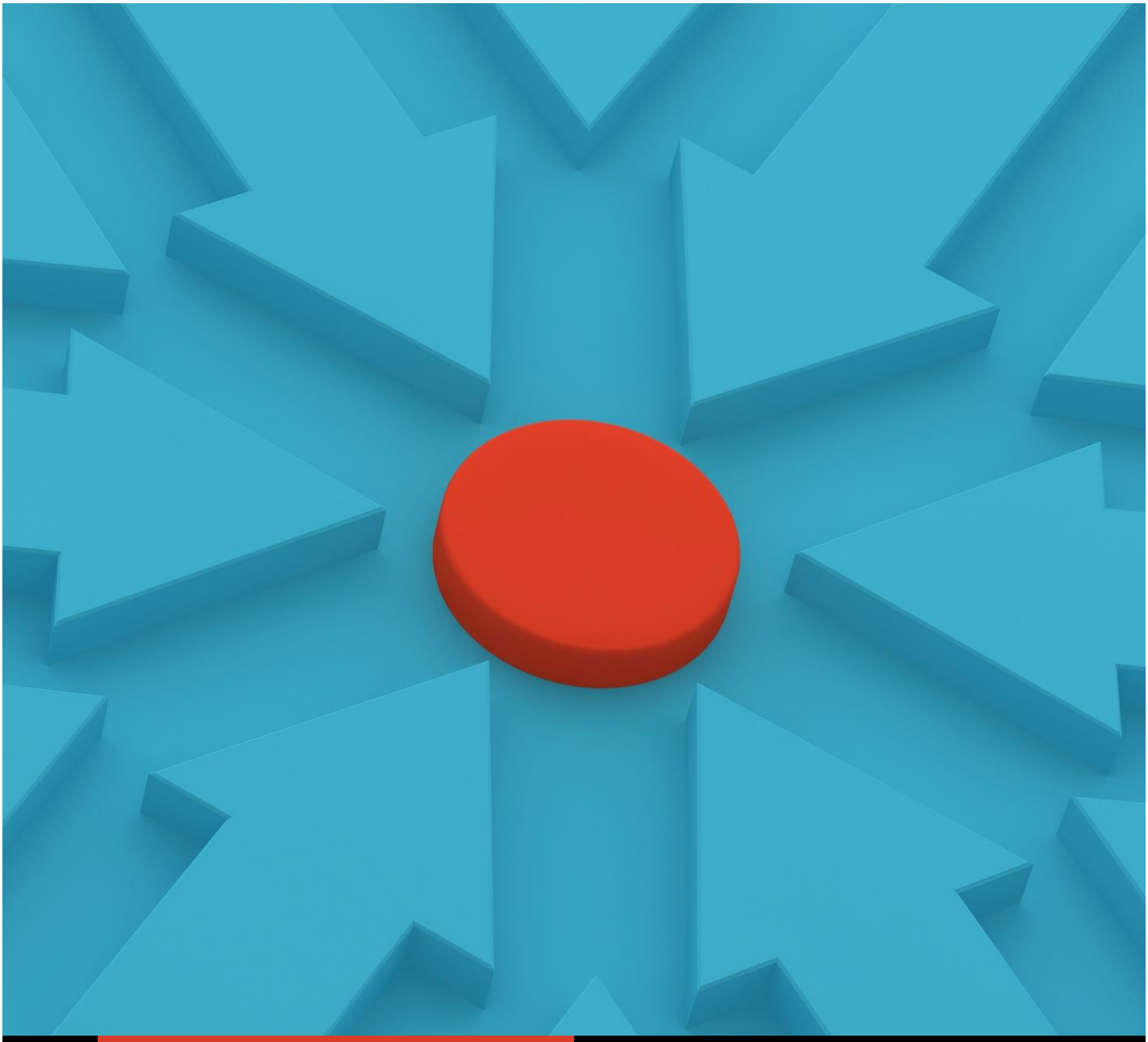

MACHINE LEARNING

PROJECT REPORT

SUNDAR RAM S
PGPDSBA
ONLINE DEC_C 2021
03-JUL-2022

Contents

TABLE OF FIGURES

TABLE OF TABLES

TABLE OF EQUATIONS

PROBLEM 1 – MODELING

Problem Statement

You are hired by one of the leading news channels CNBE who wants to analyze recent elections. This survey was conducted on 1525 voters with 9 variables. You have to build a model, to predict which party a voter will vote for on the basis of the given information, to create an exit poll that will help in predicting overall win and seats covered by a particular party.

1.1 Read the dataset. Do the descriptive statistics and do the null value condition check. Write an inference on it.

Sample of the Dataset

	vote	age	economic.cond.national	economic.cond.household	Blair	Hague	Europe	political.knowledge	gender
0	Labour	43	3	3	4	1	2	2	female
1	Labour	36	4	4	4	4	5	2	male
2	Labour	35	4	4	5	2	3	2	male
3	Labour	24	4	2	2	1	4	0	female
4	Labour	41	2	2	1	1	6	2	male

Table 1 Sample of the Dataset

The data consists of 9 variables of survey data of recent elections for 1525 records.

Dataset Description

Vote	:	Party choice - Conservative or Labour
Age	:	in years
economic.cond.national	:	Assessment of current national economic conditions, 1 to 5.
economic.cond.household	:	Assessment of current household economic conditions, 1 to 5.
Blair	:	Assessment of the Labour leader, 1 to 5.
Hague	:	Assessment of the Conservative leader, 1 to 5.
Europe	:	an 11-point scale that measures respondents' attitudes toward European integration. High scores represent 'Eurosceptic' sentiment.
political.knowledge	:	Knowledge of parties' positions on European integration, 0 to 3.
gender	:	female or male.

Variable types

The table below shows the data types of the variables.

Variable	Data Type
vote	object
age	int64
economic.cond.national	int64
economic.cond.household	int64
Blair	int64
Hague	int64
Europe	int64
political.knowledge	int64
gender	object

Table 2 Data Types of the Variables

It can be seen that there are 2 object type variables and 7 integer data types. However, from the data dictionary, we are able to understand that except age, all the other integer variables are of ordinal data type that can be converted to categorical.

Check for null values

The table below shows the count of null values in the data set. It can be seen that there are no null values in the data set.

Variable	Null value count
vote	0
age	0
economic.cond.national	0
economic.cond.household	0
Blair	0
Hague	0
Europe	0
political.knowledge	0
gender	0

Table 3 Null value count

Check for duplicated records

It can be seen that there are 8 duplicated records. However, we would not be able to verify if they are really duplicated as the data pertains to election survey of people, who might by chance can be of similar attributes. However, we would be removing these duplicated records so that the model doesn't get biased. There are 1517 records post removal of duplicated records.

	vote	age	economic.cond.national	economic.cond.household	Blair	Hague	Europe	political.knowledge	gender
67	Labour	35	4	4	5	2	3	2	male
626	Labour	39	3	4	4	2	5	2	male
870	Labour	38	2	4	2	2	4	3	male
983	Conservative	74	4	3	2	4	8	2	female
1154	Conservative	53	3	4	2	2	6	0	female
1236	Labour	36	3	3	2	2	6	2	female
1244	Labour	29	4	4	4	2	2	2	female
1438	Labour	40	4	3	4	2	2	2	male

Table 4 Duplicate Records in the dataset

Summary of the dataset

Summarizing briefly, there are 1517 records in the dataset post removal of duplicated ones. The mean age of the survey is 54.24, mean rating of assessment on national economic condition is 3.25, mean rating of assessment on household economic condition is 3.14, mean rating on assessment of Blair is 3.34, mean rating of assessment of Hague is 2.75, mean rating on attitude towards European integration is 6.74 & Mean knowledge on party's position on European integration is 1.54. On seeing the mean & median data, they are almost close to each other and hence, the data set can be considered to be normally distributed. It can also be seen that there are records with 0 knowledge on parties' position on European integration.

	age	economic.cond.national	economic.cond.household	Blair	Hague	Europe	political.knowledge
count	1517.00	1517.00	1517.00	1517.00	1517.00	1517.00	1517.00
mean	54.24	3.25	3.14	3.34	2.75	6.74	1.54
std	15.70	0.88	0.93	1.17	1.23	3.30	1.08
min	24.00	1.00	1.00	1.00	1.00	1.00	0.00
25%	41.00	3.00	3.00	2.00	2.00	4.00	0.00
50%	53.00	3.00	3.00	4.00	2.00	6.00	2.00
75%	67.00	4.00	4.00	4.00	4.00	10.00	2.00
max	93.00	5.00	5.00	5.00	5.00	11.00	3.00

Table 5 Summary of the dataset

1.2 Perform Univariate and Bivariate Analysis. Do exploratory data analysis. Check for Outliers.

Univariate Analysis – Histograms, Box Plots & Count Plots

From the below histogram having the Kernel Density function, we are able to understand the distribution of data in the age variable that is continuous in nature. There are three nodes and hence multimodal distribution This indicates presence of certain patterns in the age. However, we will not be exploring the patterns. The skewness of the distribution is 0.14 & kurtosis value is -0.94. They are more or less, closer to 0, which indicates that the data is more or less symmetrical in nature & normally distributed.

From the box plot, it evident that there are no outliers in age.

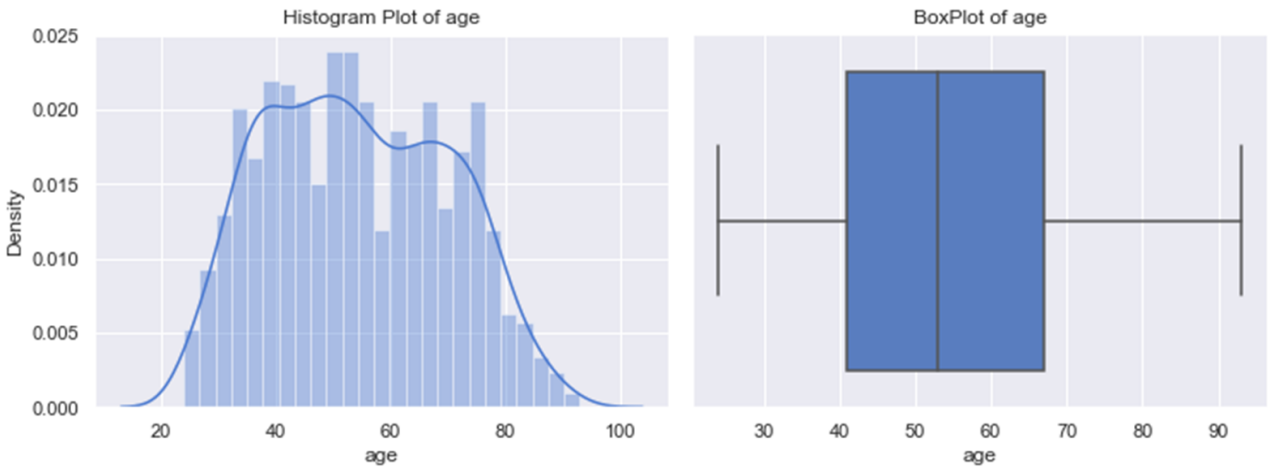

Figure 1 Histogram & Box Plots of continuous variables

Univariate visualization of Categorical variables is carried out using the count plots.

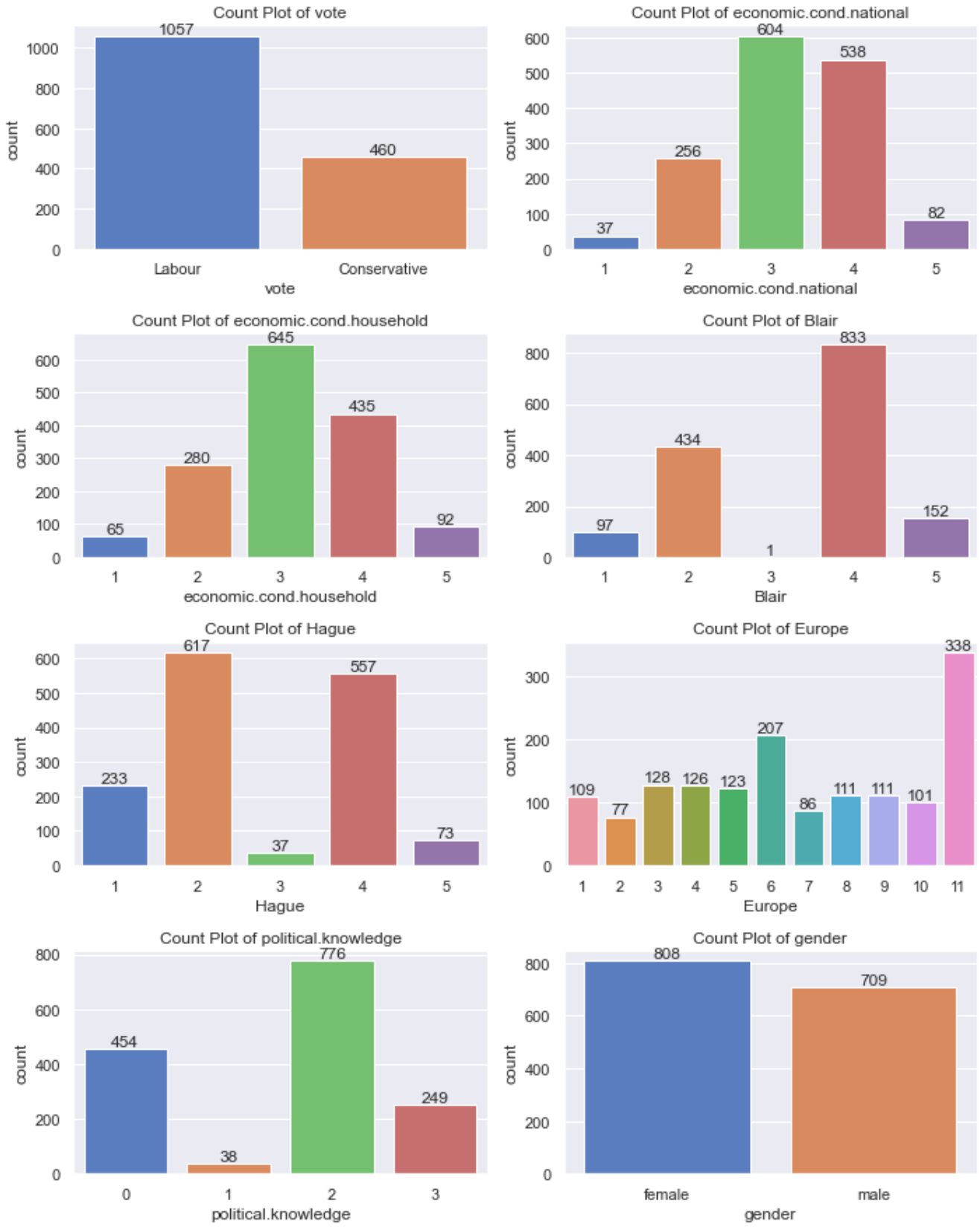

Figure 2 Count plot of ordinal/ categorical variables

From the above count plots, it is evident that there are 1057 records that have voted for Labour party and 460 records that have voted for conservative. This indicated 70 % of the records are towards the Labours and only 30 % are inclined towards the Conservatives. Assessment rating on national & household economic condition has the rating 3 as the highest mode. Most of the assessors have rated 4 for Blair while they have rated 2 for Hague. Most of the people are Eurosceptic, that can be seen from the assessment rating of European Integration, which has the mode as 11. It can be seen that the survey consists of 53% records of Female and 47 % records of Male. The mode for assessment rating of Political knowledge is 2.

From the below box plot, it can be seen that there are only 2 outliers present in the assessment of national and household economic condition. However, these need not be treated as outlier treatment remains meaningful only for continuous type variable.

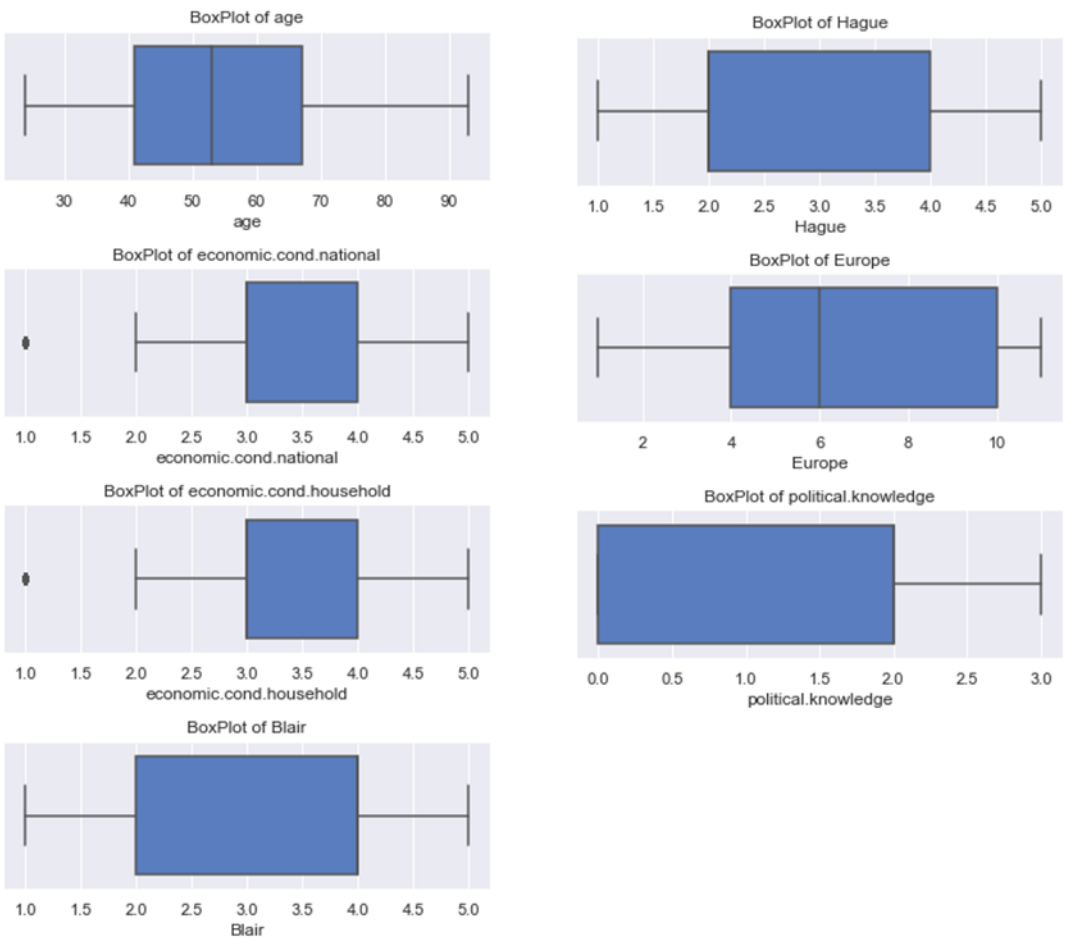

Figure 3 Box Plot - Categorical/ Ordinal variables

Bivariate Analysis – Strip plot, Count plot, Correlation Plot & Pair Plot

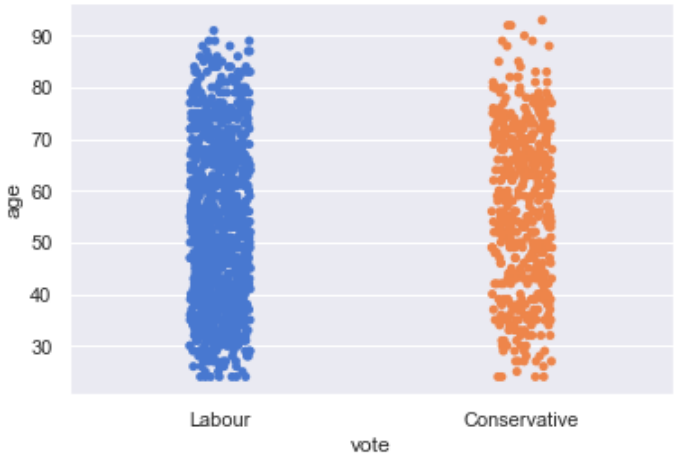

Figure 4 vote vs age – Strip plot

From the above strip plot, it can be seen that the density of records across age is more in the Labour side vote. However, this can be attributed to the 70:30 records in the labour and conservative records.

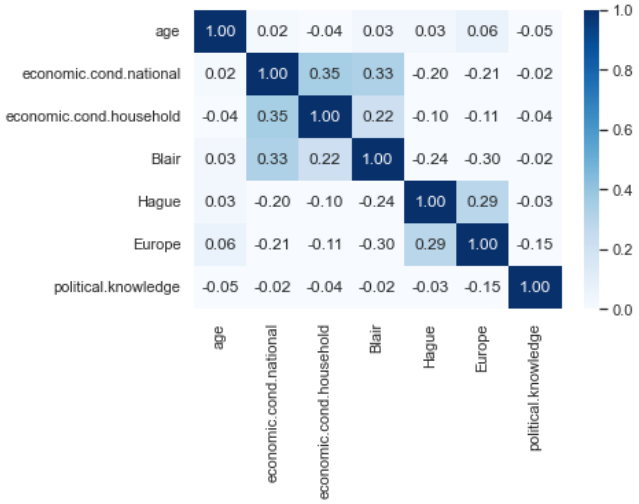

Figure 5 Correlation plot of continuous variables

The figure 5 shows the correlation plot (heat map) of all the integer variables in the dataset. The figure indicates there is very minimum or no correlation amongst the independent variables.

Bivariate analysis is done using count plots.

- It can be seen that most of the records in the labour vote has assessed 4 for national economic conditions while most of the conservative vote records have assessed 3.
- It can be seen that across labour and conservative vote records most of them have assessed 3 for household economic conditions.

- Most of the labour voters have assessed 4 for Blair, while most of the conservative voters have assessed 2 for Blair
- Most of the labour voters have assessed 2 for Hague while most of the conservative voters have assessed 4 for Hague
- It can be seen that both labour and conservative voters are Eurosceptic.
- It can also be seen that most of the voters of both labour and conservative have assessed a 2 rating on political knowledge
- Similarly, both male and female population are inclined towards labour vote.

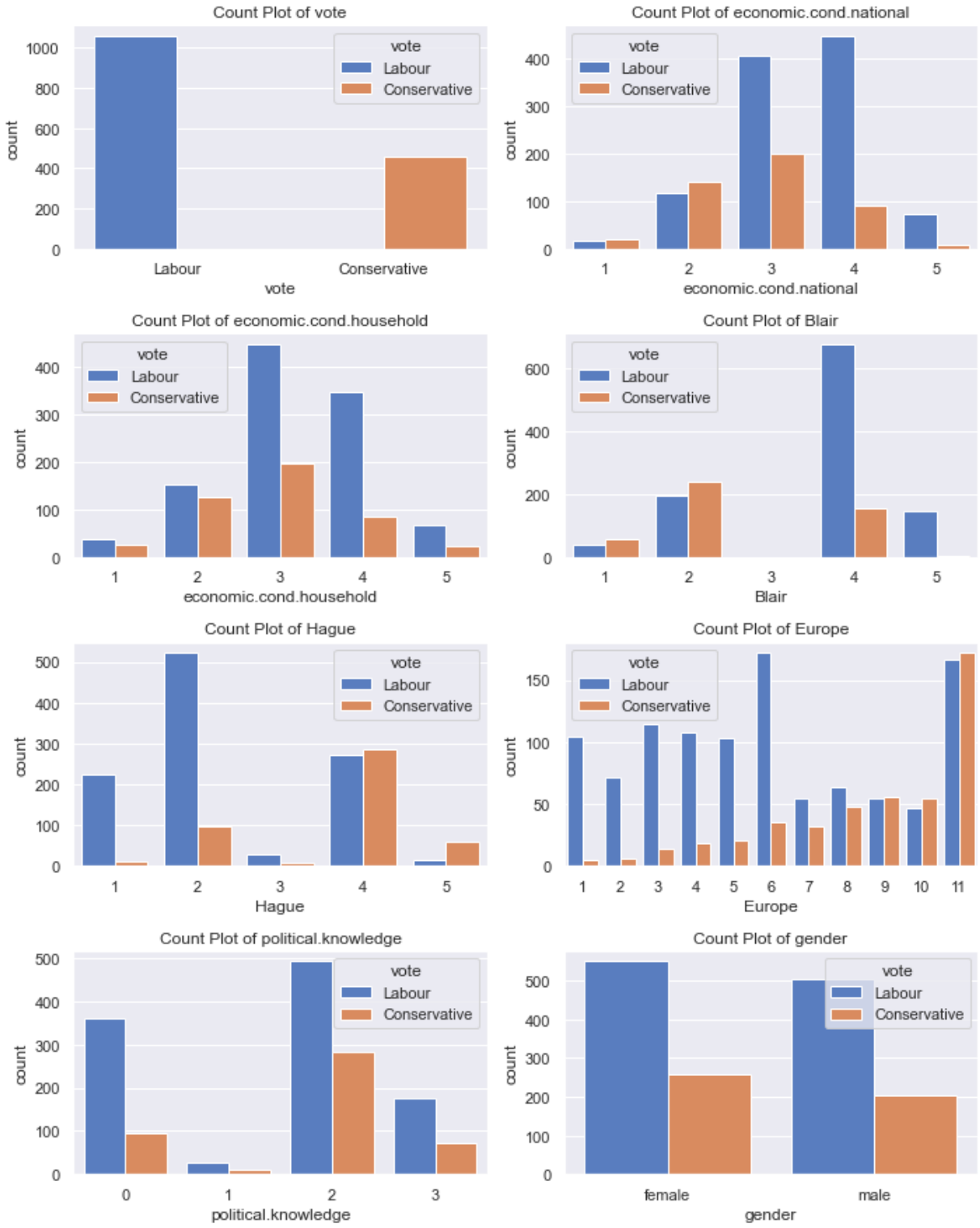

Figure 6 Bi-variate analysis using count plot

The Pair plot of the data set is as below.

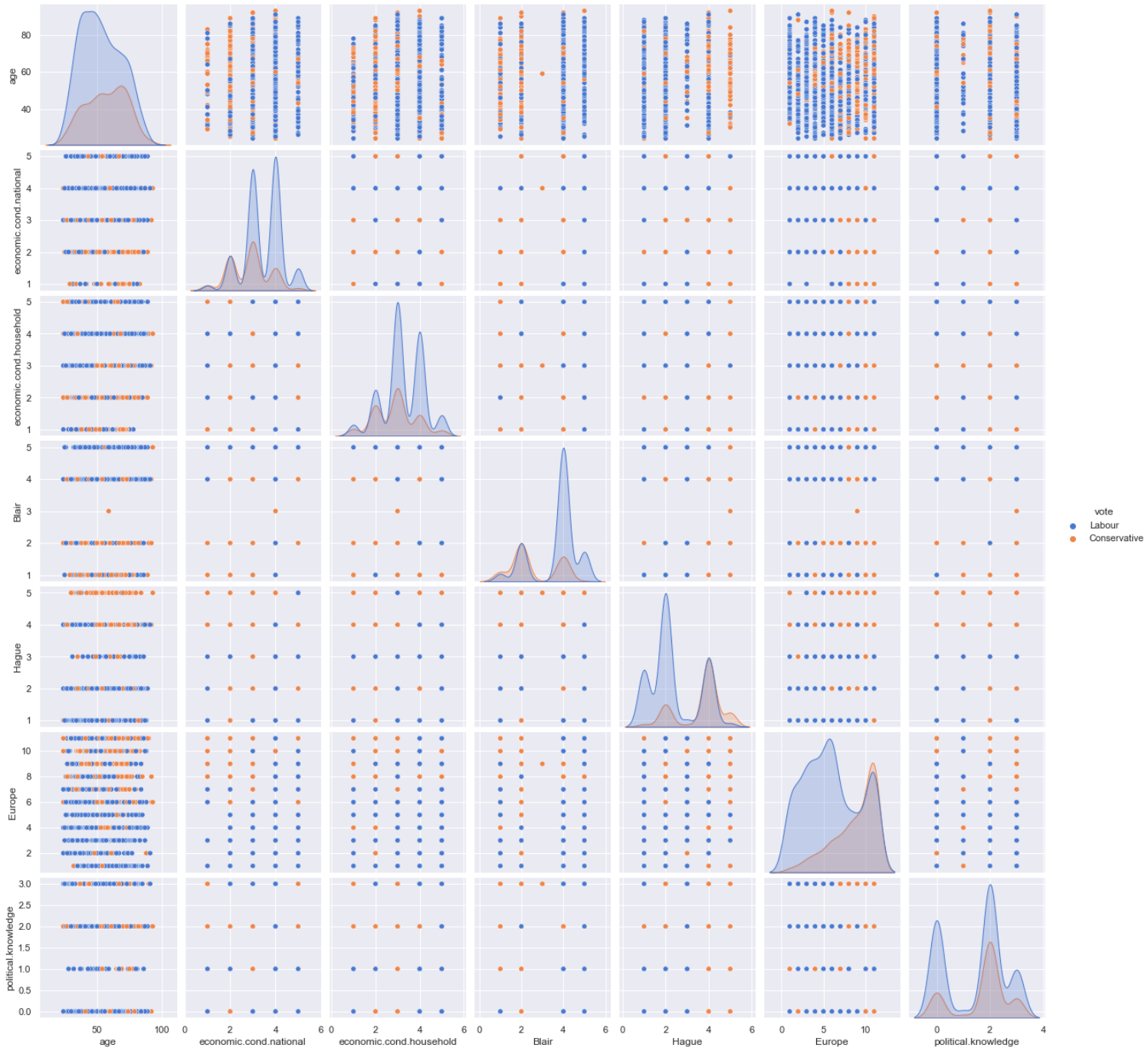

Figure 7 Pair Plot

1.3 Encode the data (having string values) for Modelling. Is Scaling necessary here or not? Data Split: Split the data into train and test (70:30).

Encoding

Modeling can be done only with numerical input. Hence the object type variables in the dataset need to be converted to numerical/ categorical type.

Here, vote is converted to numerical type by label encoding (1-Labour & 0-Conservative). Age is left as it is while the other integer type variables are converted to categorical type as they represent ordinal data type. The gender column is dummy encoded.

Scaling

In this data set, we have independent data in different scales Age is of continuous type ranging from 24 to 93, while the ordinal type variables are in a scale of 1 to 5, 1 to 11 and 1 to 3. It can also be seen that the variance is not the same across the variables. Hence the data need to be scaled. Sample of dataset after scaling using min max method is as below.

	age	economic.cond.national	economic.cond.household	Blair	Hague	Europe	political.knowledge	gender_male
count	1517.00	1517.00	1517.00	1517.00	1517.00	1517.00	1517.00	1517.00
mean	0.44	3.25	3.14	3.34	2.75	6.74	1.54	0.47
std	0.23	0.88	0.93	1.17	1.23	3.30	1.08	0.50
min	0.00	1.00	1.00	1.00	1.00	1.00	0.00	0.00
25%	0.25	3.00	3.00	2.00	2.00	4.00	0.00	0.00
50%	0.42	3.00	3.00	4.00	2.00	6.00	2.00	0.00
75%	0.62	4.00	4.00	4.00	4.00	10.00	2.00	1.00
max	1.00	5.00	5.00	5.00	5.00	11.00	3.00	1.00

Table 6 Scaled Dataset

Data Split

The Independent and dependent variables are stored as separate objects as X and y (X – dataframe, Y – Series). Using the train_test_split in SciKit learn library, the data is split as training and testing data sets to begin modeling.

MODELING

1.4 Apply Logistic Regression and LDA (linear discriminant analysis)

LOGISTIC REGRESSION MODEL

- It is a supervised learning method for classification that establishes relation between dependent class variable and independent variables using regression.
- Logistic Regression assigns probabilities to different classes.
- By using the LogisticRegression model library from sklearn.linear_model, we will be able to fit the train data in to our model.
- In defining the model, we have selected the newton-cg solver, which is the most commonly used, maximum iterations to 10000, having no penalty component, & n_jobs as 2 CPU cores to be used.
- By fitting the data, the model gets trained using the training set. By using the appropriate python codes, we will be able to predict the classes and also the probability of that record being predicted to that class
- It can be seen that the training data has an accuracy of 0.83 and testing set has an accuracy of 0.84.
- Hence, this model proves to be a good model without any under fit or over fit.

Logistic Regression Classification report

LOGISTIC REGRESSION – CLASSIFICATION REPORT

Train Data

	precision	recall	f1-score	support
0	0.74	0.64	0.69	307
1	0.86	0.91	0.88	754
accuracy			0.83	1061
macro avg	0.80	0.77	0.79	1061
weighted avg	0.83	0.83	0.83	1061

Test Data

	precision	recall	f1-score	support
0	0.76	0.74	0.75	153
1	0.87	0.88	0.88	303
accuracy			0.84	456
macro avg	0.82	0.81	0.81	456
weighted avg	0.83	0.84	0.83	456

Figure 8 Classification Report - Logistic Regression

We are able to notice that the precision & F1 Score are consistent for both 1 & 0. However, the measure in this case study is accuracy only, as we as a news channel need to be unbiased.

Logistic Regression Confusion Matrix

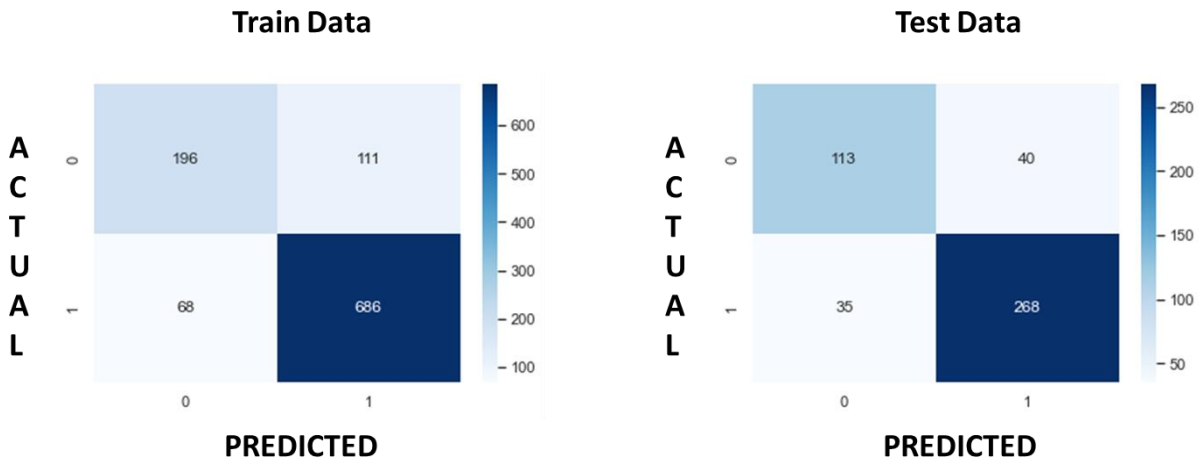

Figure 9 Confusion matrix - Logistic Regression

Logistic Regression ROC Curve

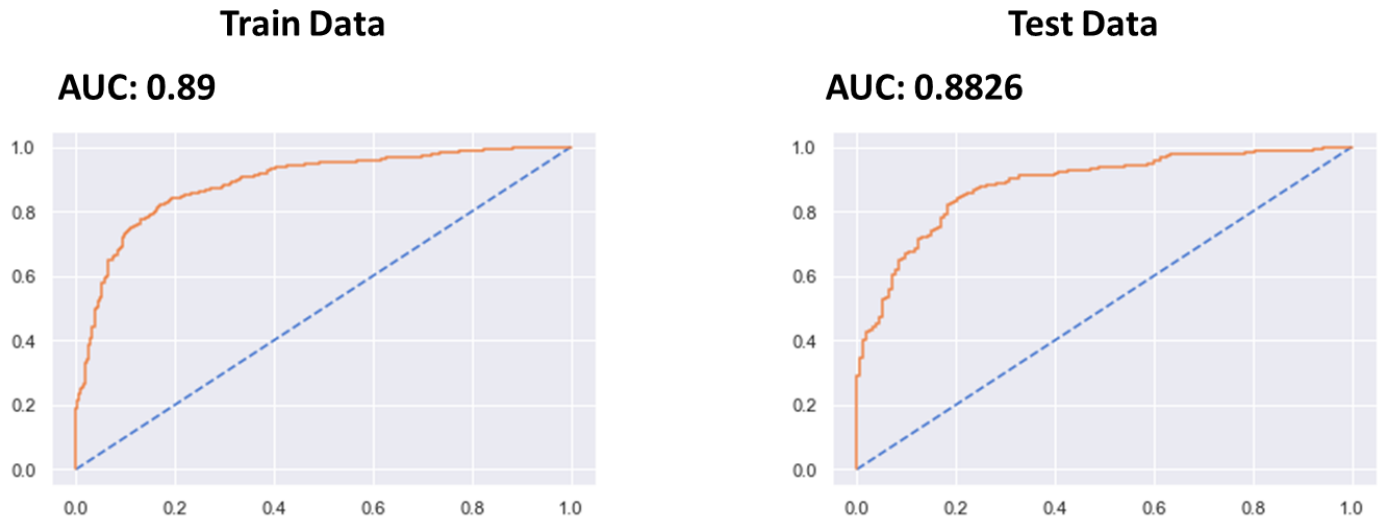

Figure 10 ROC Curve with AUC - Logistic Regression

It can be seen that the results are consistent across the training and test data.

Model Tuning

Using the Grid search cross validation with 3 fold CV, the hyperparameters solver, penalty & tolerance were given a range. Sag, lbfgs & newton-cg for solver, 12 and none for penalty & 0.0001 & 0.00001 for tolerance. The best hyper parameters turned out to be sag, 12 & 0.0001. However, on building the model with the best hyper parameters, the accuracy scores turned out be 0.83 in both test and training data. Hence, the model built with out the grid search CV remains valid.

LINEAR DISCRIMINANT ANALYSIS MODEL

- Discriminant Analysis is used for classifying observations to a class or category based on predictor (independent) variables of the data.
- Discriminant Analysis creates a model to predict future observations where the classes are known.
- LDA uses linear combinations of independent variables to predict the class in the response variable of a given observation. LDA assumes that the independent variables(p) are normally distributed and there is equal variance/ covariance for the classes. LDA is popular because it can be used for both classification and dimensionality reduction.
- By using the LinearDiscriminantAnalysis model library from sklearn.discriminant_analysis, we will be able to fit the train data in to our model.
- In defining the model, we have selected the default svd solver, which is the most commonly used & tolerance of 0.0001.
- By fitting the data, the model gets trained using the training set. By using the appropriate python codes, we will be able to predict the classes and also the probability of that record being predicted to that class
- It can be seen that the training & testing data has an accuracy of 0.83.
- Hence, this model proves to be a good model without any under fit or over fit.

LDA Classification report

Train Data

	precision	recall	f1-score	support
0	0.74	0.65	0.69	307
1	0.86	0.91	0.89	754
accuracy			0.83	1061
macro avg	0.80	0.78	0.79	1061
weighted avg	0.83	0.83	0.83	1061

Test Data

	precision	recall	f1-score	support
0	0.77	0.73	0.74	153
1	0.86	0.89	0.88	303
accuracy			0.83	456
macro avg	0.82	0.81	0.81	456
weighted avg	0.83	0.83	0.83	456

Figure 11 LDA - Classification Report

We are able to notice that the precision, recall & F1 Score are consistent for both 1 & 0. However, the measure in this case study is accuracy only, as we as a news channel need to be unbiased.

LDA Confusion Matrix

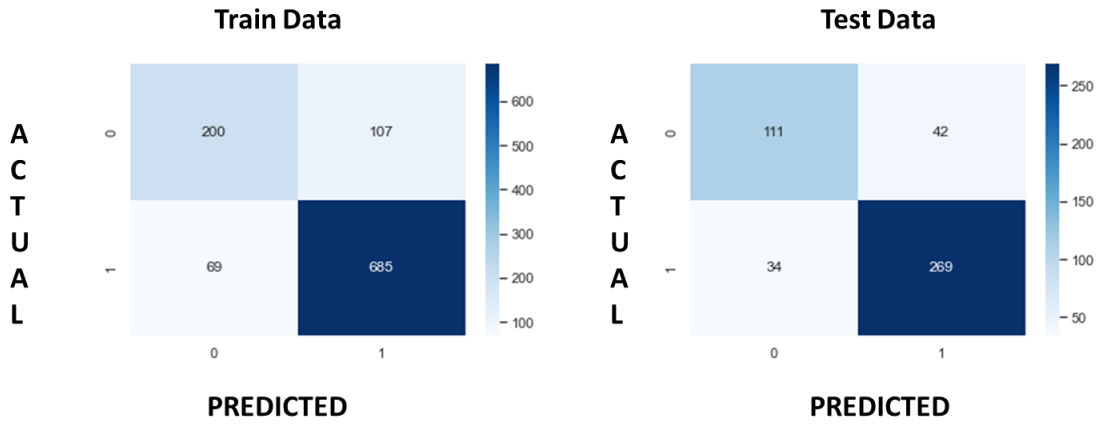

Figure 12 Confusion matrix - LDA

LDA ROC Curve

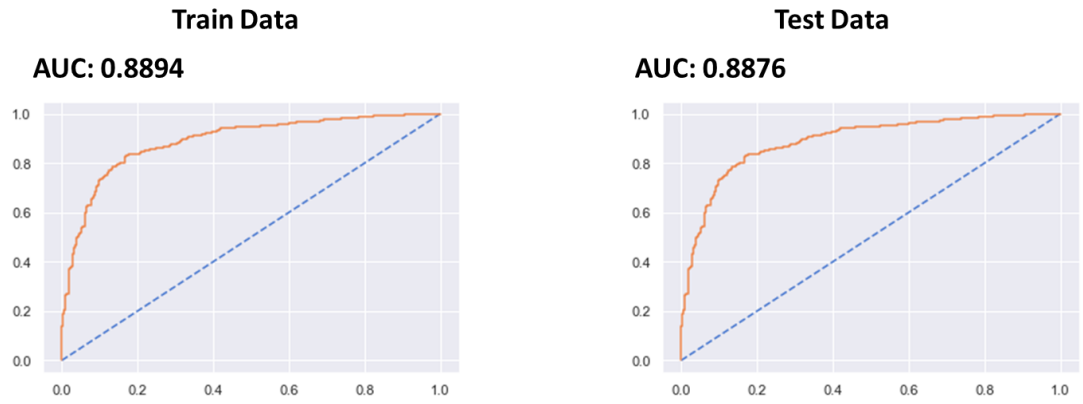

Figure 13 ROC Curve with AUC - Logistic Regression

It can be seen that the results are consistent across the training and test data.

Altering the threshold

On evaluating the model for various thresholds from 0.1 to 0.9, it can be seen that the accuracy is the highest and consistent for 0.5 & 0.6. Hence the above model is valid and holds good.

LDA V/s Logistic Regression:

From the 2 models built we can see that the accuracy of Logistic regression in test data is found to be higher than that of LDA. Hence, we can go with Logistic regression model for this cas study in comparison between these 2 models. Also, the correctly predicted values are higher for Logistic regression by 1 that is evident from True positives and true negatives of the confusion matrix.

1.5 Apply KNN Model and Naïve Bayes Model. Interpret the results

NAIVE BAYES MODEL

- Naïve Baye's Classifier works on the principle of baye's theorem with a naïve assumption of that a presence of a particular feature in a class is completely unrelated to the presence of other features(Input features are independent from each other).
- It is expressed as **Posterior Probability = (prior probability * likelihood)/ evidence**

Equation 1Naive Baye's Classifier

Naïve Baye's Working Steps

- **Step 1-** Compute the prior probabilities for given class labels.
- **Step 2-** Compute the Likelihood of evidence with each attribute for each class.
- **Step 3-** Calculate the posterior probabilities using Bayes rule.
- **Step 4-** Select the class which has higher probability for given inputs.

Model Building & fitting

- By using the GaussianNB model library from sklearn.naivebayes, we will be able to fit the train data in to our model.
- In defining the model, we have selected the default hyperparameters of priors as None. By fitting the data, the model gets trained using the training set and go with predicting.
- It can be seen that the training data has an accuracy of 0.84 and testing set has an accuracy of 0.82.
- Hence, this model proves to be a good model without any under fit or over fit.

Naïve Bayes Classification report

Train Data

	precision	recall	f1-score	support
0	0.73	0.69	0.71	307
1	0.88	0.90	0.89	754
accuracy			0.84	1061
macro avg	0.80	0.79	0.80	1061
weighted avg	0.83	0.84	0.83	1061

Test Data

	precision	recall	f1-score	support
0	0.74	0.73	0.73	153
1	0.87	0.87	0.87	303
accuracy			0.82	456
macro avg	0.80	0.80	0.80	456
weighted avg	0.82	0.82	0.82	456

Figure 14 Naive Bayes Classification Report

We are able to notice that the precision & F1 Score are consistent for both 1 & 0. However, the measure in this case study is accuracy only, as we as a news channel need to be unbiased.

Naïve Bayes Confusion Matrix

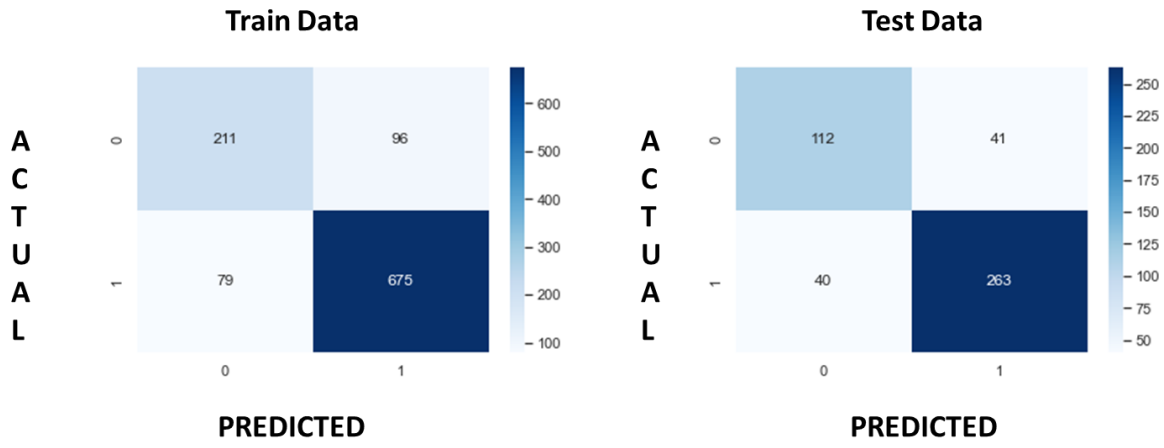

Figure 15 Confusion matrix – Naïve Bayes

Naïve Bayes ROC Curve

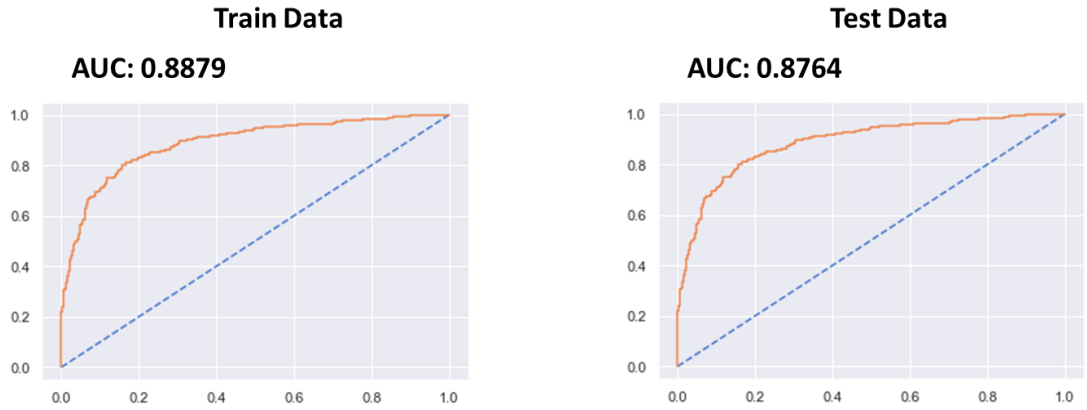

Figure 16 ROC Curve with AUC – Naïve Bayes

It can be seen that the results are consistent across the training and test data.

K – NEAREST NEIGHBORS (K-NN) MODEL

- K-NN Classifier works on the principle of distance between the query point and the nearest neighbours.
- The distance is measured using the Eucledian Distance measure.
- Determining the optimal k value is the challenge.
- Larger the value of k, suppresses impact of noise, but is prone to majority class dominating
- Usally K is taken as, **K = sqrt(n),** where n- number of records

Equation 2 Optimal K value

Model Building & fitting

- By using the KNeighborsClassifier model library from sklearn.neighbors, we will be able to fit the train data in to our model.
- In defining the model, we have selected the default hyperparameters of n_neighbors = 5, distance measure as minkowski which is generalization of Eucledian and Manhattan& weights as uniform. By fitting the data, the model gets trained using the training set and go with predicting.
- It can be seen that the training data has an accuracy of 0.85 and testing set has an accuracy of 0.81.
- However, here the optimal value of k is found by calculating the mis classification error for different values of k. From the below plot it is evident that, the optimal k value is 39 with a minimal misclassification error of 0.162.

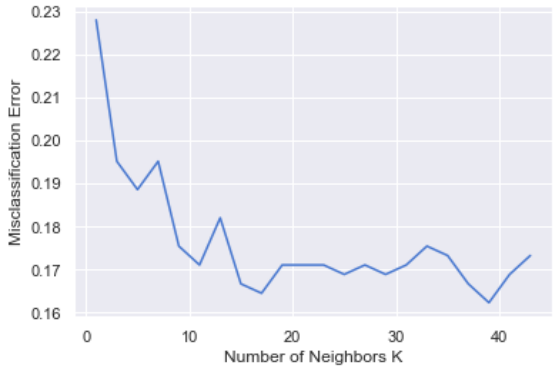

Figure 17 K V/s Misclassification Error

- The accuracy scores for this model with optimal k is found to be 0.82 & 0.84 in train & test data respectively. Hence, this model proves to be a good model without any under fit or over fit.

K-NN Classification report

Train Data

	precision	recall	f1-score	support
0	0.72	0.64	0.68	307
1	0.86	0.90	0.88	754
accuracy			0.82	1061
macro avg	0.79	0.77	0.78	1061
weighted avg	0.82	0.82	0.82	1061

Test Data

	precision	recall	f1-score	support
0	0.78	0.71	0.75	153
1	0.86	0.90	0.88	303
accuracy			0.84	456
macro avg	0.82	0.81	0.81	456
weighted avg	0.84	0.84	0.84	456

Figure 18 KNN Classification Report

We are able to notice that the precision, recall & F1 Score are consistent for both 1 & 0. However, the measure in this case study is accuracy only, as we as a news channel need to be unbiased.

KNN Confusion Matrix

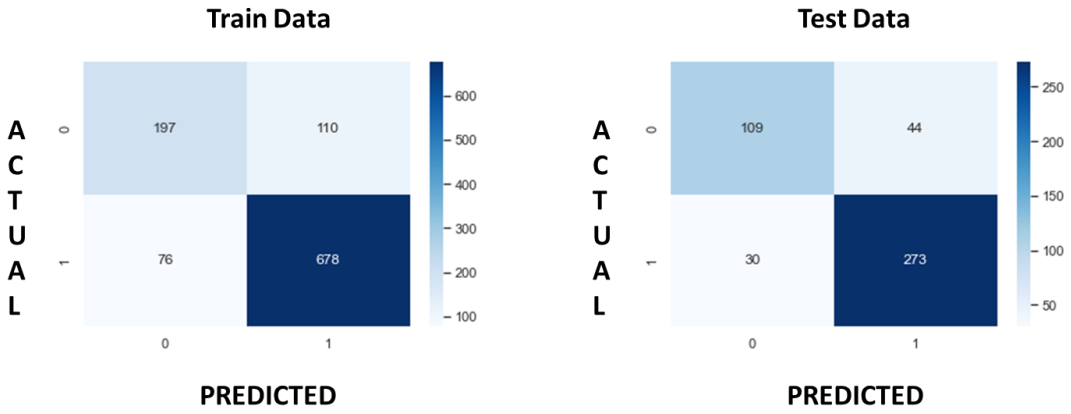

Figure 19 Confusion matrix – KNN

KNN ROC Curve

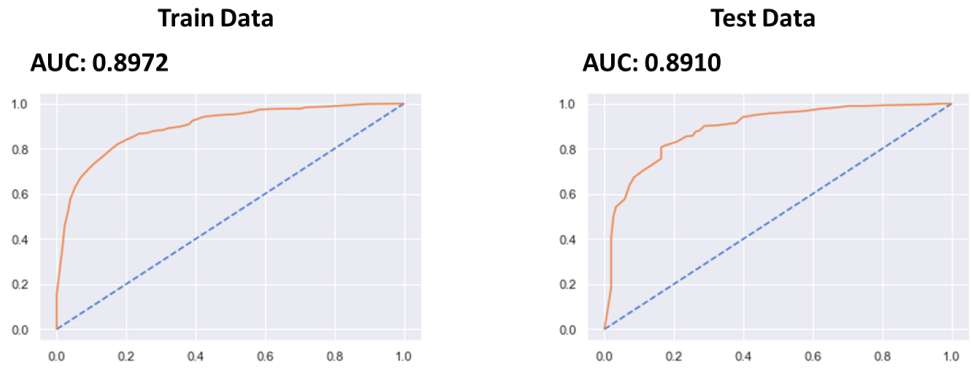

Figure 20 ROC Curve with AUC – KNN

It can be seen that the results are consistent across the training and test data.

Naïve Bayes V/s KNN:

From the 2 models built we can see that the accuracy of KNN in test data is found to be higher than that of Naïve Bayes. Hence, we can go with KNN model for this case study in comparison between these 2 models. Also, the correctly predicted values are higher for KNN by 7 that is evident from True positives and true negatives of the confusion matrix.

1.6 Model Tuning, Bagging (Random Forest should be applied for Bagging), and Boosting

RANDOM FOREST MODEL

- Random Forest is a technique, where in multiple CART models are built to get the accuracy in the predicted value.
- The model building is same except that we need to build a random forest classifier model instead of decision tree classifier model.
- We will not be able to visualize the trees in this case, as there will be multiple decision trees built. The optimum hyper parameters here are also obtained by grid search cross validation.
- In addition to the hyper parameters of Decision tree classifier, we have max_features and n_estimators.
- Max_features refer to the number of variables to be considered and the n_estimators define the number of decision trees to be built.
- Using these best hyper parameters, we shall predict the values for both training & testing dataset.
- However, for the default parameters with 100 decision trees, we can see that the model is overfit, as the training data set has an accuracy of 1 while testing data set has 0.83 accuracy.
- On tuning the model by creating a grid with a range of values for hyper parameters and running them on grid search CV, we obtain some best parameters.
- On two iterations, we receive similar results of accuracy for the test data. Hence we can go ahead with any one of the model built with best parameters.
- The model we are considering has max_depth: 10, max_features: 3, min_samples_leaf: 10, min_samples_split: 30, n_estimators: 50
- This model has a train accuracy of 0.86 & test accuracy of 0.83.
- Thus, we can see that the model is properly fit and can be considered as a good model to proceed with.

RF Classification report

Train Data

	precision	recall	f1-score	support
0	0.79	0.69	0.74	307
1	0.88	0.93	0.90	754
accuracy			0.86	1061
macro avg	0.84	0.81	0.82	1061
weighted avg	0.85	0.86	0.85	1061

Test Data

	precision	recall	f1-score	support
0	0.80	0.66	0.72	153
1	0.84	0.91	0.88	303
accuracy			0.83	456
macro avg	0.82	0.79	0.80	456
weighted avg	0.83	0.83	0.82	456

Figure 21 Classification Report - RF

We are able to notice that the precision, recall & F1 Score are consistent for both 1 & 0. However, the measure in this case study is accuracy only, as we as a news channel need to be unbiased.

RF Confusion Matrix

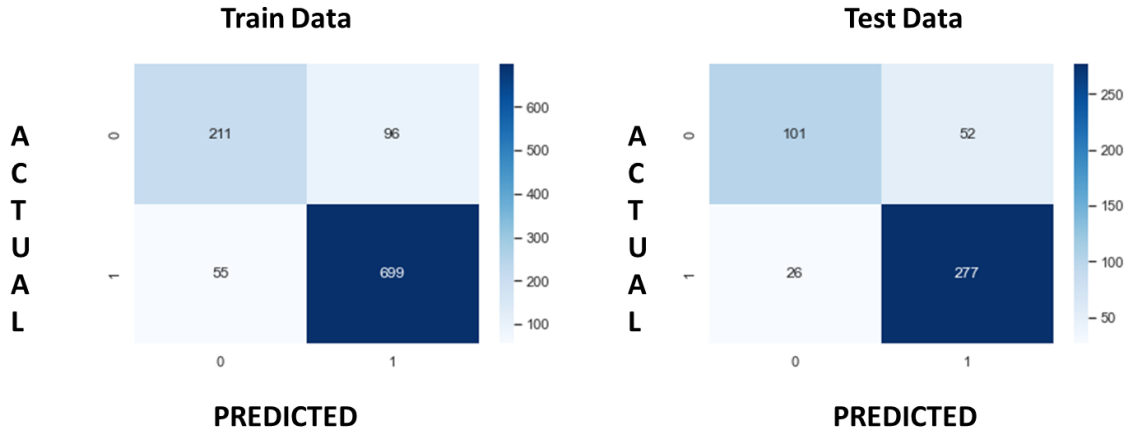

Figure 22 Confusion Matrix – RF

RF ROC Curve

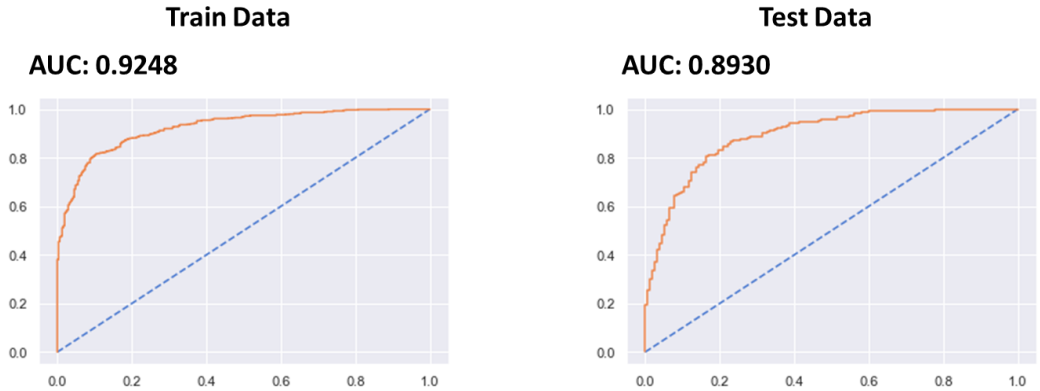

Figure 23 ROC Curve with AUC – RF

It can be seen that the results are consistent across the training and testing data

BAGGING MODEL

- Bagging is a technique of merging the outputs of various models to get a final result.
- But if we go by this method it will have higher probability that these different models generate same results as they are implemented by same input data. This problem can be mitigated by the technique known as **bootstrapping.**
- Bootstrapping is a technique of sampling by which we can create sub sample of observation with the actual dataset, with replacement.
- Bagging is also called as bootstrap aggregating.
- There will be reduced chances of over fitting by training each model only with a randomly chosen subset of the training data.
- Training can be done in parallel. Essentially trains a large number of “strong” learners in parallel (each model is an over fit for that subset of the data)
- Combines (averaging or voting) these learners together to "smooth out" predictions.

Model Building & fitting

- By using the BaggingClassifier model library from sklearn.ensemble, we will be able to fit the train data in to our model.
- In defining the model, we need to define a base estimator model for which the bagging needs to be done. In this particular case study, we will use the Random forest as base model with 100 Decision trees, which must be first created before creating the Bagging classifier model
- It can be seen that the training data has an accuracy of 0.97 and testing set has an accuracy of 0.83.
- However, this model is overfit.

Bagging Classification report

Train Data

	precision	recall	f1-score	support
0	0.98	0.90	0.94	307
1	0.96	0.99	0.98	754
accuracy			0.97	1061
macro avg	0.97	0.95	0.96	1061
weighted avg	0.97	0.97	0.97	1061

Test Data

	precision	recall	f1-score	support
0	0.78	0.68	0.73	153
1	0.85	0.90	0.88	303
accuracy			0.83	456
macro avg	0.82	0.79	0.80	456
weighted avg	0.83	0.83	0.83	456

Figure 24 Classification Report - Bagging Classifier

We are able to notice that the precision, recall & F1 Score are inconsistent for both 1 & 0. However, the measure in this case study is accuracy only, as we as a news channel need to be unbiased.

Bagging Classifier Confusion Matrix

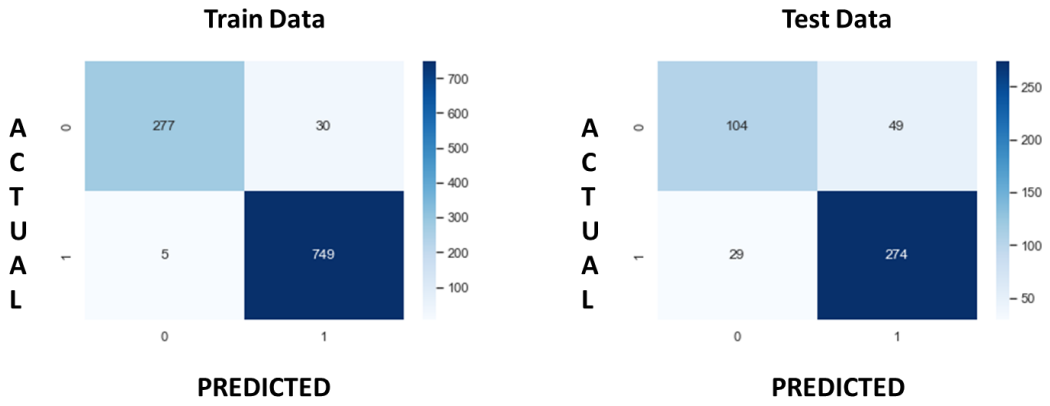

Figure 25 Confusion Matrix – Bagging

RF ROC Curve

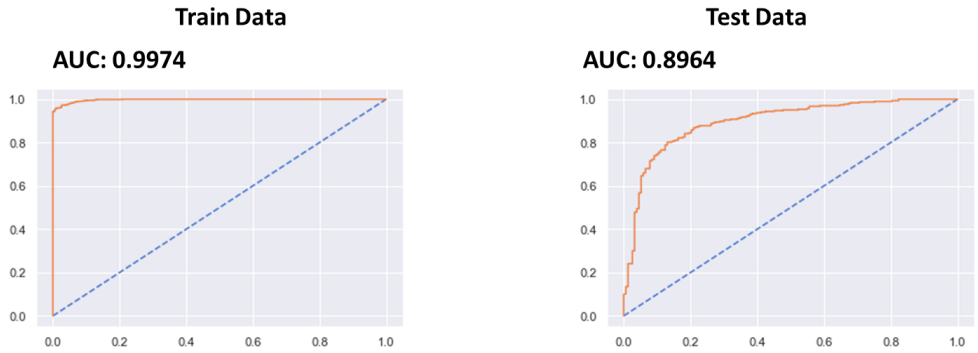

Figure 26 ROC Curve with AUC – Bagging

It can be seen that the results are inconsistent across the training and testing data

BOOSTING MODEL

- Boosting is a linear sequential process, where next or upcoming model tries to minimize the errors made by previous model in prediction.
- This method is different from bagging in the sense where each succeeding model is dependent on the previous model.
- Trains a large number of "weak" learners in sequence. A weak learner is a simple model that is only slightly better than random (E.g., One depth decision tree).
- Miss-classified data weights are increased for training the next model. So, training has to be done in sequence.
- Boosting then combines all the weak learners into a single strong learner.
- Bagging uses complex models and tries to "smooth out" their predictions, while Boosting uses simple models and tries to "boost" their aggregate complexity.
- There are 2 boosting methods, Adaptive boosting and Gradient boosting
- In Ada Boost (adaptive boosting), the successive learners are created with a focus on the ill fitted data of the previous learner.
- Each successive learner focuses more and more on the harder to fit data i.e., their residuals in the previous tree
- In Gradient boosting, each learner is fit on a modified version of original data (original data is replaced with the x values and residuals from previous learner)
- By fitting new models to the residuals, the overall learner gradually improves in areas where residuals are initially high

Ada Boost Model Building & fitting

- By using the AdaBoostClassifier model library from sklearn.ensemble, we will be able to fit the train data in to our model.

- In defining the model, we need to define number of estimators i.e., no. of decision trees, which we consider as 100
- It can be seen that the training data has an accuracy of 0.85 and testing set has an accuracy of 0.81.
- It can be seen that model performs equally well on both training & testing data.

Ada Boost Classification report

Train Data

	precision	recall	f1-score	support
0	0.76	0.70	0.73	307
1	0.88	0.91	0.90	754
accuracy			0.85	1061
macro avg	0.82	0.80	0.81	1061
weighted avg	0.85	0.85	0.85	1061

Test Data

	precision	recall	f1-score	support
0	0.75	0.67	0.71	153
1	0.84	0.88	0.86	303
accuracy			0.81	456
macro avg	0.79	0.78	0.79	456
weighted avg	0.81	0.81	0.81	456

Figure 27 Ada Boost Classification report

We are able to notice that the precision, recall & F1 Score are consistent for both 1 & 0. However, the measure in this case study is accuracy only, as we as a news channel need to be unbiased.

Ada Boost Confusion Matrix

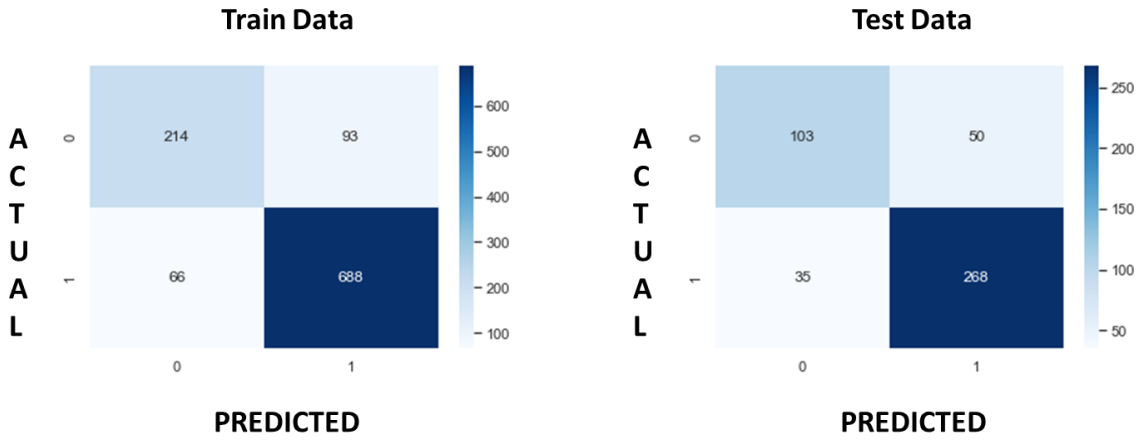

Figure 28 Confusion Matrix – Ada Boost

Ada Boost ROC Curve

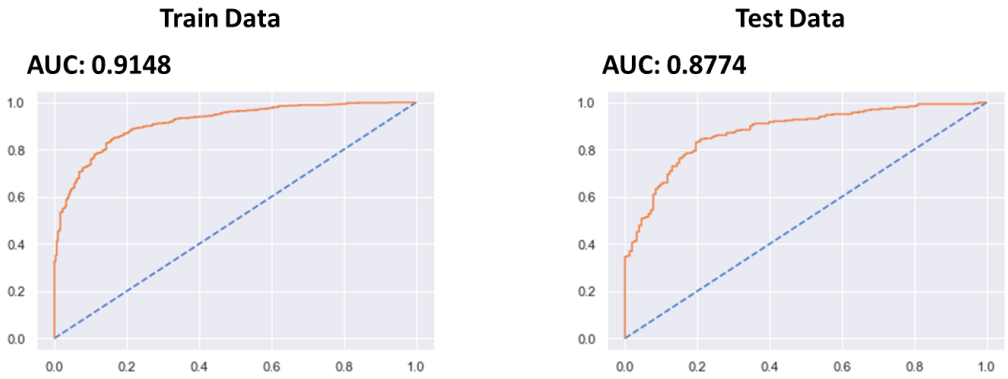

Figure 29 ROC Curve with AUC – Ada Boost

It can be seen that the results are consistent across the training and testing data

Gradient Boost Model Building & fitting

- By using the GradientBoostingClassifier model library from sklearn.ensemble, we will be able to fit the train data in to our model.
- In defining the model, we need to define the hyper parameters. We will be going with the default parameters in this case study
- It can be seen that the training data has an accuracy of 0.89 and testing set has an accuracy of 0.83.
- It can be seen that model performs equally well on both training & testing data.

Gradient Boost Classification report

Train Data

	precision	recall	f1-score	support
0	0.84	0.78	0.81	307
1	0.91	0.94	0.93	754
accuracy			0.89	1061
macro avg	0.88	0.86	0.87	1061
weighted avg	0.89	0.89	0.89	1061

Test Data

	precision	recall	f1-score	support
0	0.79	0.68	0.73	153
1	0.85	0.91	0.88	303
accuracy			0.83	456
macro avg	0.82	0.80	0.81	456
weighted avg	0.83	0.83	0.83	456

Figure 30 Gradient Boost Classification report

We are able to notice that the precision, recall & F1 Score are consistent for both 1 & 0. However, the measure in this case study is accuracy only, as we as a news channel need to be unbiased.

Gradient Boost Confusion Matrix

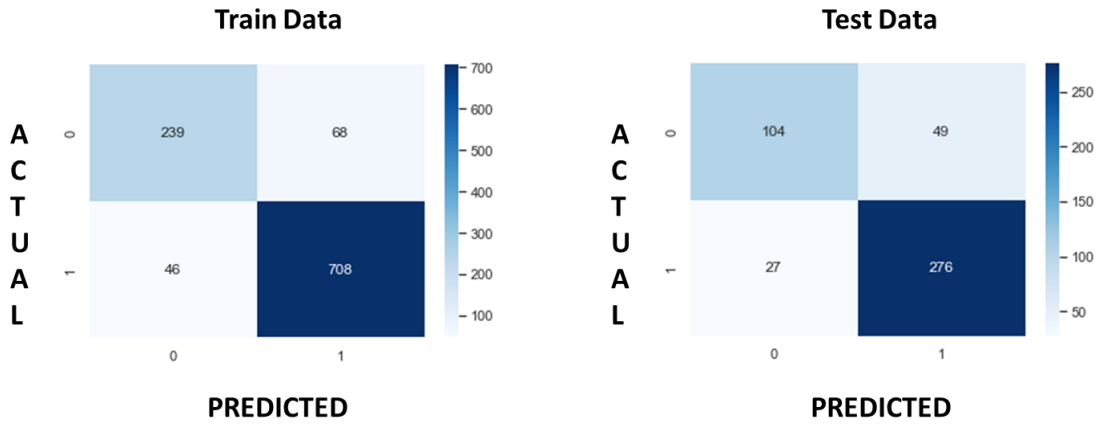

Figure 31 Confusion Matrix – Gradient Boost

Gradient Boost ROC Curve

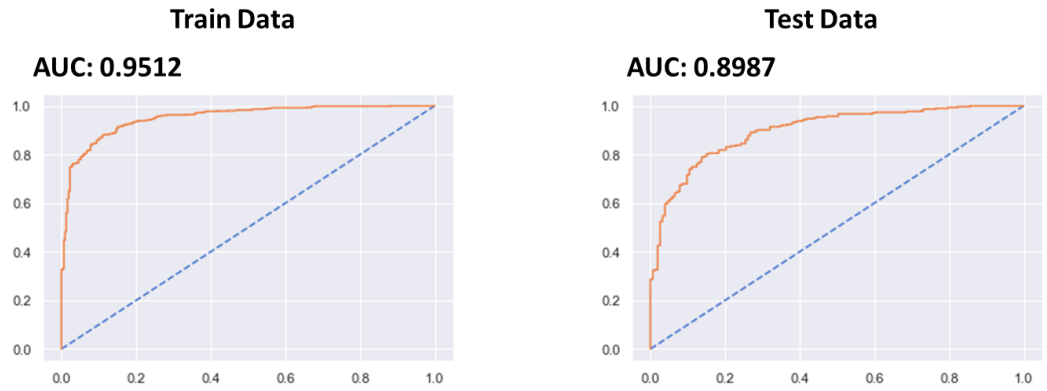

Figure 32 ROC Curve with AUC – Gradient Boost

It can be seen that the results are consistent across the training and testing data

Bagging v/s Boosting:

From the 2 models built we can see that the Bagging is over fit while boosting (both types) were better fit models. However, we can see that Gradient Boosting performs better, when comes to the accuracy score with a score of 0.83 compared to 0.81 in the ada boost model.

1.7 Performance Metrics: Check the performance of Predictions on Train and Test sets using Accuracy, Confusion Matrix, Plot ROC curve and get ROC_AUC score for each model. Final Model: Compare the models and write inference which model is best/optimized

The confusion matrix, accuracy scores, classification reports, ROC curve & AUC of all the models are already discussed. Considering the nature of problem, it is understandable that the important metric here is the accuracy score. The below table has the list of accuracy scores of train and test data for each of the models discussed above.

Model Name	Train	Test	Model Fit
	Accuracy Score	Accuracy Score	
Logistic Regression	0.83	**0.84**	Good
Logistic Regression with Grid Search CV	0.83	0.83	Good
LDA	0.83	0.83	Good
NB	0.84	0.82	Good
KNN	0.85	0.81	Good
KNN with optimal K value	0.82	**0.84**	Good
Bagging	1	0.82	Overfit
Ada Boost	0.85	0.81	Good
Gradient Boost	0.89	0.83	Good
Random Forest	1	0.83	Overfit
Random Forest with best param	0.86	0.83	Good

Table 7 Accuracy Scores & Model Fit - All models

From the above table, we can see that except Bagging & Random forest (that are overfit), all the models are having a good fit. This can be seen from their consistent accuracy scores in the prediction performance of training and testing data. It can also be seen that, Logistic regression and KNN with optimal K value models are found to have the higher accuracy in test data than in the train data. It can also be noted that these two models have the best accuracy score (0.84) in test data.

Further tabulating the total number of correct predictions in the test data for each of the model as in the table below, it can be seen that Logistic regression and KNN with optimal K value are found to have the greatest number of correct predictions (381 & 382 respectively). Hence, both these models can be considered for prediction.

Test Data									
	Logistic Regression	LDA	NB	KNN with optimal k	RF Base	RF with best param	Bagging	Ada Boost	Gradient Boost
True Negative	113	111	112	109	105	101	104	103	104
True Positive	268	269	263	273	275	277	274	268	276
Total Correct Predictions	381	380	375	382	380	378	378	371	380

Table 8 Correct number of predictions - All Models

1.8 Based on these predictions, what are the insights.

- From the given dataset it can be seen that the records are 70% inclined towards labour and 30% inclined towards conservatives.
- Across age group there is almost equivalent number of voters for both the parties
- It can also be seen that the voters for both the parties have a Eurosceptic feel, which indicates that irrespective of which party comes to power, they would be against European Integration
- Considering an assessment of 3 in both the economic conditions, each of the party would try to propose policies that would try improving the economic condition.
- However, considering the survey of only 1525 people, which is very less compared to the population, it is recommended to further extend the survey to get a better picture on predicting the results of the election

PROBLEM 2 – TEXT ANALYTICS

Problem Statement

In this particular problem, we are going to work on the inaugural corpora from the nltk in Python. We will be looking at the following speeches of the Presidents of the United States of America:

1. President Franklin D. Roosevelt in 1941
2. President John F. Kennedy in 1961
3. President Richard Nixon in 1973

Corpora Description

This corpus consists of the 59 speeches by the Presidents of the United States of America from 1789 to 2021 stored as text files.

1789-Washington.txt	1829-Jackson.txt	1869-Grant.txt	1909-Taft.txt	1949-Truman.txt	1989-Bush.txt
1793-Washington.txt	1833-Jackson.txt	1873-Grant.txt	1913-Wilson.txt	1953-Eisenhower.txt	1993-Clinton.txt
1797-Adams.txt	1837-VanBuren.txt	1877-Hayes.txt	1917-Wilson.txt	1957-Eisenhower.txt	1997-Clinton.txt
1801-Jefferson.txt	1841-Harrison.txt	1881-Garfield.txt	1921-Harding.txt	1961-Kennedy.txt	2001-Bush.txt
1805-Jefferson.txt	1845-Polk.txt	1885-Cleveland.txt	1925-Coolidge.txt	1965-Johnson.txt	2005-Bush.txt
1809-Madison.txt	1849-Taylor.txt	1889-Harrison.txt	1929-Hoover.txt	1969-Nixon.txt	2009-Obama.txt
1813-Madison.txt	1853-Pierce.txt	1893-Cleveland.txt	1933-Roosevelt.txt	1973-Nixon.txt	2013-Obama.txt
1817-Monroe.txt	1857-Buchanan.txt	1897-McKinley.txt	1937-Roosevelt.txt	1977-Carter.txt	2017-Trump.txt
1821-Monroe.txt	1861-Lincoln.txt	1901-McKinley.txt	1941-Roosevelt.txt	1981-Reagan.txt	2021-Biden.txt
1825-Adams.txt	1865-Lincoln.txt	1905-Roosevelt.txt	1945-Roosevelt.txt	1985-Reagan.txt	

Table 9 Files in the inaugural corpus

2.1 Find the number of characters, words and sentences for the mentioned documents. (Hint: use .words(), .raw(), .sent() for extracting counts).

The below table shows the number of characters, words & sentences in the 3 speeches (President Franklin D. Roosevelt in 1941, President John F. Kennedy in 1961, President Richard Nixon in 1973). These are obtained by using the codes .raw(), .words(), .sents() for characters, words & sentences respectively.

	No of characters	No of words	No of sentences
1941-Roosevelt	7571	1536	68
1961-Kennedy	7618	1546	52
1973-Nixon	9991	2028	69

Table 10 Count Table - Before Eliminating stop words

2.2 Remove all the stop words from the three speeches. Show the word count before and after the removal of stop words. Show a sample sentence after the removal of stop words.

Stop words are those words that don not give any context during text analysis (E.g., Me, Myself, etc…). There are a set of stop words that are pre-defined in the nltk (Natual Language Tool Kit) library of python. These stop words can be downloaded along with punctuation library (punkt) in python. The set of punctuations can be appended along with the stop words. On going through the text, it can be noticed that there are '- - ' in the speech that may denote a pause during the speech. These are not available in the punctuation and hence need to appended along with the stop words.

Once the set of stop words for this corpus are determined, we need to convert the text in each of the speeches to lower case, as by default the upper and lower cases are interpreted as different characters in computer language. Once, the texts are converted to lower case, we can go ahead and remove the stop words through a loop for all the 3 speeches separately. The count of words in the speeches before and after stop words removal are as in the table below.

Speech	Before	After
	No of words	No of words
1941-Roosevelt	1536	632
1961-Kennedy	1546	697
1973-Nixon	2028	836

Table 11 Word Count of Speeches before and after Stop words removal

Sample sentence before and after stop words removal is as in the table below:

Speech	Sample Sentence	
	Before	After
1941-Roosevelt	On each national day of inauguration since 1789, the people have renewed their sense of dedication to the United States.	national day inauguration since 1789 people renewed sense dedication united states
1961-Kennedy	The world is very different now.	world different
1973-Nixon	When we met here four years ago, America was bleak in spirit, depressed by the prospect of seemingly endless war abroad and of destructive conflict at home.	met four years ago america bleak spirit depressed prospect seemingly endless war abroad destructive conflict home

Table 12 Sample sentence in speeches before and after stop words removal

2.3 Which word occurs the most number of times in his inaugural address for each president? Mention the top three words. (after removing the stopwords).

The top three words that occur the greatest number of times after stop words removal are summarized in the table below. This can be found by using the .FreqDist() feature available in the nltk library.

Speech	Word 1	Frequency	Word 2	Frequency	Word 3	Frequency
1941-Roosevelt	nation	12	know	10	spirit	9
1961-Kennedy	let	16	us	12	world	8
1973-Nixon	us	26	let	22	america	21

Table 13 Word Frequency Table

2.4 Plot the word cloud of each of the three speeches. (after removing the stop words)

Word clouds are visual representation of text data. The word cloud in python can be obtained by using the wordcloud library. The word cloud post removal of stop words from each of the 3 speeches is as displayed in the figures below.

1941-Roosevelt:

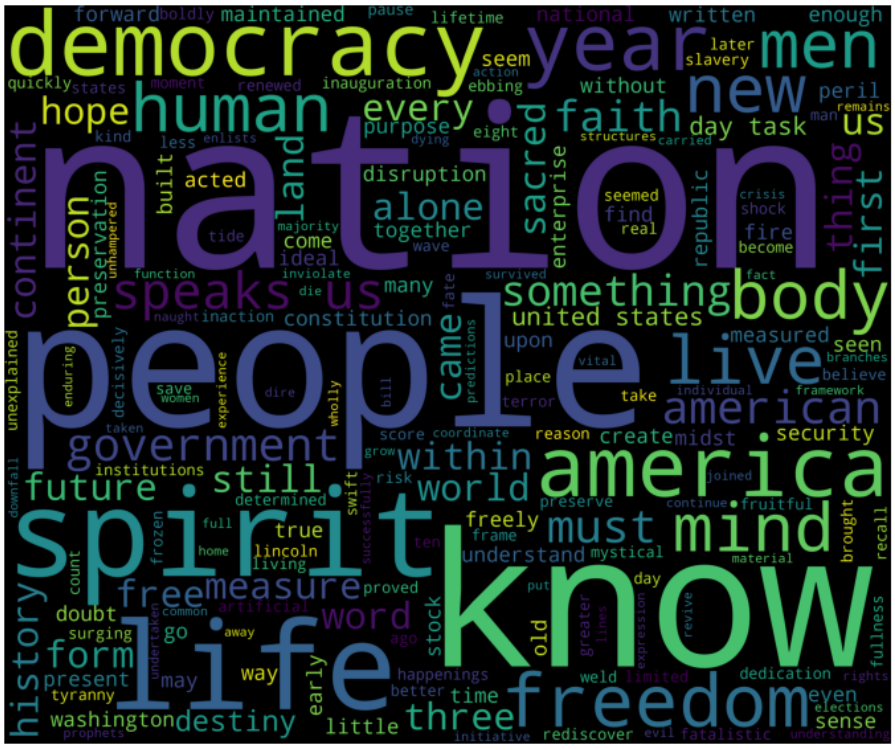

Figure 33 Word Cloud - 1941 Roosevelt

1961-Kennedy:

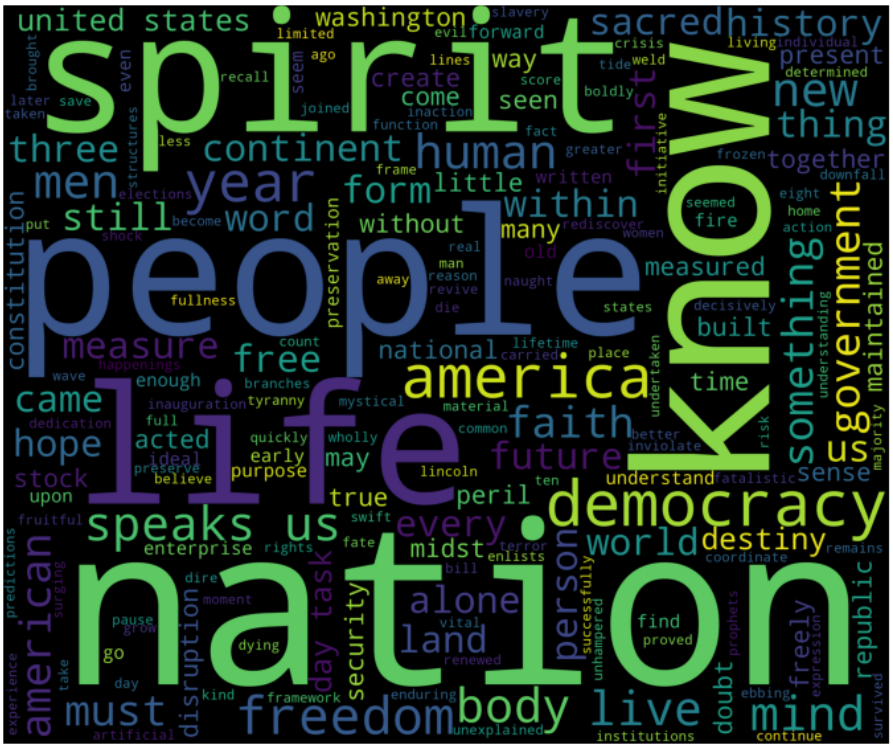

Figure 34 Word Cloud - 1961 - Kennedy

1973 - Nixon:

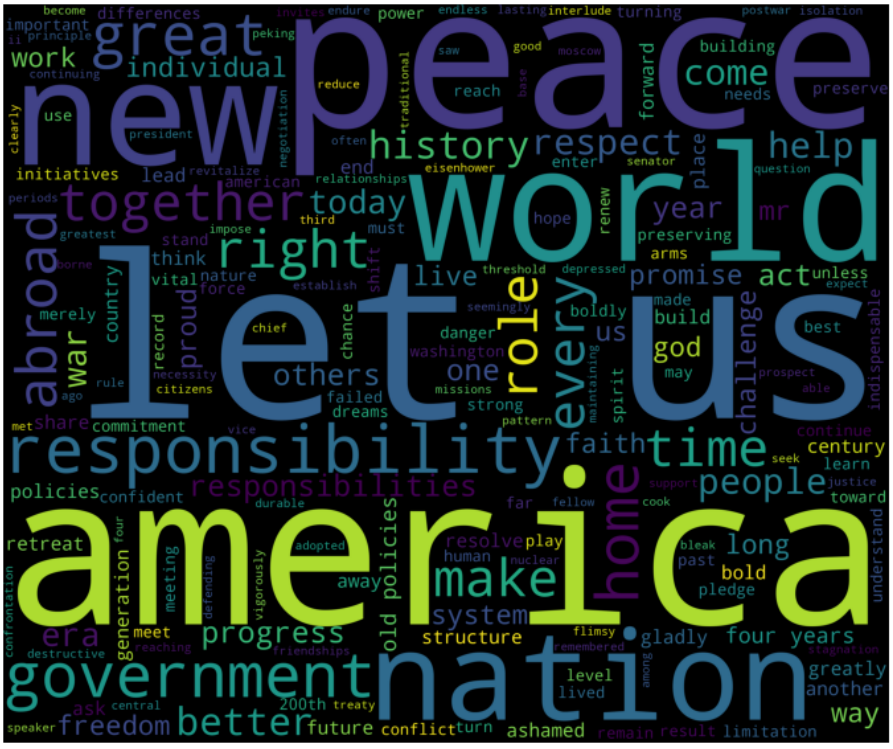

Figure 35 Word Cloud - 1973 - Nixon

END