Machine Learning - business Project

By Vinish Vincent

Great Learnings

PGP-DSBA - 23

# Table of content

Table of Content

**Problem 1**: 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. Read the dataset. Describe the data briefly. Interpret the inferences for each. Initial steps like head() .info(), Data Types, etc . Null value check, Summary stats, Skewness must be discussed.…………………………………………………………………………………..……………………………………………….2
  2. Perform Univariate and Bivariate Analysis. Do exploratory data analysis. Check for Outliers.....….4
  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)………………………………………………………………..…..……….…….13
  4. Apply Logistic Regression and LDA (linear discriminant analysis)……………………………………….........15
  5. Apply KNN Model and Naïve Bayes Model. Interpret the results……………………………………………….19
  6. Model Tuning, Bagging (Random Forest should be applied for Bagging), and Boosting……………..22
  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…………………….30
  8. Based on these predictions, what are the insights?.......................................................................31

**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:

* President Franklin D. Roosevelt in 1941
* President John F. Kennedy in 1961
* President Richard Nixon in 1973

2.1 Find the number of characters, words, and sentences for the mentioned documents…….…………32

2.2 Remove all the stopwords from all three speeches…………………………….…….………………...…………….32

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)……………………….……………………………...34

2.4 Plot the word cloud of each of the speeches of the variable. (after removing the stopwords)…..34

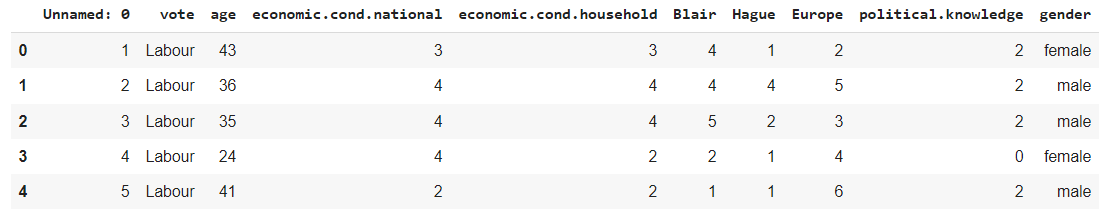
# Problem 1:

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.

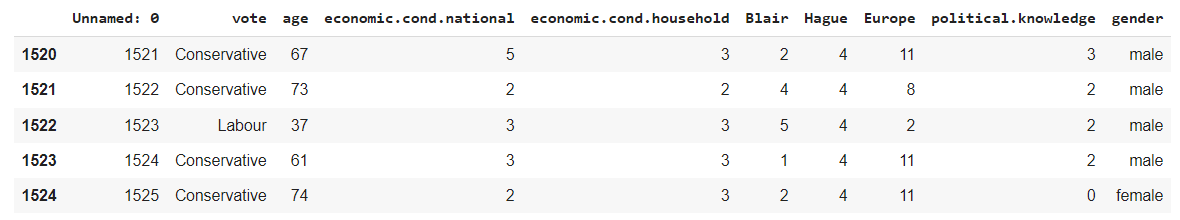
* 1. **Read the dataset. Do the descriptive statistics and do the null value condition ch eck. Write an inference on it.**

**Solution**:

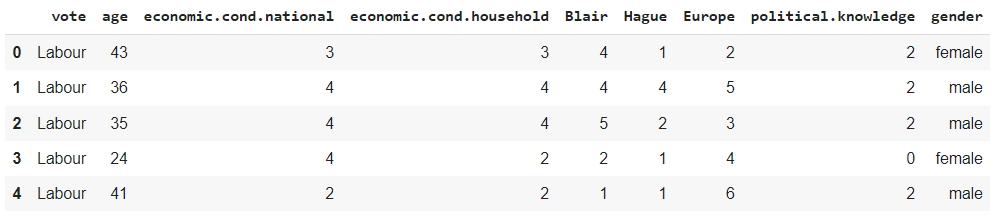
* Data top 5 records:



* Data last 5 records:

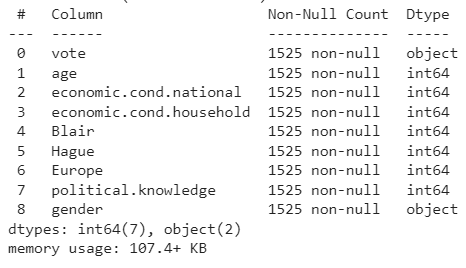


**“Unnamed: 0”** is **not needed** for our analysis as it represents index for the records, hence we are **dropping** the column

* Data top 5 records(after dropping “Unnamed: 0”):
* Data shape:

**No. of rows is 1525 and no. of columns are 9 (after dropping “Unnamed: 0”)**

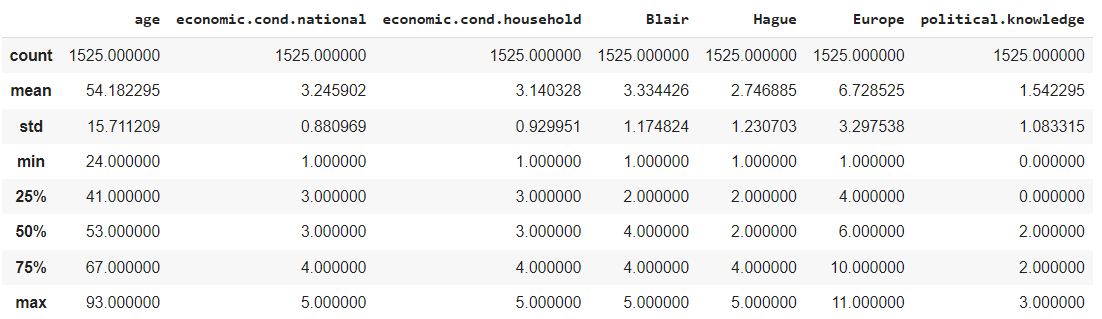
* Data info:



There are **total of 9 variables** present in the dataset

There are **no Null values** present in the dataset

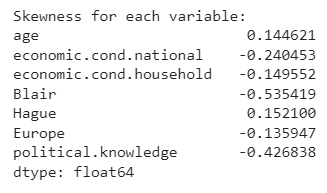
There are **2 Categorical Variables** - **vote** and **gender**; and there are **7 numeric type variables** - **age, economic\_cond\_national, economic\_cond\_household, Blair, Hague, Europe,** and **political\_knowledge**

* Data summary(5 point summary):

The values for **each field are of different scales**. For example, the field **'Europe'** shows the **values ranging between 1 and 11** while the values under the field **‘political.knowledge’ are ranging between 0 and 3**

Columns including **economic.cond.national, economic.cond.household, Blair and Hague** are **similarly rated Between 1 to 5**

* Skewness of each variable



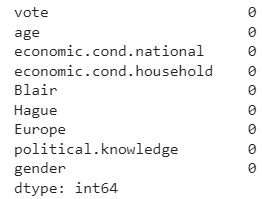
The skewness values indicate the asymmetry of the distribution for each variable. The variable **"age"** has a **positive skewness of 0.145**, indicating that the data is **slightly skewed to the right**. This suggests that there is a **slightly higher concentration of younger age people** in the dataset compared to older age people

Variables, such as “**economic.cond.national”, “economic.cond.household”, “Blair”, “Hague”, “Europe”, and “political.knowledge”**, there is a **negative skewness** suggesting that the majority of respondents **rated** these **variables** towards the **higher end**. This indicates a **tendency for positive ratings or a generally positive perception of these factors among the surveyed population**

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

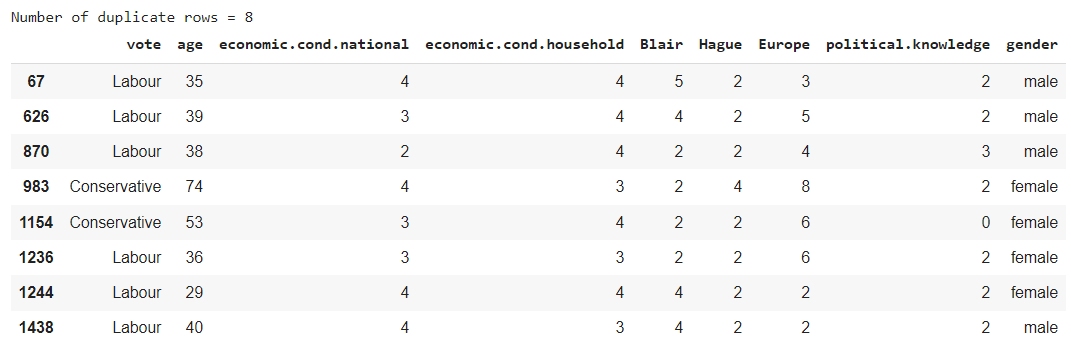
**Solution**:

* Checking for null values:



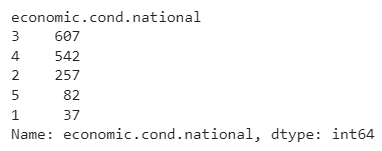
There are **no Null values** present in the dataset

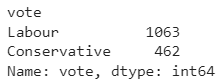
* Duplicate values:

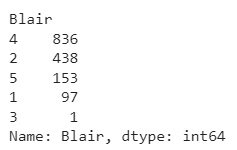
****

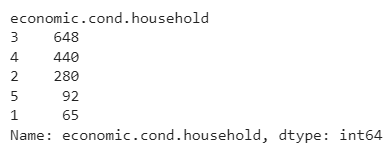
The dataset shows **8 duplicate values**. As per the dataset, there is a possibility that 2 different responses can be identical, hence we will **keep the duplicate values in our further analysis**

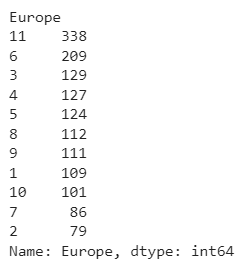
* Variable value distribution

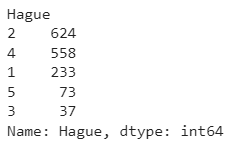


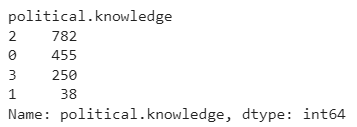


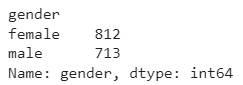






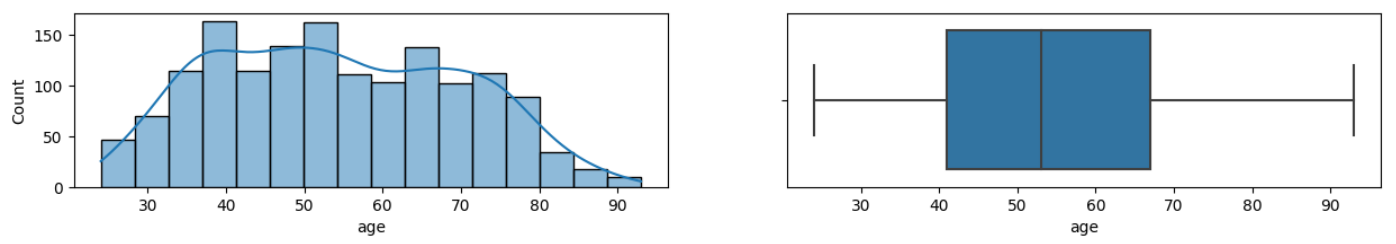




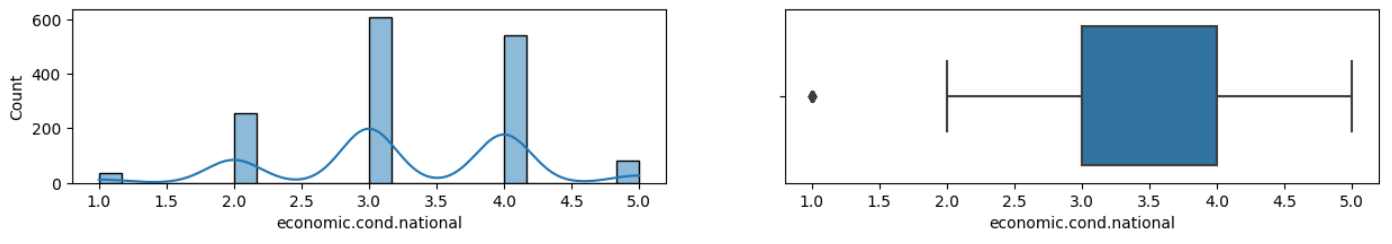


* Univariate analysis

**“Age”**

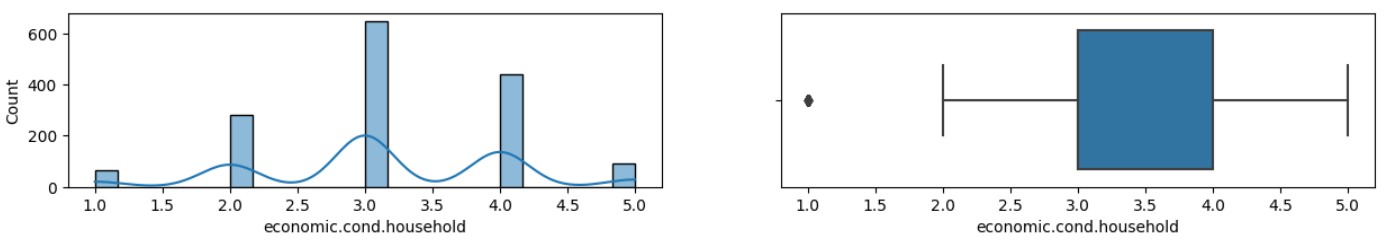
**75%** of the respondents lie in below the **age of 68** and **50%** below the **age** of **55**

**“economic.cond.national”**



Data suggests that the **majority of respondents gave a rating of 3 or 4** for the economic condition of the nation, while a **smaller number of respondents gave ratings of 2, 5, or 1**

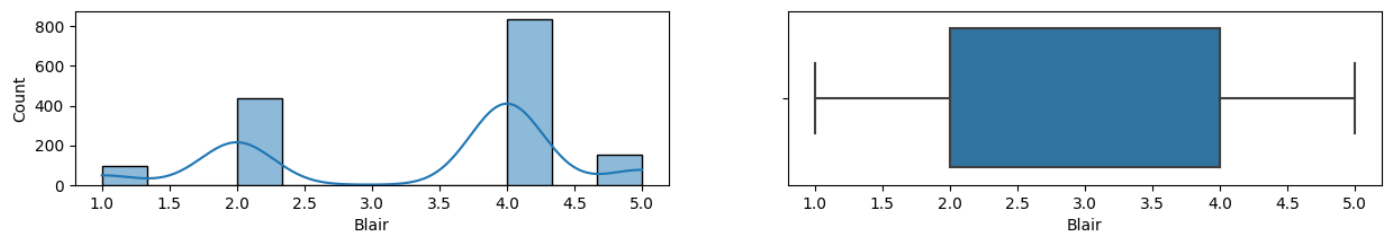
There **are some outliers** in the variable

**“economic.cond.household”**

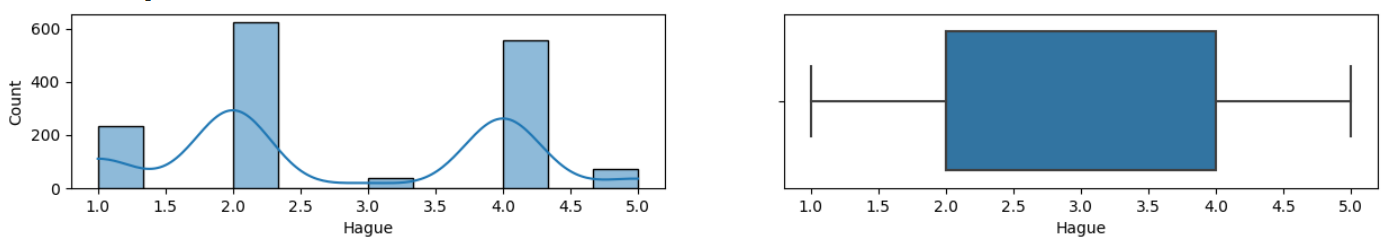
Data suggests that **majority of respondents gave a rating of** **3** (648 occurrences), followed by a rating of **4** (440 occurrences), with **fewer ratings for 2, 5, and 1** (280, 92, and 65 occurrences, respectively)

There **are some outliers** in the variable

**“Blair”**

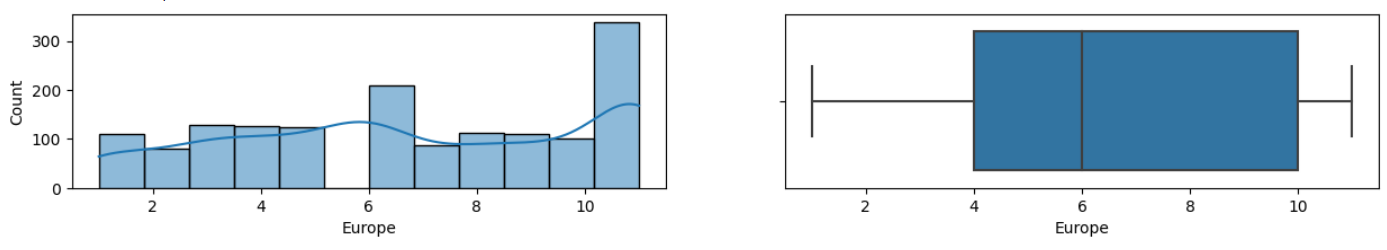
****

Data suggests that **majority of respondents gave a rating of 4** with 836 occurrences, while the **least frequent rating is 3 with only 1 occurrence**

**“Hague”**

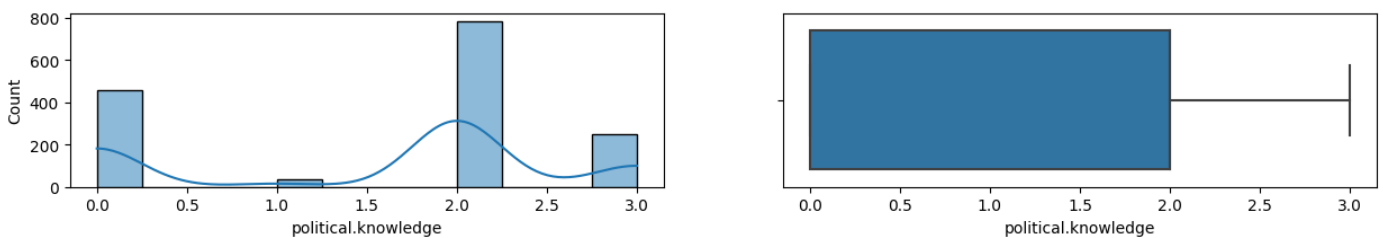
The **majority** of ratings fall into the **higher range,** with a count of 624 for rating **2** and 558 for rating **4,** while the **lowest rating, 3**, has the least count of 37

**“Europe”**

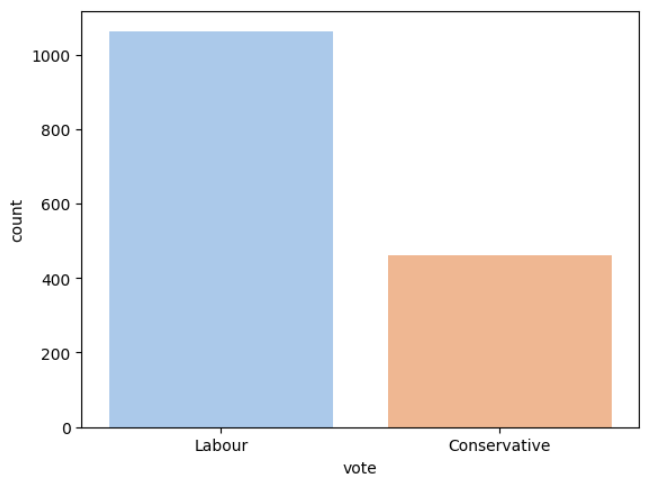
****

In **“Europe”,** the ratings range from 1 to 11, with the **highest** response count at 338 for rating11 and the **lowest** count at 79 for rating **2**

**“political.knowledge”**

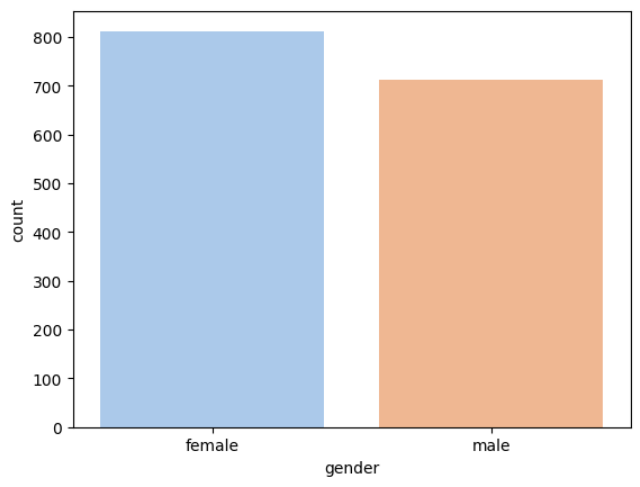
****

The **majority** of responses are for **2** (782 responses) **followed by 0** (455 responses), while the **lowest rating 1**, has the least count of 38 responses



**“vote”**

The **Labour** party received **1063 votes**, more than twice of the **Conservative party** which received **462 votes**

****

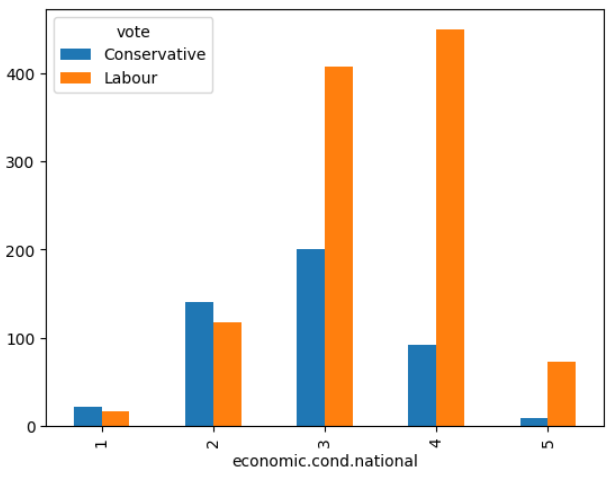
**“gender”**

The data indicates that the variable "gender" is that there are **812 females** and **713 males**

* Bivariate analysis:

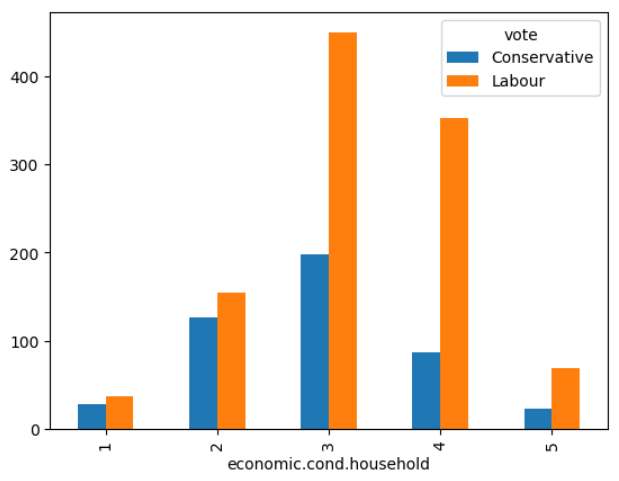
We used “vote” variable with other variable to view their relations

**“economic.cond.national”**



The **majority of voters** who rated the current national economic conditions as **3** belonged to the **Labour** party, while the **Conservative party had a smaller number of voters** in this category

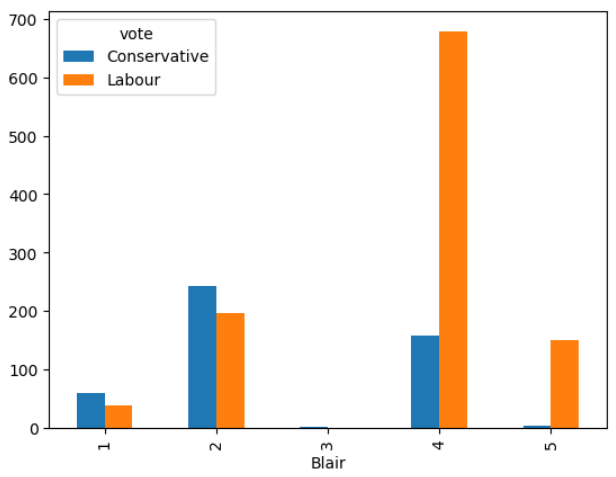
For **ratings 4 and 5**, the **Labour** party had **significantly more voters compared to the Conservative** party, indicating that a **larger proportion of Labour** party supporters were **dissatisfied** with the current economic conditions



**“economic.cond.household”**

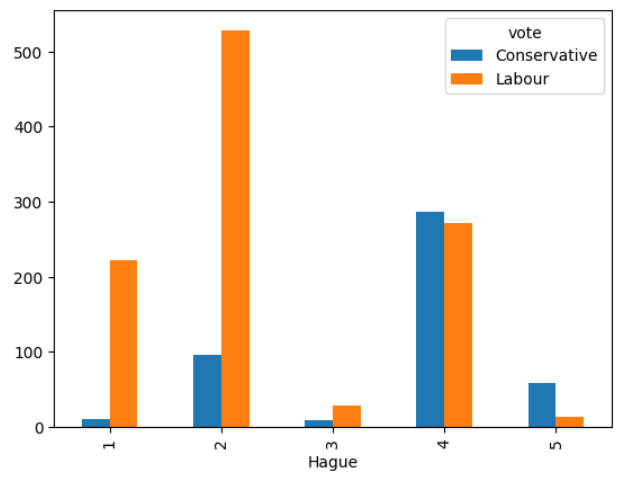
The **majority** of respondents who rated **3** chose the **Labour party, with 450 votes**, while **198 votes** chose the **Conservative** party

The **Conservative** party have **more votes** from respondents who rated **1 and 2**, while the **Labour** party has **more votes** from respondents who rated **3, 4, and 5**

**“Blair”**

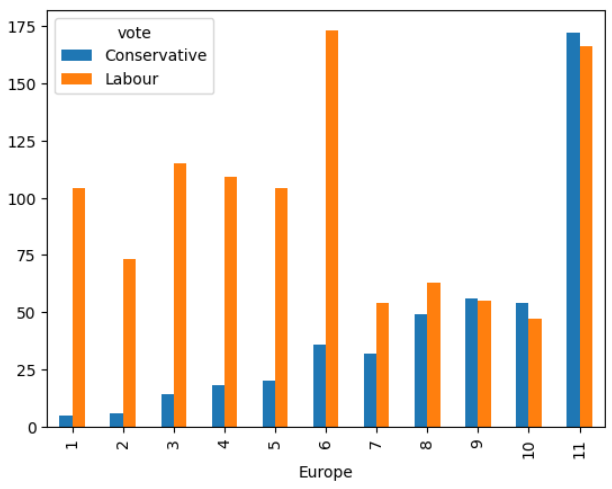
The **majority** of responses for the **Conservative** party (**59 out of 460**) rated the Labour leader with a score of **1**, indicating a relatively negative assessment

Conversely, the majority of responses for the Labour party (**679 out of 1034**) rated the Labour leader with a score of **4**, suggesting a more positive assessment compared to Conservative voters

**“Hague”**

The **majority** of respondents who gave a rating of **4** chose the **Conservative** party, with **287 respondents choosing Conservative** compared to **271 choosing Labour**

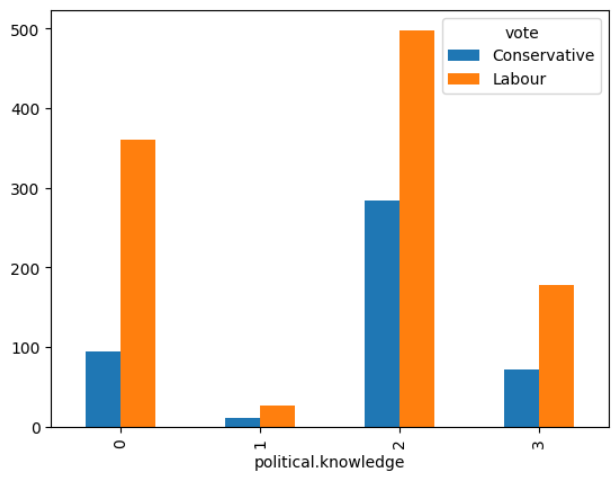
**Respondents** who rated **2** mostly **favored** the **Labour** party, **with 528 respondents** choosing Labour and only **96 choosing Conservative**

**“Europe”**

The number of respondents who voted **Conservative** and had Eurosceptic sentiment (high scores) **increased gradually from 5 to a peak of 172 at the highest scale point of 11**.

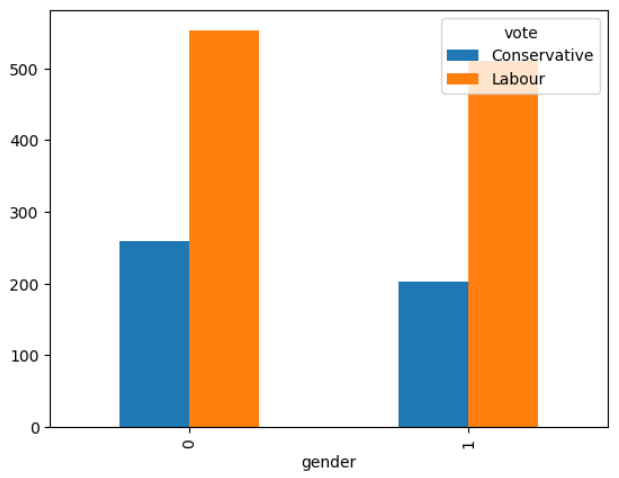
In contrast, the number of respondents who voted **Labour** and had **Eurosceptic sentiment (high scores) fluctuated** but generally remained lower than the Conservative voters, **ranging from 47 to 166 across the 11-point scale**.

**“political.knowledge”**



The **majority** of individuals with a **low** level of political knowledge (**0-1**) tend to vote for the **Labour** party, while those with a **higher level** of political knowledge (**2-3**) lean **more towards the Conservative** party.

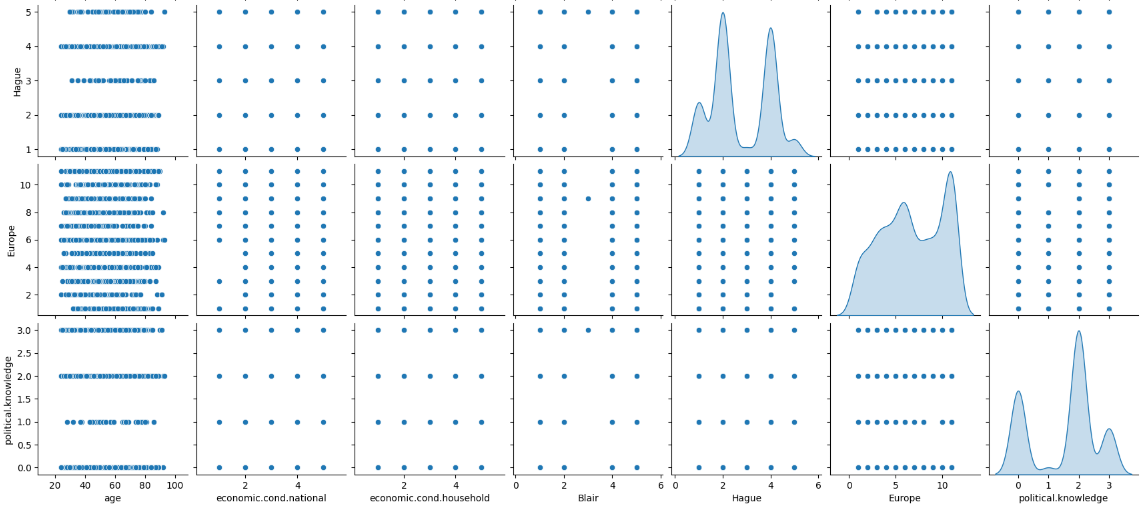
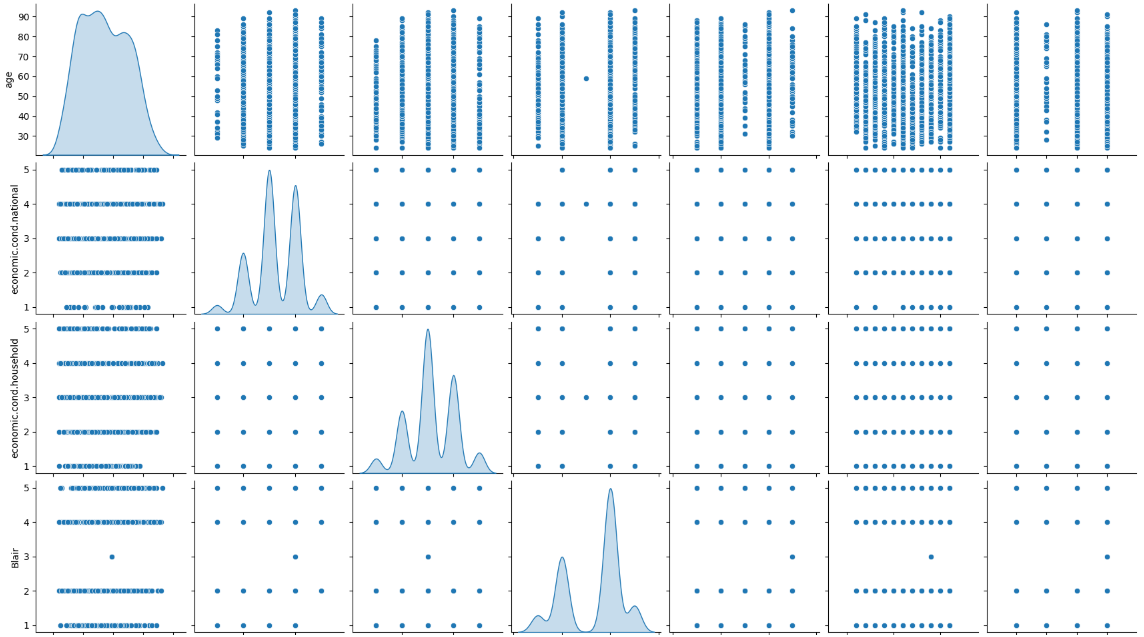
Among individuals with a **high level** of political knowledge (**2-3**), there is a **larger number of individuals** who vote for the **Labour** party **compared** to the **Conservative** party.

**“gender”**

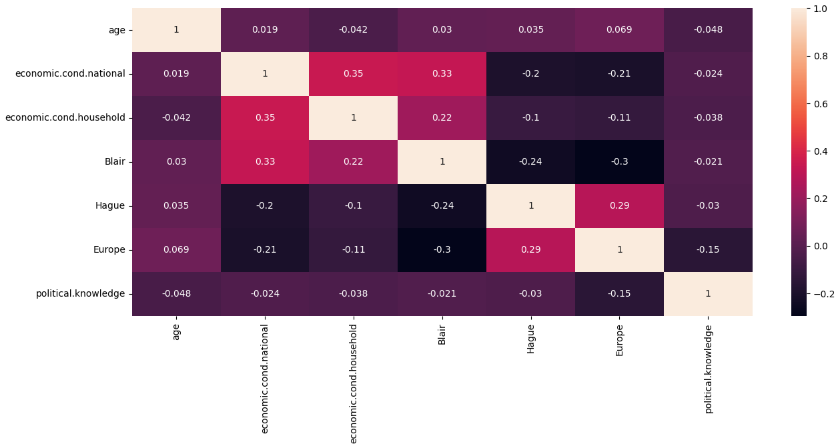
Among the respondents, **majority** voted for **Labour (1063) compared to Conservative (458)**.

The **majority** of **female respondents** voted for **Labour (553)**, while the **majority of male** respondents also **voted for Labour (510)**.

* Multivariate Analysis



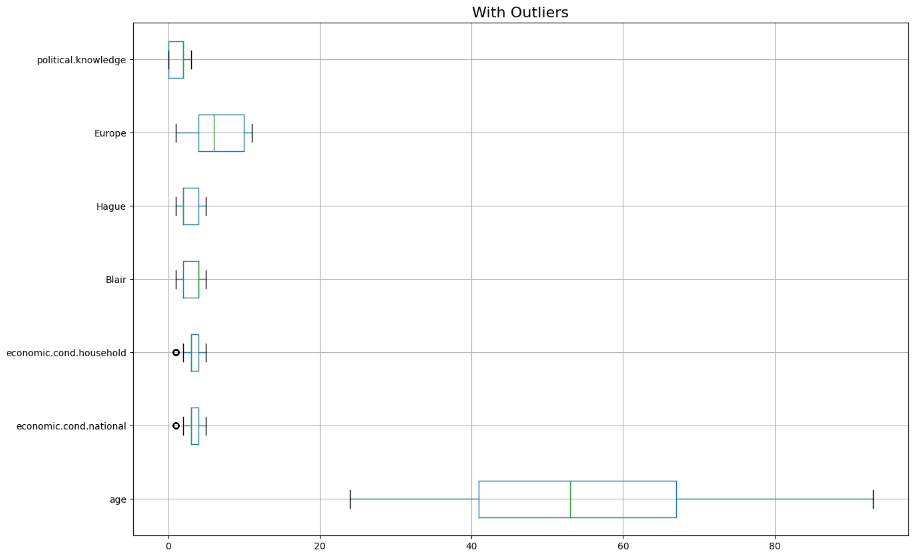
The pair plot show that there is no high correlation observed within the variables of the dataset

* Heatmap

**“economic.cond.national” has ~0.35 correlation between “economic.cond.household” and “Blair”. However this is a small value**

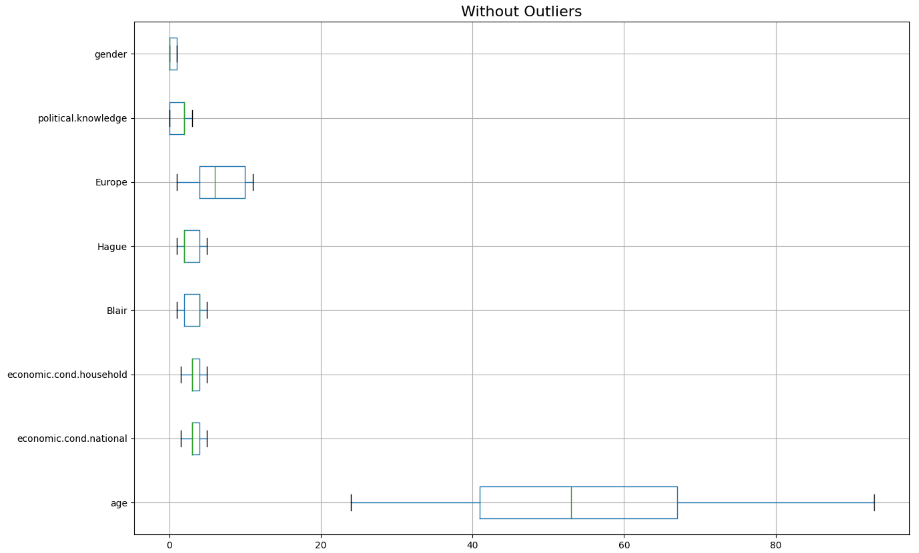
The above correlation plot shows that the **variables are having weak or low correlation between them**, i.e., the variables are not very related to each other

Checking for outliers



Two variables (economic\_cond\_national and economic\_cond\_househld) columns has outliers.

We are treating the data using interquartile range(IQR)

After Outliers treatment

* 1. **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).**

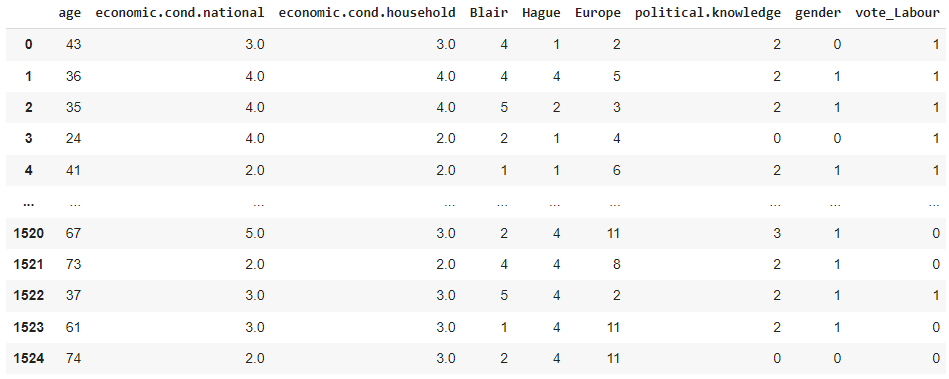
**Solution**:

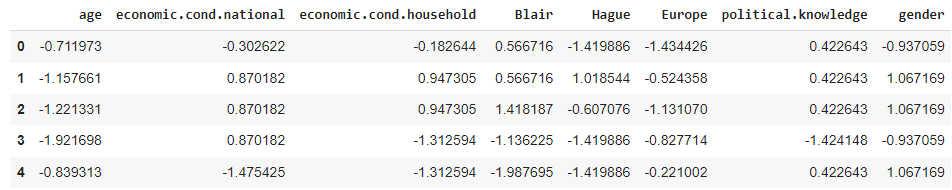
We will convert the “gender” variable and encode “female” as “0” and “male” as “1”

****We seperated the dataset into category and numeric datasets

**“vote” is the target variable** making others the independent variables

Encoding by creating dummies for “vote” variable

* Dataset after one-hot encoding:
* Scaling the numeric dataset by importing zscore from scipy.stats
* Data after scaling:

****

**Yes. Scaling is necessary** in this case. This is because all the **variables are not on same scale**. For example, the field 'Europe' shows the values ranging between 1 and 11 while the values under the field ‘political.knowledge’ are ranging between 0 and 3.

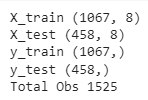
* Splitting the data

Before splitting the data into training and test sets, the final step involved is splitting the target variable from the independent variables. Here the target variable is ‘vote’ and the rest are the independent variables.

We are creating the dataset X to have all the predictor variables and the dataset y to have the dependent/target variable, “vote”.

Split the data into test and train. The X and y for the data has been formulated and split under the criteria 70:30 and random\_state = 1

Shape of the split data:

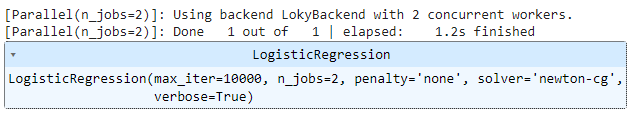


* 1. **Apply Logistic Regression and LDA (linear discriminant analysis).**

**Solution**:

* Logistic Regression Model:

We imported Logistic Regression from sklearn.



Parameters used:

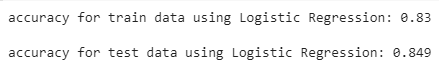
“solver”: This parameter specifies the algorithm to be used for optimization. In this case, 'newton-cg' is used, which stands for Newton Conjugate Gradient. It is a numerical optimization algorithm based on the Newton's method, designed to find the optimal coefficients in logistic regression.

“max\_iter”: This parameter determines the maximum number of iterations that the solver will perform to converge. In this case, it is set to 10,000, meaning that the solver will iterate up to 10,000 times to find the optimal solution. If convergence is not achieved within this limit, the solver will stop and return the current solution.

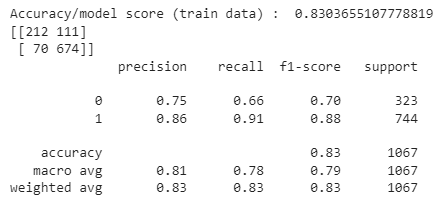
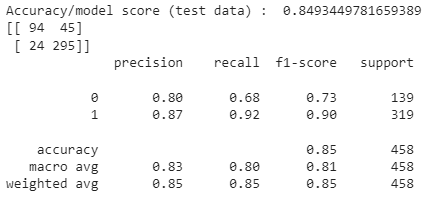
“penalty”: This parameter specifies the type of regularization to be applied to the logistic regression model. In this case, 'none' is used, indicating that no regularization is applied. Regularization is a technique used to prevent overfitting by adding a penalty term to the loss function.

“verbose”: This parameter controls the amount of output that the solver generates during the optimization process. When set to True, it provides detailed output, such as the progress of each iteration and the convergence information.

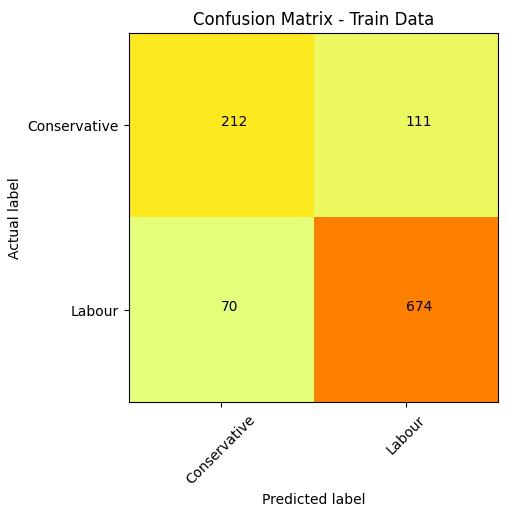
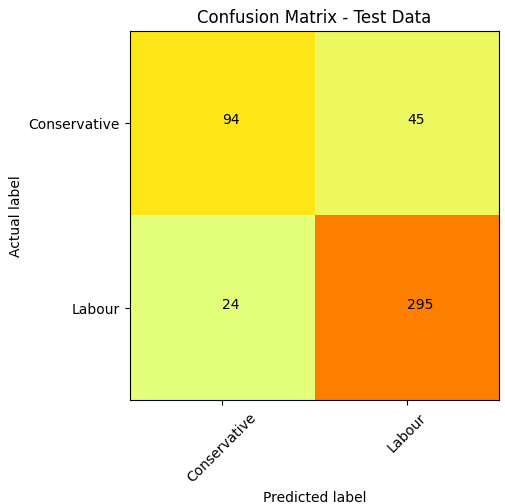
“n\_jobs”: This parameter specifies the number of CPU cores to be used for parallelizing the computation. In this case, 2 is used, which means that two CPU cores will be utilized for the computation. This can help speed up the training process, especially for large datasets.



Logistic regression Classification report

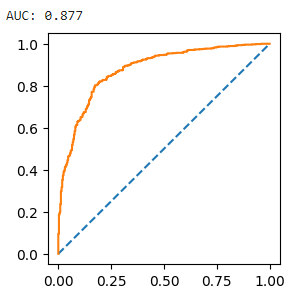
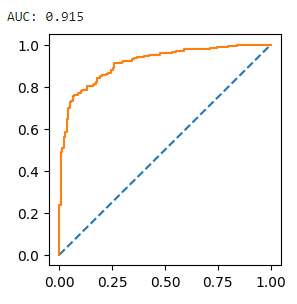
 Train data Test Data

Logistic regression Confusion matrix

 Train data Test Data

Logistic regression ROC curve and ROC\_AUC score

Train data Test Data



AUC score: 0.877 AUC score: 0.915

The model is defined with above parameters and fitted on Train and Test data. The model gives an accuracy score as 83% for Train Data and 84.90% on Test Data.

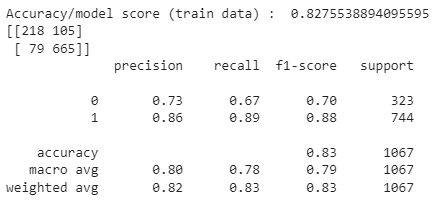
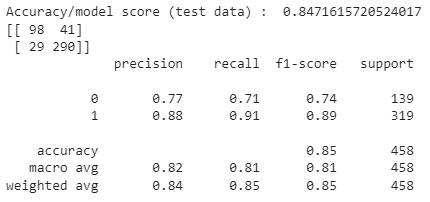
Validness - Based on the observation we see on training and testing dataset accuracy results, the Logistic regression model does not seem over or under fit.

* LDA Model:

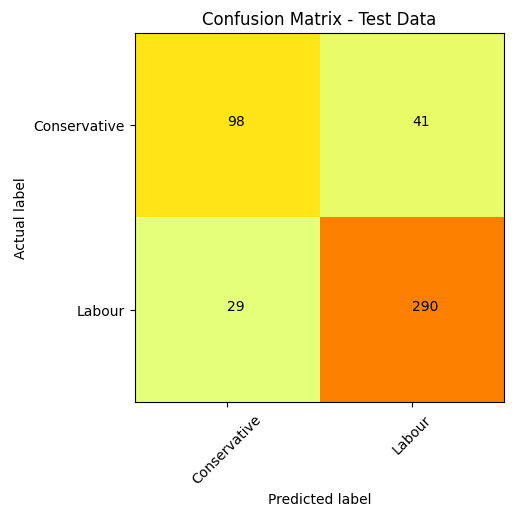
We imported LinearDiscriminantAnalysis from sklearn.discriminant\_analysis

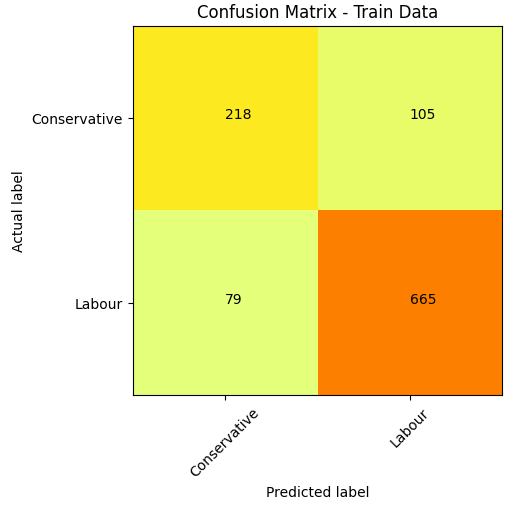
We fitted the model with train and test data

LDA Classification report

 Train data Test Data

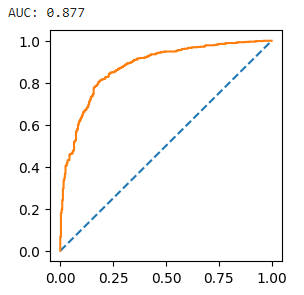
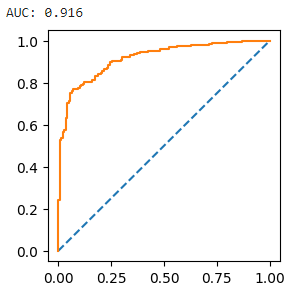
LDA Confusion matrix

 Train data Test Data



LDA ROC curve and ROC\_AUC score

Train data Test Data



AUC score: 0.877 AUC score: 0.916

The model is defined with above parameters and fitted on Train and Test data. The model gives an accuracy score as 82.8% for Train Data and 84.7% on Test Data.

Validness - Based on the observation we see on training and testing dataset accuracy results, the LDA model does not seem over or under fit.

* 1. **Apply KNN Model and Naïve Bayes Model. Interpret the results.**
* **KNN Model**

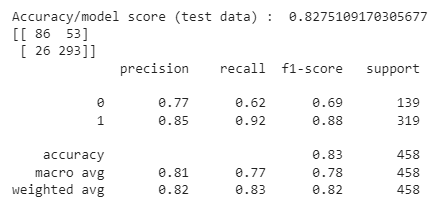
K-nearest neighbors (KNN) algorithm is a type of supervised ML algorithm which can be used for both classification as well as regression predictive problems.

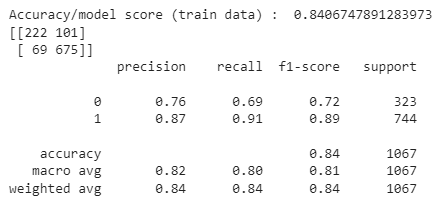
We imported KNeighborsClassifier from sklearn.neighbors

We fitted the model with train and test data



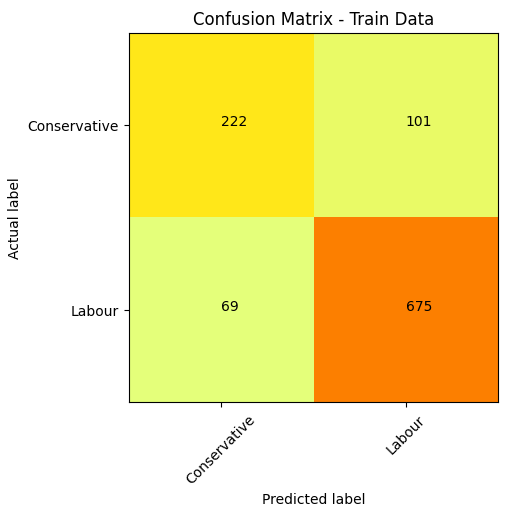
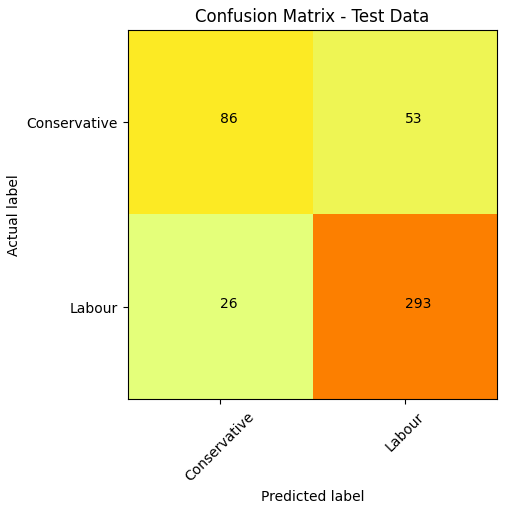
KNN model Classification report

 Train data Test Data



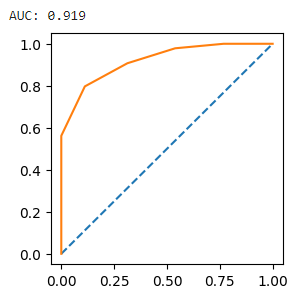
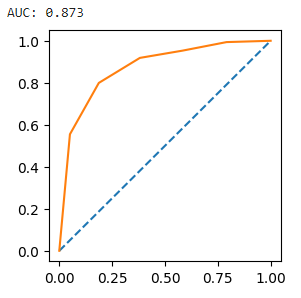
KNN model Confusion matrix

Train data Test Data



KNN model ROC curve and ROC\_AUC score

Train data Test Data



AUC score: 0.919 AUC score: 0.873

The model is defined with above parameters and fitted on Train and Test data. The model gives an accuracy score as 84.1% for Train Data and 82.8% on Test Data.

Validness - Based on the accuracy of the Train and Test set, we can confirm that there is under fitting that has happened in the model.

* **Naïve Bayes Model**

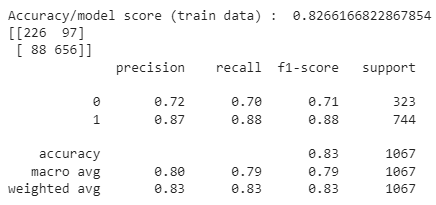
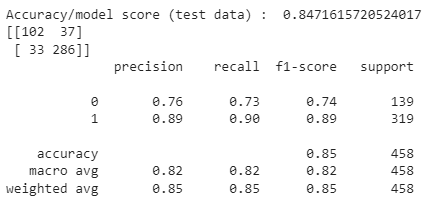
Naive Bayes classifiers is a model based on applying Bayes' theorem with strong (naïve) independent assumptions between the features.

Here the method that we are going to use is the GaussianNB() method. This method requires all the features to be in categorical type. A general assumption in this method is the data is following a normal distribution.

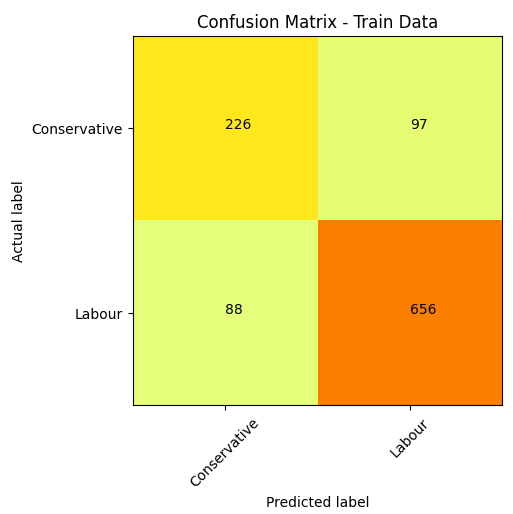
We imported GaussianNB from sklearn.naive\_bayes

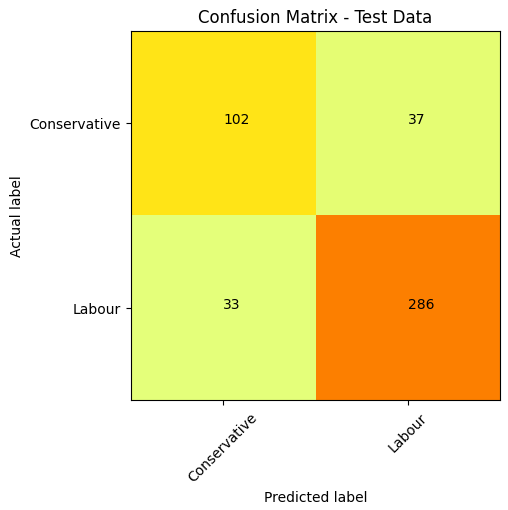
We fitted the model with train and test data

Naïve Bayes model Classification report

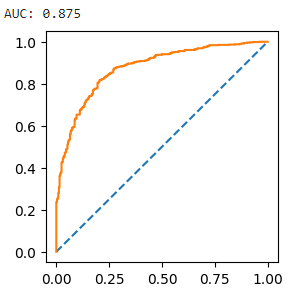
 Train data Test Data

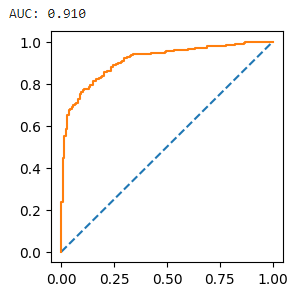
Naïve Bayes model Confusion matrix

 Train data Test Data



Naïve Bayes model ROC curve and ROC\_AUC score

 Train data Test Data



AUC score: 0.875 AUC score: 0.910

The model is defined with above parameters and fitted on Train and Test data. The model gives an accuracy score as 82.6% for Train Data and 84.7% on Test Data.

Validness - Based on the accuracy of the Train and Test set, we can confirm that model does not seem over or under fit.

* 1. **Model Tuning, Bagging (Random Forest should be applied for Bagging), and Boosting**

**Solution:**

* Model Tuning

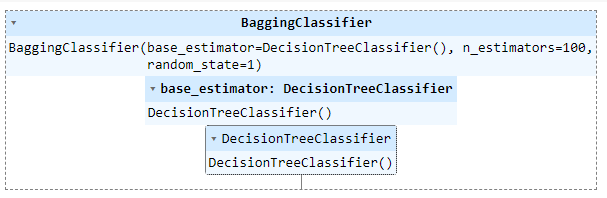
Tuning is the process of maximizing a model’s performance without overfitting or creating too high of a variance. In machine learning, this is accomplished by selecting appropriate “hyper-parameters”.

* Bagging Model (Using Decision Tree Classifier)

Random forests or random decision forests is an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time

Random forests generally outperform decision trees, but their accuracy is lower than gradient boosted trees. However, data characteristics can affect their performance.

We imported BaggingClassifier from sklearn.ensemble

We also imported DecisionTreeClassifier from sklearn.tree

Parameters used:

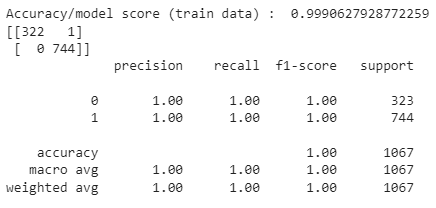
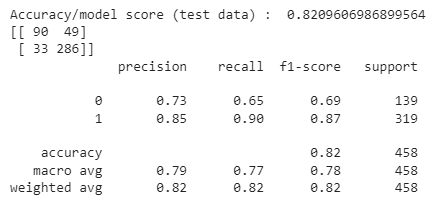
“base\_estimator”: This parameter specifies the base estimator or the model to be used as the individual weak learners in the ensemble. In this case, the base estimator is specified as cart, which likely refers to the CART (Classification and Regression Trees) algorithm. CART is a decision tree algorithm commonly used as a base estimator in bagging and boosting techniques.

“n\_estimators”: This parameter determines the number of base estimators or weak learners to be included in the ensemble. In this case, n\_estimators is set to 100, meaning that the ensemble will consist of 100 individual CART models.

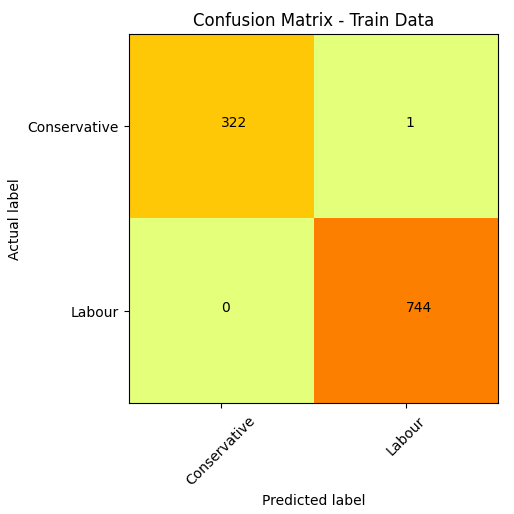
“random\_state”: This parameter is used to control the random number generator for reproducibility. It is set to 1, which means that the random state is fixed at 1, ensuring that the results of the model remain the same across different runs.

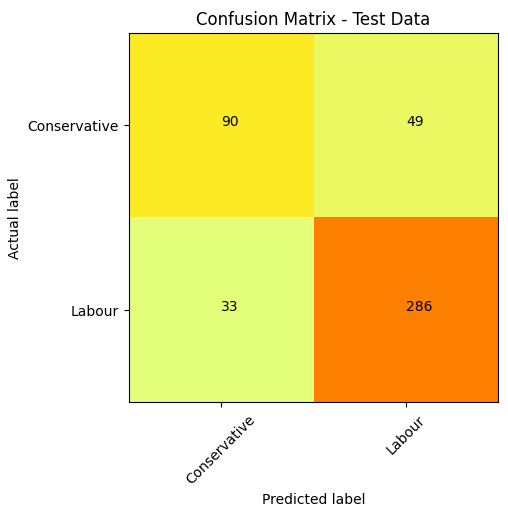
We fitted the model with train and test data

Bagging model Classification report

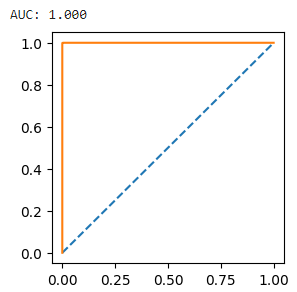
 Train data Test Data

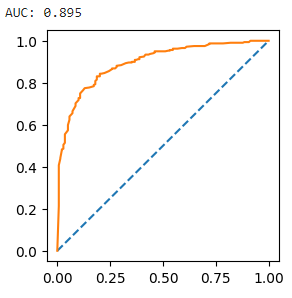
Bagging model Confusion matrix

 Train data Test Data



Bagging model ROC curve and ROC\_AUC score

 Train data Test Data



AUC score: 1.000 AUC score: 0.895

The model is defined with above parameters and fitted on Train and Test data. The model gives an accuracy score as 99.9% for Train Data and 82.1% on Test Data.

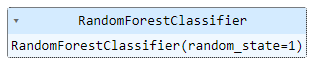
Validness - Based on the accuracy of the Train and Test set, we can confirm that the model seems to be overfitting.

* Bagging Model (Using Random Forest Classifier)

Random forests or random decision forests is an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time

Random forests generally outperform decision trees, but their accuracy is lower than gradient boosted trees. However, data characteristics can affect their performance.

We imported BaggingClassifier from sklearn.ensemble

We also imported RandomForestClassifier from sklearn.ensemble

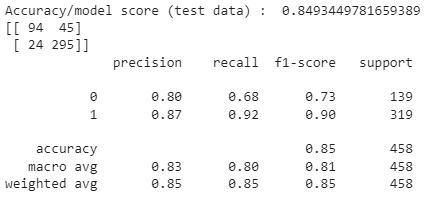
Parameters used:

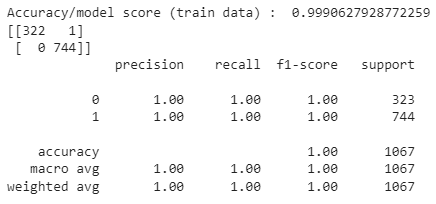
“n\_estimators”: This parameter determines the number of base estimators or weak learners to be included in the ensemble. In this case, n\_estimators is set to 100, meaning that the ensemble will consist of 100 individual CART models.

“random\_state”: This parameter is used to control the random number generator for reproducibility. It is set to 1, which means that the random state is fixed at 1, ensuring that the results of the model remain the same across different runs.

We fitted the model with train and test data

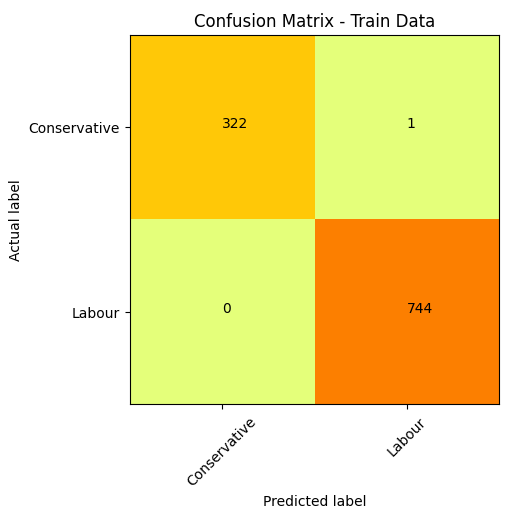
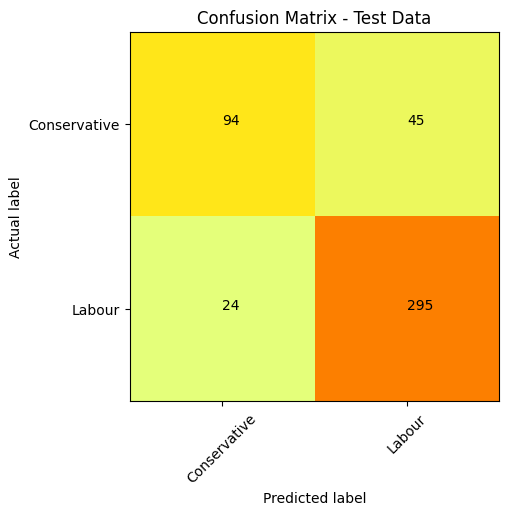
Random Forest model Classification report

 Train data Test Data

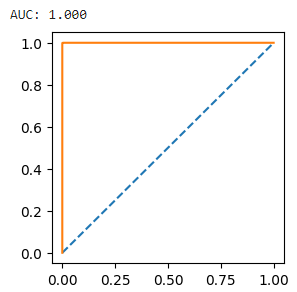


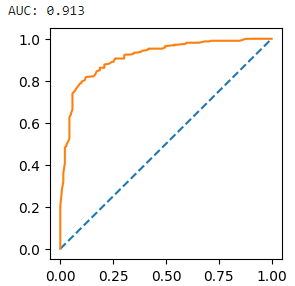
Random Forest model Confusion matrix

Train data Test Data



Random Forest model ROC curve and ROC\_AUC score

 Train data Test Data



AUC score: 1.000 AUC score: 0.913

The model is defined with above parameters and fitted on Train and Test data. The model gives an accuracy score as 99.9% for Train Data and 84.9% on Test Data.

Validness - Based on the accuracy of the Train and Test set, we can confirm that the model seems to be overfitting.

* Boosting Model (ADA Boost)

AdaBoost method operates iteratively, identifying misclassified data points and adjusting their weights to minimize the training error. The model continues optimize in a sequential fashion until it yields the strongest predictor.

We imported AdaBoostClassifier from sklearn.ensemble

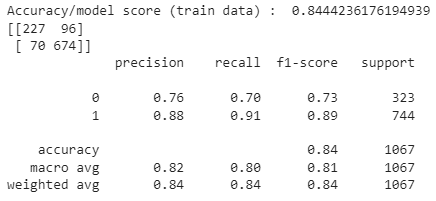
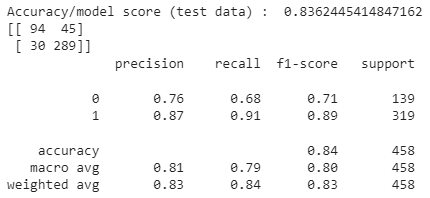
Parameters used:

“n\_estimators”: This parameter determines the number of base estimators or weak learners to be included in the ensemble. In this case, n\_estimators is set to 100, meaning that the ensemble will consist of 100 individual CART models.

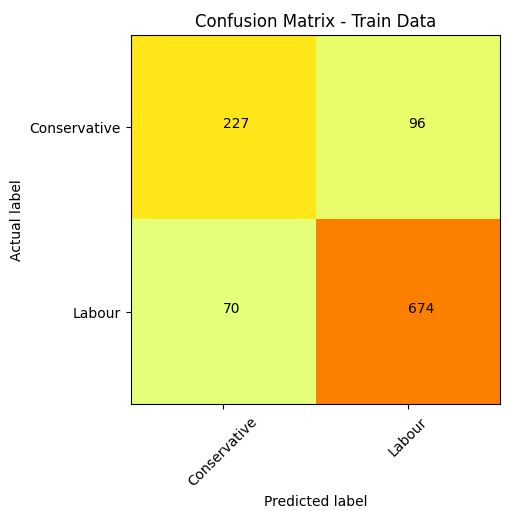
“random\_state”: This parameter is used to control the random number generator for reproducibility. It is set to 1, which means that the random state is fixed at 1, ensuring that the results of the model remain the same across different runs.

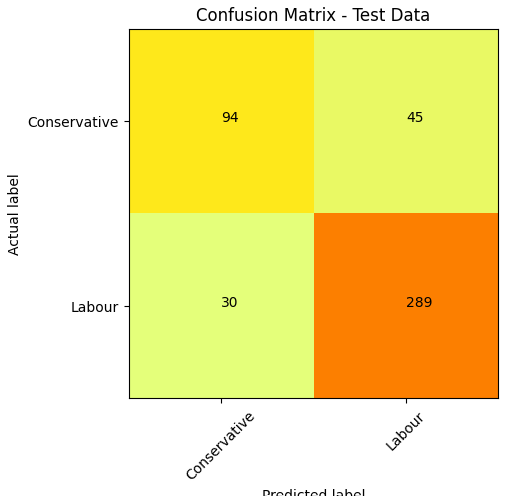
We fitted the model with train and test data

ADA Boost model Classification report

 Train data Test Data

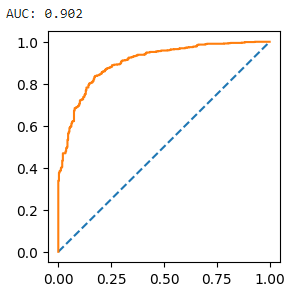
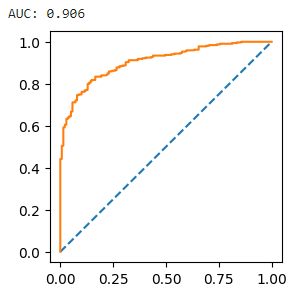
ADA Boost model Confusion matrix

 Train data Test Data



ADA Boost model ROC curve and ROC\_AUC score

Train data Test Data



AUC score: 0.902 AUC score: 0.906

The model is defined with above parameters and fitted on Train and Test data. The model gives an accuracy score as 84.4% for Train Data and 83.6% on Test Data.

Validness - Based on the accuracy of the Train and Test set, we can confirm that the model seems to be slightly overfitting.

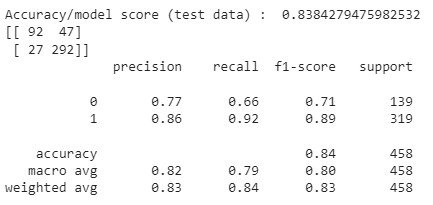
* Boosting Model (Gradient Boost)

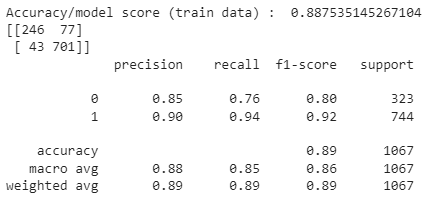
The gradient boosting trains on the residual errors of the previous predictor. The name, gradient boosting, is used since it combines the gradient descent algorithm and boosting method.

We imported GradientBoostingClassifier from sklearn.ensemble

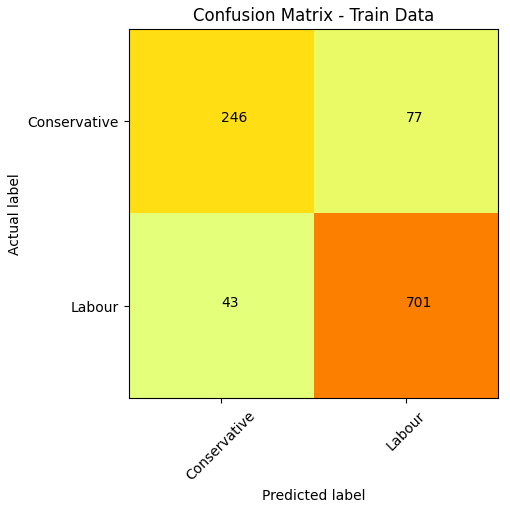
We fitted the model with train and test data

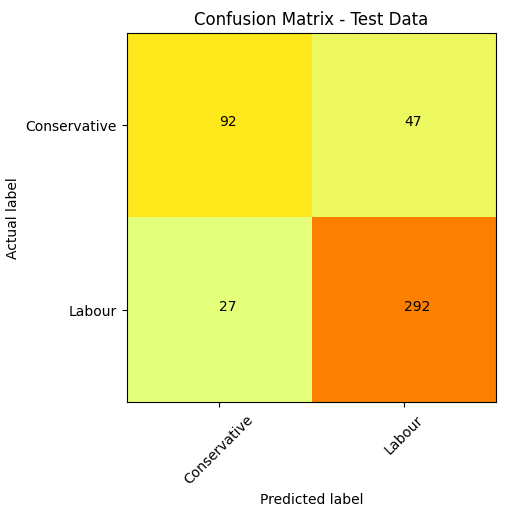
Gradient Boost model Classification report

 Train data Test Data



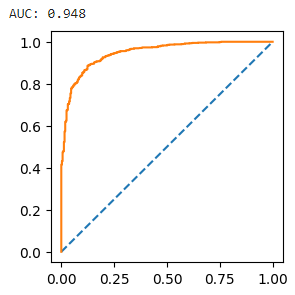
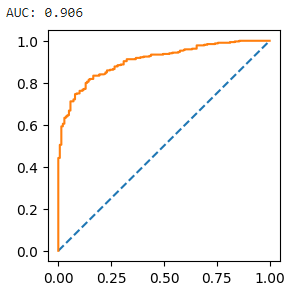
Gradient Boost model Confusion matrix

 Train data Test Data



Gradient Boost model ROC curve and ROC\_AUC score

Train data Test Data



AUC score: 0.948 AUC score: 0.906

The model is defined with above parameters and fitted on Train and Test data. The model gives an accuracy score as 88.8% for Train Data and 83.8% on Test Data.

Validness - Based on the accuracy of the Train and Test set, we can confirm that the model seems to be slightly overfitting.

* 1. **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**

**Solution: Performance table of all the models:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Model** | **Accuracy** | | **Precision** | | **Recall** | | **AUC score** | |
| **Train data** | **Test data** | **Train data** | **Test data** | **Train data** | **Test data** | **Train data** | **Test data** |
| **Logistic Regression** | 0.830 | 0.849 | 0.86 | 0.87 | 0.91 | 0.92 | 0.877 | 0.915 |
| **LDA** | 0.828 | 0.847 | 0.86 | 0.88 | 0.89 | 0.91 | 0.877 | 0.916 |
| **KNN** | 0.841 | 0.828 | 0.87 | 0.85 | 0.91 | 0.92 | 0.919 | 0.873 |
| **Naïve Bayes** | 0.827 | 0.847 | 0.87 | 0.89 | 0.88 | 0.90 | 0.875 | 0.910 |
| **Model Tuning – Decision Tree** | 0.999 | 0.821 | 1.00 | 0.85 | 1.00 | 0.90 | 1.000 | 0.895 |
| **Bagging – Random Forest** | 0.999 | 0.849 | 1.00 | 0.87 | 1.00 | 0.92 | 1.000 | 0.913 |
| **ADA Boosting** | 0.844 | 0.836 | 0.88 | 0.87 | 0.91 | 0.91 | 0.902 | 0.906 |
| **Gradient Boost** | 0.888 | 0.838 | 0.90 | 0.86 | 0.94 | 0.92 | 0.948 | 0.906 |

**Inference:**

Based on the data, it appears that the ***Gradient Boost model*** has the **highest accuracy, precision, recall, and AUC score** on both the train and test data. Therefore, based on these performance metrics, the **Gradient Boost model** can be considered the **best model** among the options given.

* **High performance metrics:** The Gradient Boost model has consistently high accuracy, precision, recall, and AUC scores on both the train and test data. This indicates that the model is performing well in terms of correctly classifying both positive and negative instances, while also achieving a high overall accuracy.
* **Low overfitting:** The Gradient Boost model's performance on the test data is comparable to its performance on the train data, suggesting that it is not overfitting the training data.
* **Robustness to imbalanced classes**: Imbalanced classes occur when the number of instances in different classes is significantly different. The Gradient Boost model's high precision, recall, and AUC scores suggest that it is effective in handling imbalanced class distributions. It is important to consider these metrics when dealing with imbalanced datasets, as accuracy alone may not provide an accurate representation of model performance.

After Gradient Boost model, ***Bagging – Random Forest*** would have been the selected model

* Random Forest performs similarly to Decision Tree on the training data but provides better generalization on the test data. It effectively reduces overfitting and demonstrates strong performance.
* The Bagging - Random Forest model achieves a perfect recall score of 1.00, meaning it identifies all positive instances correctly.
* It achieves a high AUC score of 0.913, indicating a good ability to distinguish between positive and negative instances.
* The accuracy of 0.849 is relatively high, indicating a good overall performance of the model.
* The precision of 0.87 demonstrates a decent ability to correctly classify positive instances.

Comments on other models:

* **Logistic Regression**: Logistic Regression performs reasonably well across all metrics, with good accuracy, precision, recall, and AUC scores. It shows a balanced performance on both train and test data
* **LDA (Linear Discriminant Analysis)**: LDA performs similarly to Logistic Regression in terms of accuracy, precision, recall, and AUC scores. It shows consistent performance on both train and test data
* **KNN (K-Nearest Neighbors)**: KNN shows good accuracy and recall scores, but its precision and AUC score are slightly lower compared to the top-performing models.
* **Naïve Bayes**: Naïve Bayes performs decently with balanced accuracy, precision, recall, and AUC scores. However, its performance is slightly lower compared to Logistic Regression and LDA
* **Model Tuning – Decision Tree**: Decision Tree shows excellent performance on the training data, achieving near-perfect accuracy, precision, recall, and AUC scores. However, it suffers from overfitting, as indicated by its lower scores on the test data
* **ADA Boosting**: ADA Boosting performs well with balanced accuracy, precision, recall, and AUC scores. However, its performance is slightly lower compared to the top models.
  1. **Based on these predictions, what are the insights?**

**Insights:**

* The **average age** of voters who voted for **Conservative** Party seem to be a **little higher than those who voted for Labour party**
* Voters who have given **higher ratings for current national economic conditions and current household economic conditions** show favour the **Labour** Party highly.
* Those who have **higher political knowledge** have **voted** for **conservative** party
* **Labour** party is **performing better than conservative** from huge margin. **Labour** party has **higher focus** in the **election**.
* **Female** voters turn out is **greater** than the male voters. Channel should educate and focus on strategies to target more males to vote.
* Voters who have given ratings of **3** or less for “**Hague**” are more in favour of the **Labour** party. Voters who have given a rating of **4** for “**Hague**” seem to be **equally in favour** of either party, and voters who have given a rating of **5** for “**Hague**” seem to favour the **Conservative** party more

# Problem 2: Text Analytics

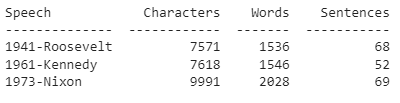
In this particular project, we are going to work on the inaugural corpora from the nltk in Python. We will be looking at the following speeches of the Presidents of the United States of America:

* President Franklin D. Roosevelt in 1941
* President John F. Kennedy in 1961
* President Richard Nixon in 1973
  1. **Find the number of characters, words, and sentences for the mentioned documents**

**Solution**:

* Number of Characters in each speech:

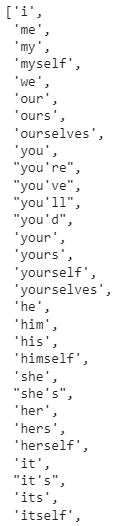


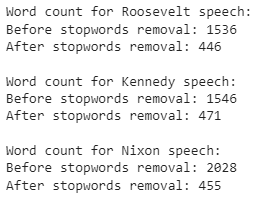
* Number of words in each speech:
* Number of sentences in each speech:
* Table:

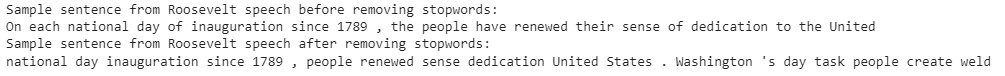
* 1. **Remove all the stopwords from all three speeches. Show the word count before and after the removal of stopwords. Show a sample sentence after the removal of stopwords.**

**Solution**: Removing all stopwords and punctuations:

Sample stopwords



* Words before and after stopwords for speeches:

Sample of first 20 words sentence of Roosevelt speech before and after removing stopwords:

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

**Solution**: Top 3 words:

* Roosevelt Speech :



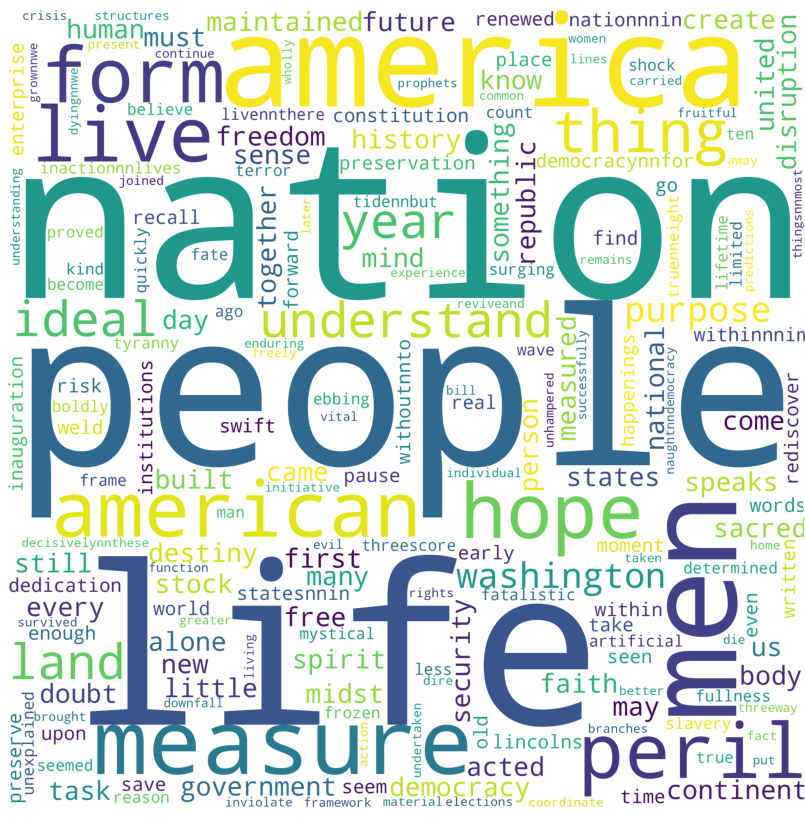
* Kennedy Speech :



* Nixon Speech :
  1. **Plot the word cloud of each of the speeches of the variable. (after removing the stopwords)**

**Solution**:

* Roosevelt Speech Wordcloud :



* Kennedy Speech Wordcloud :



* Nixon Speech Wordcloud :

**Thank you**