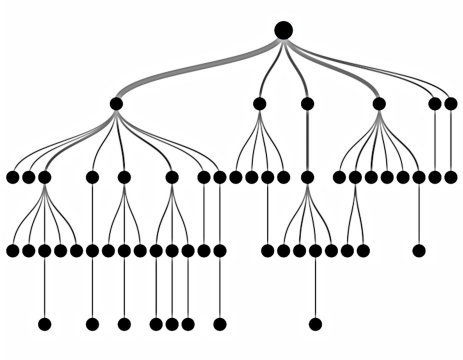
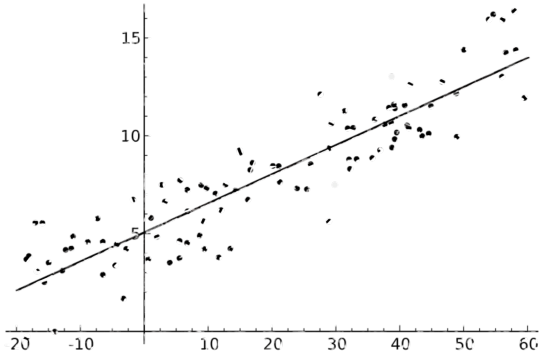
PREDICTING ONLINE NEWS POPULARITY

SPRINGBOARD CAPSTONE PROJECT PRESENTATION





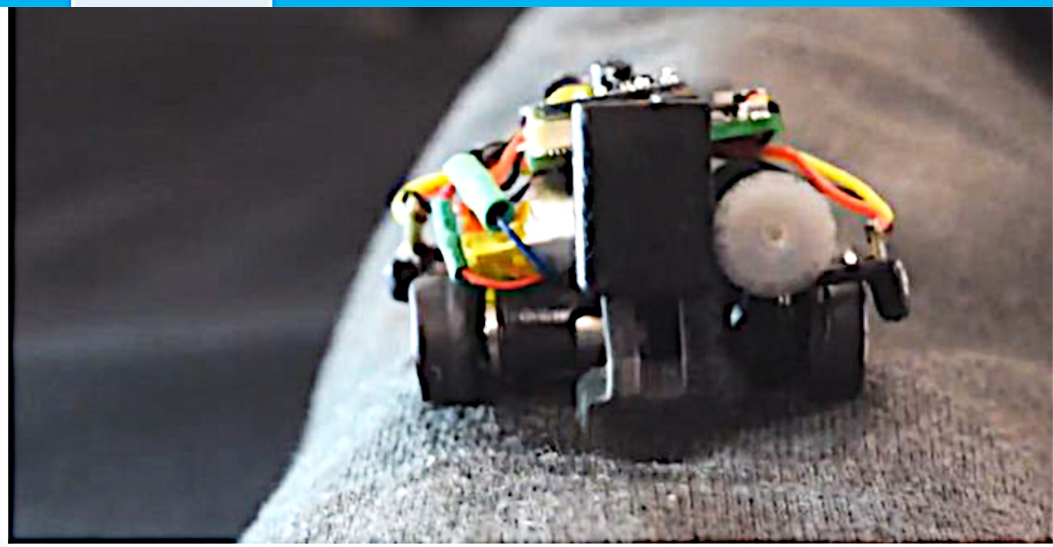
By
Tushar Prakash

INTRODUCTION

- The age of social media has brought along with it a major disruption in how news content is disseminated and consumed.
- For many, Facebook and Twitter are now the most common sources of news.
- Digital Media companies that deal specifically with curating online news and entertainment content from around the web are on the rise.
- To select the best content, they need to have an ability to accurately predict the number of shares a news article may get on social media.
- Mashable.com is one such website, that combines news and entertainment content from around the world.

ENTERTAINMENT

The best 'Civilization VI' leaders for all four victory types



TECH

Tiny wearable robots drive around your body, are extra pair of hands

Robot minions

OBJECTIVE

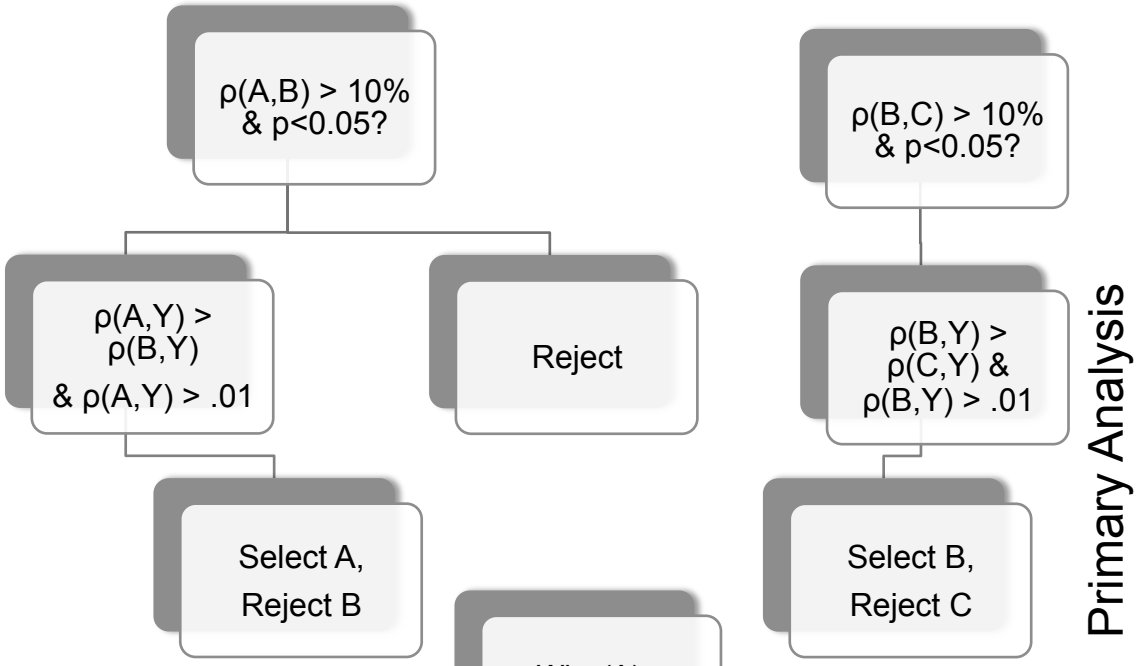
- This project deals with predicting online news popularity, using data from Mashable.com.
- Data Set is taken from UCI Machine Learning Repository.
- The stated goal is to predict whether the number of shares for a news article will be less than or greater than 1400.
- I will go beyond the stated goal to study feasibility of predicting exact values, or a more granular classification.
- I will also try to find why Mashable has drawn the lines of it's classification problem at 1400.

DATA ANALYSIS & FEATURE SELECTION

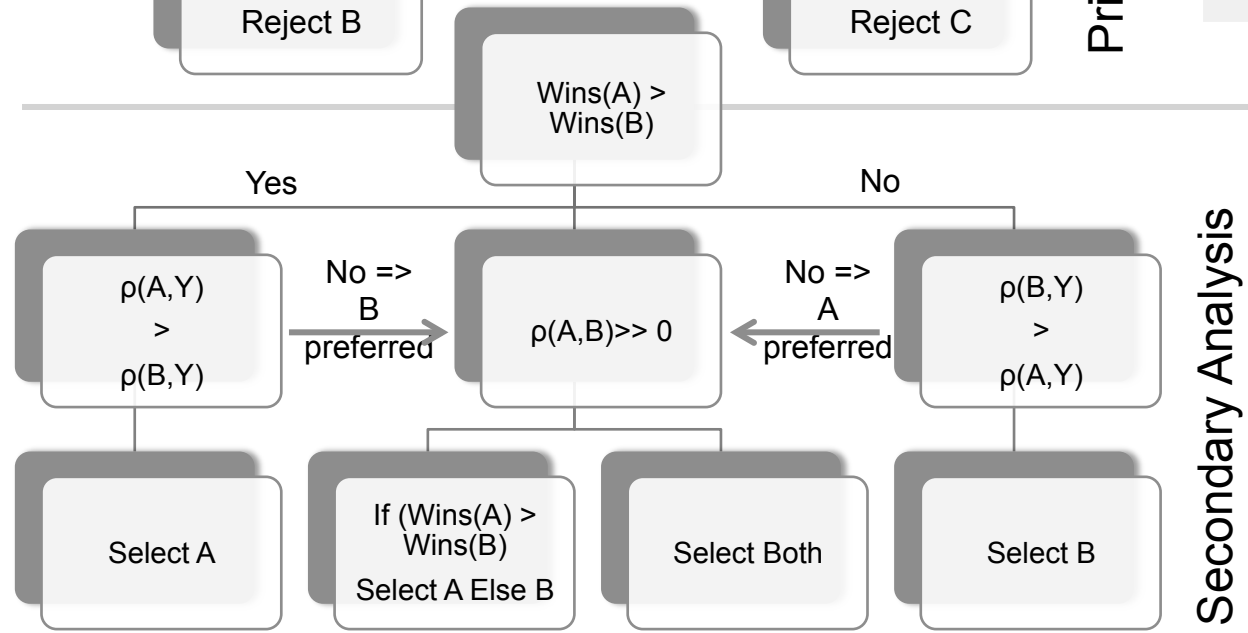
- Data contains over 39000 records containing 59 quantitative features
- Data is a combination of categorical and continuous variables
- Principal Component Analysis (PCA) was not suitable for feature selection due to multitude of data involved. I decided to forego Multi-Factor Analysis (MFA) as well due to concerns about over fitting the noise, and to keep the model interpretable.
- Devised a dimensionality reduction strategy involving correlation analysis.
- Using rcorr() function, obtained a correlation matrix. Flattened the matrix to obtain data in the below format

A	B	C	D	E	F	G	H
row	column	cor	p	rcor	ccor	selection	selection_Co
global_rate_positive_words	abs_title_sentiment_polarity	0.11969794	0	3.13682854	5.36513403	abs_title_sentiment_polarity	5.365134031
avg_positive_polarity	abs_title_sentiment_polarity	0.1173523	0	5.16295694	5.36513403	abs_title_sentiment_polarity	5.365134031
title_sentiment_polarity	abs_title_sentiment_polarity	0.48374337	0	4.23559248	5.36513403	abs_title_sentiment_polarity	5.365134031
abs_title_subjectivity	abs_title_sentiment_polarity	-0.4730971	0	0.03870663	5.36513403	abs_title_sentiment_polarity	5.365134031
n_tokens_content	avg_positive_polarity	0.10197602	0	0.06538966	5.16295694	avg_positive_polarity	5.162956938
n_unique_tokens	avg_positive_polarity	0.13929011	0	-1.3243237	5.16295694	avg_positive_polarity	5.162956938
n_non_stop_unique_tokens	avg_positive_polarity	0.13682675	0	-4.4260018	5.16295694	avg_positive_polarity	5.162956938
global_sentiment_polarity	avg_positive_polarity	0.51131201	0	3.62135507	5.16295694	avg_positive_polarity	5.162956938
global_rate_positive_words	avg_positive_polarity	0.24180487	0	3.13682854	5.16295694	avg_positive_polarity	5.162956938
global_rate_negative_words	avg_positive_polarity	0.15009199	0	0.3735878	5.16295694	avg_positive_polarity	5.162956938
rate_positive_words	avg_positive_polarity	0.18922739	0	0.31772605	5.16295694	avg_positive_polarity	5.162956938
avg_positive_polarity	min_positive_polarity	0.35672152	0	5.16295694	-0.3094281	avg_positive_polarity	5.162956938
avg_positive_polarity	max_positive_polarity	0.61961991	0	5.16295694	4.36378308	avg_positive_polarity	5.162956938
avg_positive_polarity	avg_negative_polarity	-0.2205654	0	5.16295694	-3.7841301	avg_positive_polarity	5.162956938
avg_positive_polarity	min_negative_polarity	-0.1786767	0	5.16295694	-2.6955815	avg_positive_polarity	5.162956938
avg_positive_polarity	max_negative_polarity	-0.1071208	0	5.16295694	-1.1717788	avg_positive_polarity	5.162956938

DIMENSION REDUCTION SCHEME

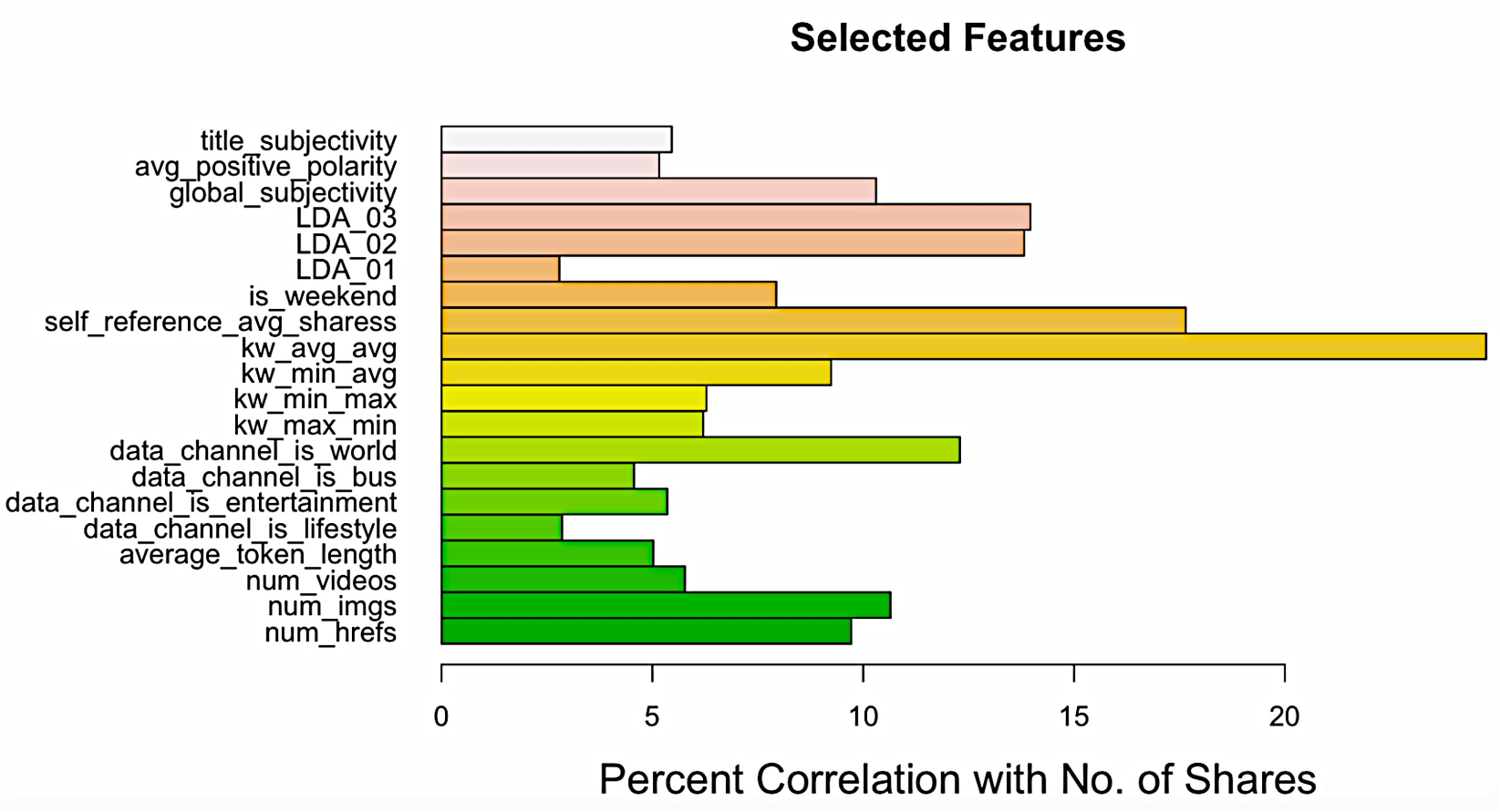


A, B, C	Independent Variables
Y	Dependent Variable
P Value	Statistical significance level of the correlation
ρ(X,Y)	Absolute value of correlation between variables X & Y
Wins(X)	No. of instances when X is the selected variable in primary analysis



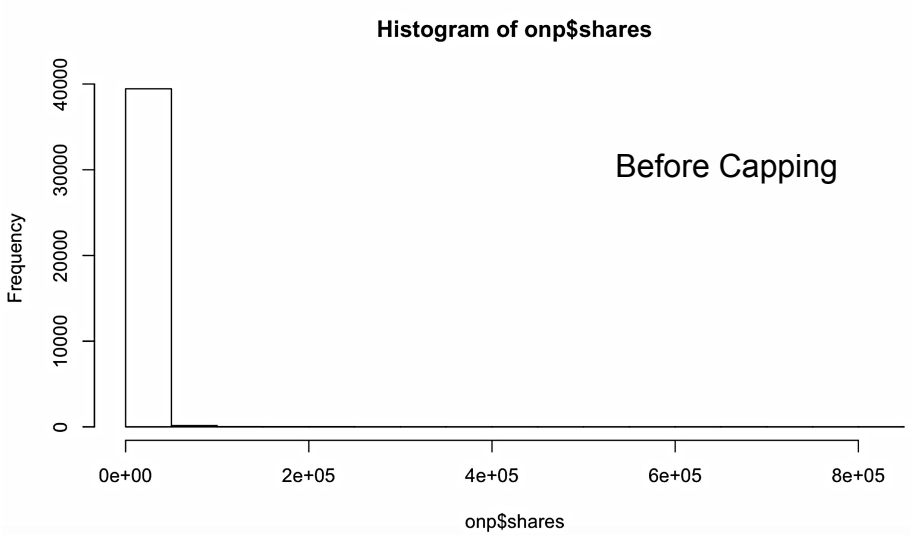
FEATURE SELECTION RESULTS

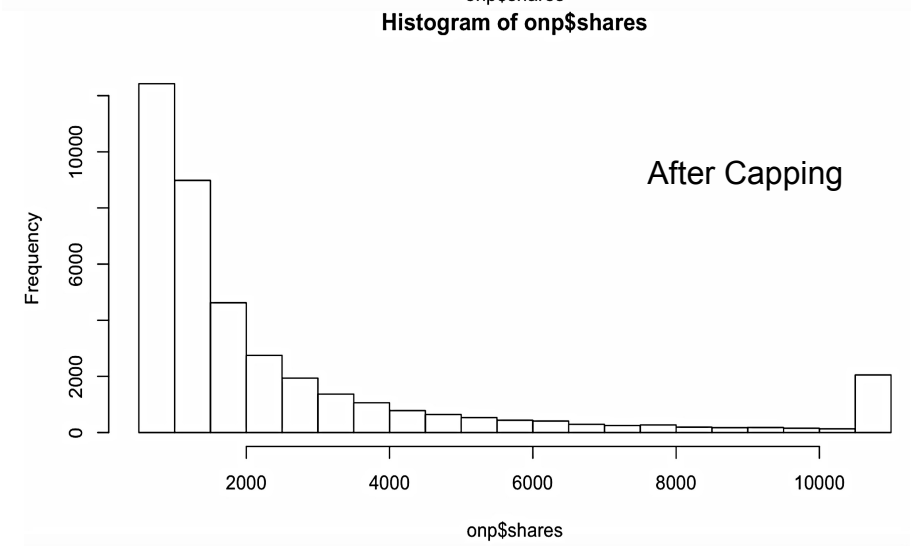
The scheme described previously is a general guiding theme of the analysis. Final list of features selected was a result of empirical and subjective valuation of the features



DATA PREPARATION

- **Capping**
  - Quantitative variables are capped to 5% and 95% quantiles
  - Outliers removed in the process as can be seen in the histogram
- **Scaling**
  - Continuous variables are scaled including target variable
- **Classification**
  - Binary classification based on 50-50 quantile distribution
  - 4-way classification based on 30-30-20-20 distribution
- **Factorization**
  - Categorical variables are factorized including class variables
- **Splitting**
  - Data is split into training and test set based on a 75-25 split ratio





Effects of Capping: Values are much better distributed after capping

DATA MODELING

- **Methods used for modeling**
  - Linear Regression for predicting exact values
  - Logistic Regression for binary classification
  - CART learning model using k-fold cross-validation on a binary classification
  - Random Forest learning model for both binary and 4-class problems
- **Results from each method**
  - Linear regression produced a low R-squared of 10% implying that trying to predict exact values is not fruitful.
  - Logistic Regression preformed better than CART model as a binary classifier.
  - Random forest gave the best results for binary classification.

Linear regression represents a least-square fit of the response to the data. It chooses the hypothesis

$$h_\theta(x) = \sum_{i=0}^{n} \theta_i x_i$$

by minimizing the cost function

$$J(\theta) = \frac{1}{2}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2$$

For logistic regression, the hypothesis is

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

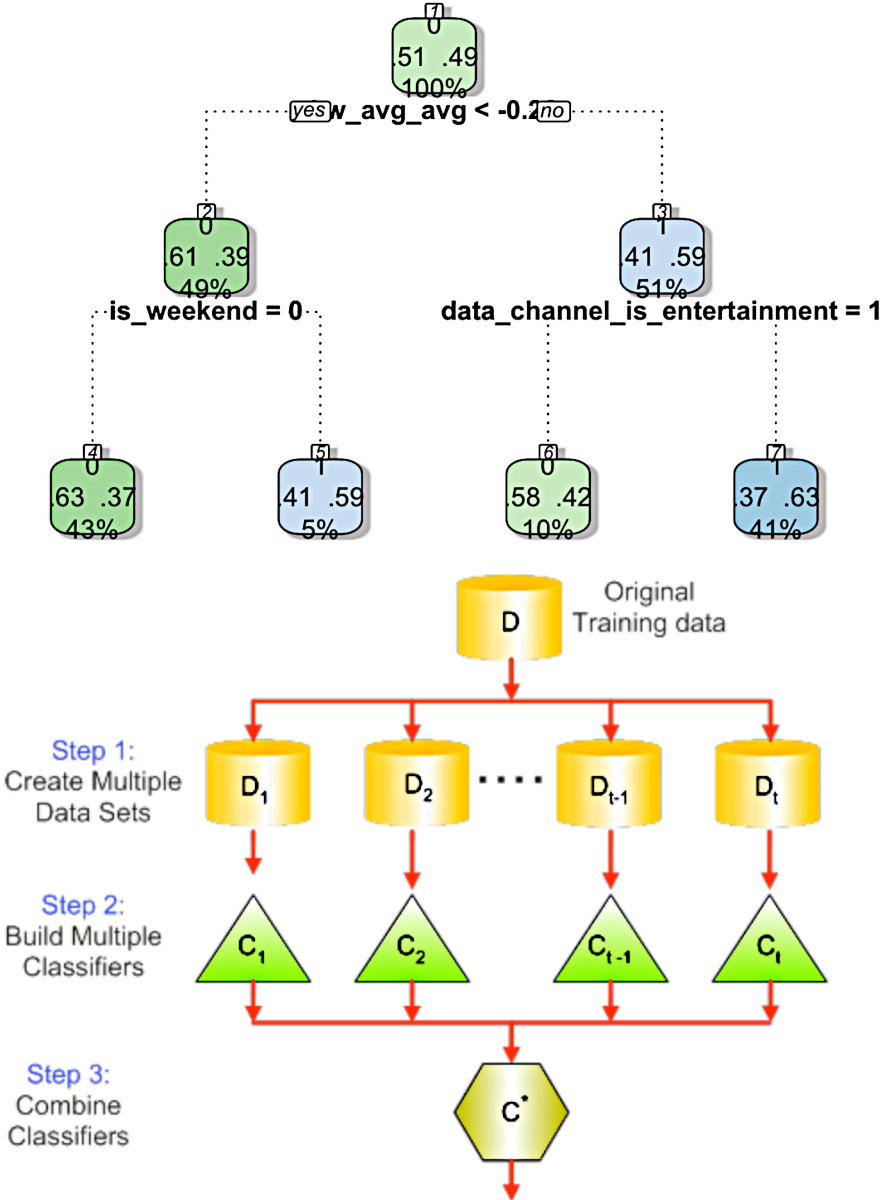
and parameters are chosen as to maximize their likelihood

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

CART (Classification & Regression Trees) is a decision tree based supervised learning model. A decision tree splits the sample into two or more homogeneous sets at every node (decision node), based on most significant splitter / differentiator in input variables, to reach a set of pre-defined classes (leaf node).
CART can be fine tuned by using a k-fold cross validation method that uses bootstrapping.

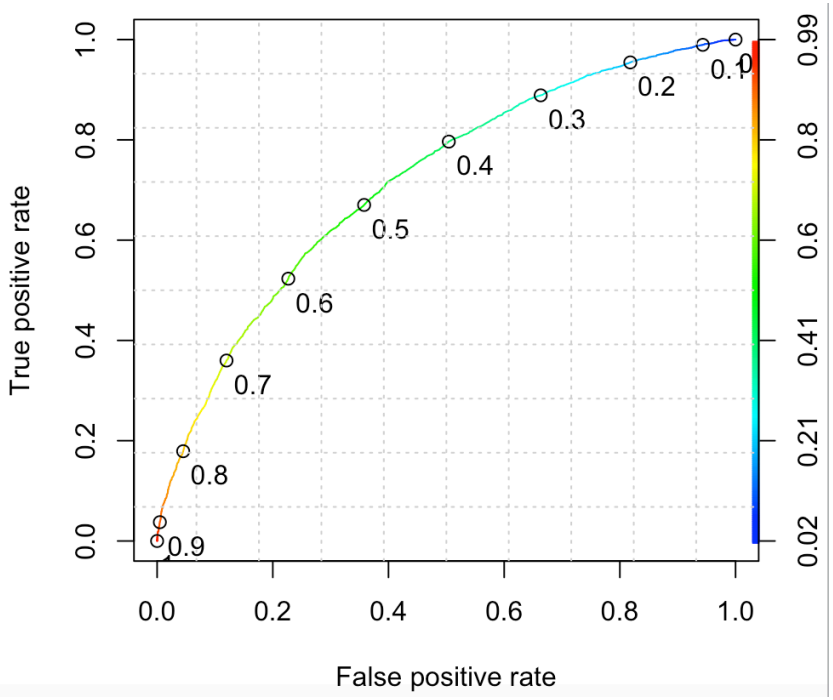
RANDOM FOREST CLASSIFIER

- Random Forest uses multiple decision trees which are built on separate sets of examples drawn from the dataset. Figure on the top right shows the actual decision tree obtained from the CART model used previously.
- Random Forest uses a technique called bagging. For each tree, we use a subset of all the features we have (with replacement). By using more decision trees and averaging the result, the variance of the model can be greatly lowered as illustrated in the figure on bottom right.
- Model parameters such as no. of trees to use, minimum leaf node size, which can govern the tree depth, and no. of features to sample at a time can all be customized and tuned.

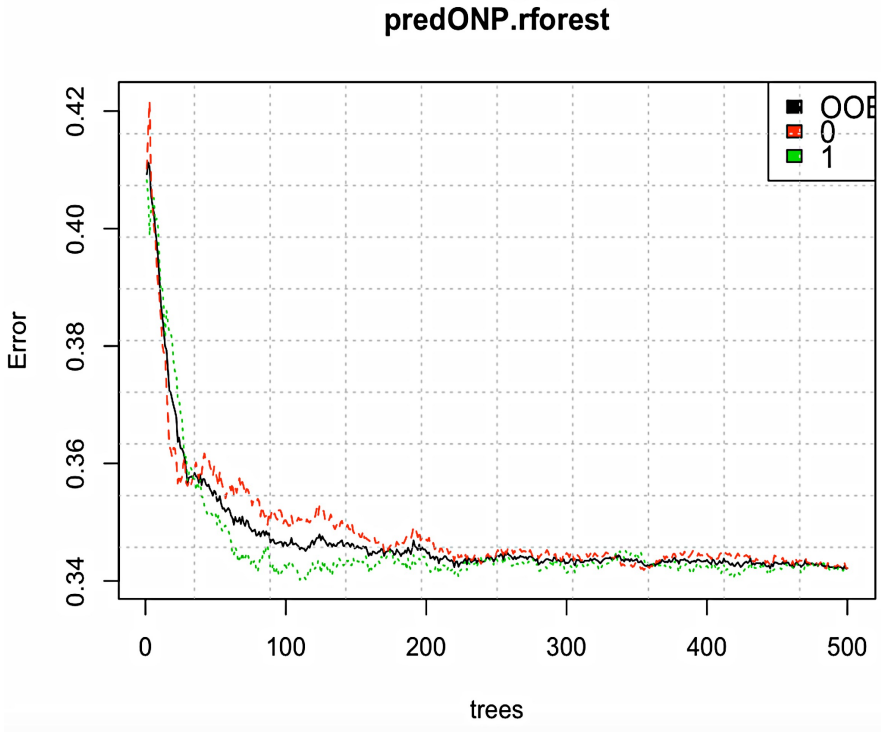


RESULTS FROM RANDOM FOREST CLASSIFIER

- Figure below shows the ROC curve showing True Positive Rate vs. False Positive Rate from at different probability thresholds as shown in color gradation.
- Area Under Curve (AUC) was 71.6%.
- Accuracy obtained was 65.5%-65.9%



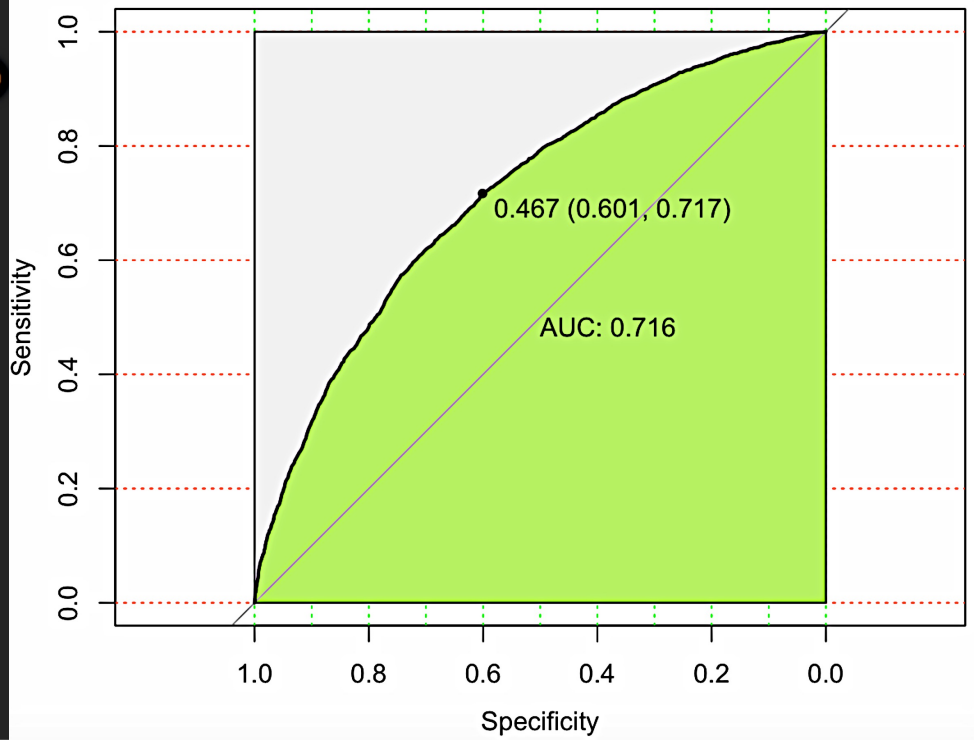
- Figure below shows the mean error of the two classes (red and green) and Out-of-Bag error (black).
- Error decreases with increasing number of trees (*ntree* parameter) and converges to a minimum at a value of 500 trees.



RESULTS FROM RANDOM FOREST CLASSIFIER

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 3238 1590
         1 1797 3286

               Accuracy : 0.6583
                 95% CI : (0.6488, 0.6676)
    No Information Rate : 0.508
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.3168
 Mcnemar's Test P-Value : 0.0004007

            Sensitivity : 0.6431
            Specificity : 0.6739
         Pos Pred Value : 0.6707
         Neg Pred Value : 0.6465
             Prevalence : 0.5080
         Detection Rate : 0.3267
   Detection Prevalence : 0.4871
      Balanced Accuracy : 0.6585
```

- On the left, a screenshot showing performance parameters for Random Forest binary classifier.
- Below, another ROC curve for the same model showing sensitivity (TPR) vs. specificity (TNR). Curve shows the values on the best threshold as per "**youden**" method, which maximizes distance from identity line (diagonal).



RESULTS FROM RANDOM FOREST 4-CLASS CLASSIFIER

- Measuring performance statistics on Multi-Class Classifiers is not well established
- Accuracy was measured here using the Exact Match Ratio. For an n-class problem this would be

$$\Sigma_1^n x_{ii} / \Sigma_1^n \Sigma_1^n x_{ij}$$

where *x* is an element of confusion matrix for the n-class model.

- AUC was measured using MAUC (Multi-Class AUC) method. MAUC calculates AUC for each class individually against all the other classes and averages the individual AUCs.

$$\mathbf{MAUC = \Sigma_1^n AUC(Class_i) / n}$$

```
> confusion.matrix
   PredictONP.forest.4class
       1    2    3    4
 1 1948  865   22  230
 2 1171 1373   76  404
 3  472  852  111  402
 4  438  732  104  711
```

```
> mauc(TestONP$classes, Predicted.4class)
$mauc
[1] 0.660704

$auc
[1] 0.7258919 0.5772050 0.6251122 0.7146068
```

- Accuracy obtained was 41.8 %
- MAUC obtained was 66.07 %

ALL MODEL SUMMARY

Linear Regression	Multiple R-squared: 0.0971 Adjusted R-squared: 0.09643 F-statistic: 145.2 on 22 and 29710 DF p-value: < 2.2e-16
Logistic Regression	Accuracy: 64.71 % Sensitivity: 65 % Specificity: 64.4 % AUC: 69.31 %
CART learning model	Accuracy: 61.6 % Sensitivity : 65.94 % Specificity : 57.12 % AUC: 62.4 %
Random Forest (2-class)	Accuracy: 65.83 % Sensitivity: 64.31 % Specificity: 67.39 % AUC: 71.6 %
Random Forest (4-class)	Accuracy: 41.8 % MAUC: 66.07 %

CONCLUSION

Recap of the project

- Took the data set from UCI Machine Learning Repository, Capped, Factored and Scaled the data and ran it through a rigrous dimensionality reduction process.
- Split the data into training and test sets, and tested 4 different regression and classification models on the data.
- Tuned the models and compared them on key performance metrics to get the best model that fits the objectives of this project.

Challenges Faced	Solution
Data had a large number of mixed variables and was unsuitable for PCA, MFA	Cleaned the data and devised my own dimension reduction strategy to get best representative features
Did not fully understand the technical complexities of the models	Dug deep into literature review, studied pros and cons of the modeling algorithms, and techniques to tune the model parameters
Model results were not easy to interpret in the raw form	Explored and used various data visualization techniques

ACKNOWLEDGEMENT & REFERENCES

Acknowledgements:

- Anirban Ghosh, my Springboard Mentor. Thank you for all your support and guidance.
- The faceless online community of data scientists and R users and bloggers. Whatever question one might have, someone has asked it before!

Resources:

- He Ren, Quan Yang. “Predicting and Evaluating the Popularity of Online News”. Link
- Johannes Ledolter, “Data Mining and Business Analytics with R”
- Rui Wang, Ke Tang, **“**An Empirical Study of MAUC in Multi-class Problems with Uncertain Cost Matrices”. Link
- R-package library at CRAN (https://cran.r-project.org/web/packages/) and other places.
- Springboard student resources