EE 559

FINAL PROJECT

BY

PRANAV GUNDEWAR

USC ID: 4463612994

EMAIL: gundewar@usc.edu

ADITYA KILLEKAR

USC ID: 20514504147

EMAIL: killekar@usc.edu

GUIDED BY

PROF. KEITH JENKINS

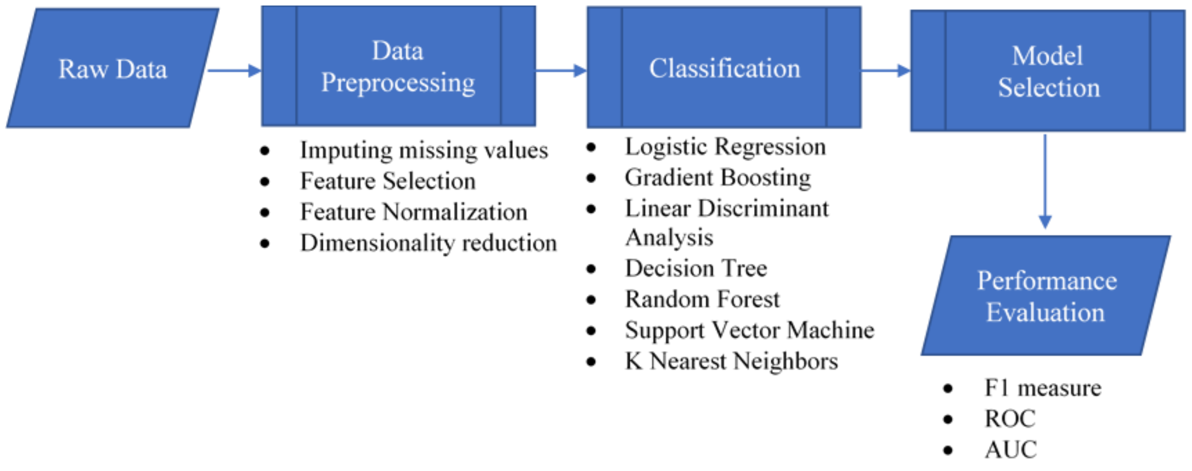
Table of Contents

ABSTRACT

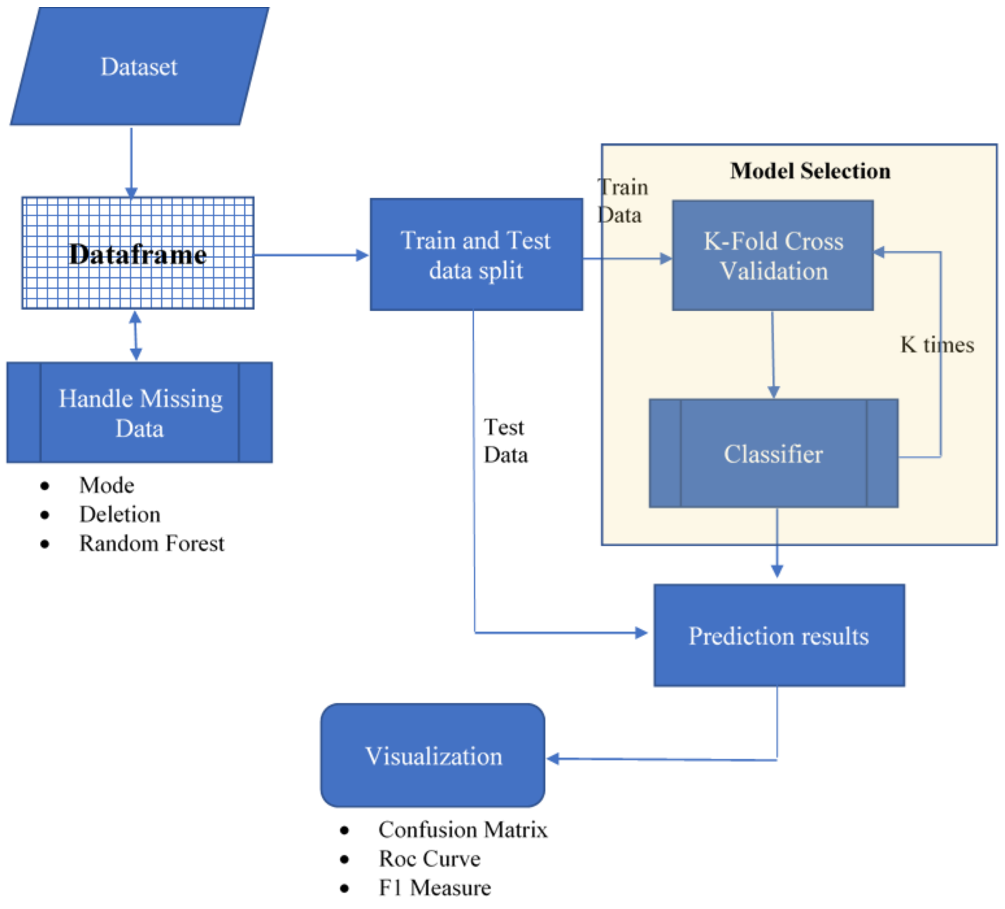
In our project, we develop a pattern recognition system that operates on the real-world datasets: Bank Marketing dataset and Online News Popularity. We implement different data imputing techniques to handle missing data, different classifiers, different feature selection techniques. The evaluation of the model is done on the following metrics: (i) For Bank Marketing dataset, F1 measure and ROC score/AUC, (ii) For Online news dataset, F1 measure and Accuracy. Finally, we conclude selecting the best model based on the above evaluation metrics. We present the results of various different experimental settings and highlight our best results.

SYSTEM OVERVIEW

The entire system block diagram can be described as



The overview of approach can be observed in the figure. The details of each block and its sub parts has been explained in following sections.



DATA INTERPRETATION

BANK MARKETING DATA SET

MARKETING INTRODUCTION

The method by which companies make value for clients and construct solid client connections in arrange to capture value from clients in return.
Marketing campaigns are characterized by centering on the client needs and their overall fulfillment. In any case, there are distinctive factors that decide whether a promoting campaign will be effective or not. There are certain factors that we got to take into thought when making a marketing campaign.
The data is related with direct marketing campaigns of a Portuguese banking institution. The marketing campaigns were based on phone calls. Often, more than one contact to the same client was required, in order to access if the product (bank term deposit) would be ('yes') or not ('no') subscribed.
The classification goal is to predict if the client will subscribe (yes/no) a term deposit (variable y).
Written in Python with scikit-learn and pandas libraries.
Data are imported as pandas dataframe, filtered by intact records, encoded on categorical variables using onehot scheme, normalized using different scaler, and divided into training and testing datasets. Various machine learning algorithms have been implemented and their performance has been compared on the basis on different parameters.

ATTRIBUTE DESCRIPTION

I. Bank Client Data:

1 - **age:** (numeric)
2 - **job:** type of job (categorical: 'admin.','blue-collar','entrepreneur','housemaid','management','retired','self-employed','services','student','technician','unemployed','unknown')
3 - **marital:** marital status (categorical: 'divorced','married','single','unknown'; note: 'divorced' means divorced or widowed)
4 - **education:** (categorical: primary, secondary, tertiary and unknown)
5 - **default:** has credit in default? (categorical: 'no','yes','unknown')
6 - **housing:** has housing loan? (categorical: 'no','yes','unknown')

7 - **loan:** has personal loan? (categorical: 'no','yes','unknown')
8 - **balance:** Balance of the individual.

II. Related with the last contact of the current campaign:

8 - **contact:** contact communication type (categorical: 'cellular','telephone')
9 - **month:** last contact month of year (categorical: 'jan', 'feb', 'mar', ..., 'nov', 'dec')
10 - **day:** last contact day of the week (categorical: 'mon','tue','wed','thu','fri')

III. Other Attributes:

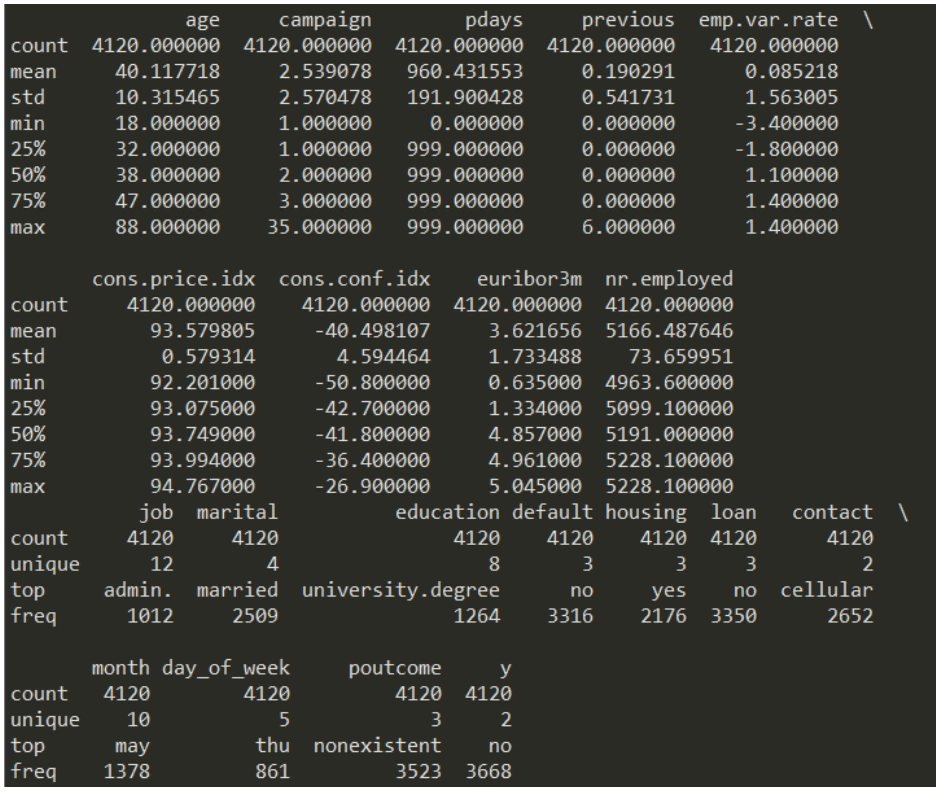
11 - **campaign:** number of contacts performed during this campaign and for this client (numeric, includes last contact)
12 - **pdays:** number of days that passed by after the client was last contacted from a previous campaign (numeric; 999 means client was not previously contacted)
13 - **previous:** number of contacts performed before this campaign and for this client (numeric)
14 - **poutcome:** outcome of the previous marketing campaign (categorical: 'failure','nonexistent','success')

IV. Social and Economic Context Attributes:

15 - **emp.var.rate**: employment variation rate - quarterly indicator (numeric)
16 - **cons.price.idx**: consumer price index - monthly indicator (numeric)
17 - **cons.conf.idx**: consumer confidence index - monthly indicator (numeric)
18 - **euribor3m**: euribor 3 month rate - daily indicator (numeric)
19 - **nr.employed**: number of employees - quarterly indicator (numeric)
20 -**Subscription:** Has client subscribed a term deposit? – class labels (binary: 'no','yes')

DATA ANALYSIS AND VISUALIZATION

For the given dataset, we can divide the features into continuous and categorical features.
Continuous Data: Index (['age', 'campaign', 'pdays', 'previous', 'emp.var.rate', 'cons.price.idx', 'cons.conf.idx', 'euribor3m', 'nr.employed'],
 dtype='object')
Categorical Data: Index (['job', 'marital', 'education', 'housing', 'loan', 'contact', 'month', 'day_of_week', 'poutcome'],
 dtype='object')



```
              age     campaign        pdays     previous  emp.var.rate  \
count  4120.000000  4120.000000  4120.000000  4120.000000   4120.000000
mean     40.117718     2.539078   960.431553     0.190291      0.085218
std      10.315465     2.570478   191.900428     0.541731      1.563005
min      18.000000     1.000000     0.000000     0.000000     -3.400000
25%      32.000000     1.000000   999.000000     0.000000     -1.800000
50%      38.000000     2.000000   999.000000     0.000000      1.100000
75%      47.000000     3.000000   999.000000     0.000000      1.400000
max      88.000000    35.000000   999.000000     6.000000      1.400000

       cons.price.idx  cons.conf.idx    euribor3m  nr.employed
count     4120.000000    4120.000000  4120.000000  4120.000000
mean        93.579805     -40.498107     3.621656  5166.487646
std          0.579314       4.594464     1.733488    73.659951
min         92.201000     -50.800000     0.635000  4963.600000
25%         93.075000     -42.700000     1.334000  5099.100000
50%         93.749000     -41.800000     4.857000  5191.000000
75%         93.994000     -36.400000     4.961000  5228.100000
max         94.767000     -26.900000     5.045000  5228.100000
            job  marital          education default housing  loan   contact  \
count      4120     4120               4120    4120    4120  4120      4120
unique       12        4                  8       3       3     3         2
top      admin.  married  university.degree      no     yes    no  cellular
freq       1012     2509               1264    3316    2176  3350      2652

       month day_of_week     poutcome     y
count   4120        4120         4120  4120
unique    10           5            3     2
top      may         thu  nonexistent    no
freq    1378         861         3523  3668
```

Figure 1 – Data Description

From figure 1, we get the basic idea about type of data, its distribution. We can get the better idea using data plots.

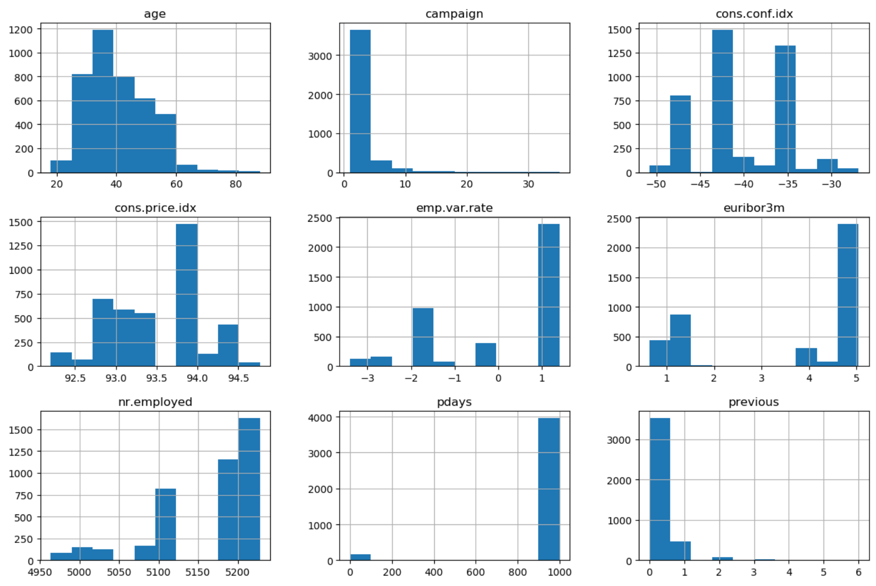


Figure 2 – Histogram of Continuous Data

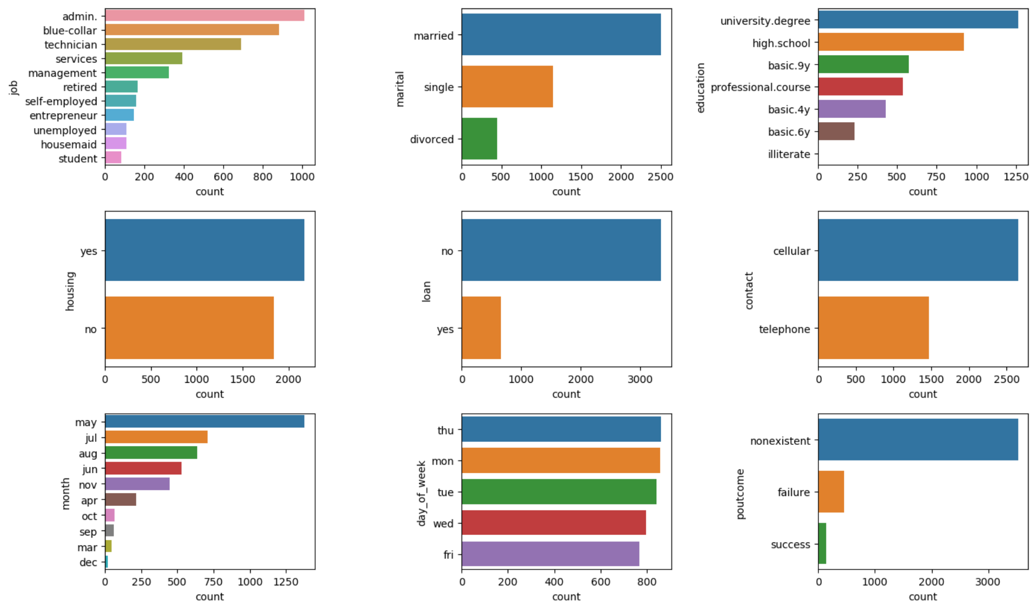


Figure 3 – Count plot of Categorical Data

By analyzing visual plots from figure 2 and 3, we can decide what are the necessary steps required for pre-processing.

Data Analysis:

Questions to determine:

1. What **percentage(%)** of potential clients accepted to subscribe to term deposits vs refused to subscribe term deposits.
2. Is there a huge difference between clients that **subscribed term deposits vs refused deposits?**

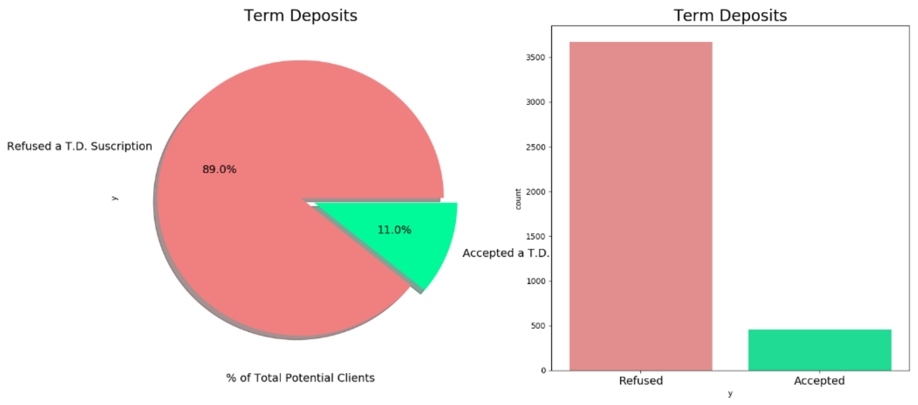


Figure 4 – Class labels density

Summary:

89% refused to subscribe to term deposits while **11% accepted** to subscribe term deposits**.** We can conclude that **data imbalance** is observed in the data which we have to take care in preprocessing.

Next Steps:
The obvious step would be to detect patterns that influence potential clients to either accept or refuse to buy term deposits. After analyzing the data, we can start with data pre-processing.

ONLINE NEWS POPULARITY DATA SET

NEWS POPULARITY INTRODUCTION

The dataset summarizes a set of features about articles published by Mashable, a well-known news website over a period of two years. The objective is to predict the number of shares depending on the features if the article to be published would be popular on the internet or no.
The reduced data has 61 attributes in all including the response attribute "shares" and 4954 observations.
Written in Python with scikit-learn and pandas libraries.
Data are imported as pandas dataframe, filtered by intact records, encoded on categorical variables using onehot scheme, normalized using different scaler, and divided into training and testing datasets. Various machine learning algorithms have been implemented and their performance has been compared on the basis on different parameters.

ATTRIBUTE DESCRIPTION

The attributes are explained below (Source: UCI Machine Learning Repository):

Col	Attribute	Information
0	url	URL of the article (non-predictive)
1	Timedelta	Days between the article publication and the dataset acquisition (non-predictive)
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_shares	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 2
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)

DATA ANALYSIS AND VISUALIZATION

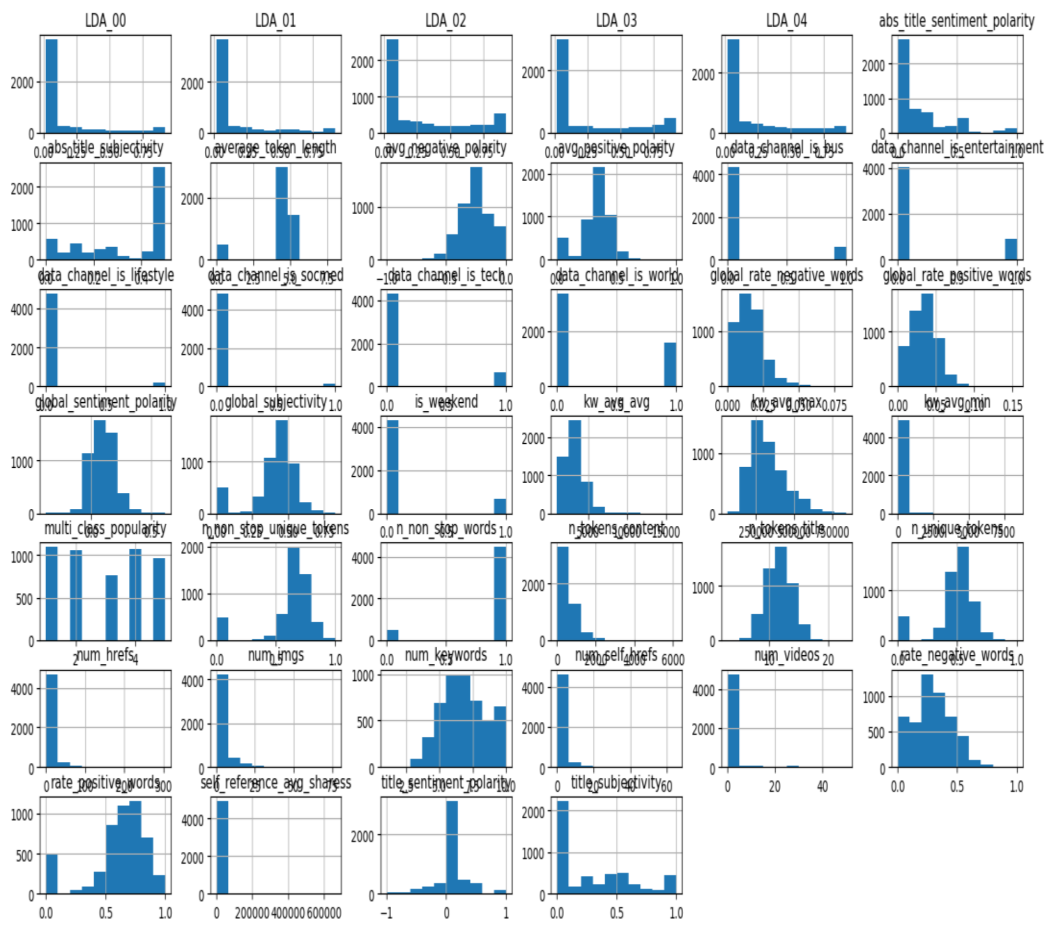


Figure 5 – Count plot of Data

Data Analysis:

Questions to determine:

1. What **percentage(%)** of news will be popular depending upon value of shares.
2. Is there a huge difference between news that are **popular vs non-popular?**

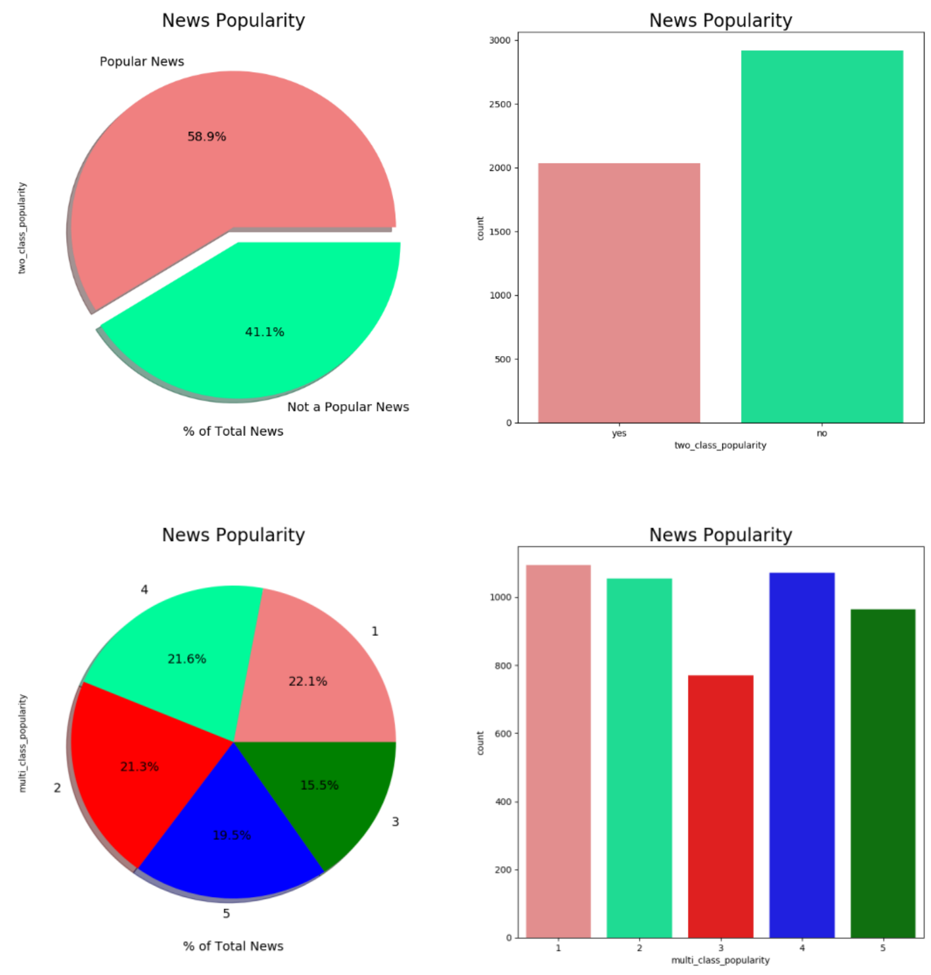


Figure 6 – Class labels density

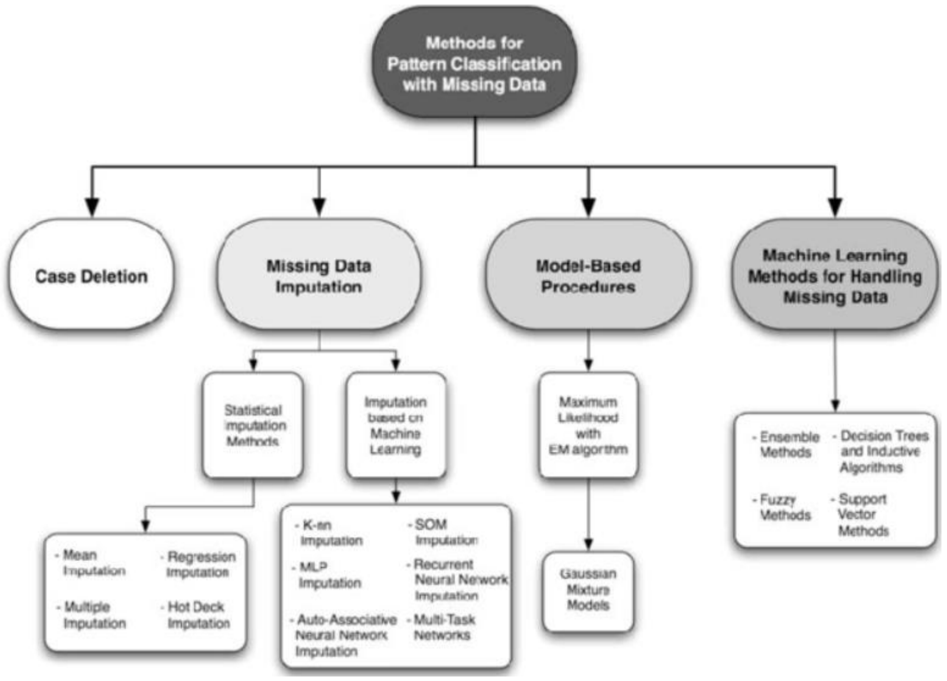
From given data set class labels, we can observe that **data imbalance is NOT present.**
Next Steps:
The obvious step would be to detect patterns that influence potential clients to either accept or refuse to buy term deposits. After analyzing the data, we can start with data pre-processing.

DATA PREPROCESSING

After data visualization, we observe certain pattern which might affect the classification. From data interpretation, we can interpret that data has imbalance due to unequal percentage of class labels.
Categorical features in data set has missing values. Hence data imputation should be precisely to tackle these
Data needs scaling because it has very high variation from mean. Due to categorical features, we need to do one-hot encoding.
After one-hot encoding, we can observe high dimensional data. So, I have used PCA for dimension reductionality.

DATA IMPUTATION

There are different solutions to tackle data imputation depending upon applications and type of data that you are imputing. From figure 4, we can see available ways to handle the data.



Methods for Pattern Classification with Missing Data
Case Deletion
Missing Data Imputation
Model-Based Procedures
Machine Learning Methods for Handling Missing Data
Statistical Imputation Methods
Imputation based on Machine Learning
Maximum Likelihood with EM algorithm
- Ensemble Methods
- Decision Trees and Inductive Algorithms
- Fuzzy Methods
- Support Vector Methods
- Mean Imputation
- Regression Imputation
- Multiple Imputation
- Hot Deck Imputation
- K-nn Imputation
- SOM Imputation
- MLP Imputation
- Recurrent Neural Network Imputation
- Auto-Associative Neural Network Imputation
- Multi-Task Networks
Gaussian Mixture Models

Figure 4 – Data Imputation Methods

Median-

For median based imputing, for every feature, we change the missing value by its median of that column. Note that this method is based on feature and does not depend on class labels. This is one of the simplest way to handle data. Disadvantage of this method is we cannot impute categorical data using median method.

Mode:

Mode based imputing is similar to median where imputing is based on feature wise imputing. Mode is one of the most used and better way to imputer the data.

Random Forest:

Unlike statistical based imputing methods, we can impute data using machine learning algorithms like random forest. For every class, we convert categorical data into multivalued labels using label encoder. The features that has missing values are now used as class labels. All the samples with the unknown values are considered as the test data and the model is trained on the remaining samples. We then use RF classifier to predict the class values for missing.

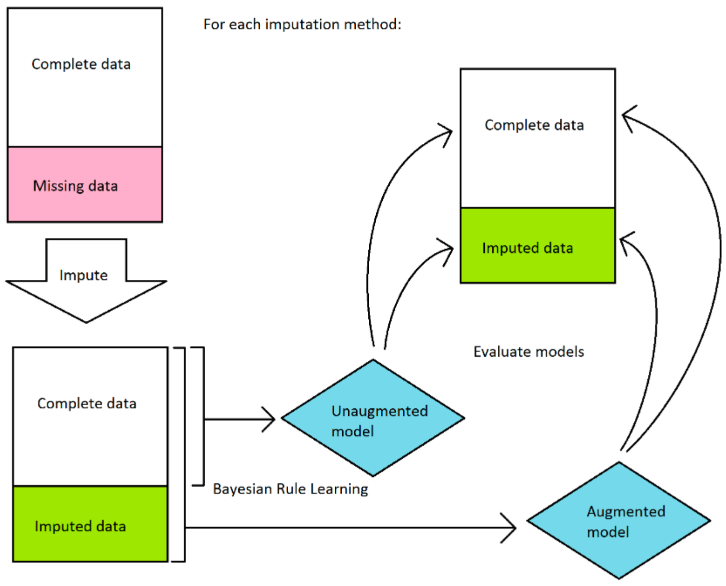


Figure 5 – Data Imputation Model

Support Vector Machines:

Just like RF classifier, we can SVM to predict the missing values of any categorical features.

Delete:

Another popular method for handling missing data is to ignore the points which has any unknowns in any features.

From figure 5, we can observe model flow for after handling missing values. I have used **Neural Network, SVM, mode and delete** methods for data imputing.

ENCODING

For categorical features, machine learning classifier cannot process the data. Hence, we need encode data in such a way that classifier understands its importance. One hot encoding is a process by which categorical variables are converted into a form that could be provided to ML algorithms to do a better job in prediction. I have used one-hot encoding. It is popular way to convert categorical data into n features with binary values. We use one hot encoder to perform "binarization" of the category and include it as a feature to train the model.

DATA IMBALANCE

Data imbalance occurs when we have more data for one class compared to another class. This is a scenario where the number of observations belonging to one class is significantly lower than those belonging to the other classes. In this situation, the predictive model developed using conventional machine learning algorithms could be biased and inaccurate. This happens because Machine Learning Algorithms are usually designed to improve accuracy by reducing the error. Thus, they do not take into account the class distribution / proportion or balance of classes.

<u>RANDOM UNDER-SAMPLING:</u>

Random Under sampling aims to balance class distribution by randomly eliminating majority class examples. This is done until the majority and minority class instances are balanced out.

- **Advantages:**
    1. It can help improve run time and storage problems by reducing the number of training data samples when the training data set is huge
- **Disadvantages:**
    1. It can discard potentially useful information which could be important for building rule classifiers
    2. The sample chosen by random under sampling may be a biased sample. And it will not be an accurate representative of the population. Thereby, resulting in inaccurate results with the actual test data set.

<u>RANDOM OVER-SAMPLING</u>

Over-Sampling increases the number of instances in the minority class by randomly replicating them in order to present a higher representation of the minority class in the sample.

- **Advantages:**

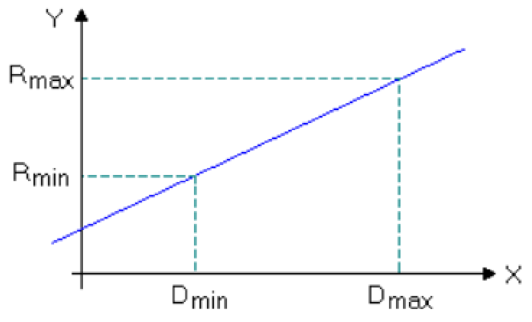
1. Unlike under sampling this method leads to no information loss.
2. Outperforms under sampling

- **Disadvantages:**
    1. It increases the likelihood of overfitting since it replicates the minority class events.

I have used under and over sampling using **randomoversampler()** and **randomundersampler()** and **SMOTEEN()** from imblearn library.

DATA SCALING

Feature scaling is a method used to standardize the range of independent variables or features of data. In data processing, it is also known as data normalization and is generally performed during the data preprocessing step.
Scaling transforms the values to another range which usually includes both a shift and a change of the scale (magnification, or reduction). The data samples are transformed according to the following equation:



$$Y = X\frac{R_{max} - R_{min}}{D_{max} - D_{min}} + \frac{R_{min}D_{max} - R_{max}D_{min}}{D_{max} - D_{min}}$$

Figure 6 – Data Scaling

Standardization (sometimes also called autoscaling, or z-transformation) is the scaling procedure which results in a zero mean and unit variance of any descriptor variable. For every data value the mean μ has to be subtracted, and the result has to be divided by the standard deviation σ (note that the order of these two operations must not be reversed):

$$Y = (X - \mu) / \sigma$$

I have used **StandardScalar, MinMaxScalar and RobustScalar** for data scaling and normalizing.

CLASSIFICATIONS

LINEAR DISCRIMINANT ANALYSIS

Linear Discriminant Analysis (LDA) is a classification method originally developed in 1936 by R. A. Fisher. It is simple, mathematically robust and often produces models whose accuracy is as good as more complex methods.
LDA is based upon the concept of searching for a linear combination of variables (predictors) that best separates two classes (targets). To capture the notion of separability, Fisher defined the following score function.

$$Z = \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_d x_d$$

Score Function,

$$S(\beta) = \frac{\beta^T \mu_1 + \beta^T \mu_2}{\beta^T C \beta}$$

$$S(\beta) = \frac{\overline{Z_1} - \overline{Z_2}}{Variance\ of\ Z\ within\ group}$$

Given the score function, the problem is to estimate the linear coefficients that maximize the score which can be solved by the following equations.

$$\beta = C^{-1}(\mu_1 - \mu_2)$$

Pooled Covariance matrix,

$$C = \frac{1}{n_1 + n_2}(n_1 C_1 + n_2 C_2)$$

where, β is the linear model coefficients, C_1,C_2 are the covariance matrices and μ_1,μ_2 are the mean vectors.
One way of assessing the effectiveness of the discrimination is to calculate the Mahalanobis distance between two groups. A distance greater than 3 means that in two averages differ by more than 3 standard deviations. It means that the overlap (probability of misclassification) is quite small.

$$\Delta^2 = \beta^T(\mu_1 - \mu_2)$$

Where, Δ is the Mahalanobis distance between the two groups.
Finally, a new point is classified by projecting it onto the maximally separating direction and classifying it as C1 if:

$$\beta^T\left(x - \left(\frac{\mu_1 + \mu_2}{2}\right)\right) > -\log\frac{\mathbb{P}(C_1)}{\mathbb{P}(C_2)}$$

Where, x is the new datapoint, $\mathbb{P}(C_1)$ and $\mathbb{P}(C_2)$ are the class probabilities.

DECISION TREE LEARNING

Decision Trees are a type of Supervised Machine Learning where the data is continuously split according to a certain parameter. The tree can be explained by two entities, namely decision nodes and leaves. The leaves are the decisions or the final outcomes. And the decision nodes are where the data is split.
There are two types of decision trees: Tree models where the target variable can take a discrete set of values are called classification trees, and Trees where the target variable can take continuous values are called regression trees.
There are many specific decision-tree algorithms, different algorithms use different metrics for measuring best split. Some examples of these metrics are: Gini impurity, Entropy, Variance reduction etc.

LOGISTIC REGRESSION

In logistic regression, we model the class conditional densities by requiring that,

$$\log\left(\frac{\mathbb{P}(1|x)}{\mathbb{P}(-1|x)}\right) = a^T x$$

Where, ‘a’ is a vector of parameters.
It is straight forward to estimate ‘a’ using maximum likelihood.
Note that,

$$\mathbb{P}(1|x) = \frac{e^{a^T x}}{1 + e^{a^T x}}$$
$$\mathbb{P}(-1|x) = \frac{1}{1 + e^{a^T x}}$$

So that we can estimate the correct set of parameters aˆ by solving for the minimum of the negative log-likelihood. That is,

$$\hat{a} = argmin_a \left[\sum_{i \in samples} \left(\frac{1 + y_i}{2}\right) a^T x - \log(1 + a^T x) \right]$$

This is a convex problem and can be easily solved by Newton’s method. In fact, when we use maximum likelihood, we are choosing a classifier boundary that minimizes the loss function.
For sample i, we write $\gamma_i = a^T x_i$. Our classifier will be,

$$choose \begin{cases} 1 & \gamma_i > 0 \\ -1 & \gamma_i < 0 \\ random & \gamma_i = 0 \end{cases}$$

Now write loss for the ith sample,

$$L(y_i, \gamma_i) = -\left[\frac{1}{2}(1 + y_i)\gamma_i - \log(1 + e^{\gamma_i})\right] = \log[1 + e^{-y_i\gamma_i}]$$

This loss is known as logistic loss. We can see from the above equation that the loss strongly penalizes a large positive γ_i if y_i is negative (or vice versa). However, there is no significant advantage to having a large positive γ_i if y_i is positive. This means that the significant component of loss function will be due to the samples that the classifier gets wrong, but also due to the samples that have γ_i near zero (that is, the sample is close to the decision boundary).
Now the total risk of applying this classifier to our set of examples is,

$$R(x) = \sum_{i\,\in samples} -\left[\frac{1}{2}(1 + y_i)\gamma_i - \log(1 + e^{\gamma_i})\right]$$

RANDOM FOREST

Random Forest Classifier is an algorithm that combines more than one calculations of same or diverse kind for classifying objects. Random forest classifier makes a set of decision trees from randomly chosen subset of training set. It at that point sums the votes from distinct decision trees to choose the final class of the test object. Fundamental parameters to Random Forest Classifier can be add up to number of trees to be created and decision tree related parameters like minimum split, split criteria etc. The leading highlight of random forest classifier is that even if one or few decision trees are prone to a noise, the overall result would tend to be correct.

SUPPORT VECTOR MACHINE

"Support Vector Machine" (SVM) is a supervised machine learning algorithm which can be used for both classification or regression challenges. However, it is mostly used in classification problems. In this algorithm, we plot each data item as a point in n-dimensional space (where n is number of features you have) with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the hyperplane that differentiate the two classes very well.
A SVM classifier is always looking for two things: (i) larger margin, and (ii) lower misclassification rate. But if we increase the margin, we will end up having larger misclassification error. On the other hand, if we decrease the margin, we will end up

having lower misclassification error. A lower error on training dataset does not guarantee a lower error on the test dataset. Therefore, to get better result on the testing dataset, SVM looks for a higher margin. The slack variable parameter C decides the width of the margin: larger value of C implies a smaller margin, and a smaller value of C implies a larger margin.
Gamma is a parameter of the RBF kernel and can be thought of as the 'spread' of the kernel and therefore, the decision region. When gamma is low, the 'curve' of the decision boundary is very low and thus the decision region is very broad. When gamma is high, the 'curve' of the decision boundary is high, which creates islands of decision-boundaries around data points.

K NEAREST NEIGHBOURS

K-nearest neighbors can be used for classification and the result is the class membership. Here the object classification is done based on the majority votes of its neighbors, with the object being classified to the class with majority votes.
The working of the algorithm is as explained: A positive integer k is specified along with the new sample. Then we select k neighbors which are closest to the new sample based on some distance metric like Euclidean distance, Manhattan distance, Minkowski distance, Mahalanobis distance etc. Now we get the most common classification outputs from the nearest neighbors and this is the class we give to the new sample.

NAÏVE BAYES CLASSIFIER

It is a classification technique based on Bayes' Theorem with an assumption of independence among predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature.
Naive Bayes model is easy to build and particularly useful for very large data sets. Along with simplicity, Naive Bayes is known to outperform even highly sophisticated classification methods

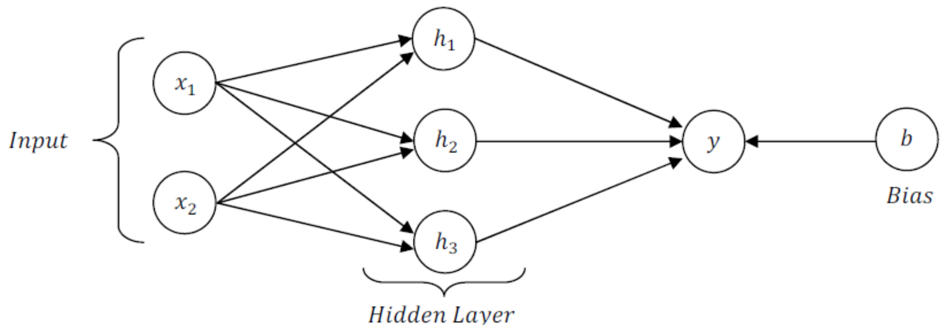
$$\underset{\text{Posterior Probability}}{P(c\mid x)} = \frac{\overset{\text{Likelihood}}{P(x\mid c)}\,\overset{\text{Class Prior Probability}}{P(c)}}{\underset{\text{Predictor Prior Probability}}{P(x)}}$$

$$P(c\mid \mathrm{X}) = P(x_1\mid c)\times P(x_2\mid c)\times\cdots\times P(x_n\mid c)\times P(c)$$

Where,
P(c|x) is the posterior probability of class (c, target) given predictor (x, attributes).
P(c) is the prior probability of class.
P(x|c) is the likelihood which is the probability of predictor given class.
P(x) is the prior probability of predictor.

MULTILAYER PERCEPTRON

A multilayer perceptron (MLP) is a deep, artificial neural network. It is composed of more than one perceptron. They are composed of an input layer to receive the signal, an output layer that makes a decision or prediction about the input, and in between those two, an arbitrary number of hidden layers that are the true computational engine of the MLP. MLPs with one hidden layer are capable of approximating any continuous function. Multilayer perceptrons are often applied to supervised learning problems: they train on a set of input-output pairs and learn to model the correlation (or dependencies) between those inputs and outputs. Training involves adjusting the parameters, or the weights and biases, of the model in order to minimize error. Backpropagation is used to make those weigh and bias adjustments relative to the error, and the error itself can be measured in a variety of ways, including by root mean squared error (RMSE).
In the forward pass, the signal flow moves from the input layer through the hidden layers to the output layer, and the decision of the output layer is measured against the ground truth labels.



In the backward pass, using backpropagation and the chain rule of calculus, partial derivatives of the error function w.r.t. the various weights and biases are back-propagated through the MLP. That act of differentiation gives us a gradient, or a landscape of error, along which the parameters may be adjusted as they move the MLP one step closer to the error minimum. This can be done with any gradient-based optimisation algorithm such as stochastic gradient descent. The network keeps playing that game of tennis until the error can go no lower. This state is known as convergence.

MODEL SELECTION

CROSS VALIDATION

The problem with machine learning models is that you won't get to know how well a model performs until you test its performance on an independent data set (the data set which was not used for training the machine learning model). Cross validation helps us estimate the performance of our model. One type of cross validation is the K-fold cross validation. It helps us understand how machine learning model would generalize to an independent dataset. We in our experiment use K-fold cross validation performing following steps:

1. Divide the original training dataset into K equal parts. Each part is called a fold. Let us call them D_{tr1}, D_{tr2}, D_{tr3} ……… D_{trk}.
2. Now keep the fold $D_{tr(i)}$ as validation set and keep the rest k-1 folds for training. Perform the learning on these k-1 folds and calculate the accuracy on the held-out validation set

i=1	D_{tr1}	D_{tr2}	D_{tr3}	…….	D_{trk}
i=2	D_{tr1}	D_{tr2}	D_{tr3}	…….	D_{trk}
i=3	D_{tr1}	D_{tr2}	D_{tr3}	…….	D_{trk}
i=k	D_{tr1}	D_{tr2}	D_{tr3}	…….	D_{trk}

Figure 7 – K Fold Cross Validation

3. Shift the validation set to new 'i' and repeat it such that every fold is atleast once used in validation.
4. Estimate the accuracy of your model by averaging the accuracies derived in all the k cases of cross validation.

FEATURE SELECTION

F Test is a statistical test used to compare between models and check if the difference is significant between the model. F-Test is useful in feature selection as we get to know the significance of each feature in improving the model.

ONLINE DATASET

The first two attributes containing the URL of the article and time delta to data acquisition are not useful for predictions. Of the rest, many are binary attributes and by visual inspection some seem to be correlated.

For example, the "is_weekday_{Saturday,Sunday}, Saturday and Sunday information is aggregated into the attribute is_weekend so, is_weekday_{Saturday,Sunday} can be safely removed.

Similarly, aggregating information from attributes is_weekday_<Monday- Friday> into a new attribute "is_weekday" will improve quality of the explanatory attribute. This we believe will create complex attributes from simple granular ones and also the construction of excessive number of dummy attributes will be reduced for regression like approaches.

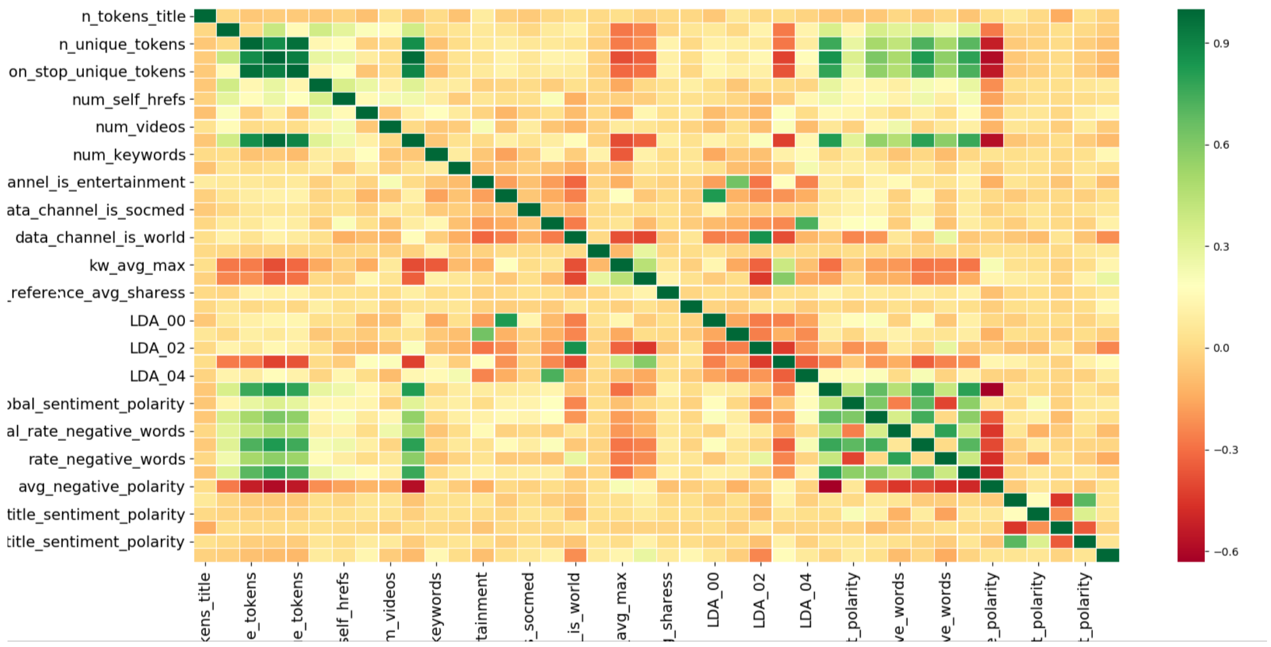


Figure 8 – Correlation Matrix

From figure 8, we can observe correlation between features. Green indicated very high correlation. We are dropping features having high correlation as it does not help in classification as both the features has almost same information. Attributes 19-27 and 50-55, in the above table, which have information about the number of keywords and positive words. Here, the attributes are average, minimum and maximum. The average encompasses information from the minimum and maximum so, there is some collinearity between these attributes. Thus, to reduce effects of singularity we removed the attributes with suffix "_min_" and "_max_" while keeping "_avg_". Thus, reducing attributes and avoiding possible ill-effects after visually inspection.

<u>BANK DATASET</u>

Scikit learn provides the Selecting K best features using F-Test. We used f_classif to detect importance of features and I have used **top 50 percentile** features for classification for feature selection and parameter tuning methods.

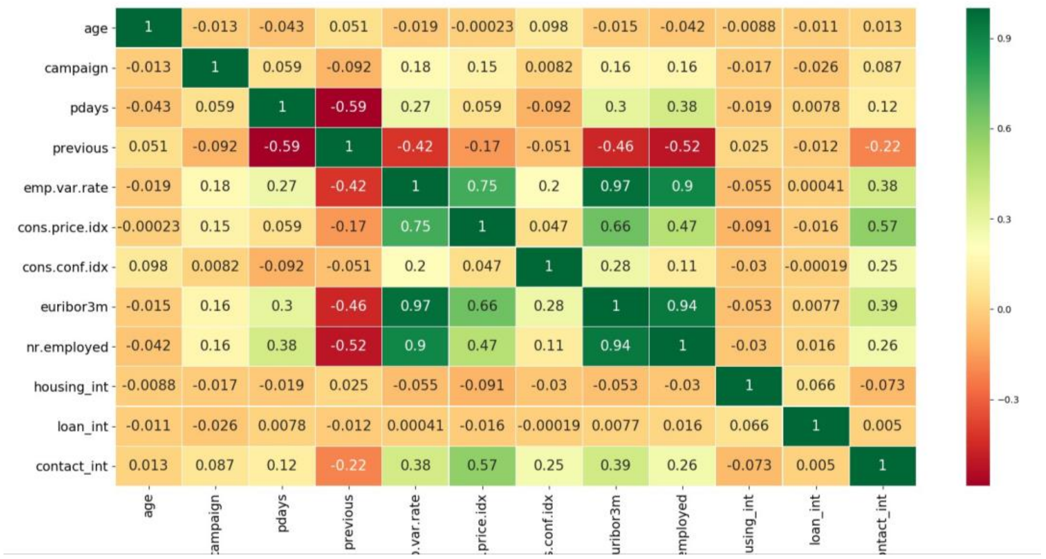


Figure 8 – Correlation Matrix

We can observe very high positive as well as negative correlation between some features from figure 8.

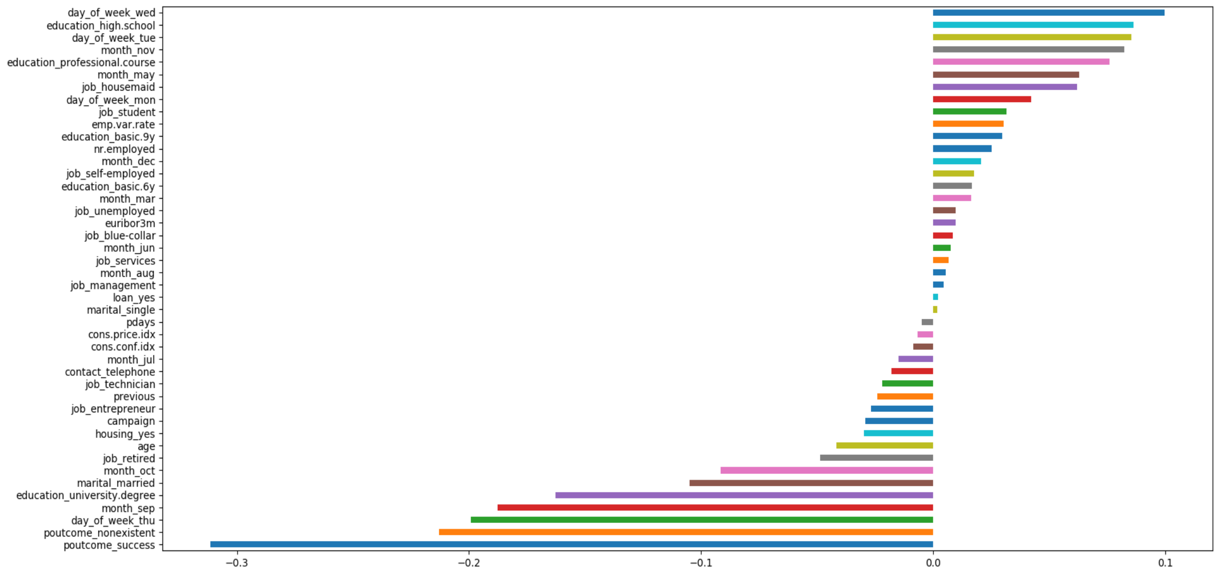


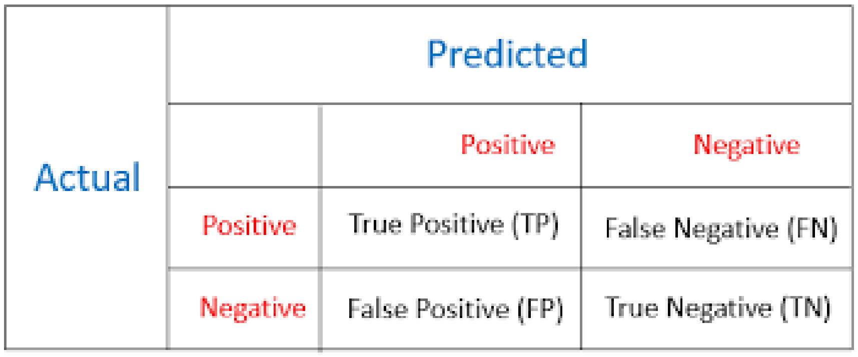
Figure 9 – Feature Importance

DIMENSIONALITY REDUCTION

Principal component analysis is a very popular dimensionality reduction technique. Here, the features are transformed into a new set of features, which are linear combination of original values. These new set of variables are known as principle components. They are obtained in such a way that first principle component accounts for most of the possible variation of original data after which each succeeding component has the highest possible variance. The principal components are sensitive to the scale of measurement, now to fix this issue we should always standardize variables before applying PCA.

F-TEST SCORE

An F statistic is a value you get when you run an ANOVA test or a regression analysis to find out if the means between two populations are significantly different. It's similar to a T statistic from a T-Test; A-T test will tell you if a single variable is statistically significant and an F test will tell you if a group of variables are jointly significant.
You can use the F statistic when deciding to support or reject the null hypothesis. In your F test results, you'll have both an F value and an F critical value.
The F critical value is also called the F statistic.
The value you calculate from your data is called the F value (without the "critical" part).



		Predicted	
		Positive	Negative
Actual	Positive	True Positive (TP)	False Negative (FN)
	Negative	False Positive (FP)	True Negative (TN)

Figure 10– F Test representation

Depending upon all the values, we can calculate F measure.

$$F_1 = \frac{2}{\frac{1}{\mathrm{recall}} + \frac{1}{\mathrm{precision}}} = 2 \cdot \frac{\mathrm{precision} \cdot \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}}.$$

ROC-AUC SCORE

In statistics, a receiver operating characteristic curve, i.e. ROC curve, is a graphical plot that illustrates the diagnostic ability of a binary classifier system as its discrimination threshold is varied. The Total Operating Characteristic (TOC) expands on the idea of ROC by showing the total information in the two-by-two contingency table for each threshold. ROC gives only two bits of relative information for each threshold, thus the TOC gives strictly more information than the ROC.

EXPERIMENTAL RESULTS

BANK MARKETING DATASET

Experiment 1:
Imputing Method='Random Forest"
Data Preprocessing = Standard Scalar

CLASSIFIER	F1 MEASURE	ROC SCORE / AUC
Logistic Regression CV with Standard Scalar	0.874	0.7742
Logistic Regression with Feature Selection	0.883	0.7695
Multi-Layer Perceptron	0.878	0.7778
K- Nearest Neighbor	0.878	0.7449
Logistic Regression with Lasso	0.880	0.7707
Logistic Regression Parameter Tuning	0.872	0.7674
Decision Trees	0.879	0.7271
Random Forest 1	0.878	0.7853
Random Forest 2	0.881	0.7783
Random Forest Final	0.888	0.7705
Under sampling with Logistic Regression	0.815	0.7697
Over sampling with Logistic Regression	0.833	0.7741
Undersampling with Random Forest	0.815	0.7634
Oversampling with Random Forest	0.833	0.7494
Support Vector Machine	0.876	0.7592

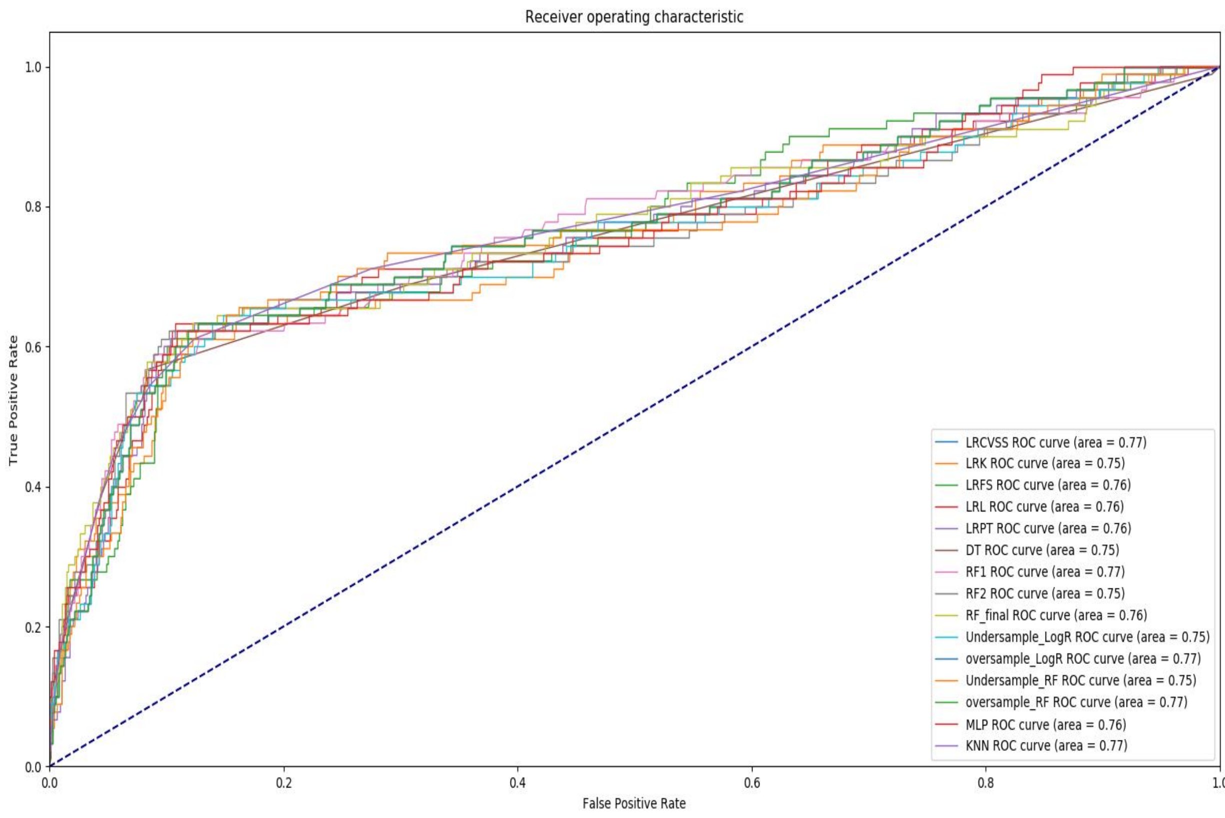
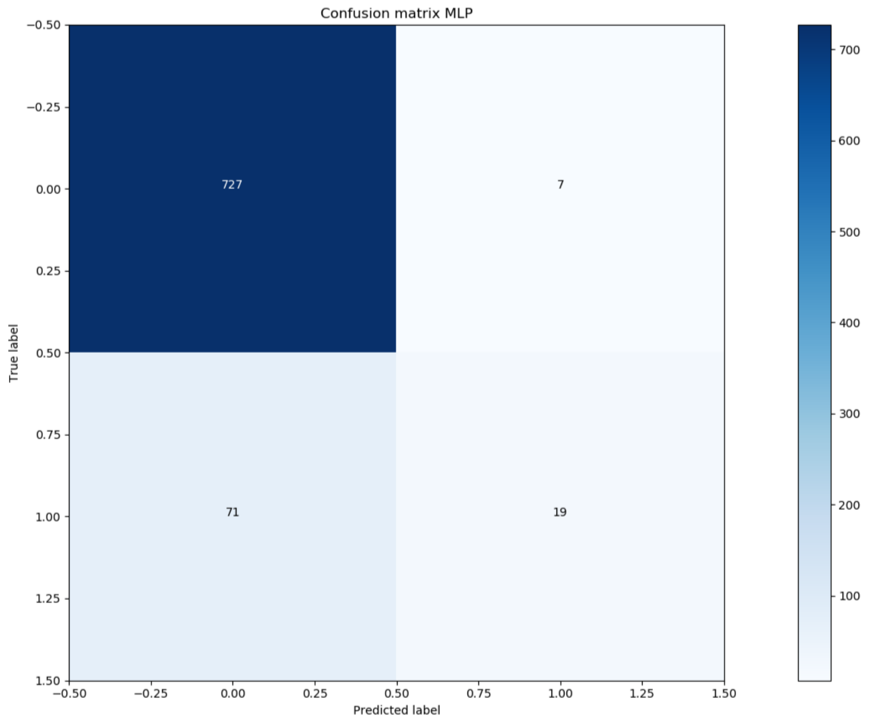
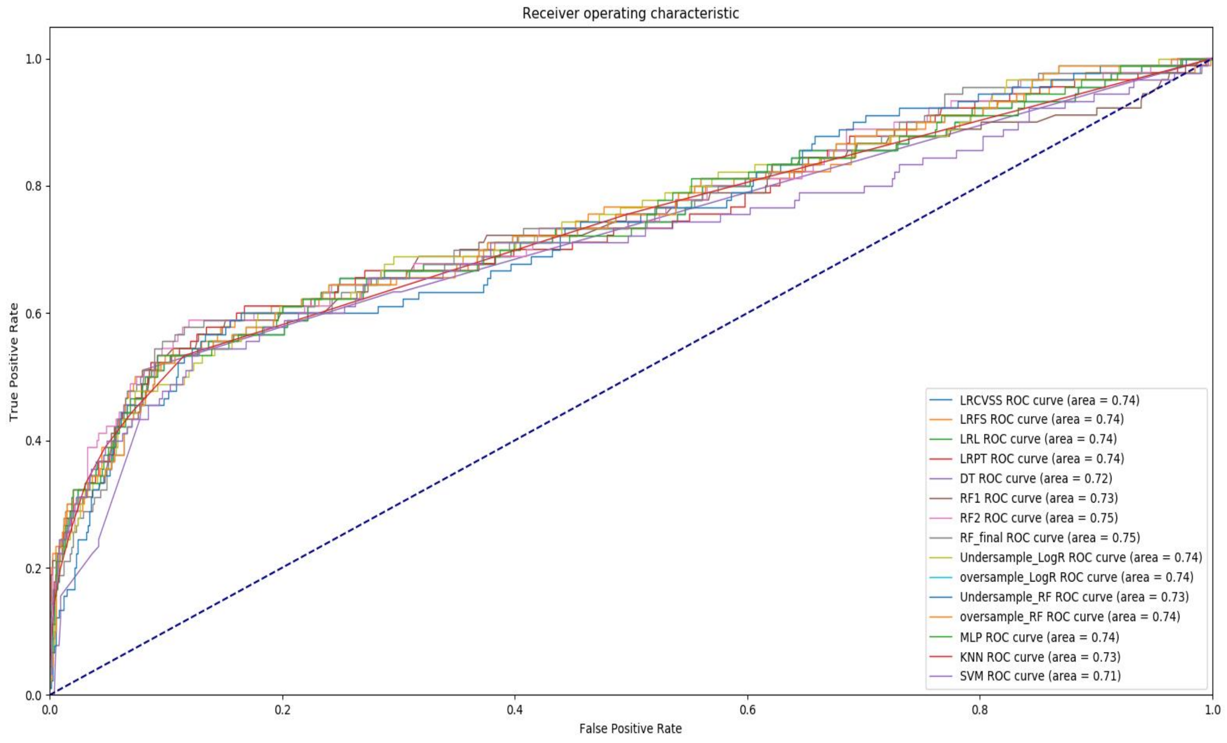


Figure 11– ROC Curve and Confusion Matrix

Experiment 2:
Imputing Method=’Random Forest”
Data Preprocessing = Min Max Scalar

CLASSIFIER	F1 MEASURE	ROC SCORE / AUC
Logistic Regression CV with Standard Scalar	0.881	0.7629
Logistic Regression with Feature Selection	0.888	0.7698
Multi-Layer Perceptron	0.881	0.8023
K- Nearest Neighbor	0.8784	0.7653
Logistic Regression with Lasso	0.881	0.7638
Logistic Regression Parameter Tuning	0.884	0.7735
Decision Trees	0.865	0.7346
Random Forest 1	0.873	0.7883
Random Forest 2	0.879	0.7773
Random Forest Final	0.877	0.7605
Under sampling with Logistic Regression	0.765	0.7803
Over sampling with Logistic Regression	0.834	0.7768
Under sampling with Random Forest	0.765	0.7621
Oversampling with Random Forest	0.834	0.7597
Support Vector Machine	0.887	0.7208

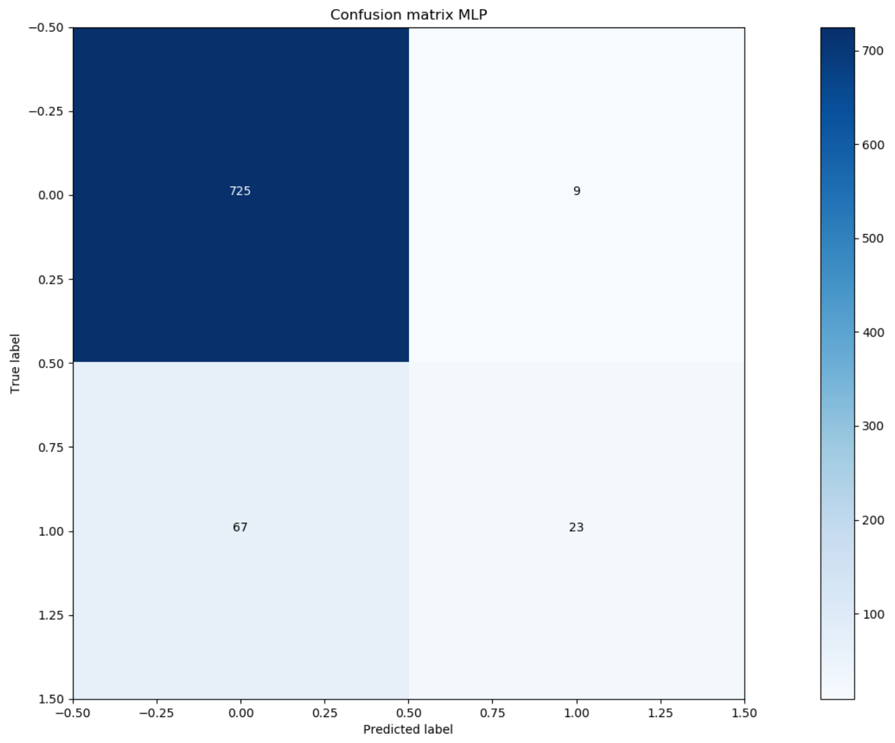




Experiment 3:

Imputing Method=”Mode”
Data Preprocessing = Min Max Scalar

CLASSIFIER	F1 MEASURE	ROC SCORE / AUC
Logistic Regression CV with Standard Scalar	0.886	0.7678
Logistic Regression with Feature Selection	0.891	0.7798
Multi-Layer Perceptron	0.890	0.8358
K- Nearest Neighbor	0.889	0.7479
Logistic Regression with Lasso	0.888	0.7698
Logistic Regression Parameter Tuning	0.891	0.7728
Decision Trees	0.895	0.7243
Random Forest 1	0.892	0.7828
Random Forest 2	0.892	0.7732
Random Forest Final	0.892	0.7635
Under sampling with Logistic Regression	0.859	0.7627
Over sampling with Logistic Regression	0.845	0.7709
Under sampling with Random Forest	0.859	0.7582
Oversampling with Random Forest	0.845	0.7410
Support Vector Machine	0.873	0.7021

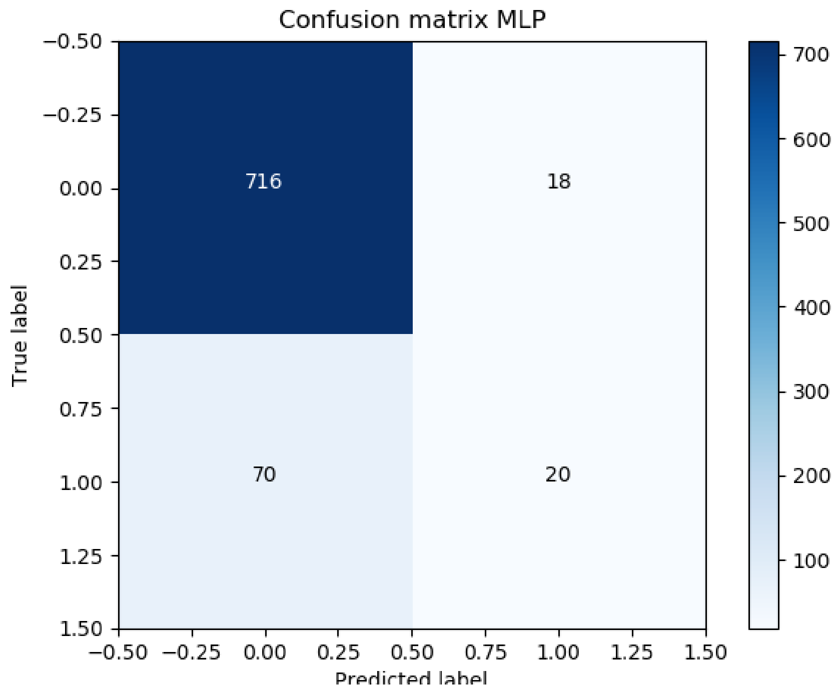


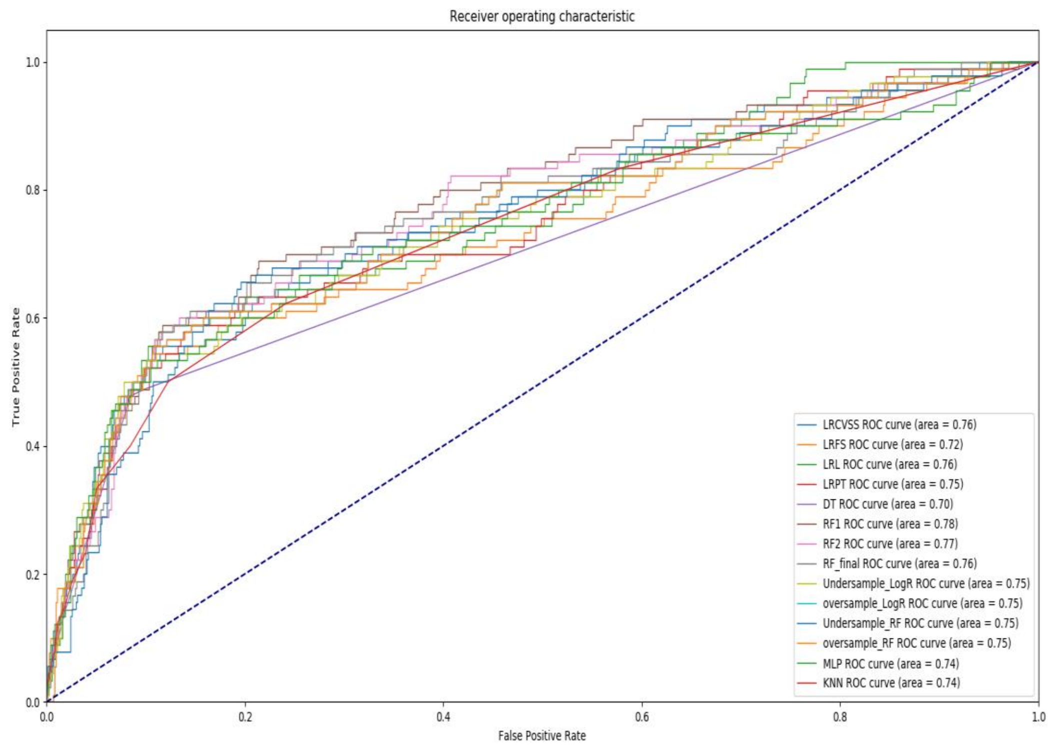
Experiment 4:

Imputing Method = ”Mode”

Data Preprocessing = Standard Scalar

CLASSIFIER	F1 MEASURE	ROC SCORE / AUC
Logistic Regression CV with Standard Scalar	0.870	0.7748
Logistic Regression with Feature Selection	0.876	0.7764
Multi-Layer Perceptron	0.877	0.7986
K- Nearest Neighbor	0.871	0.7583
Logistic Regression with Lasso	0.874	0.7769
Logistic Regression Parameter Tuning	0.875	0.7798
Decision Trees	0.874	0.7353
Random Forest 1	0.865	0.7836
Random Forest 2	0.868	0.7737
Random Forest Final	0.860	0.7613
Under sampling with Logistic Regression	0.823	0.7775
Over sampling with Logistic Regression	0.825	0.7823
Under sampling with Random Forest	0.823	0.7607
Oversampling with Random Forest	0.825	0.7321
Support Vector Machine	0.867	0.7211

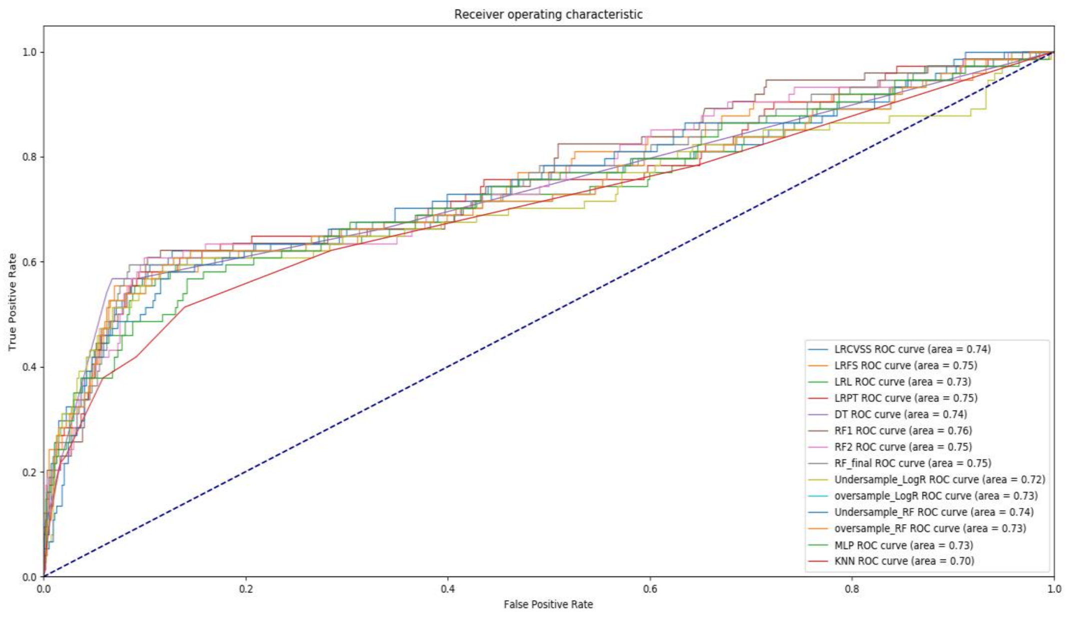


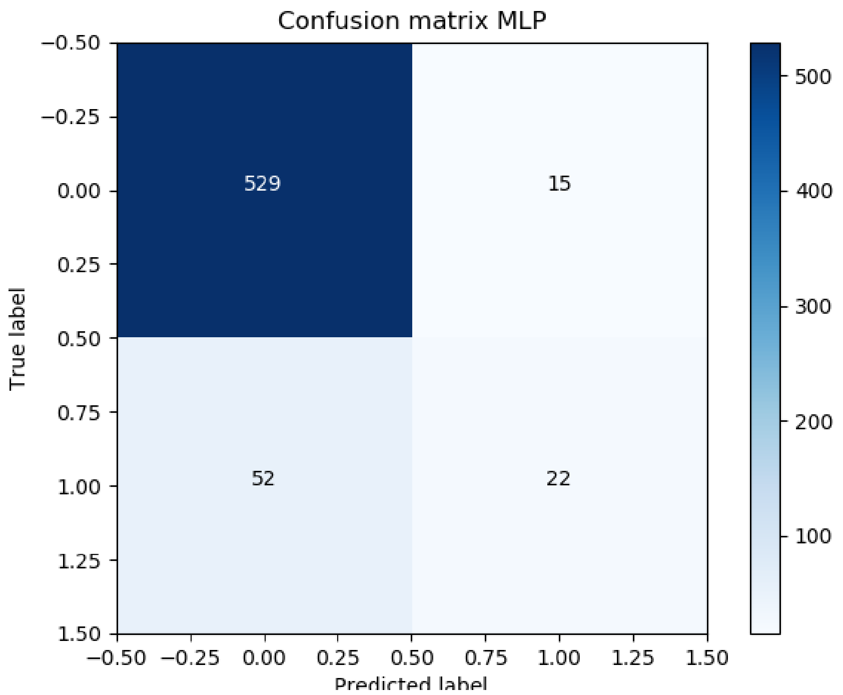


Experiment 5:
Imputing Method =" Delete"
Data Preprocessing = Standard Scalar

CLASSIFIER	F1 MEASURE	ROC SCORE / AUC
Logistic Regression CV with Standard Scalar	0.879	0.7917
Logistic Regression with Feature Selection	0.874	0.7844
Multi-Layer Perceptron	0.869	0.8129
K- Nearest Neighbor	0.867	0.7350
Logistic Regression with Lasso	0.880	0.7892
Logistic Regression Parameter Tuning	0.879	0.7896
Decision Trees	0.866	0.7593
Random Forest 1	0.870	0.8023
Random Forest 2	0.868	0.7953
Random Forest Final	0.874	0.7887

Under sampling with Logistic Regression	0.829	0.7853
Over sampling with Logistic Regression	0.800	0.7947
Under sampling with Random Forest	0.823	0.7855
Oversampling with Random Forest	0.800	0.7947
Support Vector Machine	0.861	0.7711



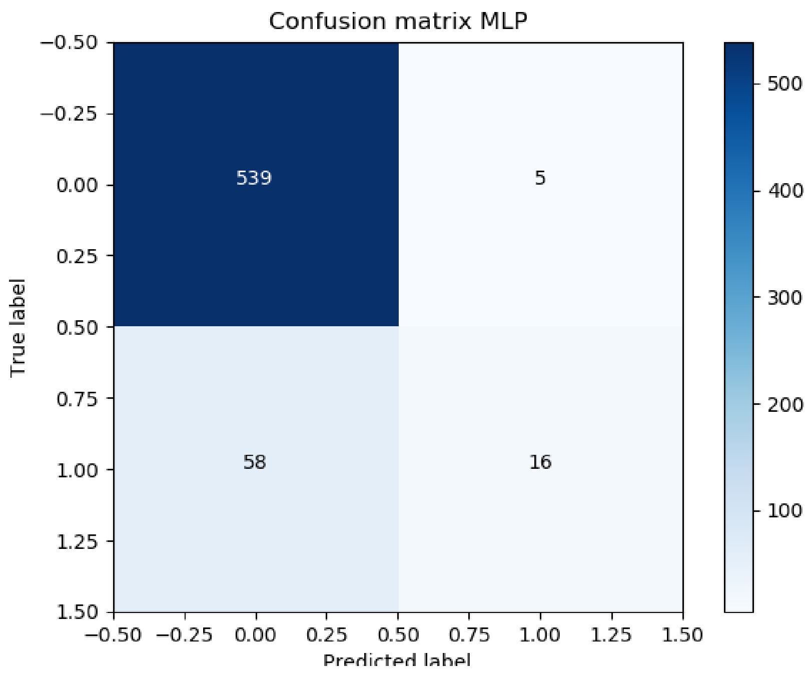


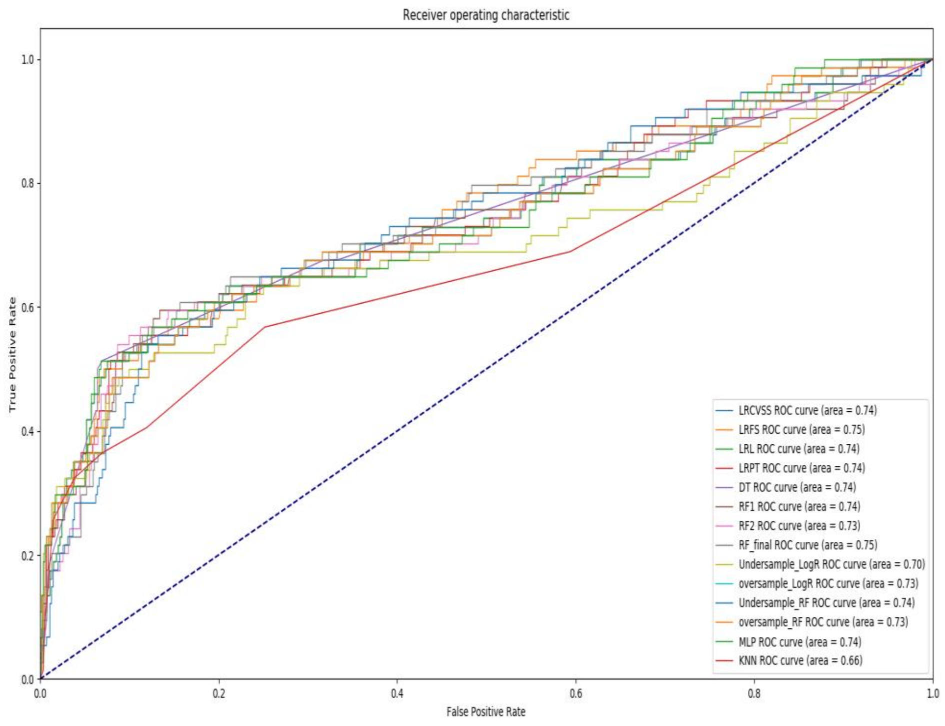
Experiment 6:

Imputing Method =” Delete”

Data Preprocessing = Min Max Scalar

CLASSIFIER	F1 MEASURE	ROC SCORE / AUC
Logistic Regression CV with Standard Scalar	0.863	0.7783
Logistic Regression with Feature Selection	0.864	0.7921
Multi-Layer Perceptron	0.871	0.8340
K- Nearest Neighbor	0.865	0.7211
Logistic Regression with Lasso	0.863	0.7770
Logistic Regression Parameter Tuning	0.855	0.7808
Decision Trees	0.863	0.7575
Random Forest 1	0.858	0.7985
Random Forest 2	0.861	0.7854
Random Forest Final	0.861	0.7721
Under sampling with Logistic Regression	0.777	0.7749
Over sampling with Logistic Regression	0.817	0.7916
Under sampling with Random Forest	0.777	0.7773
Oversampling with Random Forest	0.817	0.7647
Support Vector Machine	0.823	0.7853





ONLINE NEWS DATASET

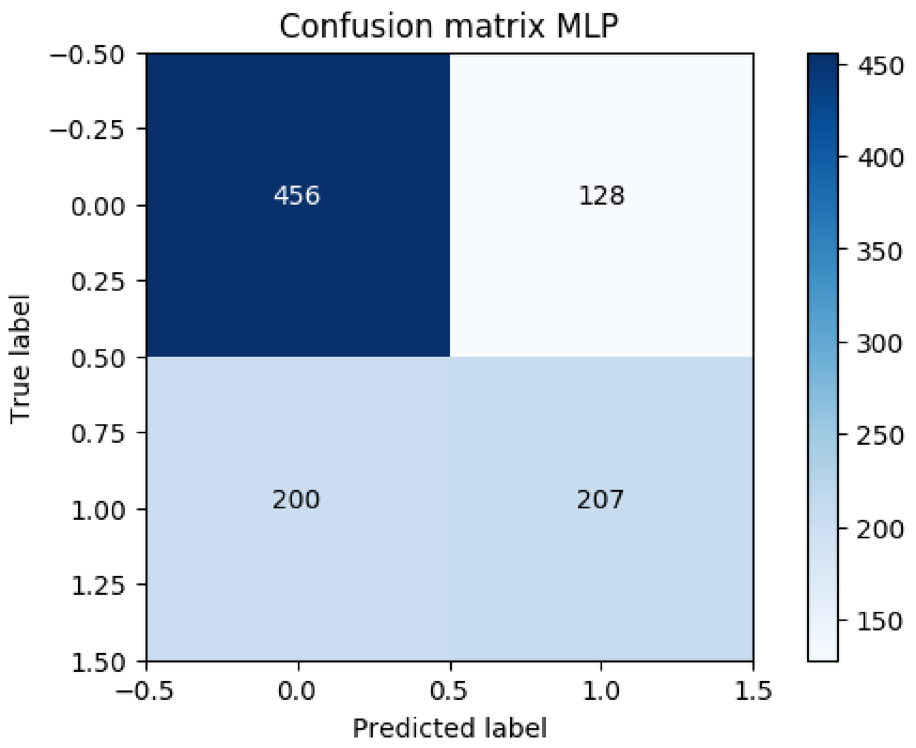
Experiment 1:

Data Preprocessing: Standard Scaler

Class: Two class

Classifier	F1 measure	Test Accuracy
Logistic Regression CV with Standard Scalar	0.644	0.6559
Logistic Regression with Feature Selection	0.653	0.6619
Multi-Layer Perceptron	0.668	0.6781
K- Nearest Neighbor	0.585	0.6014
Logistic Regression with Lasso	0.645	0.6589
Logistic Regression Parameter Tuning	0.605	0.6347
Decision Trees	0.624	0.6236
Random Forest 1	0.657	0.6680

Random Forest 2	0.654	0.6639
Random Forest Final	0.651	0.6599
Under sampling with Logistic Regression	0.564	0.5620
Over sampling with Logistic Regression	0.551	0.5702
Under sampling with Random Forest	0.596	0.5923
Over sampling with Random Forest	0.564	0.5620



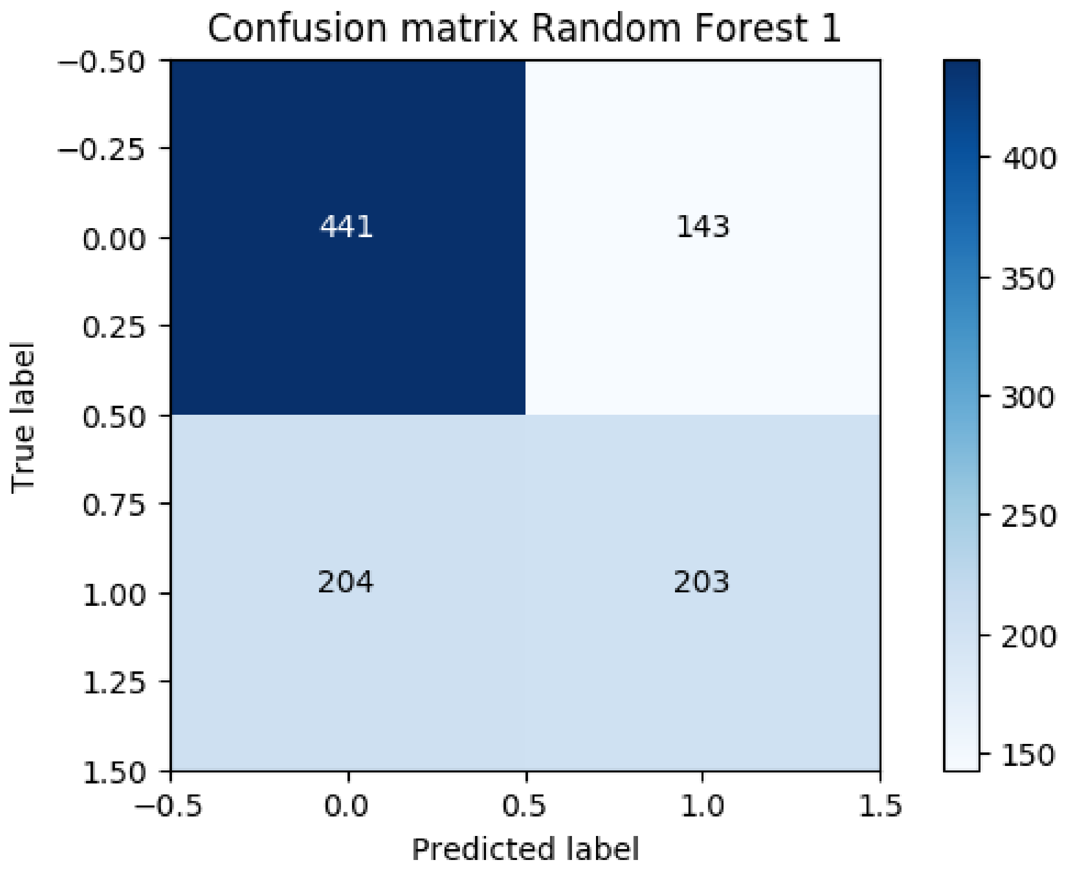
Experiment 2:

Data Preprocessing: MinMax Scaler

Class: Two class

Classifier	F1 measure	Test Accuracy
Logistic Regression CV with Standard Scalar	0.636	0.6437
Logistic Regression with Feature Selection	0.638	0.6468
Multi-Layer Perceptron	0.633	0.6326
K- Nearest Neighbor	0.558	0.5993
Logistic Regression with Lasso	0.636	0.6427
Logistic Regression Parameter Tuning	0.635	0.6458

Decision Trees	0.627	0.6316
Random Forest 1	0.651	0.6579
Random Forest 2	0.641	0.6498
Random Forest Final	0.638	0.6448
Under sampling with Logistic Regression	0.570	0.5681
Over sampling with Logistic Regression	0.584	0.5802
Under sampling with Random Forest	0.570	0.5681
Over sampling with Random Forest	0.584	0.5802



Experiment 3:

Data Preprocessing: Standard Scaler

Class: Multi Class

Classifier	F1 measure	Test Accuracy
Decision Trees	0.2933	0.3260
K- Nearest Neighbor	0.2662	0.2994
Non-Linear SVM	0.2837	0.3325
Gaussian Naïve Bayes	0.2782	0.3002

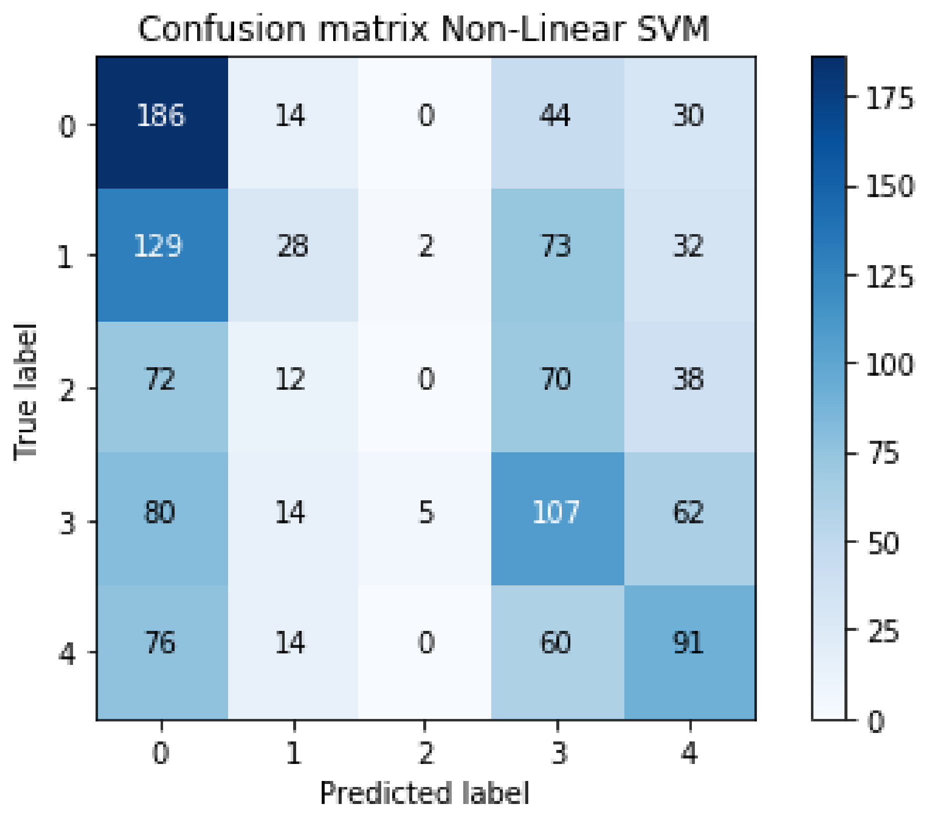


Figure 11–Confusion Matrix for best accuracy

Experiment 4:

Data Preprocessing: Min Max Scaler

Class: Multi Class

Classifier	F1 measure	Test Accuracy
Decision Trees	0.2933	0.3260
K- Nearest Neighbor	0.2830	0.2994
Linear SVM	0.2815	0.3333
Gaussian Naïve Bayes	0.2782	0.3002

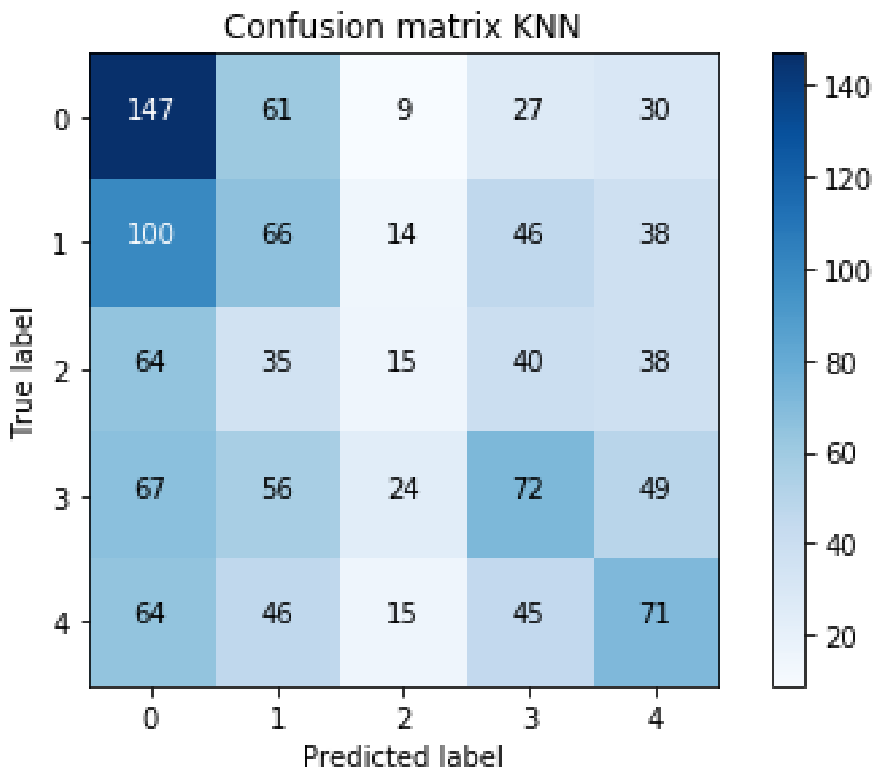


Figure 18–Confusion Matrix (highlighted model)

DISCUSSION

I have simulated classification of data using variety of approaches in pre-processing followed by various famous machine learning algorithms.

I have stated 6 cases for Bank dataset along with 4 cases for Online News dataset. I have marked best setting giving me best accuracy.

We tried many approaches while classifying data from various sampling techniques, imputed the data using 3-4 methods as well as used feature selection and feature importance to get optimized output. We even tried to reduce dimension by implementing PCA. In the end, I have stated the best parameters as well as method.

From confusion matrix, we can understand that some labels are getting confused with other classes more often

Bank data set contains combination of continuous and categorical features hence multi-layered algorithms like Multi layered perceptron, Random Forest etc. Hence, I can observe best outputs for MLP followed by random forest.

After preprocessing, I used stratifiedshufflesplit to make sure I have equal number of data for testing.

For Bank dataset,

Classifier: Multi-Layer Perceptron

Best settings: Data Imputing: Mode, Data Preprocessing: Min Max Scaler

AUC Score:0.8353, F1 Score:0.890

For Online News dataset,

Classifier: Multi-Layer Perceptron

Best settings: Data Preprocessing: Standard Scaler

Test Accuracy: 0.6781, F1 Score:0.66

WORK DISTRIBUTION

For this final project, we both have worked very hard. We got to know very new things and were successfully able to implement the machine learning algorithms and actually visualize the result.

We learned a lot about data visualization and used variety of libraries for plots. We created pipeline and grid search for all the classifiers to cross validate and get optimum output. We used various data pre-processing techniques to handle the data and I must say that was the best part about this project.

Work	Online News Popularity	Bank Additional Dataset
Data Pre-Processing	Pranav, Aditya	Pranav, Aditya
Model Selection	Pranav, Aditya	Pranav, Aditya
Classifier	Pranav, Aditya	Pranav, Aditya
Result Analysis and Conclusion	Pranav, Aditya	Pranav, Aditya

We both have worked very hard for this project and expect the good response from your side.

Thank you.