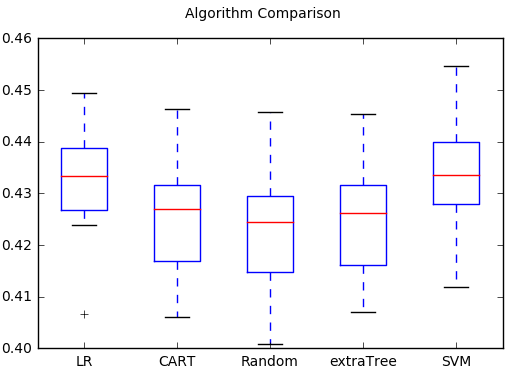
Kernel and Decision Function bar graphs do not show any prominent differences in the values as the difference in accuracies is very minor (in points).

* **COMPARISON OF ALL THE ALGORITHMS**

Two techniques have been used to compare the five implemented algorithms on the basis of accuracy:

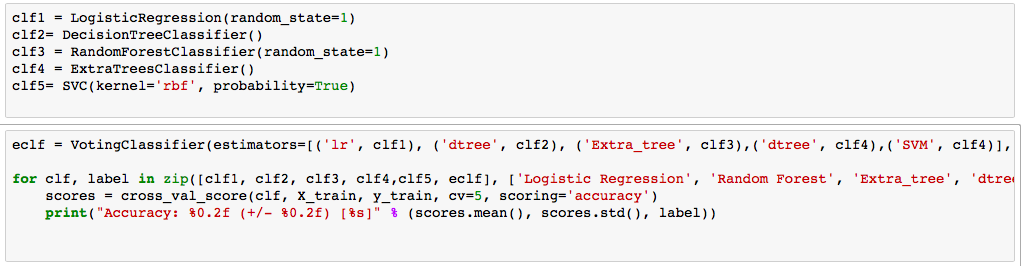
* Boxplot
* Voting
* Boxplot

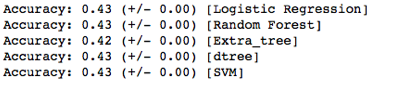
****

****

* **Voting**

Voting classifier implementation is to combine conceptually different machine learning classifiers and use a majority vote or the average predicted probabilities (soft vote) to predict the class labels. Such a classifier can be useful for a set of equally well performing model in order to balance out their individual weaknesses.



****

|  |  |  |
| --- | --- | --- |
| ALGORITHMS | Training Accuracy (%) | Test Accuracy (%) |
| Multinomial Logistic Regression | **43.2** | **43.5** |
| Decision Tree | **44.9** | **43.1** |
| Random Forest | **44.8** | **42.9** |
| Extra Tree | **44.9** | **43.1** |
| SVM | **43.7** | **44.4** |
| SVM Cross Validation | **43.4** | |
| SVM Tuned (Train-Test Samples) | **46.6** | |
| SVM Tuned (Complete Dataset) | **45.1** | |

Conclusion:

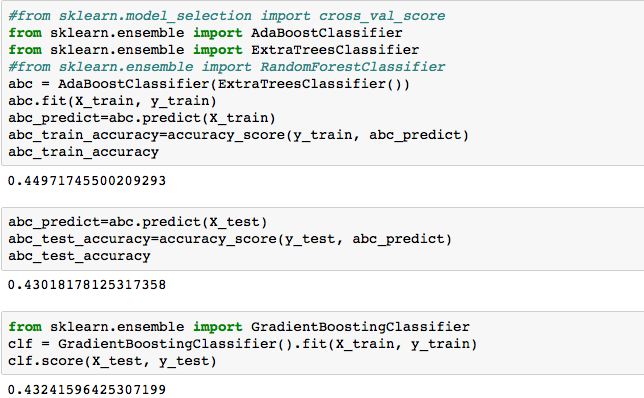
Due to the categorical nature of the data present for Traffic Violations, it is difficult to predict a higher accuracy for the model. However, it can be seen that the accuracies obtained by all the models are very close to each other. This shows that the data actually consists of very few attributes that contribute to the correct prediction of Damage Class. Even after fine tuning the parameters, the accuracy increases by a margin of 3%. It is not a significantly large percentage but holds importance in this case where accuracy of various models differs in points.

**ADDITIONAL WORK:**

We have also implemented two meta- algorithms – Bagging and Boosting. These approaches help improve the predictive force by decreasing the variance (bagging) along with bias (boosting).

1. **Boosting**

It is a two step process where first step uses subsets to produce a series of averagely performing models and then boosts the performance by combining them together using a cost function.



Analysis:

We have considered AdaBoost (Adaptive Boost) and Gradient Boost classifiers to improve the performance. The accuracy for training data is 44.9% whereas for test data it is 43.2%. This shows a slight improvement in the accuracy of the model.

1. **Bagging (Bootstrap Aggregation)**

It decreases the variance by adding additional data for training from the original dataset using combinations with repetitions.



Analysis:

We have considered Bagging classifier to improve the performance. The accuracy for training data is 44.8% whereas for test data it is 42.9%. These are very similar to the figures we obtained from boosting.