Online News Popularity Prediction Based on Shares

Sameer Ahmed Mohammed – sameerahmed1414@gmail.com

***Abstract*—In the era of digital communication getting news online though mobile or desktop as become very popular, and the online space has become a host for wide range of news. Many companies make their revenue by providing online news to the content. There is always a huge work on each and every article published and the only thing that we can measure to evaluate an article or news has become popular is when it gets more number of shares. Predicting number of shares a news would get before publishing would help the provider to analyze the article again and publish it so as it becomes popular.**

In this report we used different machine learning algorithms to predict number of shares an article would get based on different features. All these algorithms are evaluated and the results are discussed.

I. Introduction

The online news is an online version of a newspaper, it may be as a static article publication or as the online series. With the increase in use of smartphones online news has become the major source for reading news. It has become main source of income for many companies as well as bloggers. According to a survey "Nearly twice as many adults (38%) often get news online than get news in print (20%). Younger adults are especially likely to turn to the web for their news, while older Americans rely heavily on TV for their news" . The main intention of every news article is that it reaches many people i.e. to gain popularity. Every news we read has a lot of background work involved like gathering the information , preparing the content and making content attractive. After a lot of work an article is published and we should make sure that the article becomes popular and reaches many people. Therefore, predicting the popularity of online news is becoming a recent research trend . Popularity can be measured by analysing the number of features like shares, likes and comments. In this report we applied different machine learning algorithms to predict the popularity based on number of shares.

In order to predict the shares many predictive machine learning models were built and the best was selected. The data set is downloaded from UCI Machine Learning Repository .It contains a heterogeneous set of features about articles published by Mashable in a period of two years. The dataset comprises of 61 attributes of which 2 are non-predictive, 58 numeric attributes of predictive nature, and one is a class label. The class label is the shares that we are going to predict. A Quick look at data shows wide variety of features . There are few outliers and the idea is to preprocess the data to remove these outliers. Once that Data is clean, we divide the dataset into training and testing data. Then we train the models on training data using Different predictive models like linear classifier, tree based, distance based, rule based and ensemble models,. In this Project, we have taken up different methods to train models using dataset viz ,Linear regression, SVM, Random forest , Lasso, Ridge ,Bayesian, KNN, Bagging and Decision tree and the best models is evaluated based on different metrics . Further the regression model is converted into the classification model classifying weather the article is popular or not popular. Again different classification models are built and evaluated under different evaluation metrics.

II. Case Study

A. About the Dataset

The dataset taken is a Multivariate data set as is evident from the Number of Attributes which is 61.The Attributes are all Integer except URL which is Non-Predictive in nature. Total Number of Rows/Instance are 39797 .Total Number of Columns/Features are 61

Attributes:

The attributes contained in the data set are described below:

Number of Attributes: 61 (58 predictive attributes, 2 non-predictive, 1 goal field)

Attribute Information:

0. url: URL of the article (non-predictive)

1. timedelta: Days between the article publication and the dataset acquisition (nonpredictive)

2. n_tokens_title: Number of words in the title

3. n_tokens_content: Number of words in the content

4. n_unique_tokens: Rate of unique words in the content

5. n_non_stop_words: Rate of non-stop words in the content

6. n_non_stop_unique_tokens: Rate of unique non-stop words in the content

7. num_hrefs: Number of links

8. num_self_hrefs: Number of links to other articles published by Mashable

9. num_imgs: Number of images

10. num_videos: Number of videos

11. average_token_length: Average length of the words in the content

12. num_keywords: Number of keywords in the metadata

13. data_channel_is_lifestyle: Is data channel 'Lifestyle'?

14. data_channel_is_entertainment: Is data channel 'Entertainment'?

15. data_channel_is_bus: Is data channel ‘Business'?

16. data_channel_is_socmed: Is data channel 'Social Media'?

17. data_channel_is_tech: Is data channel 'Tech'?

18. data_channel_is_world: Is data channel 'World'?

19. kw_min_min: Worst keyword (min. shares)

20. kw_max_min: Worst keyword (max. shares)

21. kw_avg_min: Worst keyword (avg. shares)

22. kw_min_max: Best keyword (min. shares)

23. kw_max_max: Best keyword (max. shares)

24. kw_avg_max: Best keyword (avg. shares)

25. kw_min_avg: Avg. keyword (min. shares)

26. kw_max_avg: Avg. keyword (max. shares)

27. kw_avg_avg: Avg. keyword (avg. shares)

28. self_reference_min_shares: Min. shares of referenced articles in Mashable

29. self_reference_max_shares: Max. shares of referenced articles in Mashable

30. self_reference_avg_sharess: Avg. shares of referenced articles in Mashable

31. weekday_is_monday: Was the article published on a Monday?

32. weekday_is_tuesday: Was the article published on a Tuesday?

33. weekday_is_wednesday: Was the article published on a Wednesday?

34. weekday_is_thursday: Was the article published on a Thursday?

35. weekday_is_friday: Was the article published on a Friday?

36. weekday_is_saturday: Was the article published on a Saturday?

37. weekday_is_sunday: Was the article published on a Sunday?

38. is_weekend: Was the article published on the weekend?

39. LDA_00: Closeness to LDA topic 0

40. LDA_01: Closeness to LDA topic 1

41. LDA_02: Closeness to LDA topic 2

42. LDA_03: Closeness to LDA topic 3

43. LDA_04: Closeness to LDA topic 4

44. global_subjectivity: Text subjectivity

45. global_sentiment_polarity: Text sentiment polarity

46. global_rate_positive_words: Rate of positive words in the content

47. global_rate_negative_words: Rate of negative words in the content

48. rate_positive_words: Rate of positive words among non-neutral tokens

49. rate_negative_words: Rate of negative words among non-neutral tokens

50. avg_positive_polarity: Avg. polarity of positive words

51. min_positive_polarity: Min. polarity of positive words

52. max_positive_polarity: Max. polarity of positive words

53. avg_negative_polarity: Avg. polarity of negative words

54. min_negative_polarity: Min. polarity of negative words

55. max_negative_polarity: Max. polarity of negative words.

56. title_subjectivity: Title subjectivity

57. title_sentiment_polarity: Title polarity

58. abs_title_subjectivity: Absolute subjectivity level

59. abs_title_sentiment_polarity: Absolute polarity level

60. shares: Number of shares (target)

Few features are mostly particular to the Mashable. Genrally articles reference other articles that are published in the same mashable and they have meta-data, such as type of data channel keywords, and total number of shares received in Facebook, Twitter, LinkedIn, StumbleUpon, Google+, and Pinterest. They have extracted the minimum, average and maximum number of shares before publication of all Mashable links cited in the article. Similarly, they ranked all article keyword average shares known before publication in order to get the worst, average and best keywords. For each of these keywords, they extracted the minimum, average and maximum number of shares. The data channel categories are: “lifestyle”, “bus”, “entertainment”, “socmed” , “tech”, “viral” and “world”. They have also extracted several natural language processing features. The Latent Dirichlet Allocation (LDA) algorithm was applied to all Mashable articles order to first identify the five top relevant topic bands and then measured the closeness of current article to such topics. To compute the subjectivity and polarity sentiment analysis, Pattern web mining module was used.

The Shares is the target variable that we are going to predict. As there are many features, feature reduction and outlier removal has to be done to make data clean and build models more accurately.

B. Evaluation Metrics

1. RMSE (Root Mean Square Error)

It is the simple standard deviation of the differences between the values that are predicted and observed. calculated using this formula:

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{j=1}^{n}(y_j - \hat{y}_j)^2}$$

2. MAE

It is the average of the absolute difference between the predicted values and observed value. The MAE is a linear score which means that all the individual differences are weighted equally in the average. For example, the difference between 10 and 0 will be twice the difference between 5 and 0. Mathematically, it is calculated using this formula:

$$\mathrm{MAE} = \frac{1}{n}\sum_{j=1}^{n}|y_j - \hat{y}_j|$$

3. MSE (Mean Square Error)

MSE is the sum of squared distances between our target variable and predicted values.

$$MSE = \frac{\sum_{i=1}^{n} (y_i - y_i^p)^2}{n}$$

4. Variance

This is the amount that the estimate of the target function will change if different training data was used. The predicted value is estimated from the training data from different machine learning algorithms and we get some variance for the output predicted

III. Experimental setup and Evaluation

We used different libraries in python to Scikit-learn which is an open-source library for analyzing data mining. Python is used to analyze and create models from various regression based machine learning algorithms. Scikit-learn can also be used for preparing data in several ways: normalization, standardization, and cleaning outlier data or missing data. We also used Pandas is an open source, BSD licensed library. It provides high performance and simple tools for data analysis. It is mainly used to load data from the raw Excel .csv file

A. Data Pre-processing

The Dataset comprises of 39797 observations There were no missing values. We removed “url” (non-predictive) attributes from the Dataset. We have applied outlier detection for detecting and removing the outliers. We also did feature engineering to reduce the number of features considered only the main features

Outlier detection : Outliers are extreme values that fall a long way outside of the other observations. For example, in a normal distribution, outliers could be values on the tails of the distribution. There are different ways to identify and detect outliers.

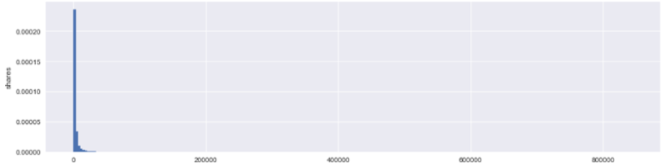


Fig 1: Graph showing distribution of shares

Example from fig 1 the shares features is distributed from a range of 0 to 80000 where only some shares are above 25000. We used formula median + two standard deviation as threshold to remove outliers. So values above the threshold were removed.

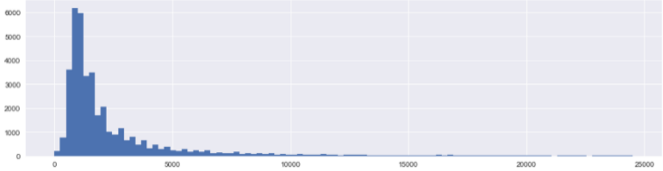


Fig 2: Graph after removing outliers

From the fig 2 we can see that after removing the outliers the shares values has been reduced and distributed between 0 to 25000.This processing has removed around 500 instances.

Correlation :

Correlation gives us the association between different quantities. By using this we can predict one quantity from other quantity. We can check the presence of casual relationship using this. There are many ways to calculate the correlation coefficient and are described below.

Correlation matrix:

It is obtained by plotting all the features against each other

We have plotted this for all the features and analyzed to see the relation between each of them. We even obtained correlation of each feature to the predicted value.

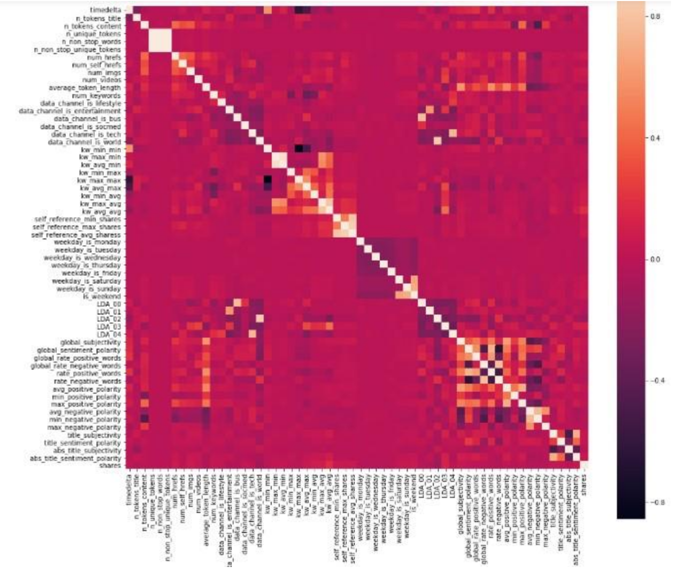


Fig 3: Correlation matrix

From fig 3 we can see that there is no high correlation between the features and also there is no high correlation with the predicted variable. So this makes that we can not remove any features and each and every feature is important for building the model. As we can see many of the features are features obtained by converting nominal to binary.

PCA:

Principle component analysis is one the technique that is widely used to find and remove unnecessary features. In order to perform PCA first we need to normalize the data i.e. scaling all the features to one range. After that compute the k principle component vectors. All the k vectors are sorted in the decreasing significance of their strength , This we can find the weak components and remove them so that we can build a better model.

Staring with the scaling we scaled all features from 0-1 and models were build . When compared to the models without normalization to the model with normalization we have found that the model without normalization is better than the model obtained through PCA. So PCA is not useful in our dataset.

SVD:
Singular value decomposition(SVD) is feature extraction method based on matrix factorization method. SVD can be used when we have all positive Features. As there are negative values in our dataset we cannot use SVD.

B. Test and Train split

Before building the model we need to split the data into training and testing .Therefore the model built using the training and validated using testing set. There are many ways in which data can be split in our project we implemented test train split and cross validation.

Test and train split:
In this method the data is divided into testing and training based on percentage split. As our data set is pretty big we tried different data splits and we got best results at 75 percent split .So Data is divided to training and testing data where 75 percent is taken as the training data and the remaining 25% is taken as testing data.

Cross validation :
The most commonly used cross validation method is k-fold were data is divided into k folds. Every time the model is built using the k-1 folds and the remaining fold is tested and this process is repeated 10 times. As our dataset has nearly 40,000 instances building models and validating using K-fold takes a lot of time.

C. Building the models

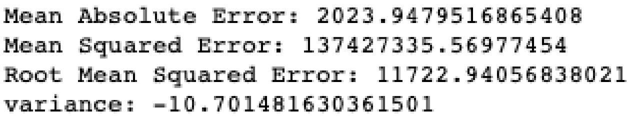
After the preprocessing of data and dividing it into test and train splits we have built different models using different kinds of algorithms. All the regression algorithms that are linear based ,tree based, rule based, probability based and ensemble models are evaluated and results are discussed.

Linear Regression :
It is an algorithm based on supervised learning which performs a regression task. It is commonly used for finding out the relationship between variables and predicting the target variable. It performs the task to predict a variable value that is depending(y) based on the given independent variable (x). So, we find a linear relation between x and y i.e. the input and output using regression technique

$$y = \theta_1 \; + \; \theta_2.x$$

Here y is the label for output and x for input . First theta is slope and second is the coefficient of X.

After applying Linear regression to the model the following results are obtained. All the four metrics discussed previously are shown in fig.4

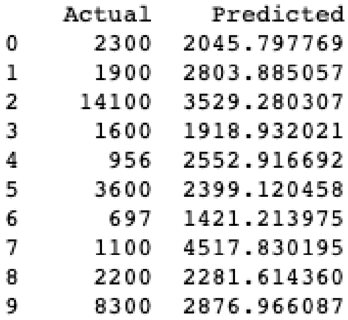


```
Mean Absolute Error: 2023.9479516865408
Mean Squared Error: 137427335.56977454
Root Mean Squared Error: 11722.94056838021
variance: -10.701481630361501
```

Fig 4: Results for linear regression

The first 10 actual values and predicted values by a the model are shown in fig 5.

A graph is also plotted between all the predicted and actual values and is shown in fig 6.The prediction is nearly accurate for the starting values and became diverse at predicting the shares at the large value.



```
   Actual    Predicted
0    2300  2045.797769
1    1900  2803.885057
2   14100  3529.280307
3    1600  1918.932021
4     956  2552.916692
5    3600  2399.120458
6     697  1421.213975
7    1100  4517.830195
8    2200  2281.614360
9    8300  2876.966087
```

Fig 5: First 10 actual vs predicted linear regression

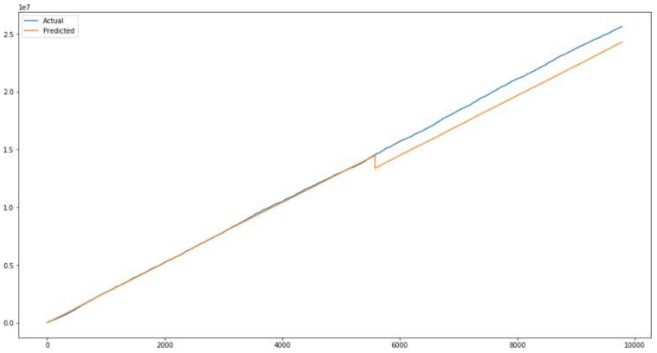
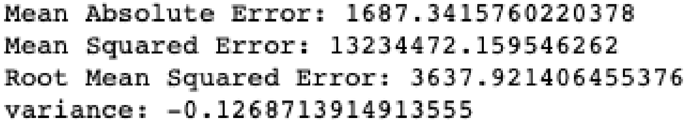


Fig 6: Graph between all actual and predicted

SVR :
We can also use Support Vector Machine regression method, maintaining all the main features that characterize the algorithm .By using Support Vector Regression we can satisfy all the principles as the SVM for classification, with only a few minor differences. As the output is a integer value it becomes very difficult to predict the information easily, as it has many possibilities. Where as in regression, a margin is set in approximation to the SVR as per the problem. Basically our idea always is the same to minimize error, getting the hyperplane which maximizes the margin.

After applying to the SVR model the following results are obtained. All the four metrics discussed previously are shown in fig.7

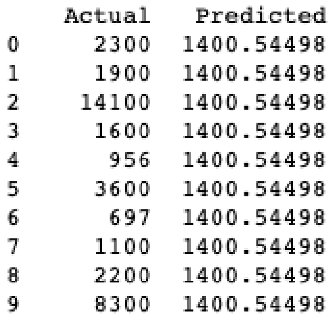


```
Mean Absolute Error: 1687.3415760220378
Mean Squared Error: 13234472.159546262
Root Mean Squared Error: 3637.921406455376
variance: -0.1268713914913555
```

Fig 7: Results for SVR

The first 10 predicted values vs actual values are shown in fig 8.

A graph is also plotted between all the predicted and actual values and is shown in fig 9.



```
   Actual   Predicted
0    2300  1400.54498
1    1900  1400.54498
2   14100  1400.54498
3    1600  1400.54498
4     956  1400.54498
5    3600  1400.54498
6     697  1400.54498
7    1100  1400.54498
8    2200  1400.54498
9    8300  1400.54498
```

Fig 8: First 10 actual vs predicted SVR

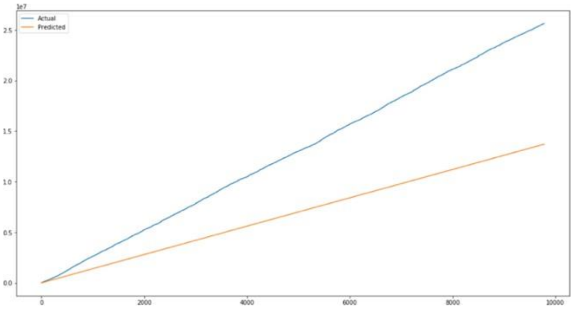
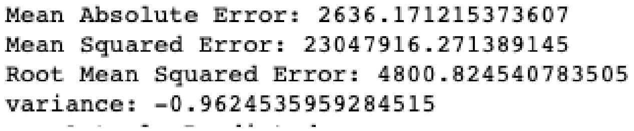


Fig 9: Graph between all actual and predicted

Decision Tree :

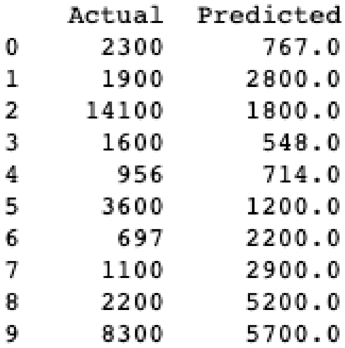
By using decision tree a regression model can be built in the form of a tree structure. Many small subsets are created and at the same time an complete decision tree is developed towards the end. Finally a tree is built with all the decision and leaf nodes. A decision node has two or more branches , each branch represents the values for the attribute that is tested. The node at the end i.e. leaf node represents a decision on the numerical target. By using this we can handle both the numerical categorical data.

After applying to the decision tree model the following results are obtained. All the four metrics discussed previously are shown in fig.10



```
Mean Absolute Error: 2636.171215373607
Mean Squared Error: 23047916.271389145
Root Mean Squared Error: 4800.824540783505
variance: -0.9624535959284515
```

Fig 10: Results for Decision tree



```
   Actual  Predicted
0    2300      767.0
1    1900     2800.0
2   14100     1800.0
3    1600      548.0
4     956      714.0
5    3600     1200.0
6     697     2200.0
7    1100     2900.0
8    2200     5200.0
9    8300     5700.0
```

Fig 11: First 10 actual vs predicted Decision tree

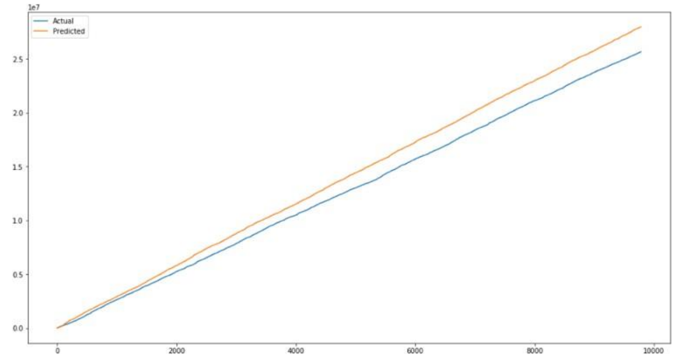
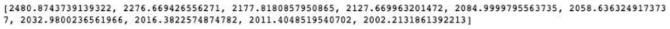


Fig 12: Graph between all actual and predicted

KNN :

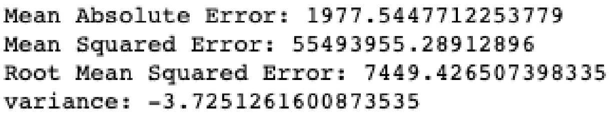
K nearest neighbors stores all training data and predict the target based on the distance measurement functions. There are different distance calculation methods like Euclidean ,Manhattan , Minkowski .We can use these three distance measures for continuous variables. After calculating the distance , best K value is selected by first inspecting the data. Another way to determine a best value for K is by using Cross validation to an independent data set to validate the K value. Generally K value for most datasets is greater than 10,which produces much better results when one near neighbor is taken.



```
[2480.8743739139322, 2276.669426556271, 2177.8180857950865, 2127.669963201472, 2084.9999795563735, 2058.636324917373
7, 2032.9800236561966, 2016.3822574874782, 2011.4048519540702, 2002.2131861392213]
```

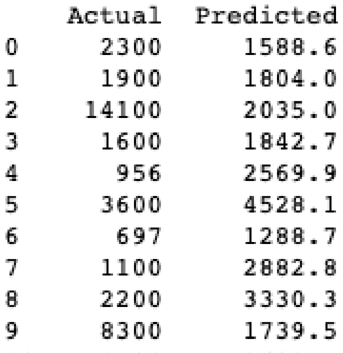
Fig 13: MAE for 1 to 10 neighbors

After applying to the KNN model for 10 neighbours the following results are obtained. All the four metrics discussed previously are shown in fig.14



```
Mean Absolute Error: 1977.5447712253779
Mean Squared Error: 55493955.28912896
Root Mean Squared Error: 7449.426507398335
variance: -3.7251261600873535
```

Fig 14: Results for KNN



```
   Actual  Predicted
0    2300     1588.6
1    1900     1804.0
2   14100     2035.0
3    1600     1842.7
4     956     2569.9
5    3600     4528.1
6     697     1288.7
7    1100     2882.8
8    2200     3330.3
9    8300     1739.5
```

Fig 15: First 10 actual vs predicted KNN

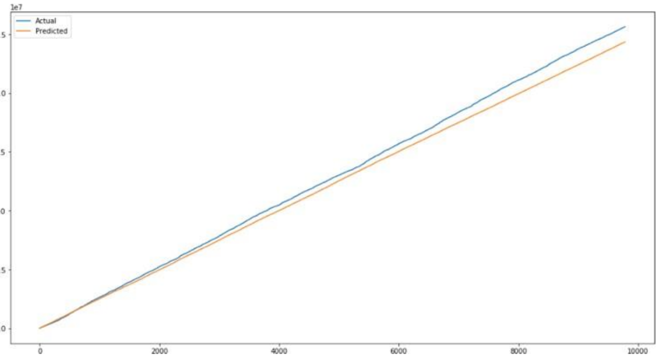
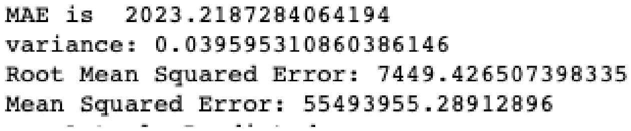


Fig 16: Graph between all actual and predicted

Random Forest :

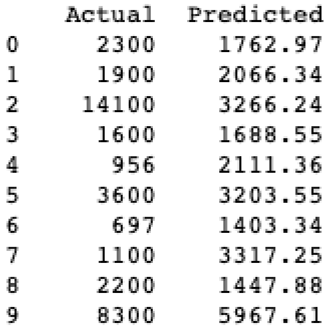
Random Forest ,from the name we can say that creates a forest and makes it random. The forest that is built, is a combination of Decision Trees. It is trained using the bagging method which is the combination of different models so that the overall result is increased. It has almost the same parameters as a decision tree. While growing the trees It adds some randomness to the model . It searches for the best feature among a random subset of features, instead of searching for the important feature while splitting a node, This way we can get different range of results and a better model.

After applying to the Random Forest model the following results are obtained. All the four metrics discussed previously are shown in fig.17



```
MAE is  2023.2187284064194
variance: 0.039595310860386146
Root Mean Squared Error: 7449.426507398335
Mean Squared Error: 55493955.28912896
```

Fig 17: Results for Random Forest



```
    Actual  Predicted
0     2300    1762.97
1     1900    2066.34
2    14100    3266.24
3     1600    1688.55
4      956    2111.36
5     3600    3203.55
6      697    1403.34
7     1100    3317.25
8     2200    1447.88
9     8300    5967.61
```

Fig 18: First 10 actual vs predicted Random Forest

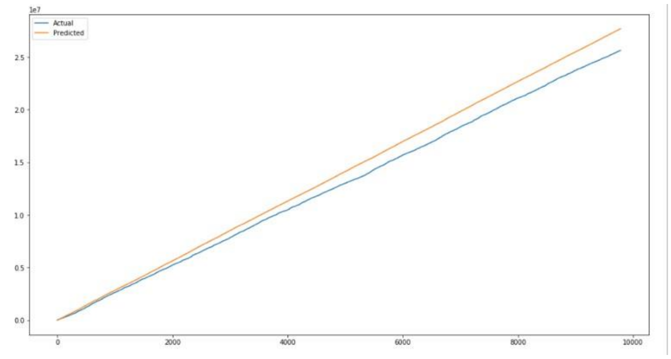
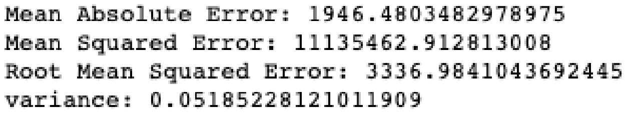


Fig 19: Graph between all actual and predicted

Bayesian Ridge :

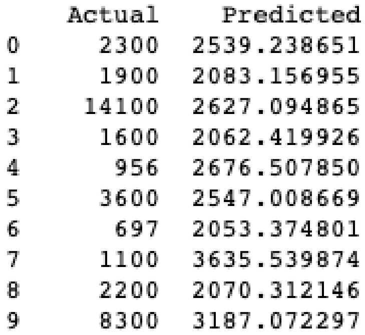
In the view of Bayesian , this is similar to linear regression using probability distributions instead of point estimates. The output, y, is not calculated as a single value, but it is assumed to be calculated from a probability distribution. The predicting value y generated from a Gaussian Distribution by a mean and variance. The mean for linear regression is the transpose of the weight matrix multiplied by the predictor matrix. The variance is the square of the standard deviation σ.The Goal of this model is determine the posterior distribution for the model parameters rather than finding the single best value of the model parameters.

After applying to the Bayesian Ridge model the following results are obtained. All the four metrics discussed previously are shown in fig.20



```
Mean Absolute Error: 1946.4803482978975
Mean Squared Error: 11135462.912813008
Root Mean Squared Error: 3336.9841043692445
variance: 0.05185228121011909
```

Fig 20: Results for Bayesian ridge



```
   Actual    Predicted
0    2300  2539.238651
1    1900  2083.156955
2   14100  2627.094865
3    1600  2062.419926
4     956  2676.507850
5    3600  2547.008669
6     697  2053.374801
7    1100  3635.539874
8    2200  2070.312146
9    8300  3187.072297
```

Fig 21: First 10 actual vs predicted Bayesian Ridge

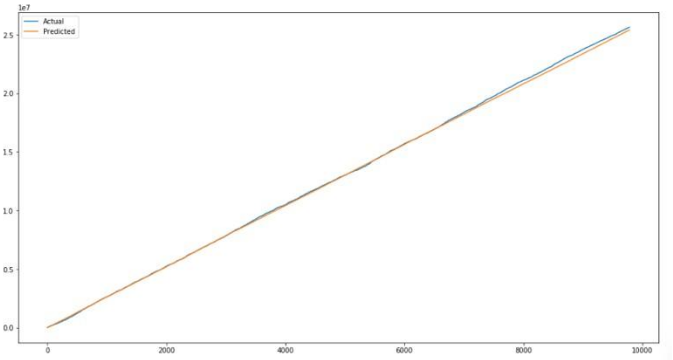
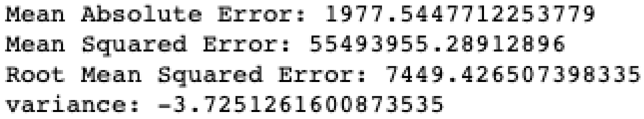


Fig 22: Graph between all actual and predicted

Ridge:

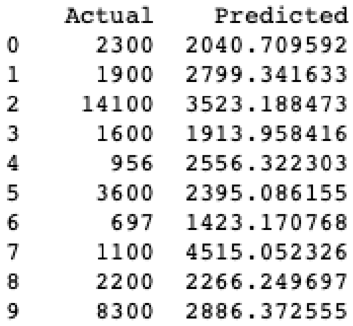
By using ridge we can analyze multiple regression data that is suffering from multicollinearity. When multicollinearity occurs, least squares estimates are unbiased, but their variances are large so they may be far from the true value. By adding a degree of bias to the regression estimates, ridge regression reduces the standard errors.

After applying to the Ridge model the following results are obtained. All the four metrics discussed previously are shown in fig.23



```
Mean Absolute Error: 1977.5447712253779
Mean Squared Error: 55493955.28912896
Root Mean Squared Error: 7449.426507398335
variance: -3.7251261600873535
```

Fig 23: Results for Ridge



```
     Actual    Predicted
0      2300  2040.709592
1      1900  2799.341633
2     14100  3523.188473
3      1600  1913.958416
4       956  2556.322303
5      3600  2395.086155
6       697  1423.170768
7      1100  4515.052326
8      2200  2266.249697
9      8300  2886.372555
```

Fig 24: First 10 actual vs predicted using ridge

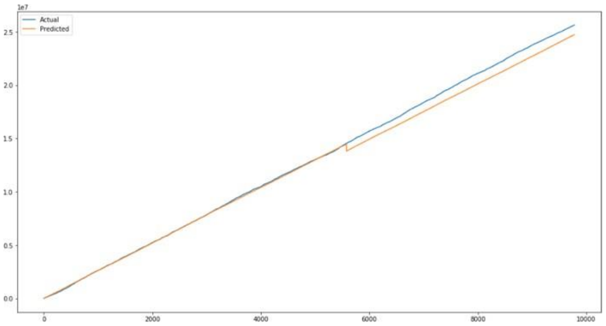
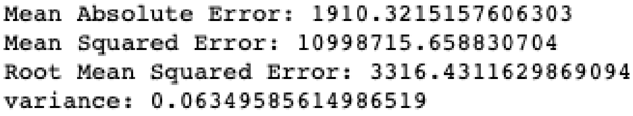


Fig 25: Graph between all actual and predicted

Lasso :

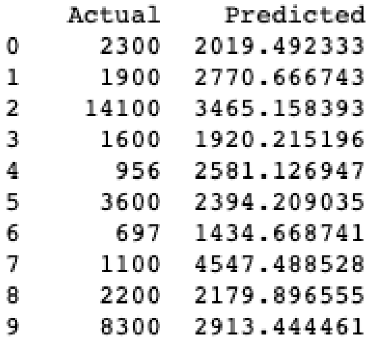
In this analysis occurrence of both variable selection and regularization takes place simultaneously .Coefficients of regression value is effected by the penalty . More coefficients becomes zero with increase in penalty and vice Versa. It uses L1 normalization technique in which tuning parameter is used as amount of shrinkage. When the tuning parameter decreases then variance increases and when it increases then bias increases and as is. When is tends to infinity then all the coefficients will be zero and when it is constant then no coefficients are zero.

After applying to the Lasso model the following results are obtained. All the four metrics discussed previously are shown in fig.26



```
Mean Absolute Error: 1910.3215157606303
Mean Squared Error: 10998715.658830704
Root Mean Squared Error: 3316.4311629869094
variance: 0.06349585614986519
```

Fig 26: Results for lasso



```
     Actual    Predicted
0      2300  2019.492333
1      1900  2770.666743
2     14100  3465.158393
3      1600  1920.215196
4       956  2581.126947
5      3600  2394.209035
6       697  1434.668741
7      1100  4547.488528
8      2200  2179.896555
9      8300  2913.444461
```

Fig 27: First 10 actual vs predicted using ridge

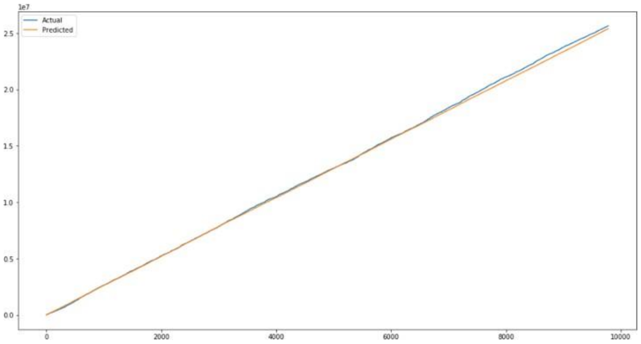
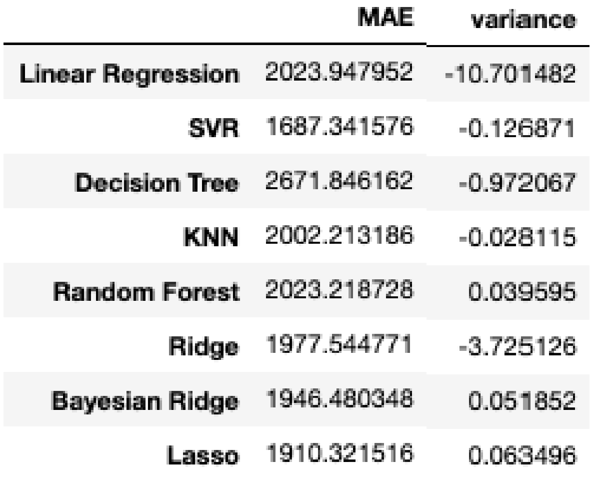


Fig 28: Graph between all actual and predicted

D. Evaluation of the models

We have evaluated the models based on the metrics discussed in the previous section.



	MAE	variance
Linear Regression	2023.947952	-10.701482
SVR	1687.341576	-0.126871
Decision Tree	2671.846162	-0.972067
KNN	2002.213186	-0.028115
Random Forest	2023.218728	0.039595
Ridge	1977.544771	-3.725126
Bayesian Ridge	1946.480348	0.051852
Lasso	1910.321516	0.063496

Fig 29:Results of prediction models

After building all the models and calculating the metrics we can see that Mean absolute error for SVR is very less. When we consider variance the variance is lowest for the random forest. From fig 28 we can also see that Lasso also has an good prediction rate.

E. Solving as Classification

By using regression metrics we cannot estimate model which is better and cannot find metrics like accuracy, recall, precision, area under curve and roc. In order to get an idea of all the techniques discussed in class we have turned the regression model into classification and evaluated models based on all metrics discussed in the lecture's.

1. Data conversion:

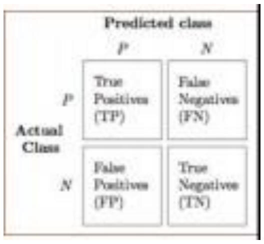
In order to perform the classification we have taken the data that is obtained after the pre-processing done for the regression. The shares attributes which is to be predicted is converted into binary classification. Median value of the shares attribute is taken and the dataset is split based on the median value all the values below the median value are taken as on group and the remaining as the other value. In our dataset median value of number of shares is 1400 so conversion is based on this value. Shares below median value are converted to 0 and above are converted to 1.Now the data is converted into binary classification where the output 0 means the article will be not popular and 1 means it will become popular.

2. Test and Train split:

As discussed previously there are many ways to split the data and validate in the classification we have used 10 fold cross validation and models are evaluated.

3. Evaluation metrics :

Confusion Matrix : A confusion matrix is a table that is often used to describe the performance of a classification model (or "classifier") on a set of test data for which the true values are known. It has the form:



		Predicted class	
		P	N
Actual Class	P	True Positives (TP)	False Negatives (FN)
	N	False Positives (FP)	True Negatives (TN)

Fig 30: Confusion Matrix

For instance, TP : correctly predicted positive class and FN: positive class instances which are predicted as negative by classifier.

Accuracy : Classification accuracy is the ratio of correct predictions to total predictions made , It is often presented as a percentage.

Formula : Accuracy = TP+TN/Total * 100%

Precision: True positives over total predicted positive class is called Precision or Positive predictive value (PPV)

Formula : Precision = TP / TP + FP

Recall : True positives over total number of positives is called Recall or True positive rate or Sensitivity.

Formula : Recall = TP / TP + FN

4. Building Models :

Different classification models are build covering all types like the rule based ,tree based, probability based, Logistic and ensemble models. We have tried different algorithms and the results of the best are discussed below.

Logistic Regression :

Logistic regression is a statistical method for analyzing a dataset in which there are one or more independent variables that determine an outcome. The outcome in this case would be a discrete value as opposed to continuous value in linear regression task.

Rather than choosing parameters that minimize the sum of squared errors (like in ordinary regression), estimation in logistic regression chooses parameters that maximize the likelihood of observing the sample values.

Logistic Regression can be easily understood if one has idea about logistic function which is also referred to as sigmoid function.

Sigmoid function takes any real input and outputs a value between 0 and 1 .

$$\sigma(t) = \frac{e^t}{e^t + 1} = \frac{1}{1 + e^{-t}}$$

Where t can be assumed as a hypothesis function.

$$t = \beta_0 + \beta_1 x$$

Logistic function can now be written as :

$$p(x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}}$$

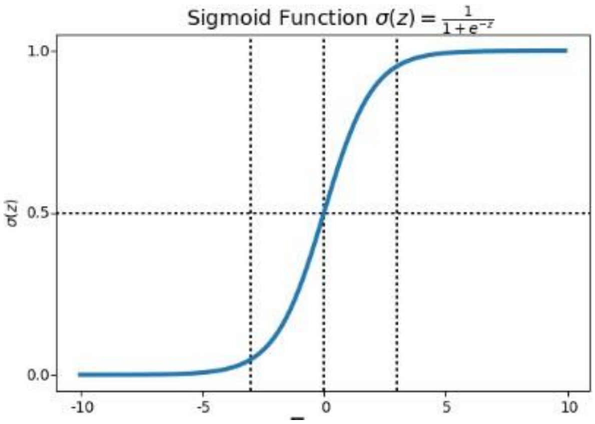
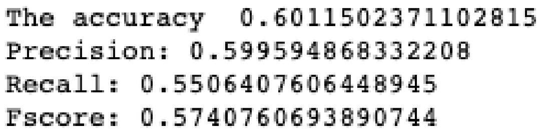


Fig 31: Sigmoid function

Sigmoid function graph y belongs to (0 , 1) inclusive

After applying Logistic regression to the dataset the following results are obtained :



```
The accuracy  0.6011502371102815
Precision: 0.599594868332208
Recall: 0.5506407606448945
Fscore: 0.5740760693890744
```

Fig 32: Results after Logistic Regression

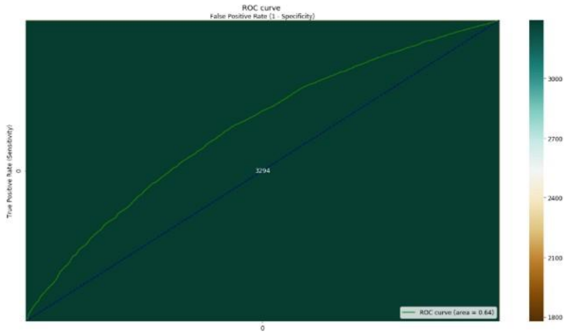


Fig 33: ROC Curve

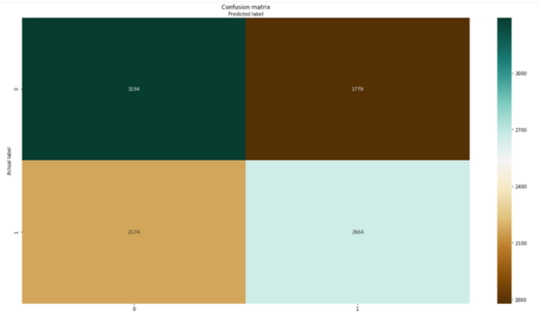


Fig 34: Confusion Matrix

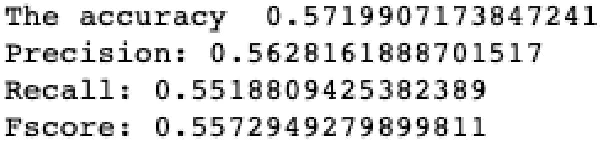
The area under the curve after Logistic regression is 0.64 form fig 33.

KNN :

This algorithm can be summarized as :

A positive integer k is specified , along with a new sample.
Model then selects k entries in dataset which are closest to new sample.
Most common classification class out of these k is found.
We classify this new instance into the majority class out of those K instances chosen
KNN does not learn any model.
Makes predictions just in time by calculating similarity between sample and training instances.

After applying to the KNN model for 10 neighbors the following results are obtained. All the four metrics discussed in evaluation metrics section.



```
The accuracy  0.5719907173847241
Precision: 0.5628161888701517
Recall: 0.5518809425382389
Fscore: 0.5572949279899811
```

Fig 35: Results after KNN

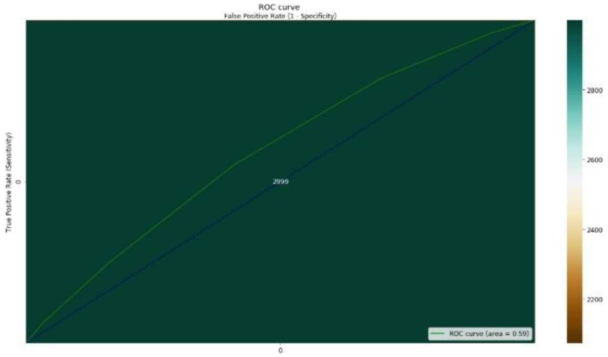


Fig 36: ROC Curve

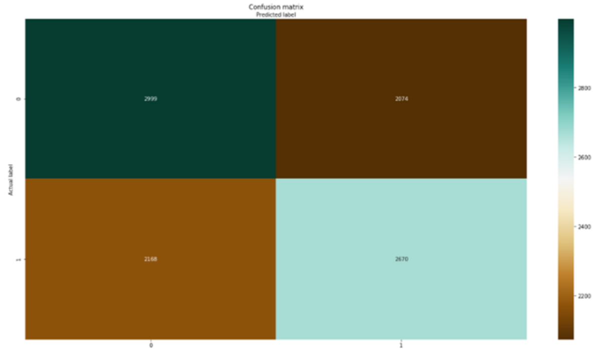


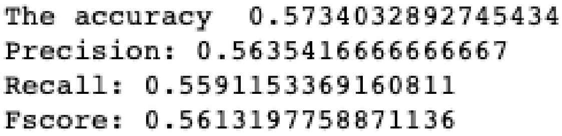
Fig 37: Confusion Matrix

The area under the curve after KNN is 0.59 form fig 36.

Decision tree :

Decision tree learning uses a decision tree to go from observations about an instance to conclusion about that instances target value. Tree models where target variable can take a discrete set of values are known as classification trees in these structures leaves represent class labels and branches represent conjunction of features that lead to these class labels.

After applying to the decision tree model the following results are obtained. All the four metrics are discussed previously .



```
The accuracy  0.5734032892745434
Precision: 0.5635416666666667
Recall: 0.5591153369160811
Fscore: 0.5613197758871136
```

Fig 38: Results After decision tree

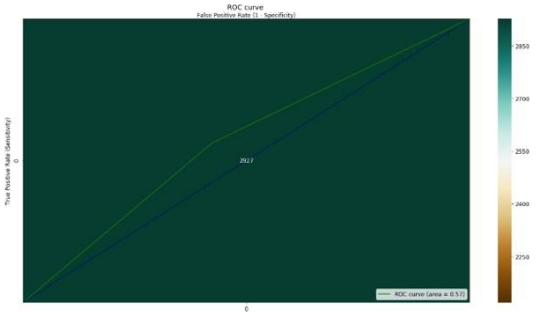


Fig 39: ROC Curve

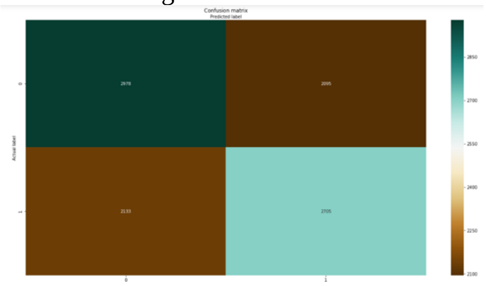


Fig 40: Confusion Matrix

The area under the curve after Decision trees is 0.57 form fig 39.

Naive Bayes :

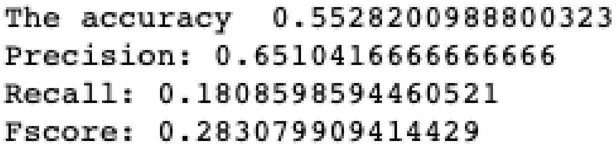
A Naive Bayes classifier is a probabilistic machine learning model that's used for classification task. The crux of the classifier is based on the Bayes theorem.

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Using Bayes theorem, we can find the probability of A happening, given that B has occurred. Here, B is the evidence and A is the hypothesis. The assumption made here is that the predictors/features are independent. That is presence of one particular feature does not affect the other. Hence it is called naïve.

They are fast and easy to implement but their biggest disadvantage is that the requirement of predictors to be independent. In most of the real life cases, the predictors are dependent, this hinders the performance of the classifier.

After applying Naïve Bayes model the following results are obtained. All the four metrics are discussed previously



```
The accuracy  0.5528200988800323
Precision: 0.6510416666666666
Recall: 0.1808598594460521
Fscore: 0.283079909414429
```

Fig 41: Results after Naïve Bayes

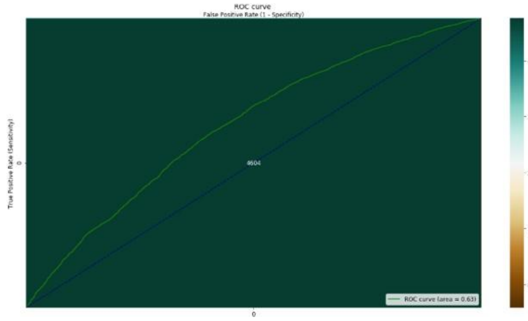


Fig42: ROC Curve

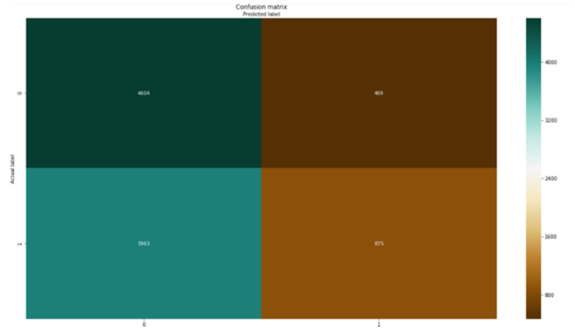


Fig 43: Confusion Matrix

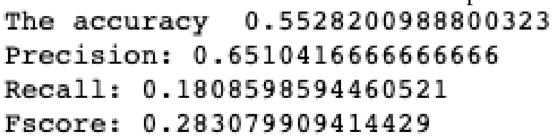
The area under the curve after Naïve bayes is 0.63 form fig 42.

Random Forest :

Random Forest algorithm randomly selects observations and features to build several decision trees and then averages the results.

Deep decision trees might suffer from overfitting. Random Forest prevents overfitting most of the time, by creating random subsets of the features and building smaller trees using these subsets. Afterwards, it combines the subtrees. Note that this doesn't work every time and that it also makes the computation slower, depending on how many trees random forest builds.

After applying Ensemble model of trees the following results are obtained. All the four metrics are discussed previously



```
The accuracy  0.5528200988800323
Precision: 0.6510416666666666
Recall: 0.1808598594460521
Fscore: 0.283079909414429
```

Fig 44: Results after Random Forest

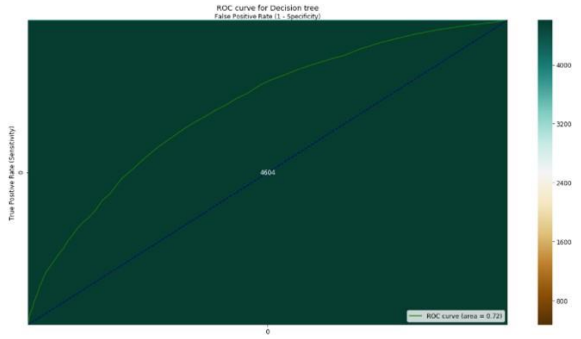
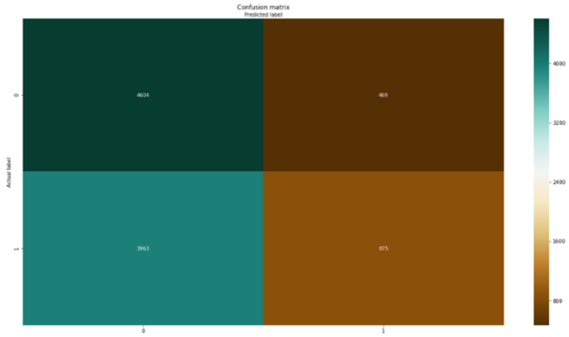


Fig 45: ROC Curve



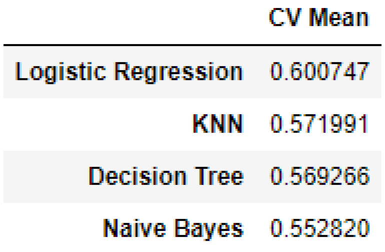
	0	1
0	4604	469
1	3963	875

Fig 46: Confusion Matrix

The area under the curve after Random Forest is 0.72 form fig 45.

5. Evaluation of Classification models :

   Summary of Classifier models when cross validated with 10 folds, clearly RF classifier is performing better when compared to other classifiers.



	CV Mean
Logistic Regression	0.600747
KNN	0.571991
Decision Tree	0.569266
Naive Bayes	0.552820
Random Forest Classifier	0.64

By considering the accuracy and also the area under the curve from the ROC clearly Random forests perform better than other models

F. *Discussion and Lessons learnt*

After doing both the regression and classification on the dataset we have Learnt a lot of things. Starting with the preprocessing we have gained knowledge on different correlation techniques like PCA,SVD and also understood their advantages and disadvantages . Different splitting techniques for dividing the data into training and testing are implemented and the best is selected.As the lectures were mostly focused on classification we have learnt many new regression methods. Different evaluation metrics for the regression are studied and all the models are evaluated based on those metrics. Many regression models their working, pros and cons is understood. In order to get an hands on working we converted the regression problem into classification and evaluated the classification models based on the metrics discussed in the class. Finally we have learnt the flow of the machine learning process with a real time dataset gaining practical knowledge.

IV. Conclusion

Predicting the popularity of the article is very useful for the content providers and company and this study can help many companies to publish news that will be popular. We have applied different regression models and were able to get good results building a good model. Turning the prediction into classification we have we were also able to classify the article to be popular or not popular and achieved an average accuracy of 0.64 using random forest. The results would have been better if the dataset has more features to work on rather than the simple categorical data. There is no explanation on how the Natural language processing is applied to the articles and how the values like polarity, LDA are calculated would have been good if those were known as it would been helpful in the feature extraction. Moreover the dataset is limited to the mashable website and may not be accurate for the other websites with slight change of content or genre. Even though we have achieved a less accuracy we can enhance it by making some changes at the stage of data collection and also by implementing many more ensemble models. Finally there is still a room ,where we can try to achieve more accuracy.

.

V. Future Work

We can extend this work by implementing Kfold validation on all the regression models which might result in better prediction of values . Apply ensemble algorithms by mixing two or more algorithms , this will also improve the model. Try to get data from other websites and also include some of the features like the author popularity , uniqueness of the article and also applying better natural language processing for the feature extraction of the article.

VI. References

[1] Fernandes, K., Vinagre, P. and Cortez, P., 2015, September. A proactive intelligent decision support system for predicting the popularity of online news. In Portuguese Conference on Artificial Intelligence (pp. 535-546). Springer, Cham.

[2] Shreyas, R., Akshata, D.M., Mahanand, B.S., Shagun, B. and Abhishek, C.M., 2016, August. Predicting popularity of online articles using random forest regression. In 2016 Second International Conference on Cognitive

Computing and Information Processing (CCIP) (pp. 1-5). IEEE.

[3] Ren, H. and Quan, Y., 2015. Predicting and Evaluating the Popularity of Online News. Standford University Machine Learning Report.

[4] Harrington, P., 2012. Machine learning in action (Vol. 5). Greenwich: Manning.

[5] Hall, M.A., 2000. Correlation-based feature selection of discrete and numeric class machine learning.

[6] Chmielewski, M.R. and Grzymala-Busse, J.W., 1996. Global discretization of continuous attributes as preprocessing for machine learning. International journal of approximate reasoning, 15(4), pp.319-331.

[7] Wong, T.T., 2015. Performance evaluation of classification algorithms by k-fold and leave-one-out cross validation. Pattern Recognition, 48(9), pp.2839-2846.

[8] https://www.analyticsvidhya.com/blog/2017/06/a-comprehensive-guide-for-linear-ridge-and-lasso-regression/

[9] https://medium.com/usf-msds/choosing-the-right-metric-for-machine-learning-models-part-1-a99d7d7414e4

[10] https://github.com/LJANGN/Predicting-the-Popularity-of-Online-News/blob/master/onlinenewspopularity-version_1.ipynb

[11] https://towardsdatascience.com/introduction-to-bayesian-linear-regression-e66e60791ea7

[12] https://machinelearningmastery.com/linear-regression-for-machine-learning/

[13] https://scikit-learn.org/stable/

[14] https://pandas.pydata.org/

[15] https://www.anaconda.com/

[16] Rizos, G., Papadopoulos, S. and Kompatsiaris, Y., 2016, April. Predicting news popularity by mining online discussions. In Proceedings of the 25th International Conference Companion on World Wide Web (pp. 737-742). International World Wide Web Conferences Steering Committee.