EE 660
Fall 2017
Final Project : Online News Popularity

Dixith Reddy Gomari
USC ID:3098766483
gomari@usc.edu

University of Southern California

Abstract

One of the amazing phenomena which is reshaping and changing the perspective of the world is Social Media. With rapid growth of the internet, popularity is what all internet marketers buzz over and over. Predicting and evaluating how popular an online news could provide key information about how marketers should promote their product to reach out to many people. Determining if there are well-defined features in posts that accurately predict a posts popularity is both an interesting and a useful endeavor. One measure to evaluate the popularity is the number of shares an online news gets. Here, I have taken the data from Mashable website, a popular digital media website which provides online news regarding various things. The dataset is provided by UCI machine learning repository, originally acquired and processed by K.Fernandes et al. It is a multivariate dataset with 39797 instances and 61 attributes. I have done Classification, Regression and Clustering analysis on the data. I have used classifiers based on supervised learning which belong to both distribution free and statistical classification. The classifiers I have used in this problem is SVM, Decision Trees, Random forest Classifier and Logistic Regression. For regression analysis I have just used linear regression and explained why the regression analysis is a poor fit for the data. As for clustering I have used K-Means clustering for the improvement of the model. Appropriate pre-processing techniques have been explained in the discussion of the problem solving.

Introduction

In this era, it is important to attain people's attention to gain popularity to promote any news or product. Predicting the popularity of an online news becomes essential for marketers to make strategies about how to promote their news and products. The current data is taken from mashable with over 39000 articles. The main purpose of this project is to analyze and compare the performance of several machine learning algorithms.

Even though the predictive results in the data "shares" is a regression problem, it is converted to a classification problem by putting a threshold on the number of shares by dividing into desired number of classes. Since the number of features are very high, it becomes an essential part to reduce the dimensions of the samples. All available features can be summed up into the given table and a little snippet of the data is shown.

Aspects	Features
Words	Number of words of the title/content; Average word length; Rate of unique/non-stop words of contents
Links	Number of links; Number of links to other articles in Mashable
Digital Media	Number of images/videos
Publication Time	Day of the week/weekend
Keywords	Number of keywords; Worst/best/average keywords (#shares); Article category
NLP	Closeness to five LDA topics; Title/Text polarity/subjectivity; Rate and polarity of positive/negative words; Absolute subjectivity/polarity level
Target	Number of shares at Mashable

	url	timedelta	n_tokens_title	n_tokens_content	n_unique_tokens	n_non_stop_words	n_non_sto
0	http://mashable.com/2013/01/07/amazon-instant-...	731.0	12.0	219.0	0.663594	1.0	
1	http://mashable.com/2013/01/07/ap-samsung-spon...	731.0	9.0	255.0	0.604743	1.0	
2	http://mashable.com/2013/01/07/apple-40-billio...	731.0	9.0	211.0	0.575130	1.0	
3	http://mashable.com/2013/01/07/astronaut-	731.0	9.0	531.0	0.503788	1.0	

Prior and related work - None

Project formulation and setup:

Regression:

Linear Regression:

Linear regression is a statistical method that helps us summarize the relationship between two continuous variables (x and y). It represents a least square fit of the response of the data. Linear models are used to determine the relation between observations y and independent variables x. The hypothesis is given by:

$$\mathbf{y} = X\boldsymbol{\beta} + \boldsymbol{\varepsilon}$$

Where y is the response variable, X is the input variable, β is the dimensional parameter vector and ε is the error term.

Kernel Ridge Regression:

In order to operate at higher dimensions, we need to replace x with $\phi(x)$. The solution can be given by $\beta = (\Phi^T \Phi + \lambda I)^{-1} y$. Where λ is the regularizer term, which needs to be optimized for the optimal fit of the data.

Classification:

Logistic Regression:

It is a regression model where the dependent variable is categorical. For a logistic regression model, the hypothesis is

$$h_\theta(x) = g(\theta^T x) = \frac{1}{1 + e^{-\theta^T x}}$$

Where the parameters are chosen to maximize their likelihood

$$\prod_{i=1}^{m} p(y^{(i)} | x^{(i)}; \theta)$$

Support Vector Machine:

I have used both linear and nonlinear kernels to make predictions, we can formulate the predictions with the following:

$$\begin{aligned} w^T x + b &= \left(\sum_{i=1}^{m} \alpha_i y^{(i)} x^{(i)} \right)^T x + b \\ &= \sum_{i=1}^{m} \alpha_i y^{(i)} \left\langle x^{(i)}, x \right\rangle + b \end{aligned}$$

The kernel can be replaced by higher order complex ones like polynomial or gaussian.

Decision tree classifier:

The goal of building a decision tree is to build a tree of depth h that achieves minimum misclassification errors on the training data. A trivial solution could be to create a decision tree with one path from root to leaf for each training example, but it would not generalize to new data points. Other way of doing it is to find the smallest tree h that minimizes error.

Random forest classifier:

To improve upon the result of the decision trees, I have used random forest classifier. They use multiple decision trees which are built on separate sets of examples drawn from the dataset. By using more number of decision trees and averaging the result, the variance of the model can be greatly lowered.

Adaptive boosting classifier:

AdaBoost refers to a method of training a boosted classifier. The sign of weak learner output identifies the predicted object class and absolute value gives the confidence in that classification.

Clustering Analysis:

K-Means Clustering:

I have used clustering algorithm as the prediction values are spread over a vast range, we could make a good use of the clustering analysis. K means is an unsupervised learning technique where the data is grouped into clusters based on the location on the dimensionality space. This could be used to find the cluster centers in order to predict the labels of the data.

Implementation and Methodology:

Preprocessing:

Preprocessing wasn't difficult on this data,as it was already preprocessed by K.Fernandes et al.[1] and uploaded onto UCI machine learning repository. The original data was categorical but K.Fernandes et al.[1] have encoded the data before hand. There were no missing or NULL data. I have performed normalization and scaling, but there was no significant difference in the performance of the models.

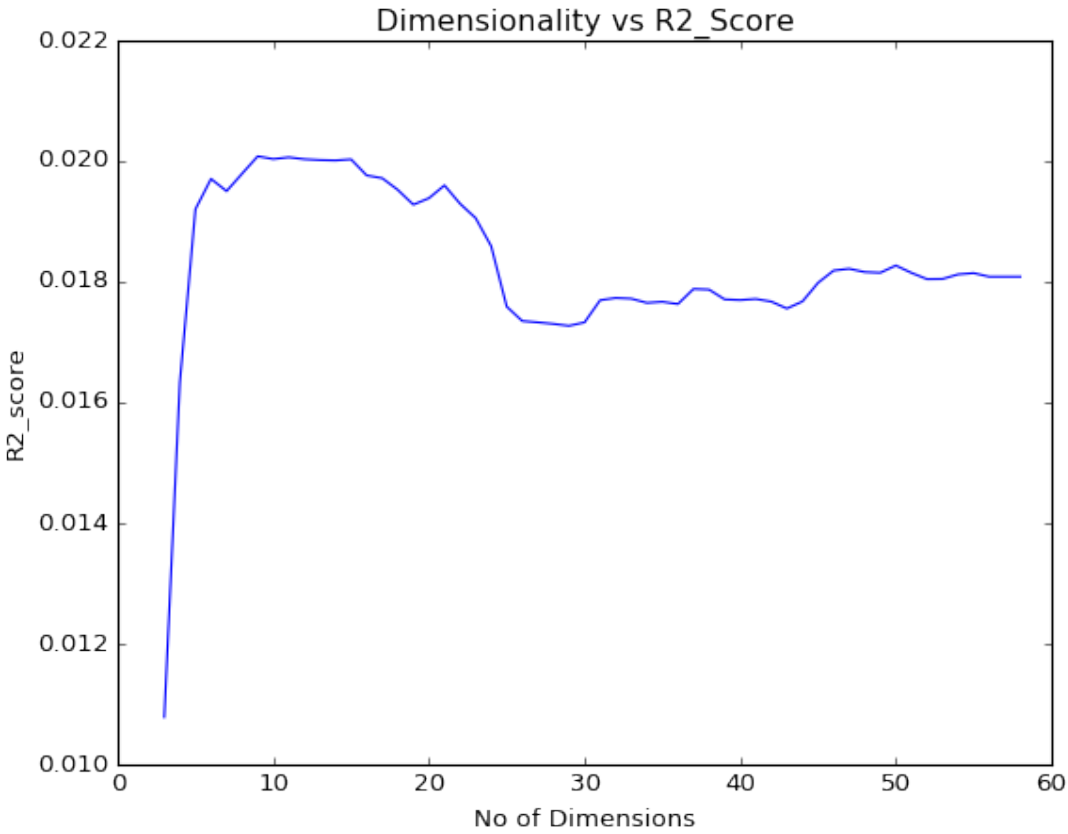
Feature Selection and Dimensionality reduction:

PCA:

A full feature may include unwanted noise. I have attempted PCA for the dimensionality reduction. It did not provide any improvements to the algorithms I have used. Even though PCA is a commonly used dimensionality reduction technique, the results only made the models perform worse. This is because the original feature set has very limited correlated information.

LDA:

LDA is closely related to PCA, where LDA considers the classes of the data, while PCA on the other hand does not take any difference in class into account. I have used cross validation on linear regression in order to select the number of features I should use for further processing.



Based on this result, we can see that a range between 10 and 15 gives good results. For a better fit, I chose 15 dimensions to perform further processing of the data.

Train Test Split:

I split the data into train and test data with 70-30 proportion of the total respectively.

Regression Analysis:

Linear Regression:

Since it is a regression problem, I have started with a simplest regression model to start with. Due to high variance of the target variable, application of linear regression performs very poorly. The results also suggests that. They are as shown:

```
Average Cross-validation R2-Score: 0.026203997583
R2-Score of the test set: 0.0179487794688
```

Since it was performing poorly I opted to perform regression for higher order kernels with regularizing terms.

Kernel Ridge Regression:

R2 score was much worse for higher dimensional kernels for kernel ridge regression. The cross validation results for the respective kernels were like following:

Degree of the kernel	Average CV R2 score
2	-0.03363
3	-0.03736
4	-0.39975
5	-2.00594
6	-142.262

The results clearly indicate that regression performs very poorly for the given data as negative R2 score indicates worse fit over data. Negative R2 score suggests that the chosen model performs much worse than a horizontal line(null hypothesis).

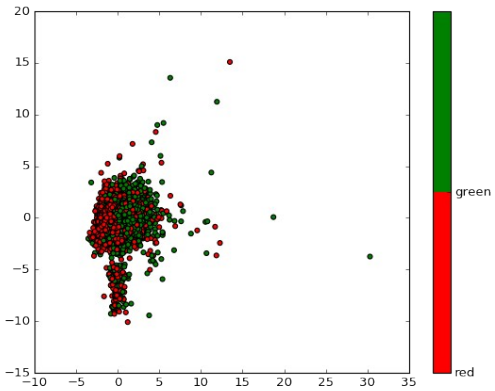
Classification analysis:

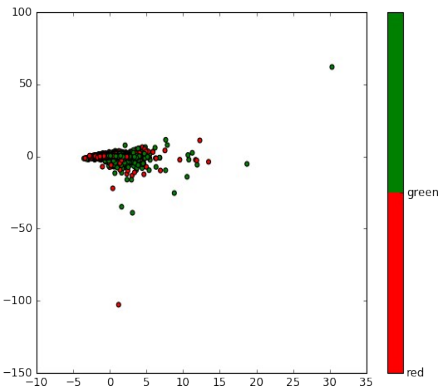
Division of classes:

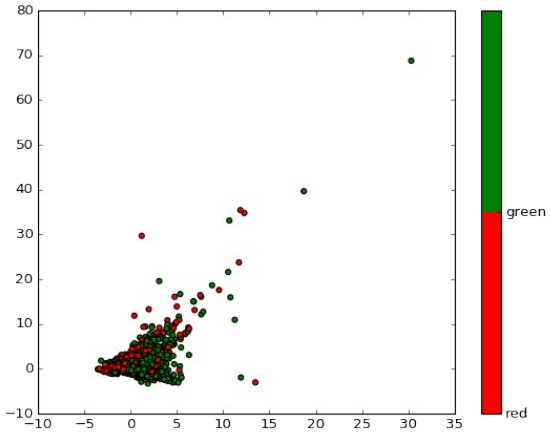
I have started out with a 2 class problem with 1400 as threshold, where regression values falling under 1400 belong to class 0 and values with greater than 1400 belong to class 1. The reason for choosing 1400 samples as a threshold is because out of over 27000 training samples, there is an equal division of samples with such a threshold.

	0-1400 shares	>1400 shares
Class label	0	1

A 2 d representation between 2 random features can be seen below:







Logistic Regression:

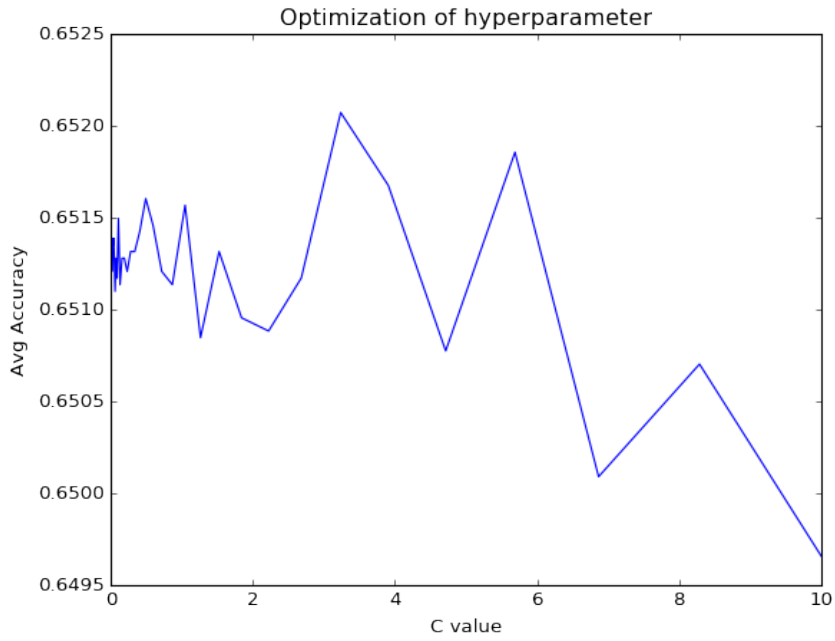
I have used classification models to improve the accuracy. I tried to optimize over the parameters over the model with cross validation error. The parameters do not change the accuracy drastically. For multinomial classification the model uses logarithmic function to estimate the relative probability of each category with reference to the kth category.For a 2 class problem the average cross validation accuracy and test accuracy is as shown below:

For K=2 Classes

Cross validation accuracy	64.207%
Testing accuracy	63.855%

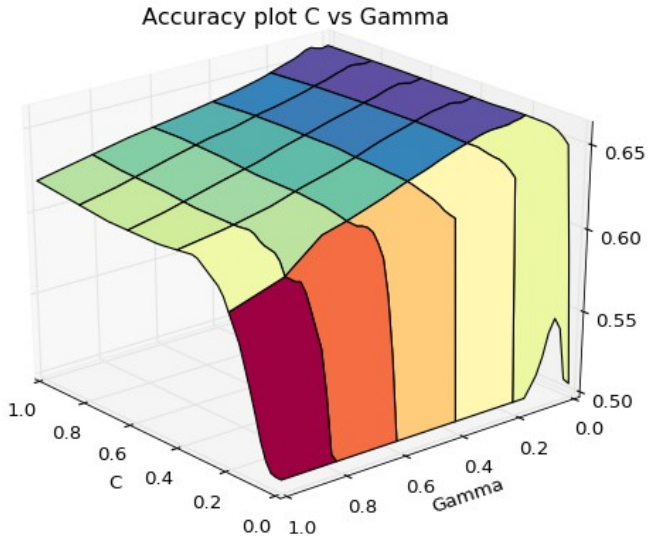
Support Vector Machine:

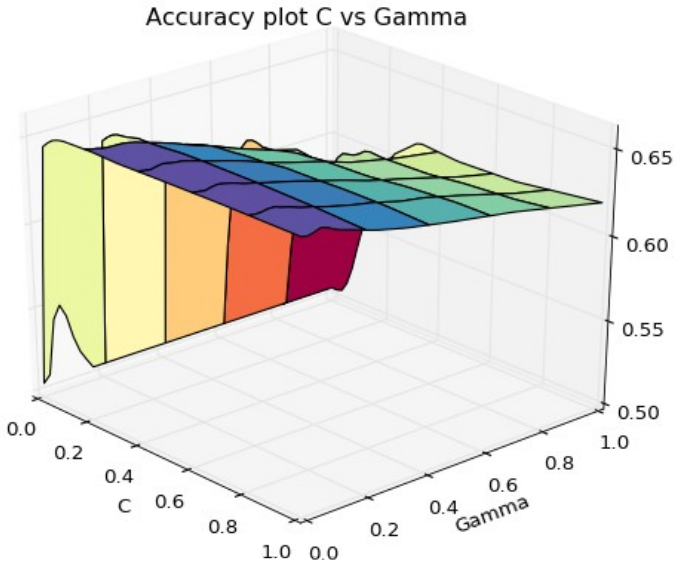
I started using SVM with a linear kernel and followed up with higher dimensional kernel. Since linear kernel has a high bias problem when compared to higher dimensional kernel, I have tried even the higher dimensional kernel. For a linear kernel, I have optimized over slack variable C. The distribution of cross validation accuracy over logarithmic spacing of slack variable is shown below. I have optimized over a range of 0.001 to 10 of slack variable C.



Optimal C value is 2.4770
The test accuracy for optimal C value is 65.15%

For higher dimensional kernel:
I have optimized over the slack variable and gamma variable over a range of logarithmic space of 0 and 1. I have 5 fold cross validation over the space of values. The cross validation accuracy of samples over the grid of values of the C and gamma is given below:

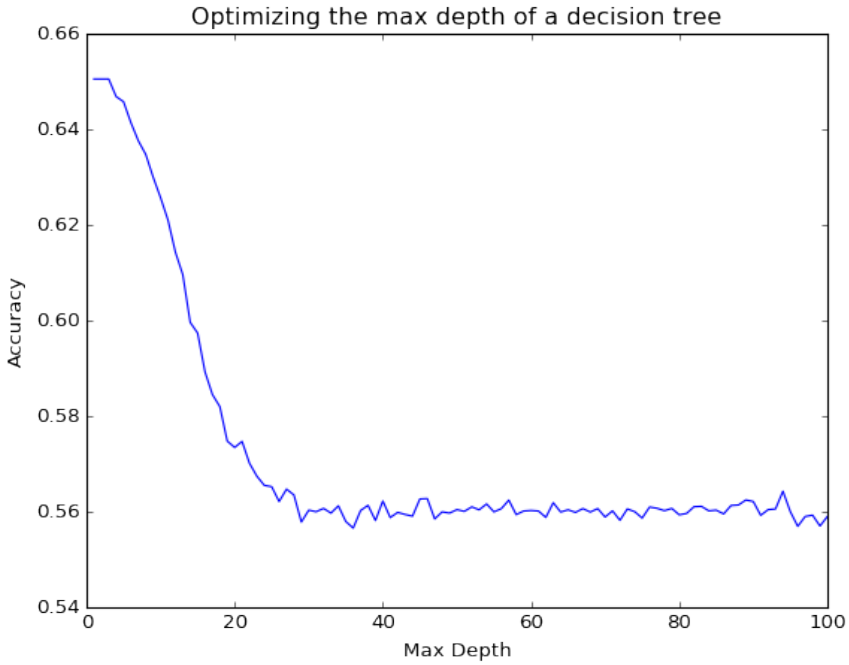




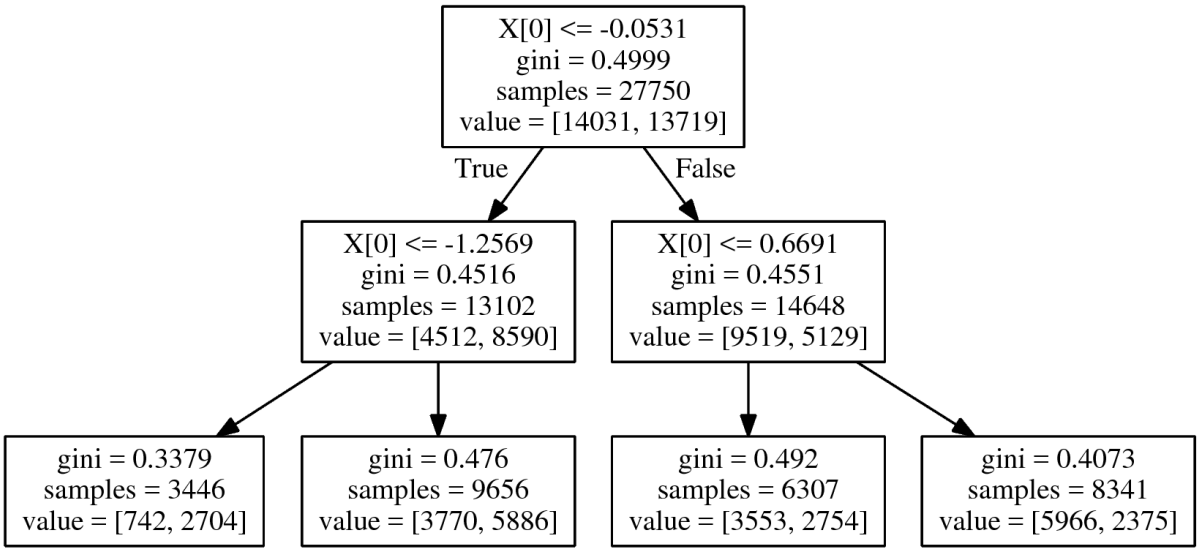
Optimum value of C is 1.0
And gamma is 0.0545559
Test accuracy for the optimal values is 64.66%

Decision tree classifier:

Depth of the decision tree affects the classification accuracy the most. So I have tried to optimize the depth of the tree over the cross validation accuracy. The distribution could be shown below:



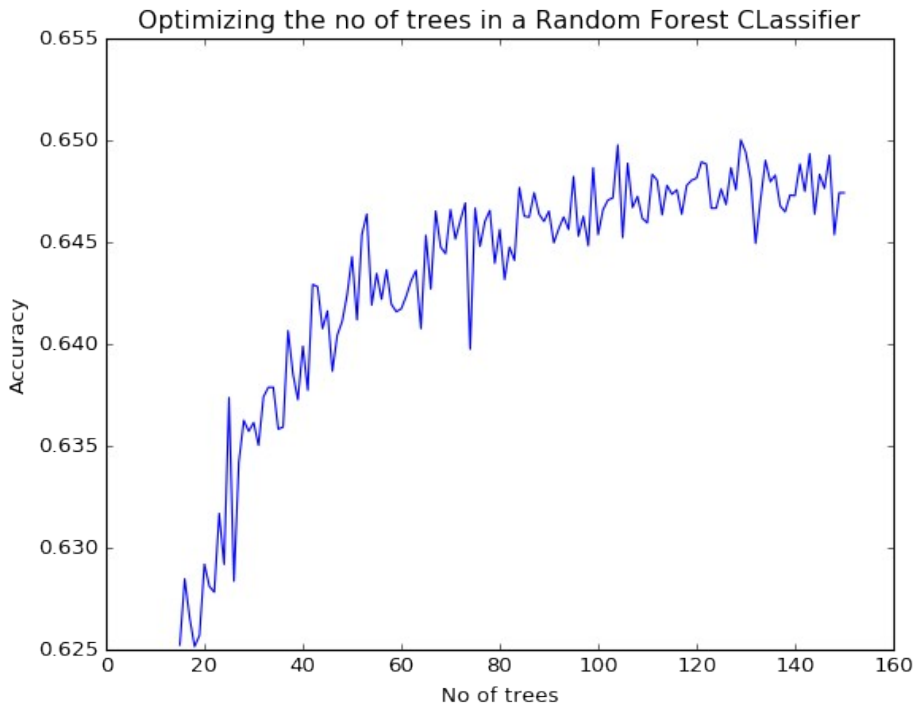
The optimum value of maximum depth is 2.
The decision tree constructed over training samples for a depth 2 is given below:



Test accuracy for a max_depth 2 for Decision tree classifier is 65.327%

Random Forest Classifier:

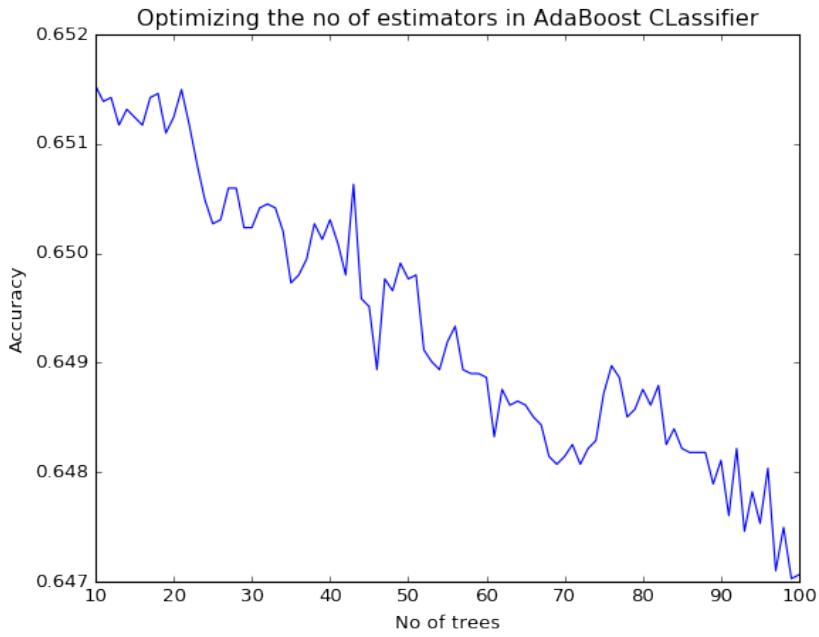
The next logical step was to find the accuracy over multiple decision trees and try to vote for the classification label. So I have performed the random forest classifier. No of trees in a random forest affect the accuracy of a random forest classifier. I tried to optimize over that parameter to achieve good accuracy results.



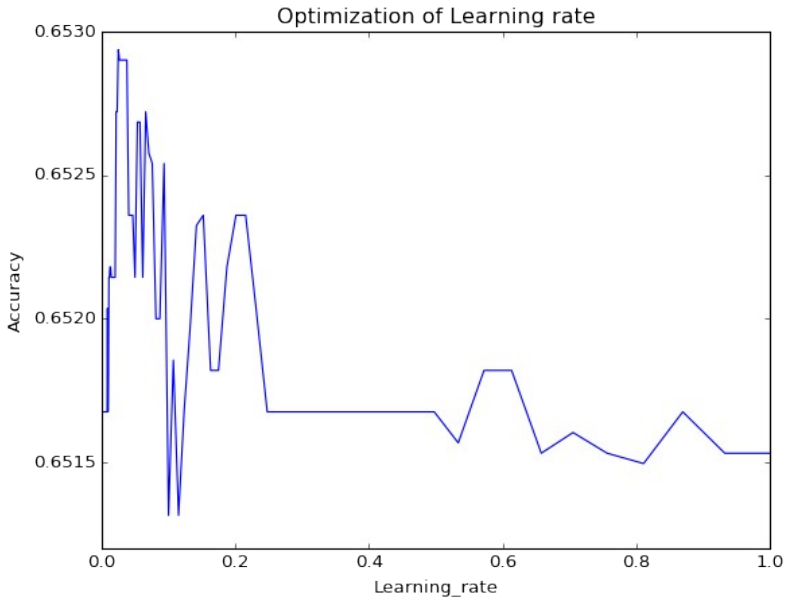
I have used the optimum value of 80 trees and a result of the previous classifier of optimum max_depth of 2 and applied it on the random forest classifier.
Surprisingly, the classification accuracy on the test set is 67.369%

Adaptive Boosting Classifier:

There were various parameters which we could optimize in AdaBoost Classifier. Learning rate and Number of estimators were two good parameters which we could optimize over. The optimization on n_estimators is as shown below:



Learning rate also drastically affects the performance of the classifier. For an optimum value of the number of estimators, I tried to optimize over the learning rate in the range of 0.001 to 1. The accuracy of the cross validated samples are as shown below:



For an estimate of the number of estimators and learning rate AdaBoost classifier gives an accuracy of 65.78% over test set.

Clustering Analysis:

K-Means Clustering:

As shown in previous figures, the distribution of samples. I wanted to try out clustering in order to achieve better accuracy. I have divided the class labels based on so that number of samples within a range are almost equal.

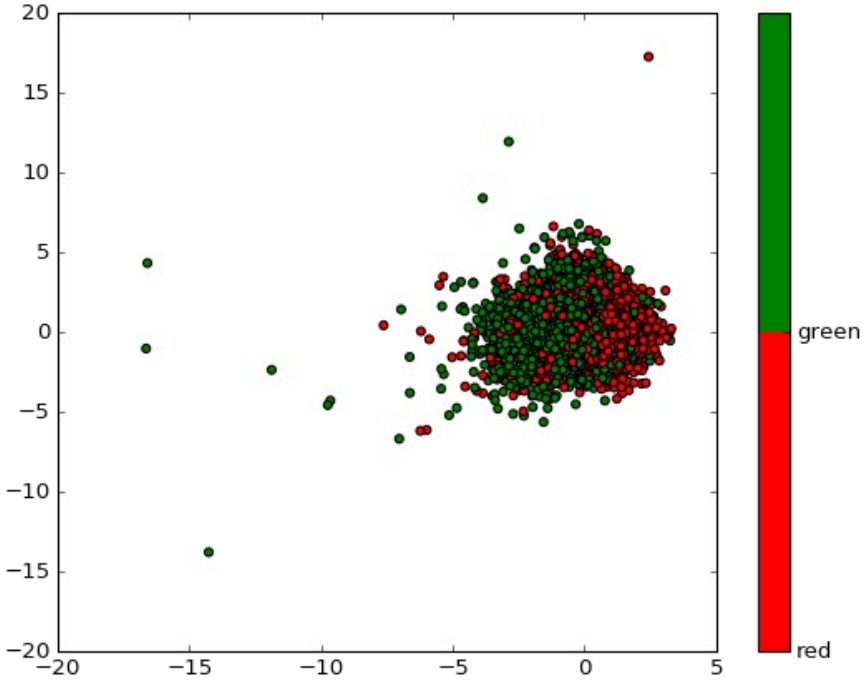
For no of clusters=2

	0-1400	>1400
Class label	0	1

Accuracy of classification: 64.594%

A 2D scatter plot for 2 random features is given below:

Where Red: Class 0 and Green: Class 1



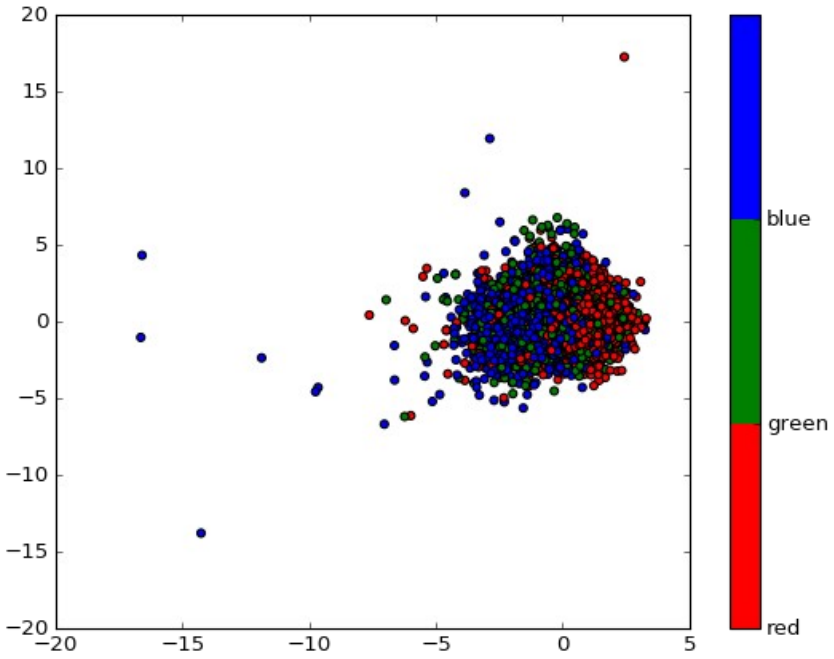
For no of clusters=3

	0-1100	>1100-2100	>2100
Class label	0	1	2

Accuracy of classification: 32.69%

A 2D scatter plot for 2 random features is given below:

Where Red: Class 0 , Green: Class 1 and Blue: Class2

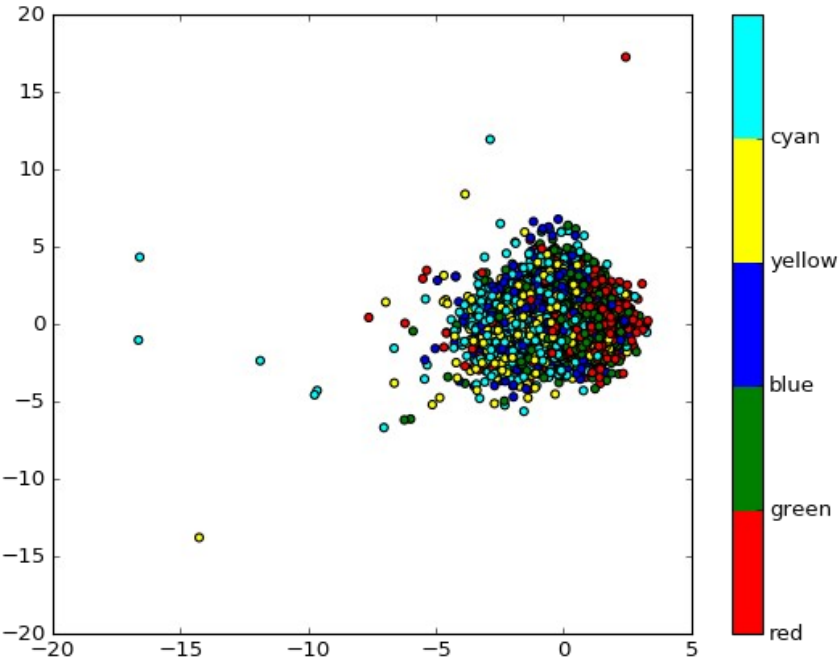


For no of clusters=5

	0-868	>868-1200	>1200-1800	>1800-3600	>3600
Class label	0	1	2	3	4

2D scatter plot for 2 random features is given below:

Where Red: Class 0 , Green: Class 1, Blue: Class2, Yellow: Class2, Cyan: Class4



Classification accuracy:17.61%

For higher class problems, I have tried out the best performing classifiers, i.e., random forest classifier.
I have divided classes based on the thresholds suggested in clustering analysis:

For Random forest classifier

No of Classes	Accuracy of Classification
2	67.36%
3	48.17%
5	30.06%
6	26.46%
10	16.07%

Result:

Regression Analysis with their R2 scores

Linear Regression	0.02
Kernel Ridge Regression	-142.2 to -0.030

Classification analysis with their Classification accuracy

Logistic Regression	63.855%
Linear SVM	65.15%
Kernel SVM	64.66%
Decision Tree	65.32%
Random Forest	**67.36%**
AdaBoost	65.78%

Clustering analysis over number of clusters in K means clustering

Number of clusters	Accuracy
2	64.59%
3	32.69%

5	17.61%

As we can see, over all the algorithms applied, Random forest classifier performs really well over such kind of data. This type of data is a bit difficult to deal with as there was samples were closely packed. Linear SVM performed a decent enough job even though it had high bias. Even when the bias was narrowed down in the higher order kernels of SVM, it did not affect the performance vastly. This suggests that data is not separable enough for SVM to handle even for extremely high degree polynomial kernels.

Future work:

One way of dealing with such type of data is improvement in the feature selection as there is very little room in model selection. We could utilise other state-of-the-art feature selection methods to improve the selection of combination of features for better performance.

As the number of samples are high enough for a neural network to learn. We could build a neural network which could learn the training data and fine tuning can be done for the hyper parameters of the network for better performance. With advancements in neural networks and deep neural network, we could utilize state-of-the-art methodologies for a better evaluation of the samples.

References:

[1] K. Fernandes, P. Vinagre and P. Cortez. A Proactive Intelligent Decision Support System for Predicting the Popularity of Online News. Proceedings of the 17th EPIA 2015 - Portuguese Conference on Artificial Intelligence, September, Coimbra, Portugal.

[2] Hensinger, Elena, Ilias Flaounas, and Nello Cristianini."Modelling and predicting news popularity." Pattern Analysis and Applications 16.4 (2013): 623-635.

[3] Tatar, Alexandru, et al. "Predicting the popularity of online articles based on user comments." Proceedings of the International Conference on Web Intelligence, Mining and Semantics. ACM, 2011.

[4] James, Gareth, et al.An introduction to statistical learning.New York: springer, 2013.

[5] http://network.colaberry.com/2015/07/28/online-news-popularity-analysis-with-tableau/

[6] http://www.nathalievilla.org/doc/pdf/dataanalytics2016-correction.pdf

[7] http://scikit-learn.org/stable/

[8] https://matplotlib.org/

[9] http://www.graphviz.org/