Machine Learning

Project

Rateesh Upendran

August 2020

Problem - 1

You are hired by one of the leading news channel 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 Descriptive Statistics

Summary of the original data

	count	mean	std	min	25%	50%	75%	max
age	1525.0	54.182295	15.711209	24.0	41.0	53.0	67.0	93.0
economic.cond.national	1525.0	3.245902	0.880969	1.0	3.0	3.0	4.0	5.0
economic.cond.household	1525.0	3.140328	0.929951	1.0	3.0	3.0	4.0	5.0
Blair	1525.0	3.334426	1.174824	1.0	2.0	4.0	4.0	5.0
Hague	1525.0	2.746885	1.230703	1.0	2.0	2.0	4.0	5.0
Europe	1525.0	6.728525	3.297538	1.0	4.0	6.0	10.0	11.0
political.knowledge	1525.0	1.542295	1.083315	0.0	0.0	2.0	2.0	3.0

- The given data-set is of 1525 rows of observations from an exit poll, across 9 dimensions of type integer and objects

	count	unique	top	freq
vote	1525	2	Labour	1063
gender	1525	2	female	812

- Only the 'age' variable is found to be numerical and continuous in nature and rest of the integer type variables are ordinal in nature, representing a voter's assessment on different factors which influenced his/her vote

```
1  df.isnull().sum()
```

```
vote                    0
age                     0
economic.cond.national  0
economic.cond.household 0
Blair                   0
Hague                   0
Age_Group               0
Europe                  0
political.knowledge     0
gender                  0
```

- The categoric variables 'vote' and 'gender' represent which party did he/she voted for and their gender
- No variables were found to be having 'null' values and there were only 8 duplicate rows which were retained as is

- The ordinal variables in integer type were converted to object type for ease of analysis and visualization
- A new dimension, 'Age_Group' was created from the continuous variable 'age', where the grouping was done as follows

Order	Age group
1	Up to 35 years
2	36 to 50 years
3	51 to 65 years
4	66 to 80 years
5	Above 80

- From the descriptive summary of the dataset it can be inferred that the average age of the voters is 54 years and median is 53 years, which indicates a normal distribution of age
- The youngest of the voter is of 24 years. Which is the minimum age of voting in UK and the eldest is of 93 years
- 50% of the voters are between the age of 41 years and 67 years
- An average voter's assessment of the national and household economic condition is around 3, with 607 and 648 respondents respectively choosing 3
- The voters assessment on the leaders of the Labour and Conservative parties, Mr. Tony Blair and Mr. William Hague draws a contrasting picture. While majority (836) ranked Blair at 4, Hague was ranked 2 by 624 voters

Summary of data converted to categoric

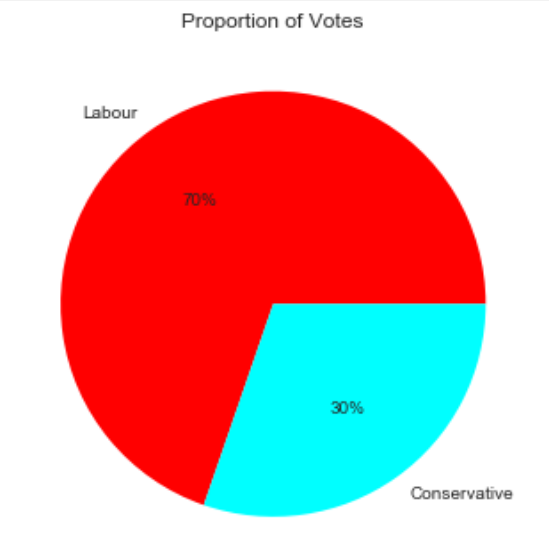
	count	unique	top	freq
vote	1525	2	Labour	1063
economic.cond.national	1525	5	3	607
economic.cond.household	1525	5	3	648
Blair	1525	5	4	836
Hague	1525	5	2	624
Europe	1525	11	11	338
political.knowledge	1525	4	2	782
gender	1525	2	female	812
Age_Group	1525	5	2	479

- While all the afore mentioned ordinal factors showed a minimal deviation of approximately 1%, the factor 'Europe' shows a variance above 3%
- 'Europe' assessed the level of scepticism of a voter on influence of European union, from 1 to 11, where 11 being highly 'Eurosceptic'
- While a significant group of voters (338) were highly Eurosceptic (11) an average voter was found to be ranked little above 6
- The voters seem to be less aware on political party's position on European integration of UK, as the factor 'political_knowledge' shows a mean of little above 1, and max frequency 2, out of 3
- Based on this exit polls, Labour party is getting a clear mandate with 1063 votes out of 1525 respondents

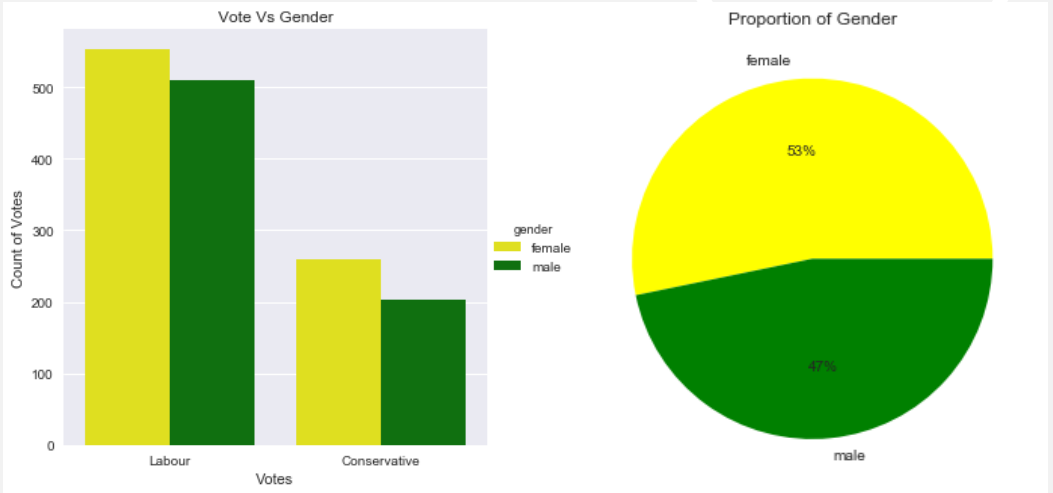
1.2 Exploratory Data Analysis

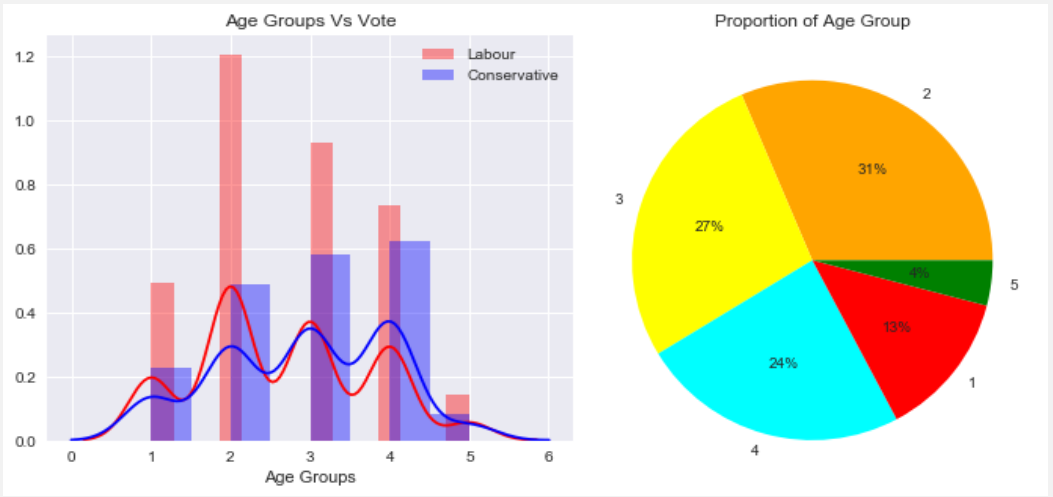
Demographic factors

- The exit poll data clearly shows a winning mandate for the Labour party with 70% of votes in their favour, which is a whooping lead of 40% over Conservatives
- From the data, one can infer that 53% of the poll respondents are female and 47% male. On analysing the voting patterns, both Labours and Conservatives got more votes from women than men



- The age group distribution of voters provide us interesting insights on the voting patterns
- 57% of the voters are found to be above 55 years of age
- 31% of the voters are in the age group between 36 and 50 years, where as those between 24 and 35 are only 13%
- The group between age 51 and 65 years make 27%. The elderly voters between 66 and 80 years make an impressive 24% and above 80 years 4%
- On analysing the votes polled, it can be observed that the Labour party got more votes from all age groups than the Conservatives
- The Conservative party is observed to be clearly favoured by the elderly voters than the younger ones. The average age of a Conservative voter is 57 years
- The Labours are found to be favoured by the younger voters with an average age of 53 years. The voters between age 36 and 50 years makes the pillar stone of Labour's mandate
- 58% of all voters are between 36 years and 65 years of age and they make the key portfolio of voters who decide the electoral fortune of the parties

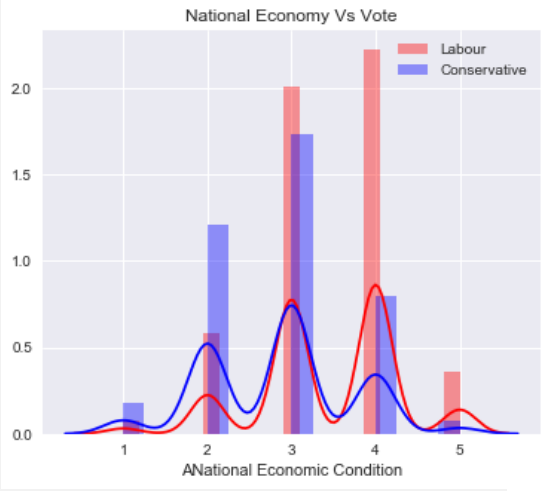


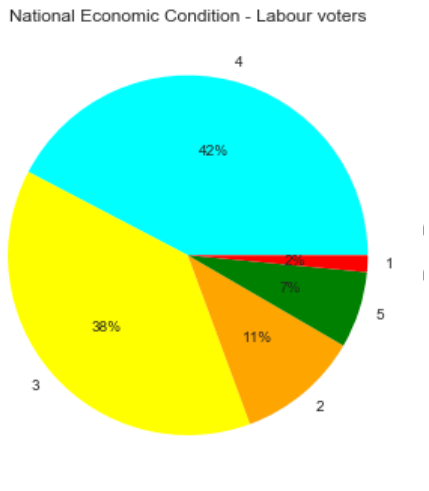


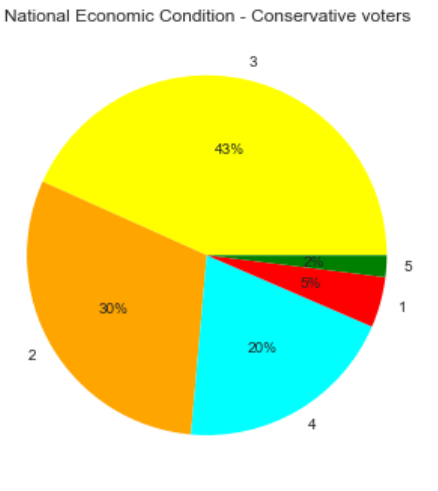
1.2 Exploratory Data Analysis

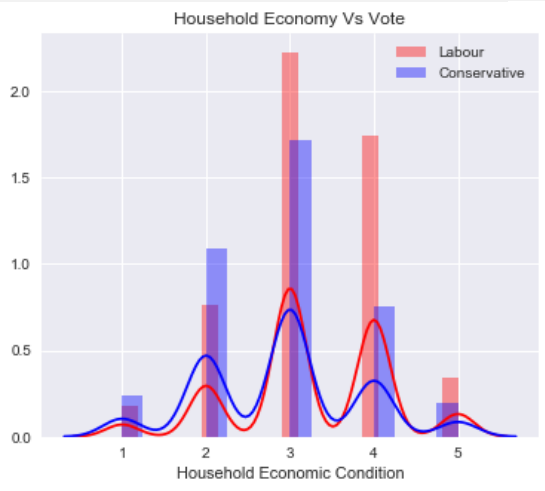
Economic factors

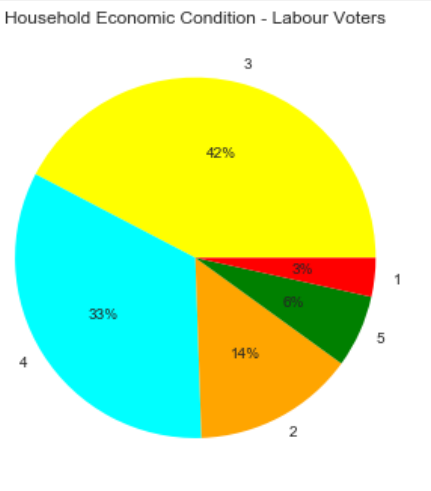
- The features representing every voters assessment on the national and household economic condition gives us an insight into the prevailing economic situation of the period in which the elections were conducted
- 75% of the Labour voters ranked the national economic condition on or above 3, on a scale 1 to 5 were 5 being the highest rank
- Interestingly 68% of the Conservative voters also ranked the national economic condition on or above 3. Only 35% of the conservative voters believe the condition is bad
- 81% of the Labour voters have ranked the household economic condition on or above 3, on a scale of 1 to 5
- 67% of the Conservative voters also ranked the household economic condition on or above 3. Only 33% of Conservative voters ranked the condition below 3
- 81% and 77% of all the voters have ranked the national and household economic conditions respectively on or above 3, which indicates that the prevailing economic condition of the period to be great
- From the distribution of the economic rankings, it can be inferred that those who ranked both the national and household conditions high, i.e. 4 or 5 are more likely to vote for Labour
- And those ranked the national and household economic condition to be lower than 3 are more likely to vote for Conservatives

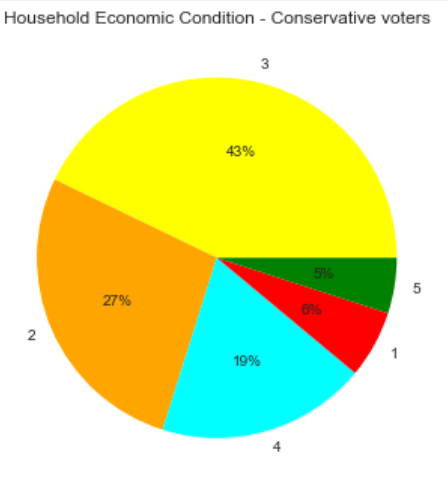








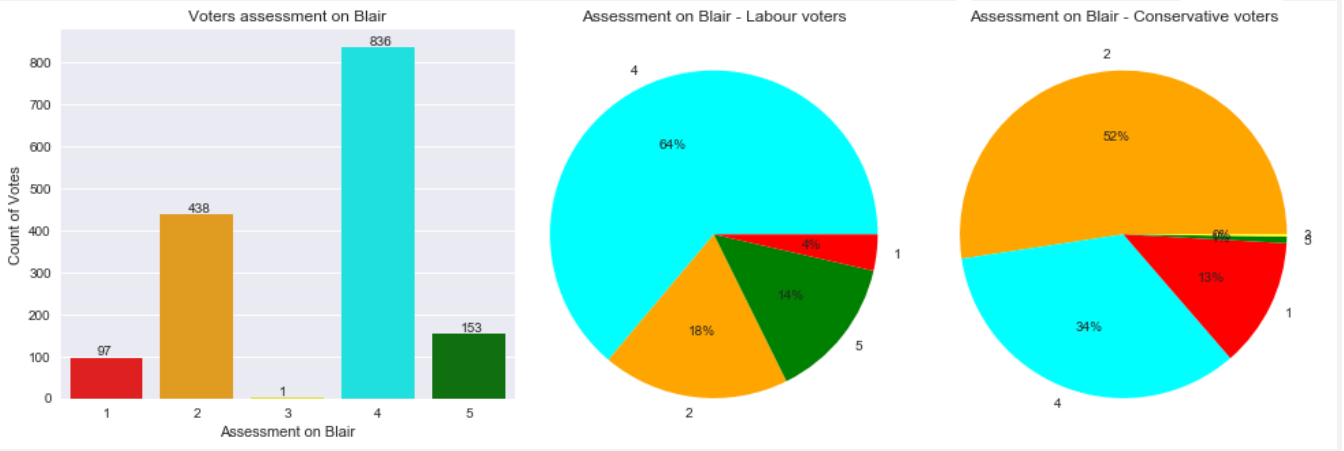


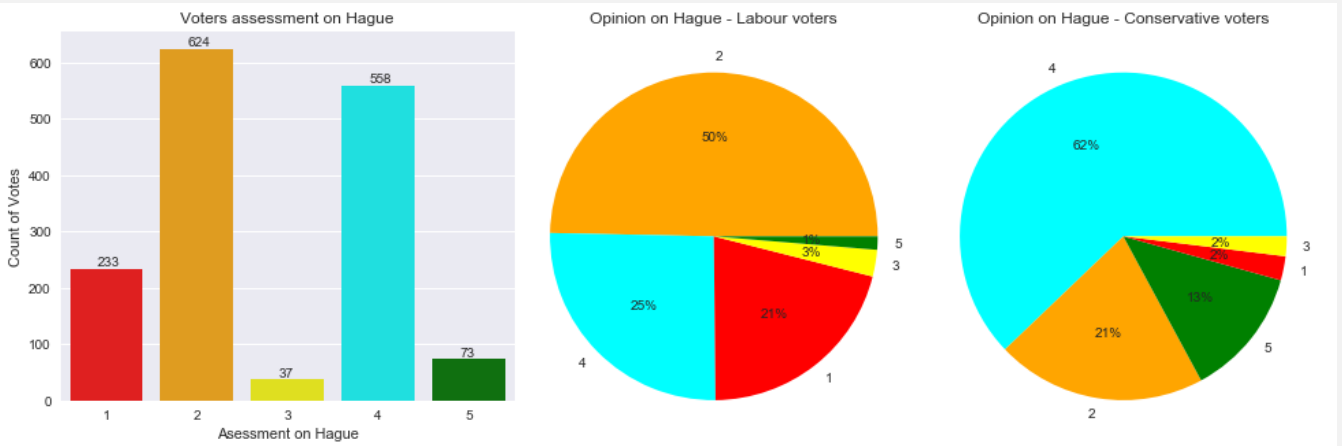


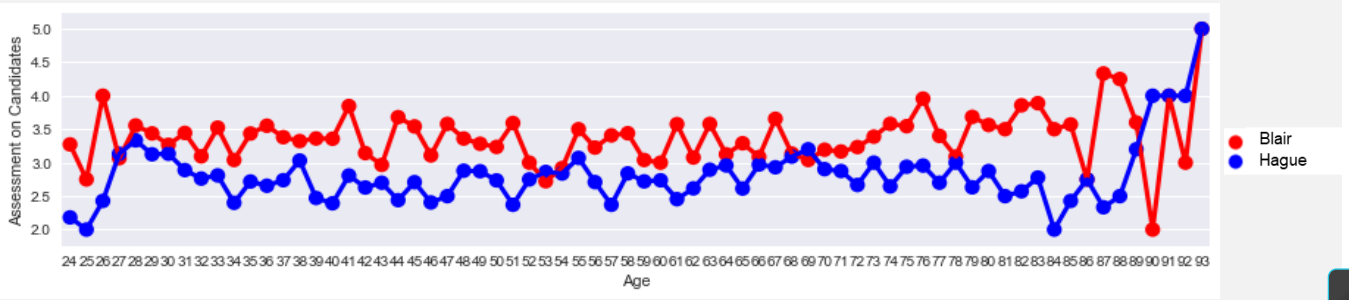
1.2 Exploratory Data Analysis

Tony Blair Vs William Hague

- The features 'Blair' and 'Hague' represent voters assessment on the Labour and Conservative party leaders and prime-ministerial candidates of 2011 UK general elections, Mr Tony Blair and Mr William Hague respectively
- From the data it can be inferred that irrespective of party affiliations, 65% of voters have rated Mr Blair above ranking of 3
- 40% of the Conservative voters and 78% of Labour voters rated Mr Blair above 3
- 65% of Conservative voters and 22% of Labour voters rated Mr Blair's performance as below average
- 55% of all the voters have assessed Mr Hague below 3, which indicates that a majority of voters didn't approve his performance as Conservative leader
- While 71% of Labour voters rated Mr Hague below 3, only 28% rated at 3 or above
- 75% of Conservative voters has assessed Mr Hague above 3, 25% assessed 3 and below
- From plotting the assessment on party leaders against age distribution of voters, one can infer that except among the elderly voters above 87 years, Mr Blair is rated consistently above Mr Hague in all age segments
- Mr Blair is rated high among youth and those between 75 and 85 years
- Mr Hague is rated lowest among 40 and 50 years and 80 to 85 years



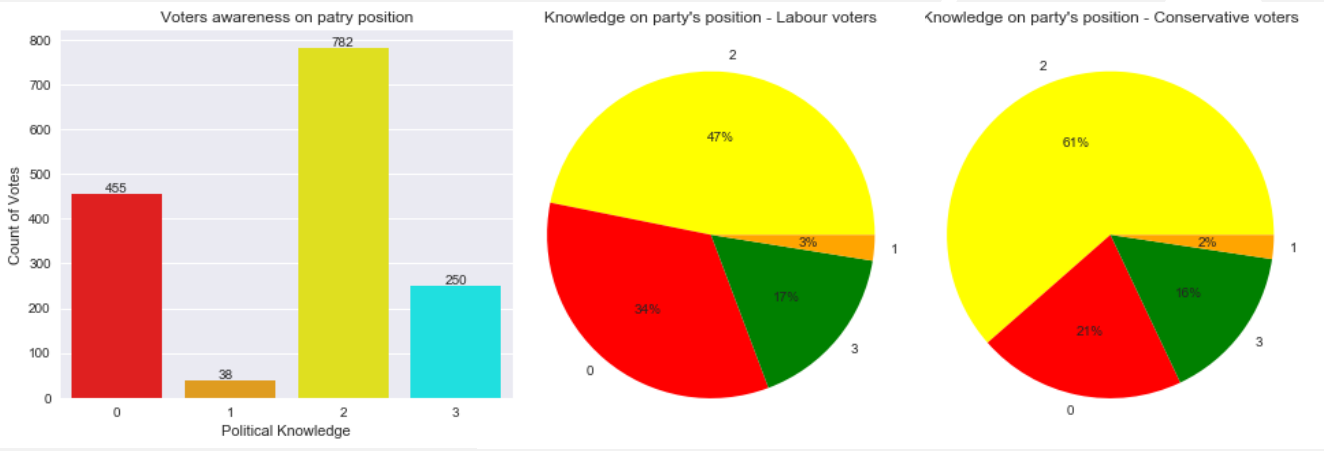


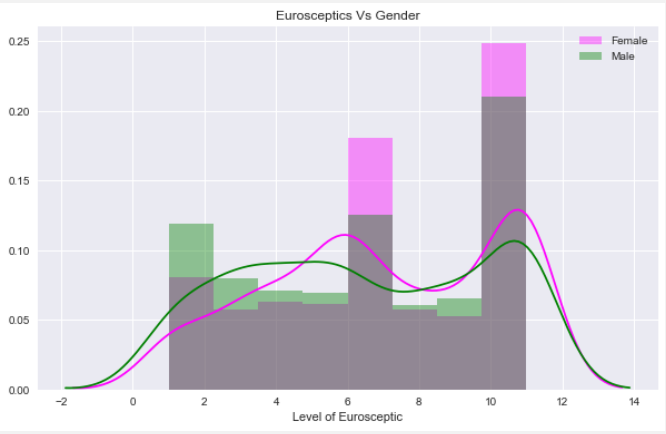


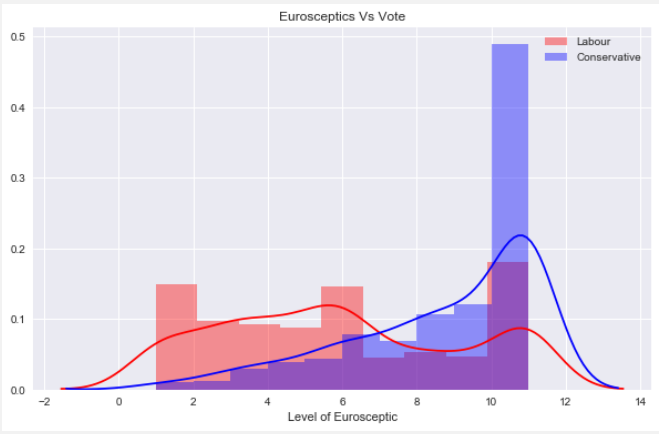
1.2 Exploratory Data Analysis

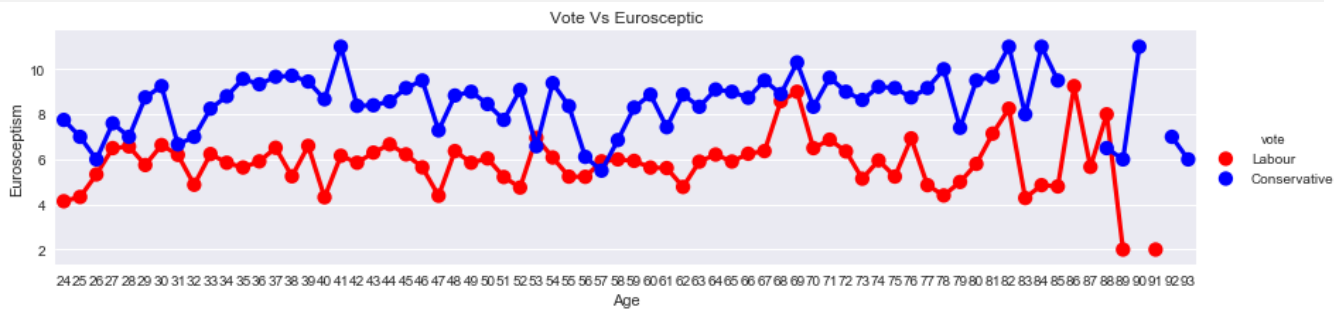
European Integration

- The increasing influence of European Union and UK's integration with it had been the most controversial issue in the 2011 general election
- The exit poll had recorded the voters knowledge on the political parties position on European Integration. 67% of all the voters have expressed their awareness on the issue on or above a rank of 2 on a scale of 0 to 3
- Only 32% of voters consider themselves unaware of the position of Labours and Conservative on European Integration
- While 68% of the Labour voters were confident on their awareness, 78% of Conservative voters rated themselves high
- The factor 'Europe' gauged the level of "Euroscepticism" of the voter on a scale of 0 to 11, where 11 indicates the person is highly sceptic on increasing influence of European Union
- The Conservative voters appears to be highly Eurosceptic than the Labour ones. While the average level of a Conservative voter is above 8, the Labour voter appears to be a little below 6
- The female Labour voter are observed to be more Eurosceptic than the male voters, whereas both male and female Conservative voters share same level of Euroscepticism
- Highly Eurosceptic voters are more likely to vote for Conservative, whereas a Labour voter is more likely to be less Eurosceptic
- The younger lot of the voters appear to be less concerned about the issue than the older ones, especially among the Conservative voters
- The Labour voters do not consider European Integration and increasing influence of EU as an election issue, whereas for a Conservative voter is a major consideration







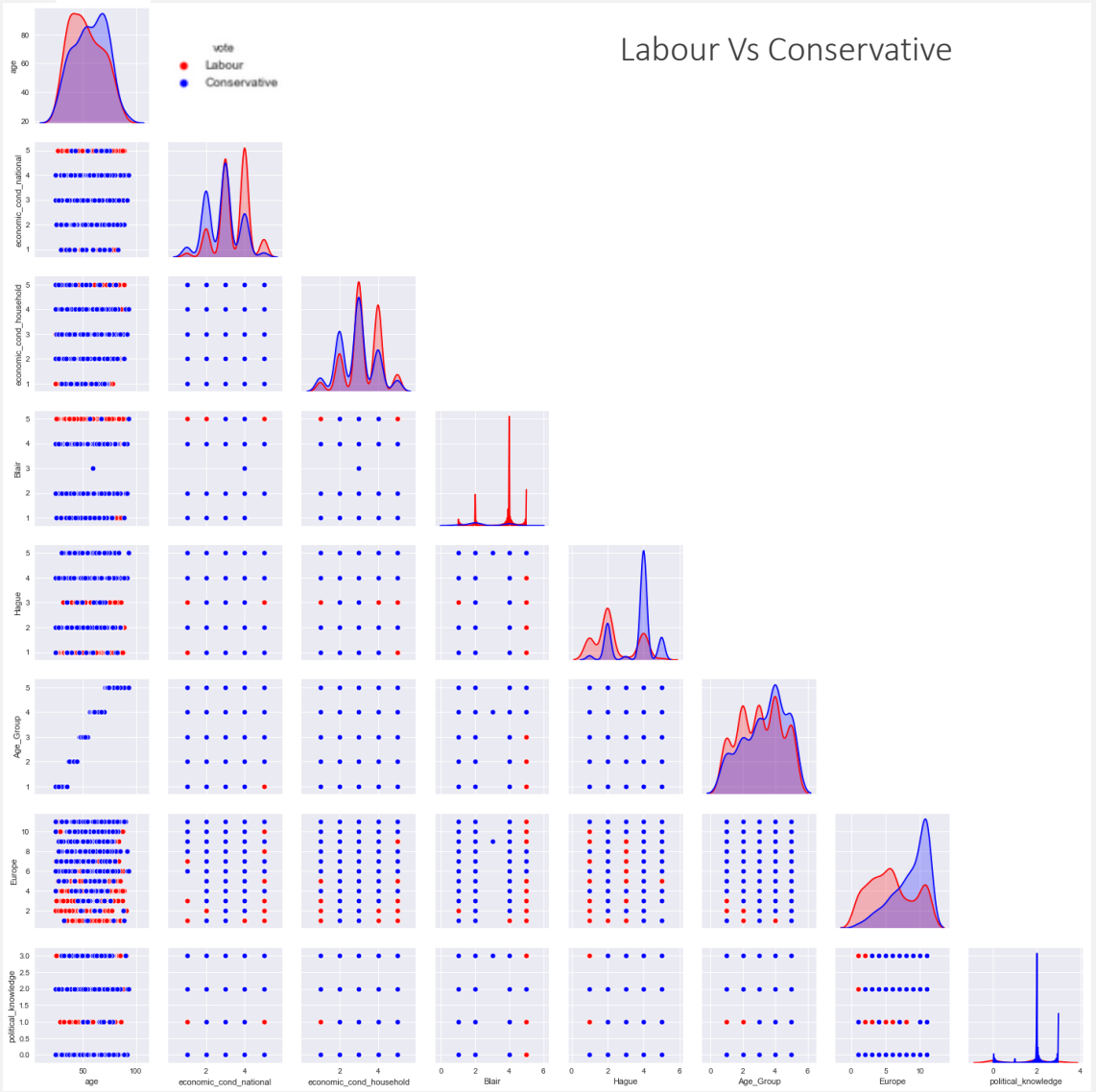


1.2 Exploratory Data Analysis

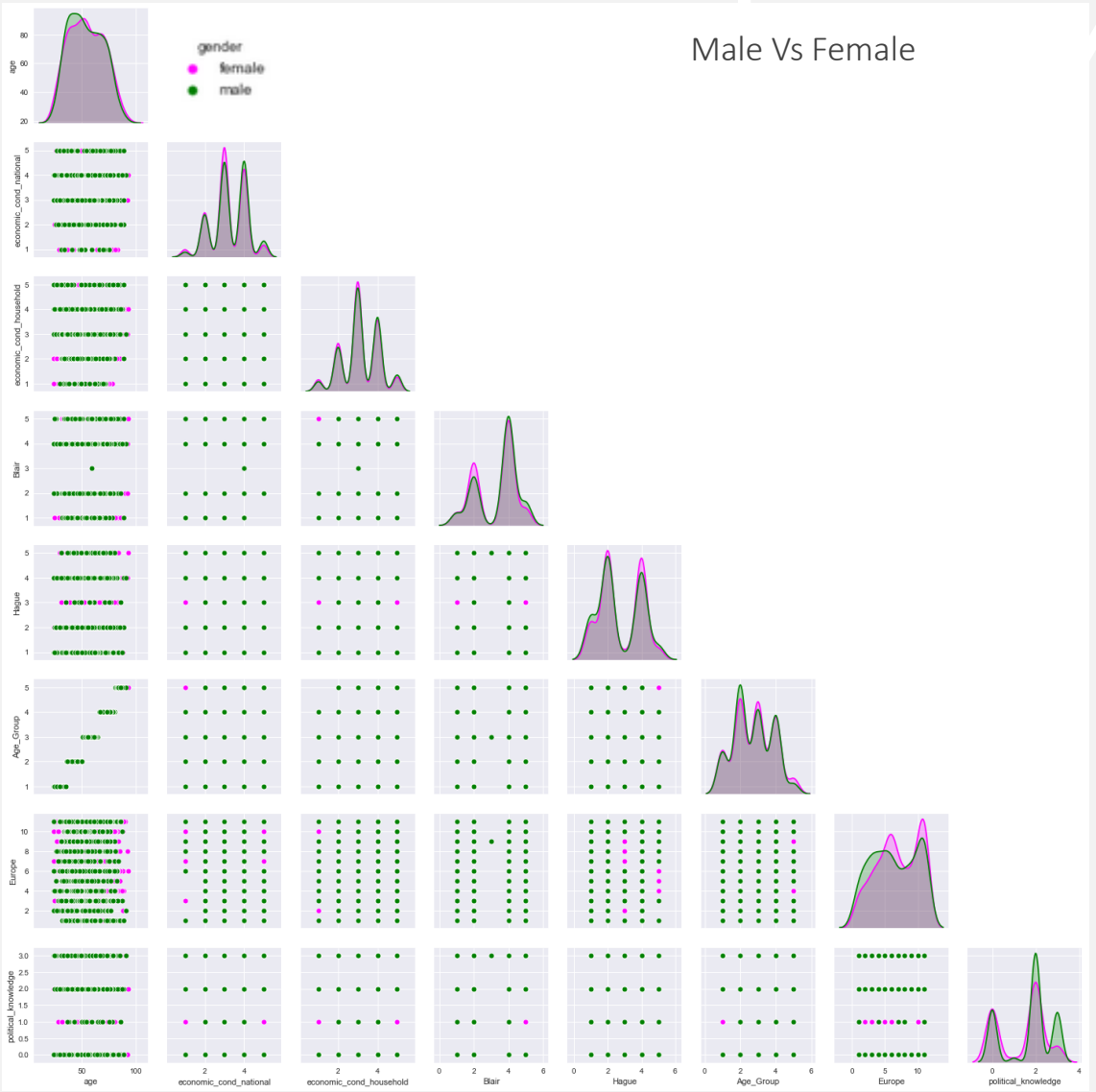
Pair plots

Note: Correlation heat-map is not shown as there is no significant correlation between features to be shown

Labour Vs Conservative

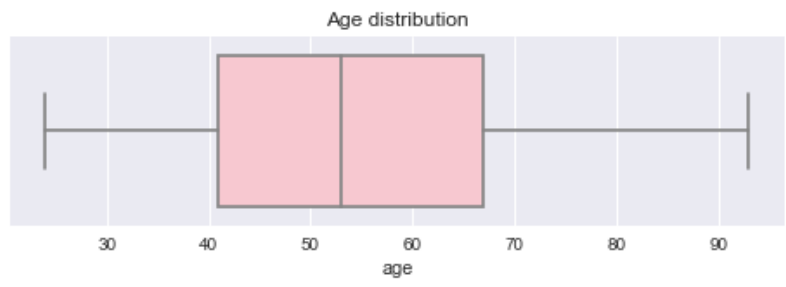


Male Vs Female

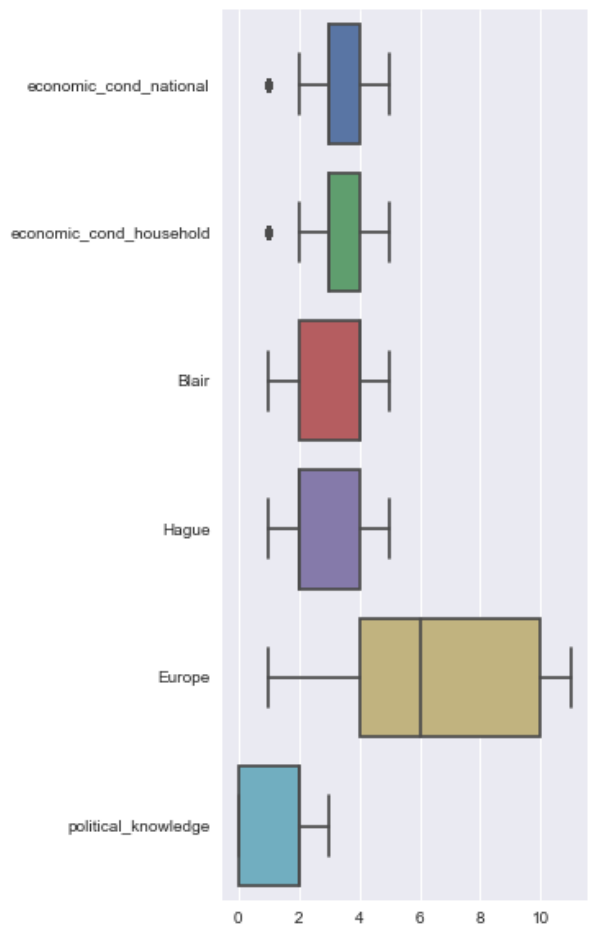


1.2 Exploratory Data Analysis

Summary of EDA



- The age distribution of voters follows a normal distribution with mean of 54 years and median of 53 years. There are no outliers
- A voter between 30 and 50 years is more likely to vote for Labour, whereas a voter between 65 and 75 years is more likely to be a conservative voter
- A voter below 30 is not as decisive as other age groups as they are more unlikely to vote in the election than those above
- Those rated national and household economic condition below 2 are an exception, thus outliers
- A voter who rated the economic conditions to be lower is more likely to vote for Conservative, where as those rated high may vote for Labour
- Mr Blair got higher acceptance among both Labour and Conservative voters, whereas Mr Hague got poorly rated among Labour
- Mr Hague got higher acceptance from the Conservatives
- A highly Eurosceptic voter is highly probable to vote for Conservatives, whereas a lesser Eurosceptic voter is more likely to vote for Labour
- For a Labour voter the flourishing economic condition and acceptance of Mr Blair as Prime Minister is observed to be major a motivation to vote for Labour
- Whereas concerns on EU's increasing influence appears to be the major consideration for a Conservative voter



1.3 Data pre-processing

Binning

- The only numerical variable in the data set 'age' is binned to create the categoric variable 'age_group' to classify the voters into 5 different groups in increasing order of their age
- The variable age is dropped after binning the values into a categoric variable, as keeping both the variables will bring collinearity and redundancy between these variables

Data Encoding

- One hot encoding method is used to encode the categoric variables in the dataset in order to improve the explainablity of the models using feature importance and coefficients
- Except 'gender', rest of the categoric variables are encoded with *drop_first = False* to retain all the categories in the variables including the first category, so that all categories are explainable in the feature set
- The final dataset including the target class variable for 42 features

Scaling

- Scaling is not required in this business case as there are no numerical variables in different scales present in the dataset after dropping 'age' from the dataset after binning and one-hot encoding
- Binning and one-hot encoding has ensured that all factors are on uniform scale and all models are trained using the same data

Data Splitting

- Data has been split into Train an Test sets on 70:30 ratio, with *random_state=27*

Binning of age factor

Order	Age group
1	Up to 35 years
2	36 to 50 years
3	51 to 65 years
4	66 to 80 years
5	Above 80

```
category = pd.cut(df.age,
                  bins=[0,35,50,65,80,99],
                  labels=[1,2,3,4,5])
df.insert(6,'Age_Group',category)
df['Age_Group'].value_counts()
```

```
2    479
3    416
4    367
1    201
5     62
```

Final dataset for predictive modelling

```
RangeIndex: 1525 entries, 0 to 1524
Data columns (total 42 columns):
vote                          1525 non-null int8
Age_Group_1                   1525 non-null uint8
Age_Group_2                   1525 non-null uint8
Age_Group_3                   1525 non-null uint8
Age_Group_4                   1525 non-null uint8
Age_Group_5                   1525 non-null uint8
economic_cond_national_1      1525 non-null uint8
economic_cond_national_2      1525 non-null uint8
economic_cond_national_3      1525 non-null uint8
economic_cond_national_4      1525 non-null uint8
economic_cond_national_5      1525 non-null uint8
Europe_1                      1525 non-null uint8
Europe_2                      1525 non-null uint8
Europe_3                      1525 non-null uint8
Europe_4                      1525 non-null uint8
Europe_5                      1525 non-null uint8
Europe_6                      1525 non-null uint8
Europe_7                      1525 non-null uint8
Europe_8                      1525 non-null uint8
Europe_9                      1525 non-null uint8
Europe_10                     1525 non-null uint8
Europe_11                     1525 non-null uint8
economic_cond_household_1     1525 non-null uint8
economic_cond_household_2     1525 non-null uint8
economic_cond_household_3     1525 non-null uint8
economic_cond_household_4     1525 non-null uint8
economic_cond_household_5     1525 non-null uint8
Hague_1                       1525 non-null uint8
Hague_2                       1525 non-null uint8
Hague_3                       1525 non-null uint8
Hague_4                       1525 non-null uint8
Hague_5                       1525 non-null uint8
Blair_1                       1525 non-null uint8
Blair_2                       1525 non-null uint8
Blair_3                       1525 non-null uint8
Blair_4                       1525 non-null uint8
Blair_5                       1525 non-null uint8
political_knowledge_0         1525 non-null uint8
political_knowledge_1         1525 non-null uint8
political_knowledge_2         1525 non-null uint8
political_knowledge_3         1525 non-null uint8
gender_male                   1525 non-null uint8
dtypes: int8(1), uint8(41)
```

1.4 Logistic Regression & Linear Discriminant Analysis

Logistic Regression

```
class_weight = dict({0:2, 1:1})
clf = LogisticRegression(class_weight = class_weight)
grid_values = {'penalty': [ 'l2', 'none']
               ,'C':np.logspace(-3,3,20)
               ,'solver' : ['newton-cg', 'lbfgs', 'sag' ,'saga' ]
              }
logit_model3 = GridSearchCV(clf, param_grid = grid_values,cv = 10)
logit_model3.fit(X_train, y_train)
```

```
1  logit_model3.best_params_
```

```
{'C': 1.438449888287663, 'penalty': 'l2', 'solver': 'newton-cg'}
```

- The final model selected after all the iterations of model tuning is as follows
- The minority class (Conservative) and the majority class (Labour) is on a ratio of 1:2.3, which is used to set the *class_weight* parameter in Logistic Regression
- The function *logspace()* is used to find the regularisation parameter '*C*' which returned a value on log scale
- The *GridsearchCV()* returned hyperparameters for '*penalty*' and '*solver*' algorithm, *L2* and *newton-cg* respectively
- The proportional values to be applied for class weight parameter is finalised after trial and error iterations, where the ideal performance matrices were evaluated for both minority and majority classes
- The feature coefficients are plotted to generate insights based on feature ranking
- The resulting model is a right fit one with acceptable score of recall of minority class

Linear Discriminant Analysis

```
clf = LinearDiscriminantAnalysis()
grid_values = {'solver': ['svd', 'lsqr', 'eigen'],
               'tol': [0.0001, 0.001, 0.01, 0.1]
              }
LDA_model3 = GridSearchCV(clf, param_grid = grid_values, cv = 10)
LDA_model3.fit(X_train, y_train)
```

```
1  LDA_model3.best_params_
```

```
{'solver': 'svd', 'tol': 0.0001}
```

- *GridsearchCV()* was applied on the LDA model to identify the right *solver* algorithm and *tolerance* value
- The default algorithm *SVD* and the default *tolerance* value was returned as the ideal hyperparameter
- Thus both the first iteration and the final iteration returned same set of performance matrices
- The model scores are found to be right fit, but the recall of minority class is lower

Note: Refer slide 13 (Model tuning) for further details on model tuning and overall model selection criteria followed

1.5 K Nearest Neighbors, Naïve Bayes & Support Vector Machine

K Nearest Neighbors Model

```
clf = KNeighborsClassifier()
grid_values = {'n_neighbors': range(5,20),
               'weights' : ['uniform'],  #['uniform','distance'],
               'metric': ['minkowski', 'euclidean', 'canberra']
              }
KNN_model3 = GridSearchCV(clf, param_grid = grid_values, cv = 5)
KNN_model3.fit(X_train, y_train)
```

```
1  KNN_model3.best_params_
```

```
{'metric': 'minkowski', 'n_neighbors': 18, 'weights': 'uniform'}
```

- For tuning the hyperparameters of the KNN model *GridsearchCV()* was applied to find ideal values for *n_neighbours*, *weights* and *matric* parameters
- Though '*distance*' was found to be the ideal value for the *weight* function, the model found to be highly overfitting with the model capturing almost 100% of the variance in train
- Hence '*uniform*' weight is used to create a right fit model and default *'minkowski'* metric is used for distance measure. Only *n_neighbors* value is optimised using *GridsearchCV()*
- The model was found to be right fit, but the recall of minority class was lowest

Multinomial Naïve Bayes Model

```
from sklearn.naive_bayes import MultinomialNB
MNB_model = MultinomialNB()
MNB_model.fit(X_train, y_train)
```

- For building a Naïve Bayes model, Multinomial Naïve Bayes algorithm is used. The reason being that the dataset contain only nominal factors and it meets the basic assumption of Naïve Bayes that all the factors are non-colinear
- Neither *GridsearchCV()* nor any further tuning was applied on the model as the function does not have too many hyperparameters to be optimised
- The resulting model was found to be right fit, with acceptable recall rate of both classes

Support Vector Machine Model

```
class_weight = dict({0:2.3, 1:1})
clf = svm.SVC(probability=True, class_weight = class_weight)
grid_values = {'C': np.logspace(-1,1,20),
               'kernel': ['linear', 'poly', 'rbf', 'sigmoid']
               }
SVM_model3 = GridSearchCV(clf, param_grid = grid_values, cv = 3)
SVM_model3.fit(X_train, y_train)
```

```
1  SVM_model3.best_params_
```

```
{'C': 0.20691380811147897, 'kernel': 'linear'}
```

- SVM which is a kernel based model allow us to specify class weightage using the hyperparameter *class_weight*. The weightage for the minority class was applied after trial and error process
- Logspace() is also used to find the optimal value for the regularisation parameter 'C', which returned a optimal value in logscale
- *'linear'* was identified as the optimal kernel algorithm
- Though the model returned same level of accuracy, the recall od the minority class was significantly improved

Note: Refer slide 13 (Model tuning) for further details on model tuning and overall model selection criteria followed

1.6 Model Tuning

- Unlike usual business cases where there is a positive and negative class outcomes, in this case the classifications are neither positive nor negative and both holds equal importance in terms of the predictability of the model
- **Class imbalance:** The majority class (Labour) is 70% and the minority class (Conservative) is 30% of the total target class (ratio of 2.3:1). The machine learning classification algorithms tend towards the majority class in prediction resulting in what is known as accuracy paradox
- **Accuracy paradox** is when the model gives a very high accuracy score but the underlying class distribution is skewed favouring a higher recall and precision of the majority class
- **The approach** taken for model tuning and selection is to first balance the precision and recall of the minority class with that of the majority class in train and test datasets before accuracy and area under curve of the model is assessed
- The model tuning measures applied are as follows:

1. SMOTE
    - Synthetic minority oversampling technique was applied on the train dataset in second iteration to assess if the performance of the models will improve.
    - However it was found that the model performance remained same or got depreciated in all models (accuracy and recall of minority class in test)

Model	Before SMOTE		After SMOTE	
Logit	.82	.68	.82	.71
LDA	.82	.71	.70	0.00
KNN	.80	.63	.81	.65
SVM	.82	.67	.70	0.00
MNB	.83	.74	.75	.45

Accuracy

Recall

2. Class weights
    - The algorithms such as Linear Regression, Support Vector Machine and Random Forest (for Bagging) allow us to set weightage on the classes such that the performance matrices of the minority class can be improved
    - A dictionary of class weightage was initially set based on the original class ratio (1:2.3) and later tweaked iteratively by validating the performance metrices and not letting the model overfit on train
3. GridsearchCV
    - Except for the Multinomial Naïve Bayes and Bagging classifier, GridsearchCV was applied to optimise the hyperparameters of all the models
    - Logspace() was used in Logistic regression and SVM models to find the optimal value of the regularisation parameter 'C' in log scale
    - The model accuracy and recall of minority class after applying class_weight and GridsearchCV is as below

Model	Before tuning		After tuning	
Logit	.82	.68	.83	.79
KNN	.80	.63	.82	.66
SVM	.82	.67	.80	.85

Accuracy

Recall

4. Cross validation
    - 10 fold cross validation of all the final selected models were applied to find out the mean accuracy score and standard deviation of the
    - Mean accuracy score and standard deviation of the scores will be used in final model selection

1.6 Bagging & Boosting

Bagging with Random Forest classifier

```
from sklearn.ensemble import BaggingClassifier
from sklearn.ensemble import RandomForestClassifier
class_weight = dict({0:4, 1:1.5})
rfcl = RandomForestClassifier(class_weight=class_weight,
                              min_samples_leaf=2,
                              min_samples_split=4)
Bagging_model=BaggingClassifier(base_estimator=rfcl,n_estimators=50,random_state=1)
Bagging_model.fit(X_train, y_train)
```

- The ratio of the target classes (1:2.3) was used and later tuned iteratively to set the *class_weight* parameter of the Random Forest Classifier. The ideal weightage for classes 0 and 1 was finally set as 4 and 1.5 respectively
- The values for *min_samples_leaf* and *min_samples_split* was also tuned iteratively based on the accuracy score and recall of classes
- The Rndom Forest model was used as *base_estimator* for the Bagging Classifier and *n_estimator* was optimised so that model wont overfit
- Without the *class_weight* values the model was initially found to be highly overfitting and the recall of the minority class was found to be too low
- The final Bagging classifier model was found to be slightly overfit but acceptable as it returned equally high recall rate for both majority and minority classes

Boosting (XGBoost)

```
import xgboost as xgb
XGB_model = xgb.XGBClassifier(max_depth = 5,
                              min_child_weight = 3,
                              learning_rate =0.01,
                              n_estimators=1000)
XGB_model.fit(X_train, y_train)
```

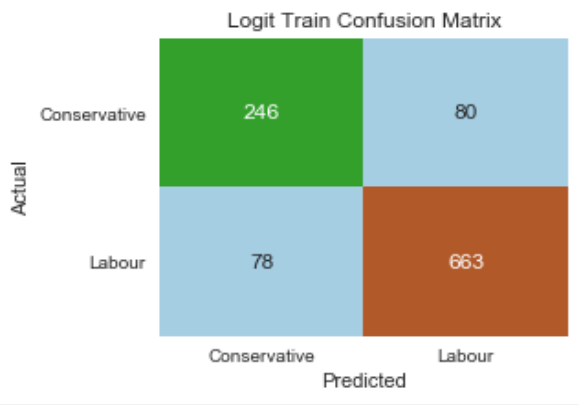
- Extreme Gradient Boosting or known as XGBoost is used to build a Boosting model
- GridsearchCV was initially used to optimize the hyperparameters of the model. But it was found to have created an extremely overfitting model with no significant improvement of performance in test
- The hyperparameters are tuned iteratively so that the model is right-fit and the recall of minority class is acceptable along with the majority class
- The final model accuracy score was found to be slightly overfit but acceptable, but the recall in test was found to be lowest among all models
- XGBoost is observed to be more appropriate for very large datasets than smaller ones as in the given business case

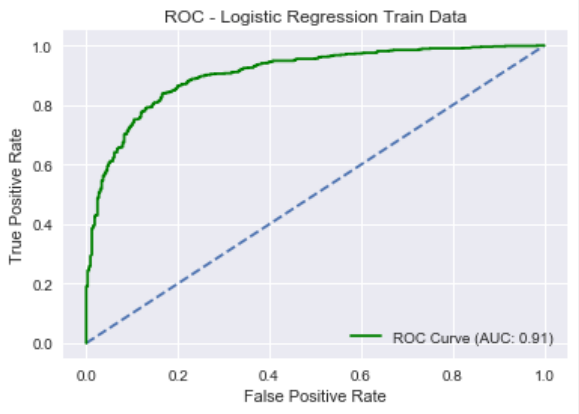
1.7 Performance Metrics

Logistic Regression

Train

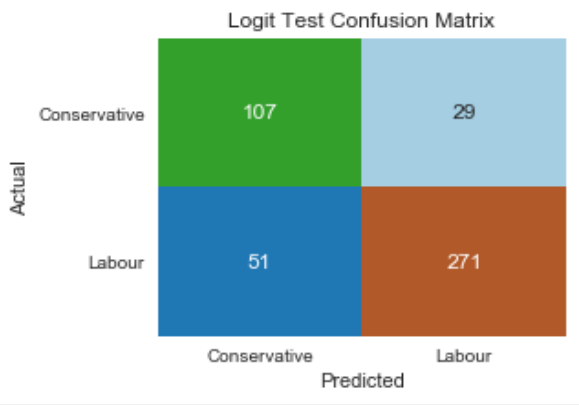
	precision	recall	f1-score	support
0	0.759259	0.754601	0.756923	326.000000
1	0.892328	0.894737	0.893531	741.000000
accuracy	0.851921	0.851921	0.851921	0.851921

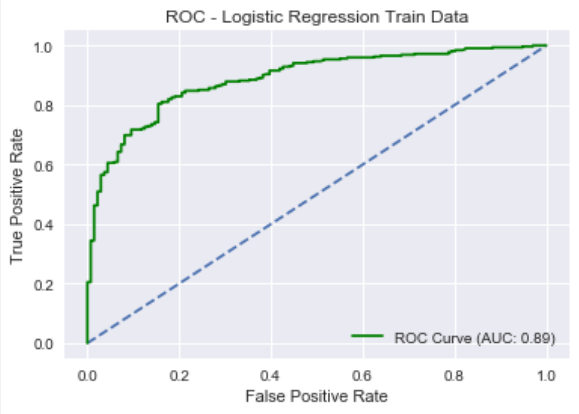




Test

	precision	recall	f1-score	support
0	0.677215	0.786765	0.727891	136.000000
1	0.903333	0.841615	0.871383	322.000000
accuracy	0.825328	0.825328	0.825328	0.825328

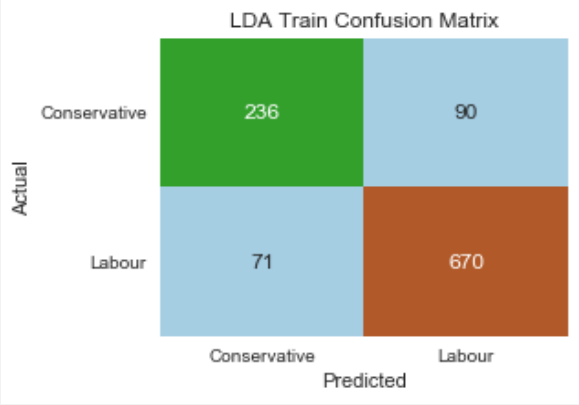


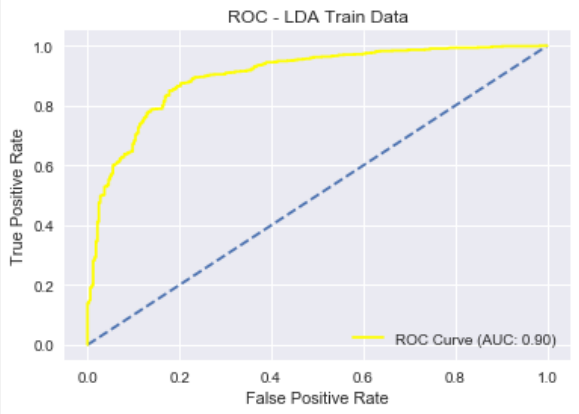


Linear Discriminant Analysis

Train

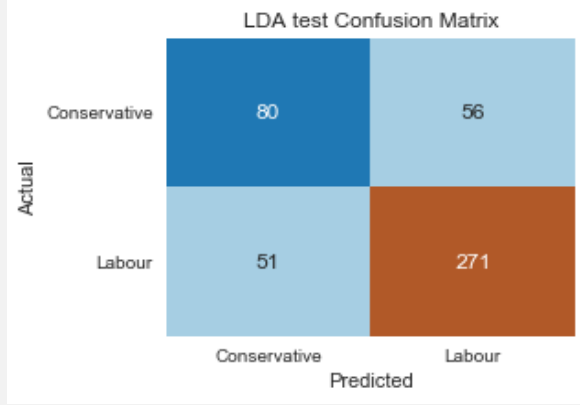
	precision	recall	f1-score	support
0	0.768730	0.723926	0.745656	326.00000
1	0.881579	0.904184	0.892738	741.00000
accuracy	0.849110	0.849110	0.849110	0.84911





Test

	precision	recall	f1-score	support
0	0.610687	0.588235	0.599251	136.000000
1	0.828746	0.841615	0.835131	322.000000
accuracy	0.766376	0.766376	0.766376	0.766376

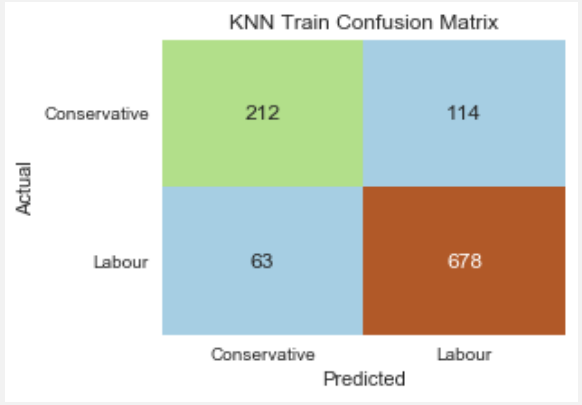


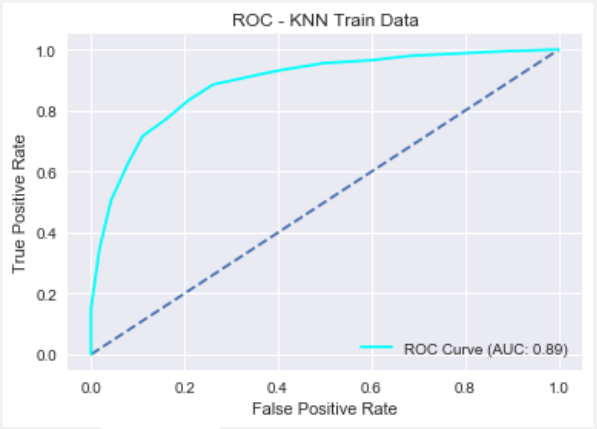
1.7 Performance Metrics

K Nearest Neighbors

Train

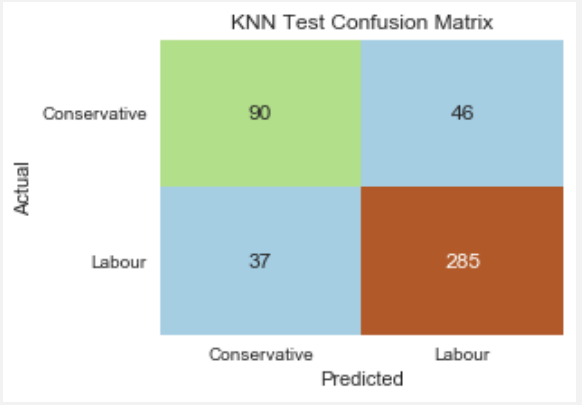
	precision	recall	f1-score	support
0	0.770909	0.650307	0.705491	326.000000
1	0.856061	0.914980	0.884540	741.000000
accuracy	0.834114	0.834114	0.834114	0.834114

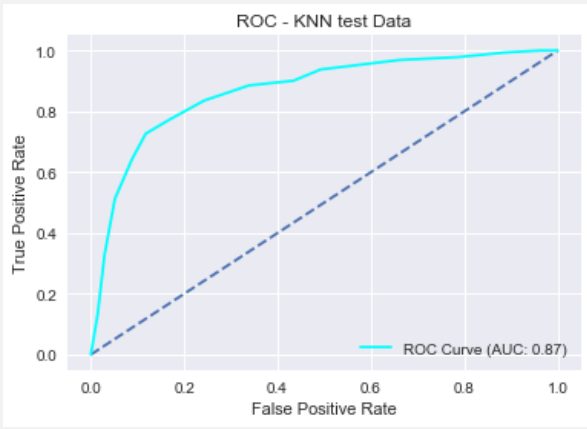




Test

	precision	recall	f1-score	support
0	0.708661	0.661765	0.684411	136.000000
1	0.861027	0.885093	0.872894	322.000000
accuracy	0.818777	0.818777	0.818777	0.818777

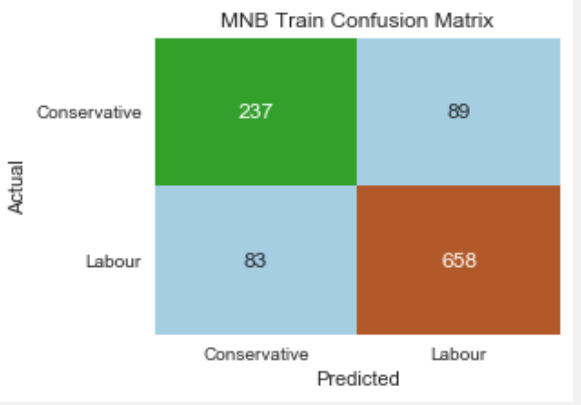


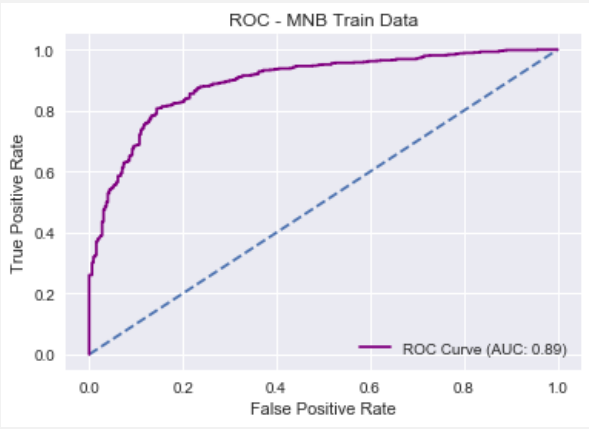


Multinomial Naïve Bayes

Train

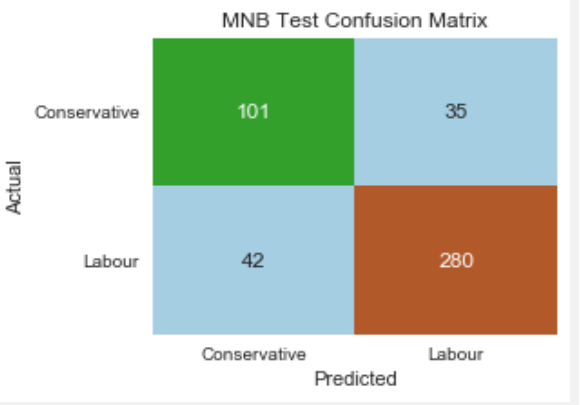
	precision	recall	f1-score	support
0	0.740625	0.726994	0.733746	326.0000
1	0.880857	0.887989	0.884409	741.0000
accuracy	0.838800	0.838800	0.838800	0.8388

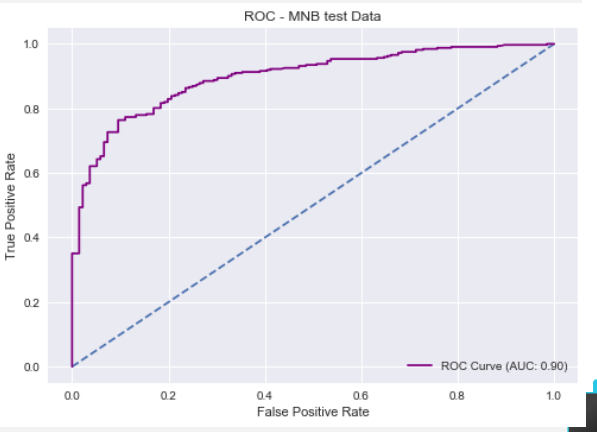




Test

	precision	recall	f1-score	support
0	0.706294	0.742647	0.724014	136.000000
1	0.888889	0.869565	0.879121	322.000000
accuracy	0.831878	0.831878	0.831878	0.831878



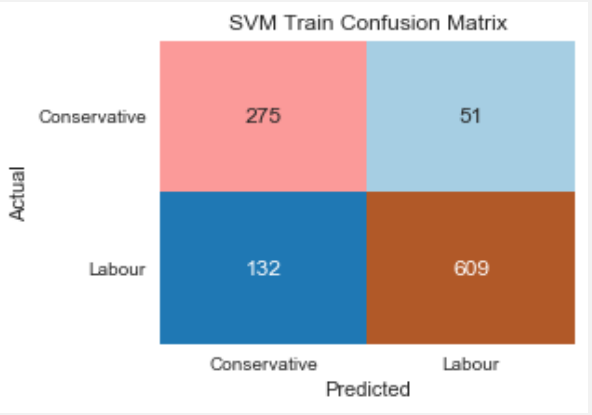


1.7 Performance Metrics

Support Vector Machine

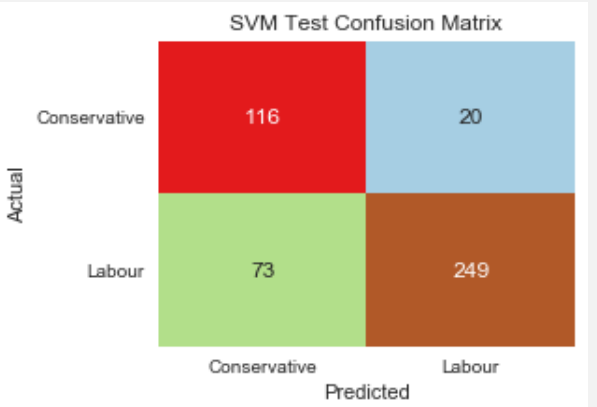
Train

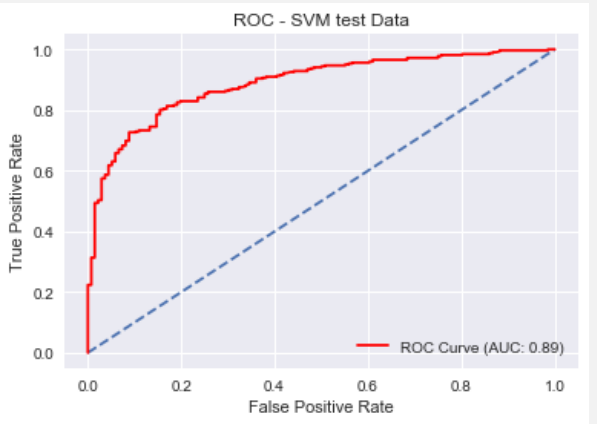
	precision	recall	f1-score	support
0	0.675676	0.843558	0.750341	326.000000
1	0.922727	0.821862	0.869379	741.000000
accuracy	0.828491	0.828491	0.828491	0.828491



Test

	precision	recall	f1-score	support
0	0.613757	0.852941	0.713846	136.000000
1	0.925651	0.773292	0.842640	322.000000
accuracy	0.796943	0.796943	0.796943	0.796943



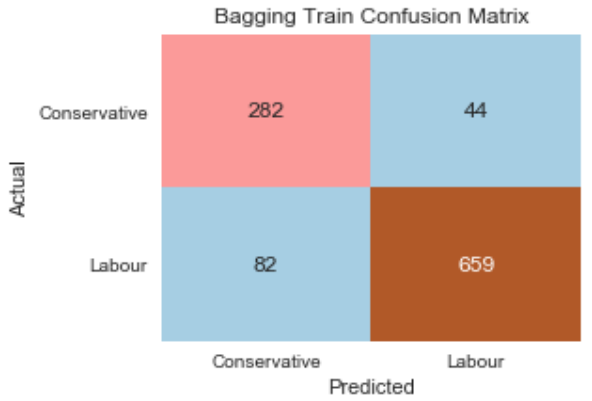


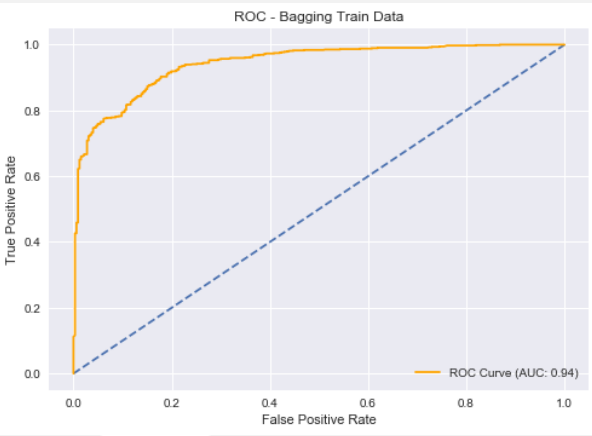
1.7 Performance Metrics

Bagging with Random Forest classifier

Train

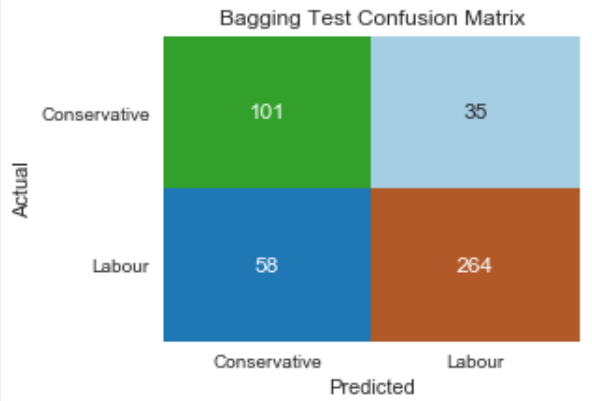
	precision	recall	f1-score	support
0	0.774725	0.865031	0.817391	326.000000
1	0.937411	0.889339	0.912742	741.000000
accuracy	0.881912	0.881912	0.881912	0.881912

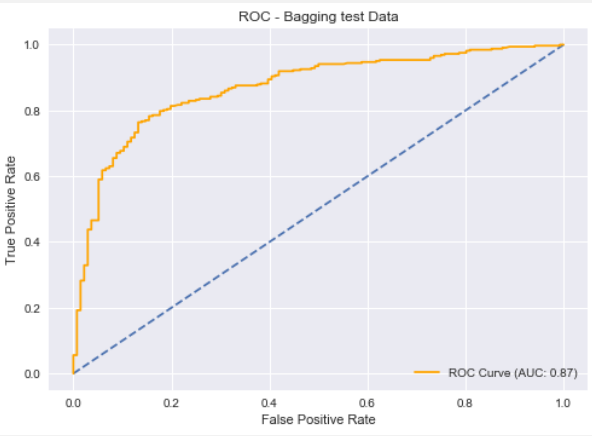




Test

	precision	recall	f1-score	support
0	0.635220	0.742647	0.684746	136.000000
1	0.882943	0.819876	0.850242	322.000000
accuracy	0.796943	0.796943	0.796943	0.796943

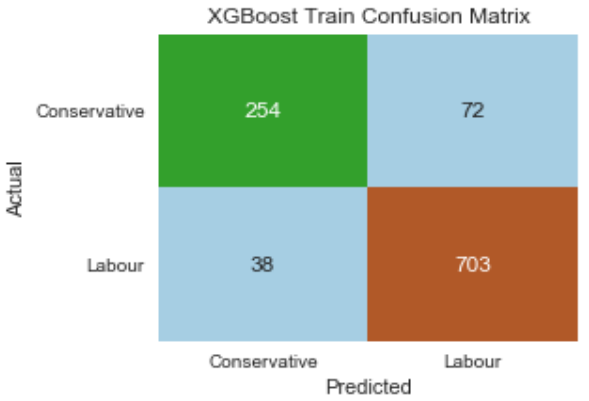


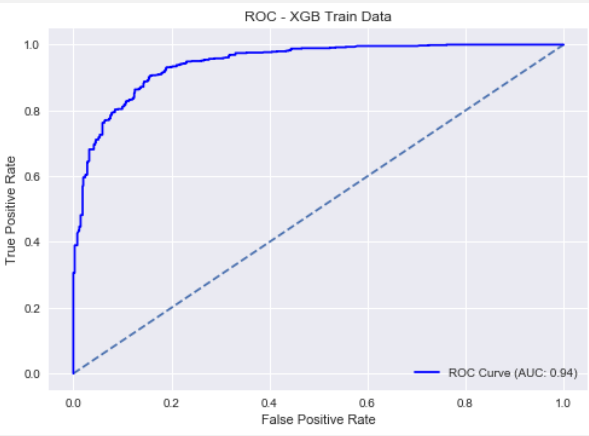


Boosting (XGBoost)

Train

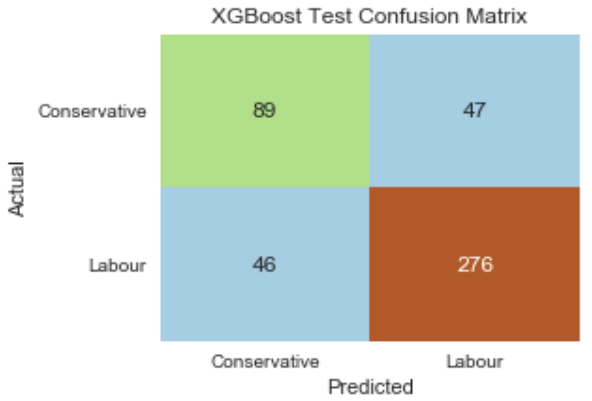
	precision	recall	f1-score	support
0	0.869863	0.779141	0.822006	326.000000
1	0.907097	0.948718	0.927441	741.000000
accuracy	0.896907	0.896907	0.896907	0.896907

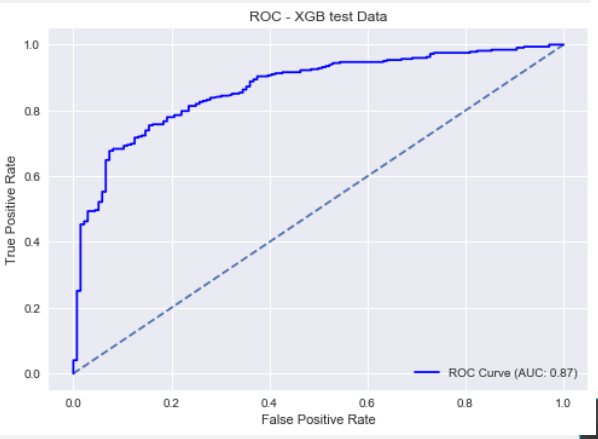




Test

	precision	recall	f1-score	support
0	0.659259	0.654412	0.656827	136.000000
1	0.854489	0.857143	0.855814	322.000000
accuracy	0.796943	0.796943	0.796943	0.796943





1.7 Final Model Selection

Accuracy

- While Bagging and XGBoost classifiers delivered the highest accuracy of 89% in train, Logistic Regression and Naïve Bayes gave the highest accuracy of 83% in test
- KNN and SVM is least with 83% in train and in test Bagging is least at 81%

Recall

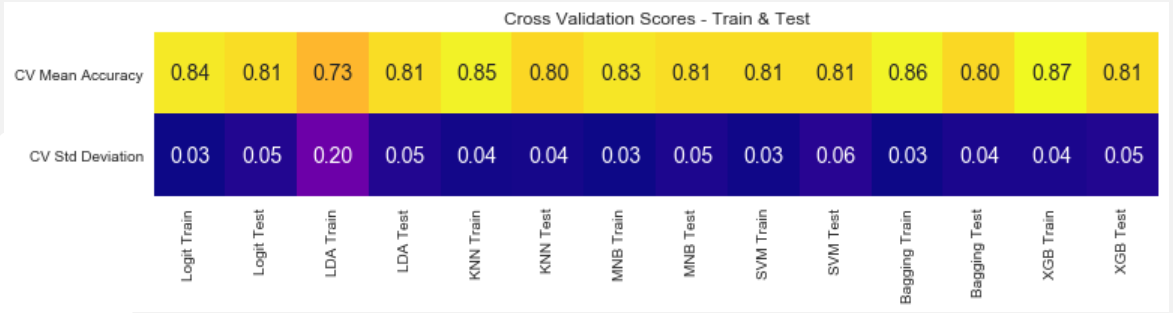
- In train, Bagging has the highest recall score for the minority class at 86%, followed by SVM at 84%. For the majority class, XGBoost got 94% recall, followed by KNN AT 91%
- In test, SVM gave highest recall of minority class at 85%, followed by Bagging and Logit at 79%. On the majority class KNN gave 89% recall, followed by XBG (88%) and MNB (87%)
- SVM, Logit and Bagging gave most balanced recall of both the classes than other models

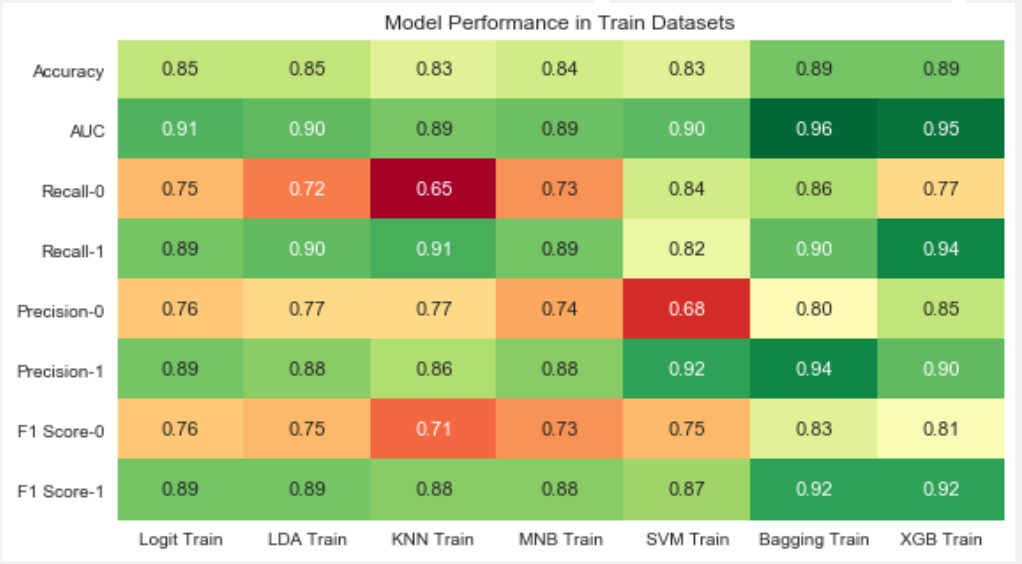
F1 Score

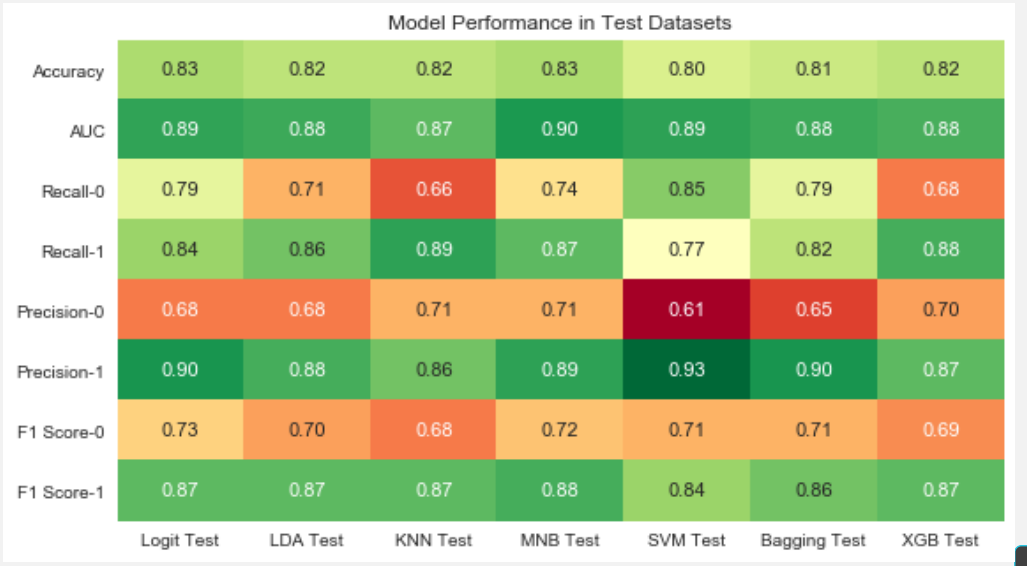
- The harmonic mean of recall and precision, in train is high for Bagging and XGBoost at 83% and 81% for minority class and 92% for majority class
- In test, Logit is highest for minority class at 73% and 87% by Logit, LDA, KNN and XGBoost for majority class

Cross Validation

- The 10 fold cross validation shows that the highest mean accuracy in train for XGBoost (87%) and lowest for LDA (73%)
- In test, except Bagging (80%) all other models gave a mean accuracy of 81%
- LDA model got highest inconsistency with 20% std deviation. Logit, MNB, SVM & Bagging got lowest std dev of 3%
- While in test KNN & XGB got 4% deviation and rest at 5% and 6% (SVM)







1.7 Final Model Selection

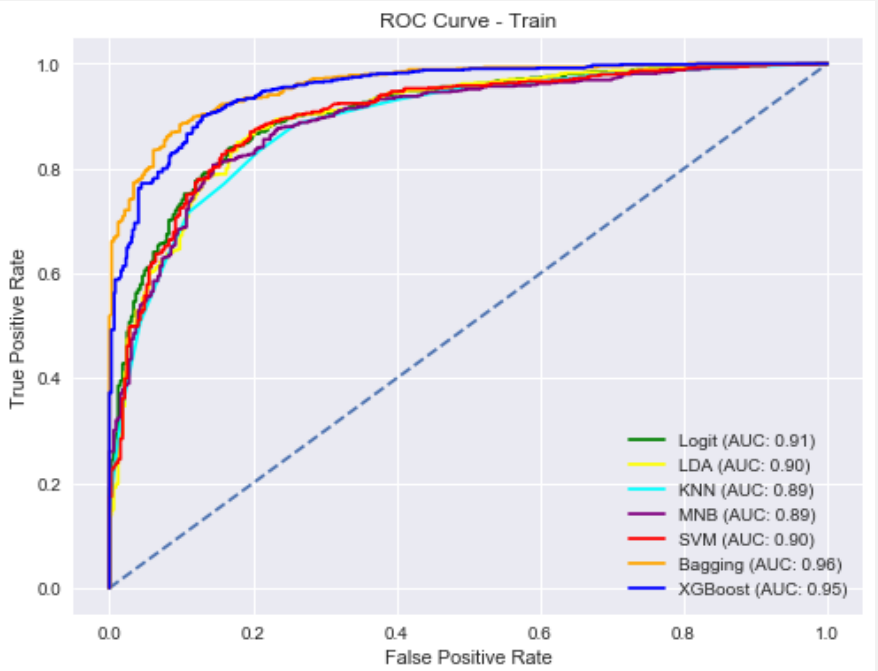
AUC score

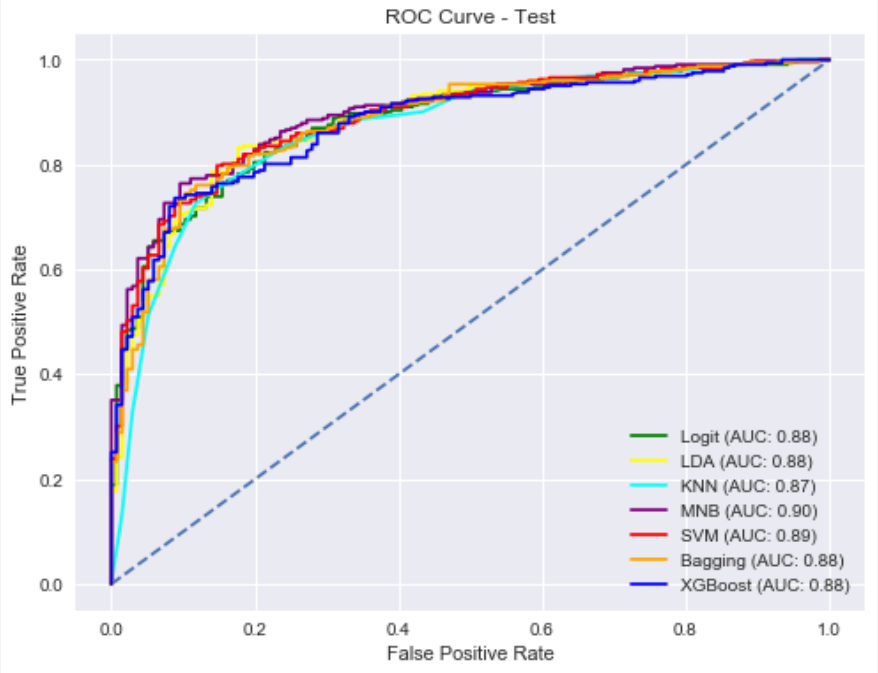
- Bagging and XGBoost gave the highest AUC score on train, 96% and 95% respectively, but on test they gave 88% each
- On test, Naïve Bayes delivered the highest AUC score at 90%, followed by SVM and Logit at 89%
- All the models have consistently given an AUC score above 89% in train and above 88% in test, except KNN. KNN gave the least AUC score in with 87%

Conclusion

- Bagging and XGBoost models tend towards defining high variance in train, but fail to perform in test, creating overfit models
- Gradient Boosting models like XGBoost would perform extremely well with large datasets than the smaller ones as given in this case
- Bootstrap aggregation (Bagging) require higher computational resources and may pose low performance challenges
- Whereas parametric models like Naïve Bayes and Logit showed better consistency in class predictions and computational performance
- LDA showed least consistency in 10 fold cross validation and KNN gave the least recall of minority class in train & test
- Considering the robustness of model on cross validation, recall across both the target classes and consistent accuracy on both train and test datasets, Logistic Regression, SVM and MNB could be chosen as the final model
- MNB has delivered highest AUC score in test and right-fit in train and test without any complex model tuning required. The recall rate of the target classes is also on par with Logit
- SVM model has delivered right-fit in train and test with high recall score on minority class, but at the cost of low recall of majority class leading to least of precision among these models
- Logit model has produced very consistent and balanced performance in terms of minority and majority class prediction and overall accuracy in train and test
- Multinomial Naïve Bayes is recommended as the final model considering the low size of the given dataset and the model being fast and highly scalable

Final model: Multinomial Naïve Bayes

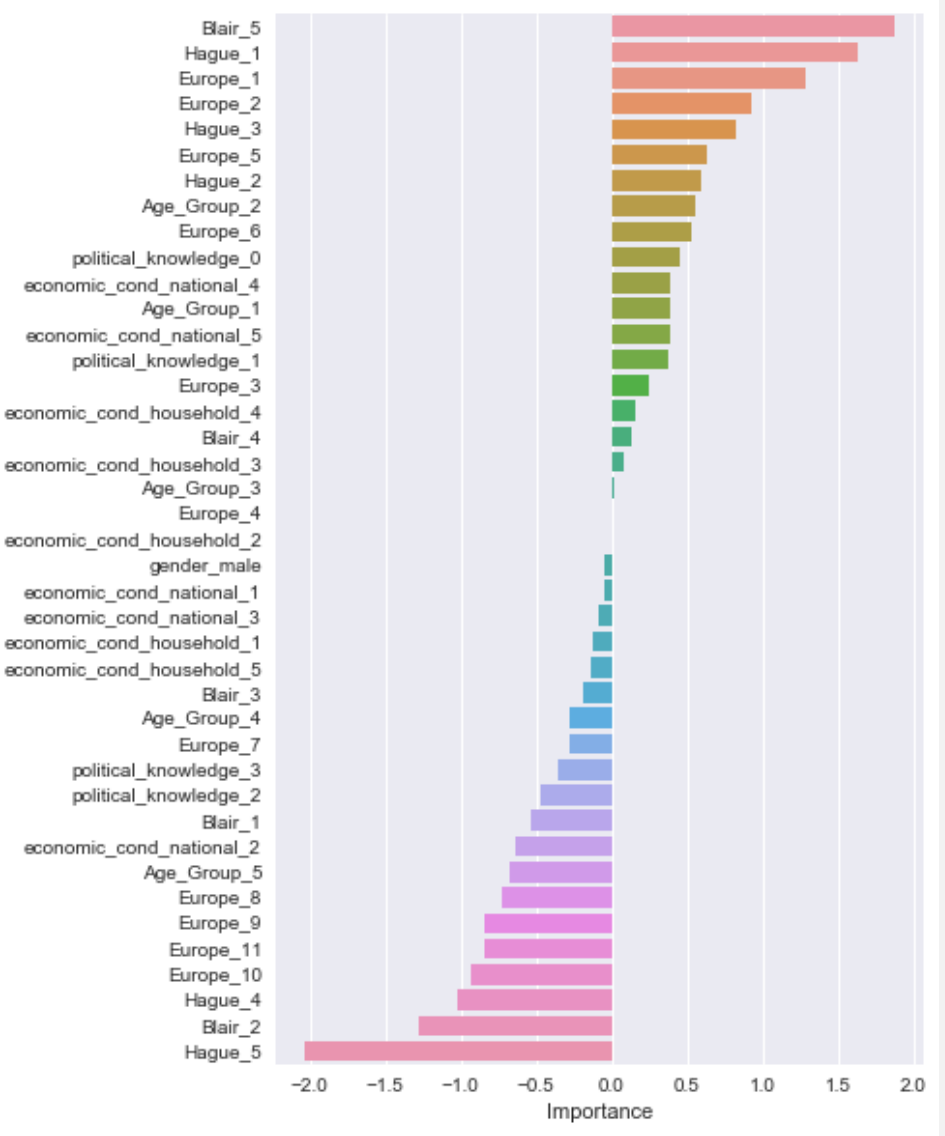




1.8 Insights

- The models applied in the exercise such as Logistic Regression and Gradient Boosting allow us to rank features based on feature coefficient and feature importance values of the respective models
- Here, feature coefficients from the Logistic Regression is used to draw insights on the predictions, by plotting the values as a barplot. Key inferences are as follows.
- The assessment/opinion of voters on the Labour and Conservative leaders, Mr Blair and Mr Hague is the most decisive factor in the poll than any other factors, making the election a mandate on these two political leaders
- A voter who rated Mr Blair as 5 and Mr Hague as 1 is more likely to vote for Labour, whereas the one who rated Hague as 5 and Mr Blair as 2/1 is more likely to vote for Conservative
- The most important political factor of the poll is observed to be European integration of UK, but it is second only to voter's confidence/ opinion on Mr Blair and Mr Hague
- While the Eurosceptic voters chose to vote for Conservative party, the lesser sceptics or those who are not concerned on the increasing influence of EU chose to vote for Labour
- The national and household economic condition prevailed during the period is found to be a non-issue during the poll, which is an indictor of the fledging economic situation of the period
- The age of the voter appears to be a decisive factor in predicting votes, as those from group 2 (36 to 45 years) and lesser (35 and below) appears to have favoured Labour party and those from group 5 and 4 (66 years and above) are more likely to vote for Conservative party
- Those who are confident on their awareness on political parties position on European integration are most likely to be a Conservative voter, where as for a Labour voter is was least of their concerns
- Gender of the voter does not appear to have a significant influence on the outcome of the election
- The feature importance from the XGBoost also returned similar patterns of feature ranking (not shown here) where the rating on Mr Blair and Mr Hague stood out to be the most significant factor of the election mandate
- Using the prediction probability from classification, with 83% confidence it can confirmed that Labour would get 70% votes

Feature Ranking from Logistic Regression



Problem - 2

In this particular project, 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 1963

2.1 Number of characters, words and sentences

Character count (inc. space)

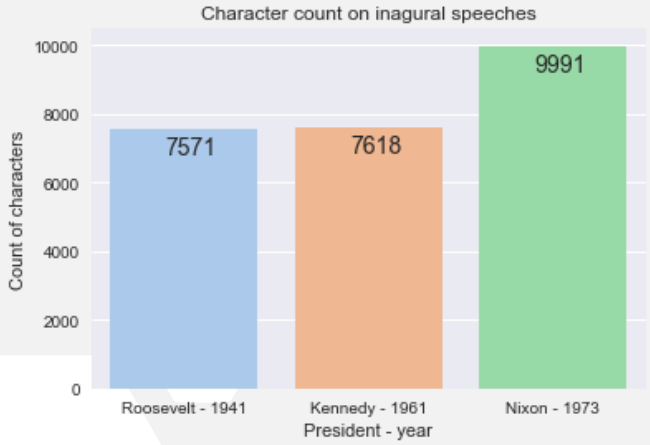
- President Roosevelt's inaugural speech in 1941 consists of 7571 characters
- In 1961 President Kennedy gave similarly long inaugural speech of 7618 character long
- President Nixon appears to gave the longest of the inaugural speech of the three, which is of 9991 characters long

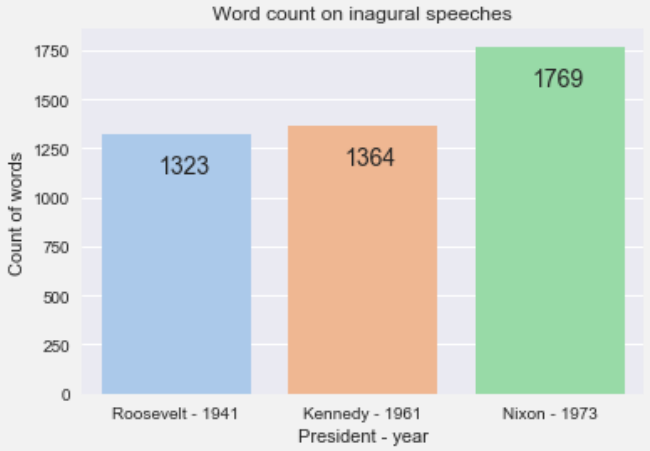
Word count

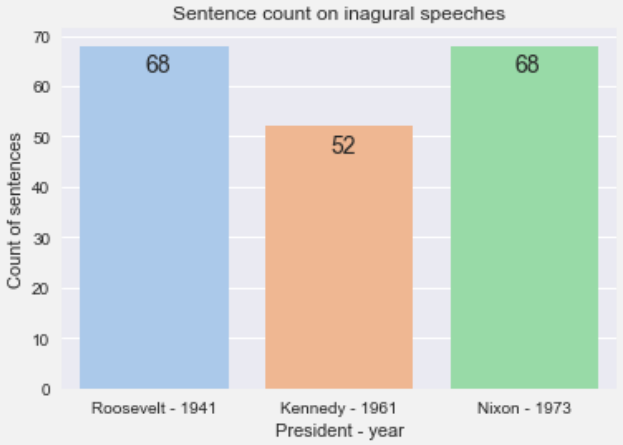
- In terms of number of words used in the speech, President Nixon stands out with 1769 words used in his 1973 inaugural speech
- Followed by President Kennedy's inaugural speech in 1961 with 1364 words
- President Roosevelt's inaugural speech in 1841 consists of 1323 words

Sentence count

- It looks like President Nixon favours longer sentences while delivering his speeches, because with larger word base he delivered equal number of sentences as other Presidents
- Both Presidents Roosevelt's and Nixon's inaugural speeches contained 68 sentences
- Followed by President Kennedy with 52 sentences in delivering his inaugural speech



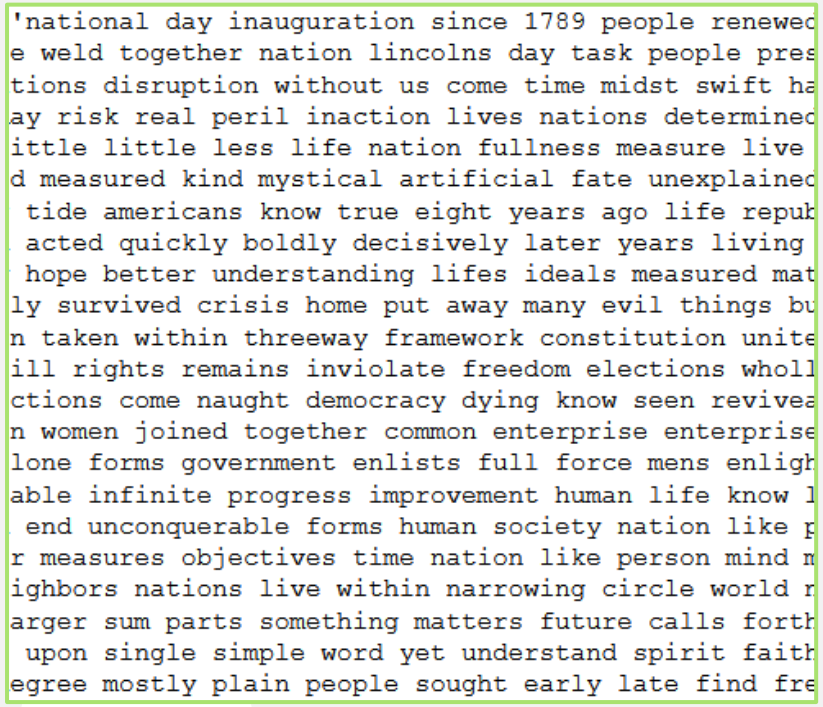




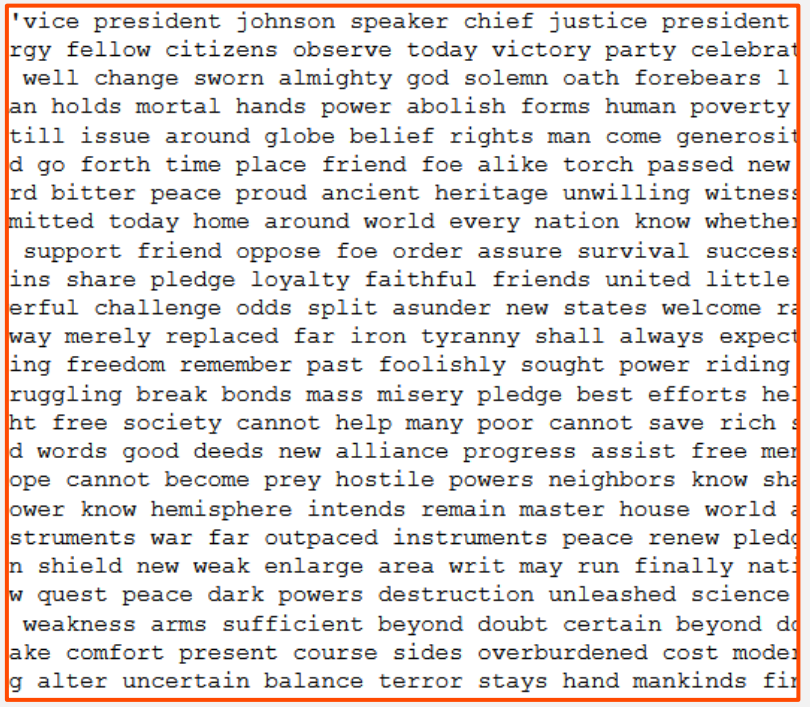
2.2 Remove the stopwords

```
from nltk.corpus import stopwords
stop_words = stopwords.words('english')
add_to_stop_words = ['mr','on','it','the','in','let','to','us','shall']
stop_words.extend(add_to_stop_words)
stop_words=set(stop_words)
```

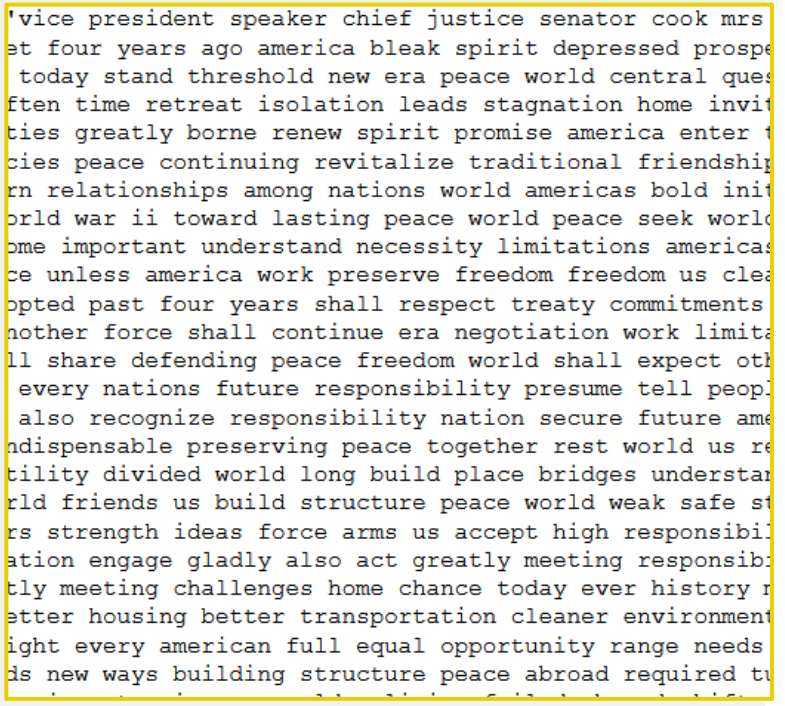
- Before removing the stopwords all the alphabets in the inaugural speech texts were converted to lower case
- After lower case conversion the punctuations including all special characters were removed from the speeches
- A set of words were added next to the corpus of stop words, as shown here
- Then stopwords are removed from the speeches, sample screen shots of the final texts are as below



President Roosevelt - 1941



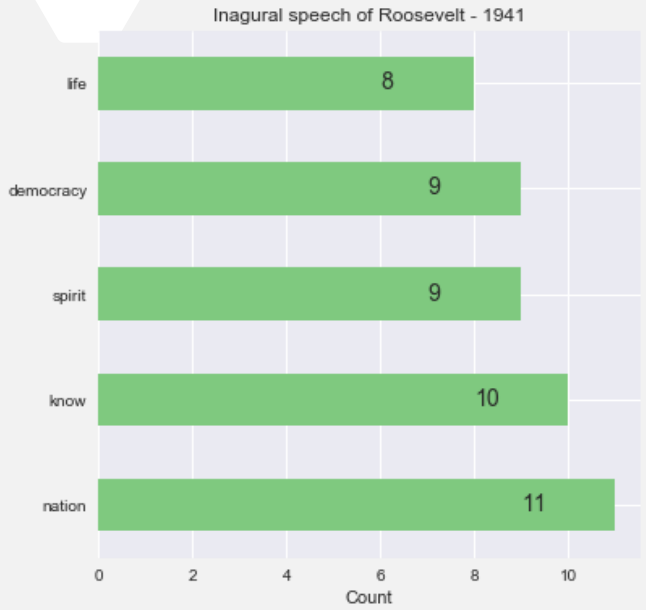
President Kennedy - 1961



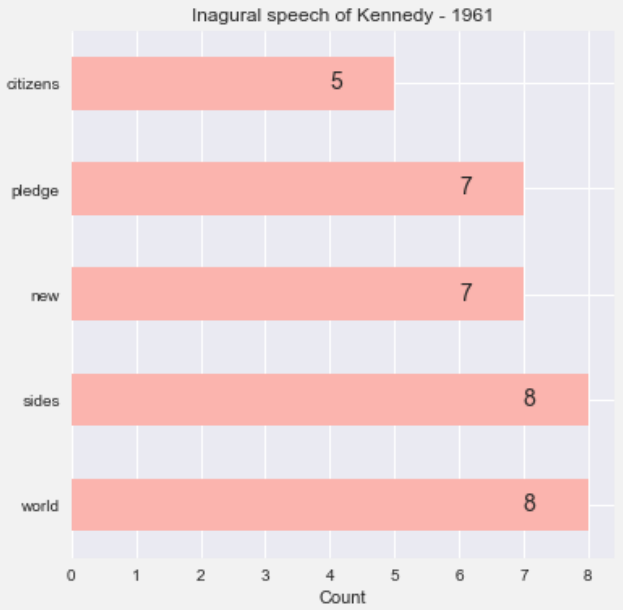
President Nixon - 1973

2.3 Most occurring words

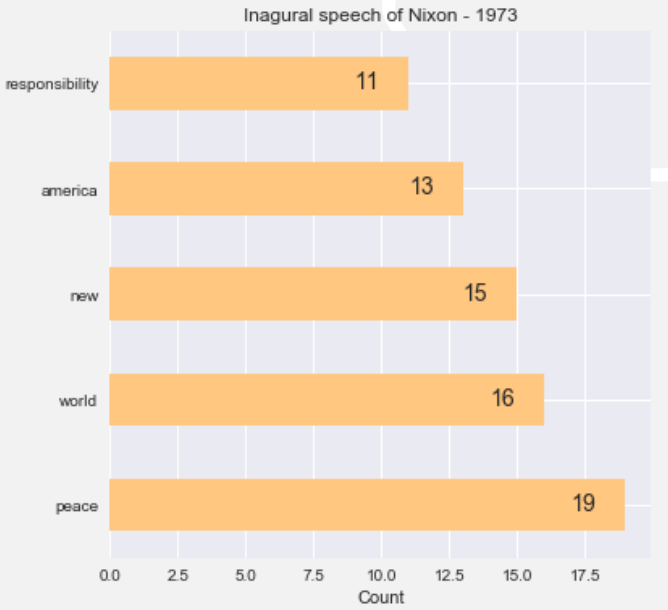
* Considering only nouns and verbs, pronouns were removed using stopword removal



Top 3 words	Frequency
Nation	11
Know	10
Spirit	9
Democracy	9



Top 3 words	Frequency
Sides	8
World	8
Pledge	7
New	7
Citizens	5



Top 3 words	Frequency
Peace	19
World	16
New	15

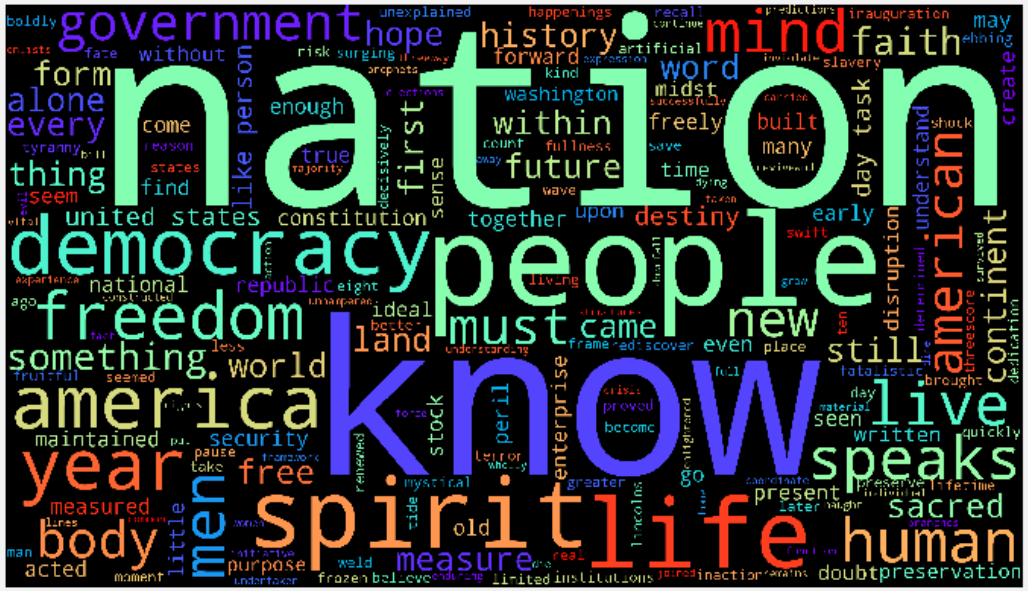
- In President Roosevelt's inaugural speech in 1941, 'nation' is the most often used word – 11 times, followed by 'know' occurring 10 times
- The words spirit and democracy came third occurring 9 times

- President Kennedy in his 1961 inaugural speech, 'world' and 'sides' were used most often, 8 times
- Followed by 'new' and 'pledge' used 7 times each and 'citizens' used 5 times in the speech

- In his 1973 inaugural speech, President Nixon used the word 'peace' 19 times, followed by 'world' 16 times and 'new' 15 times
- President Nixon used the pronoun 'us' 26 times in this speech

2.5 Plot the word-clouds

President Roosevelt - 1941



President Kennedy - 1961

President Nixon - 1973

- On analysing the word-cloud of the given inaugural speeches, we can draw various inferences on the prevailing world order and the challenges of the period in which these presidents took over the leadership of USA
- ‘freedom’, ‘peace’, ‘world’, ‘nation’, ‘new’ are few of the most common stated words used by the Presidents Roosevelt, Kennedy and Nixon
- When President Roosevelt took oath of office 1941, World War II was looming over the horizon and the nation officially entered into the war by end of the year. The words ‘nation’, ‘democracy’, ‘freedom’, ‘spirit’, ‘life’ etc..` are reflective of the period
- Whereas, President Kennedy took over the presidency during the height of cold war with USSR in 1961, where the world order had taken sides as blocs of powers. The words ‘world’, ‘sides’, ‘power, ‘new’, ‘freedom’, ‘peace’, ‘war’, ‘pledge’, ‘hope’ etc.. are indicative of the new world order
- In 1973 President Nixon won his second term after US called off offensive action in Vietnam. The words ‘peace’, ‘world’, ‘new’, ‘responsibility’, ‘freedom’, ‘history’ etc are reflective of the mandate he won in bring peace in Vietnam


ThankYou