|  |  |
| --- | --- |
| A picture of a winding road and trees  MACHINE LEARNING  Project | **Created by-**  **SHUBHASREE SARKAR** |

**INDEX**

|  |  |  |
| --- | --- | --- |
| **Sr No** | **Topic** | **Page No** |
| **1** | **Technique: Machine Learning Techniques**  **Topic:**  **Election Data** | **4-63** |
| **2** | **Technique: Text Analysis**  **Topic:** | **64-68** |

***Topic –* Election Data**

**List of Figures**

|  |  |
| --- | --- |
| **Description** | **Page no.** |
| Check for*Distribution plots (histogram) - Figure 1* | **11** |
| *Check for Box plots (without outlier treatment) for the continuous columns- Figure 2* | **11** |
| *Check for Bar plots for the categorical columns- Figure 3* | **12** |
| *Check Bar plots for the categorical columns w.r.t. Vote- Figure 4* | **14** |
| *Check Bar plots for the categorical columns w.r.t. economic.cond.national – Figure 5* | **15** |
| *Check Bar plots for the categorical columns w.r.t. economic.cond.Household– Figure 6* | **16** |
| *Check Bar plots for the categorical columns w.r.t. Blair– Figure 7* | **17** |
| *Check Bar plots for the categorical columns w.r.t. Hague– Figure 8* | **18** |
| *Check Bar plots for the categorical columns w.r.t. Europe– Figure 9* | **19** |
| *Check Bar plots for the categorical columns w.r.t. political.knowledge– Figure 10* | **20** |
| *Check Bar plots for the categorical columns w.r.t. Gender– Figure 11* | **21** |
| *Check the Heat map with only continuous variables- Figure 12* | **22** |
| *Check with the pair plots for Bi variate data- Figure 13* | **23** |
| *Check the heatmap with only numerical variables after encoding- Figure 14* | **25** |
| *Confusion Matrix for Each Model- Figure 15* | **46-49** |
| *ROC -AUC Curve for each model – Figure 16* | **52-54** |
|  |  |

**List of Tables**

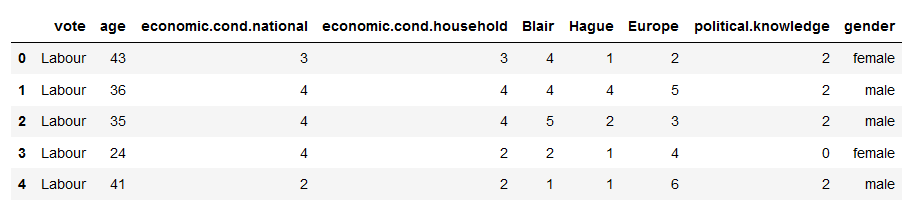
|  |  |
| --- | --- |
| **Description** | **Page no.** |
| *Head/Sample of the dataset – Table 1* | **5** |
| *info – Table 2.1,2.2* | **5** |
| *Check the datatypes of the dataset- Table 3* | **6** |
| *Check for Summary stats- Table 4* | **6** |
| *Check for Summary stats- Table 5* | **8** |
| *Check for the Null Values- Table 6* | **9** |
| *Check for the head Values after Label Encoding- Table 7* | **24** |
| *Check with the output of scaled data-Trained data- Table 8* | **27** |
| *Check with the output of scaled data-Test data- Table 9* | **28** |
| *Train and Test accuracies for each model – Table - 10* | **45-46** |
| *ROC -AUC Score for each model- Table -11* | **51-52** |
| *Train and Test Precision for each model- Table -12* | **56-57** |
| *Train and Test Recall for each model- Table -13* | **58-59** |
| *Train and Test F1 score for each model- Table -14* | **60-61** |
| *Train and Test scores for each model- Table -15* | **62** |

**Topic: Election Data**

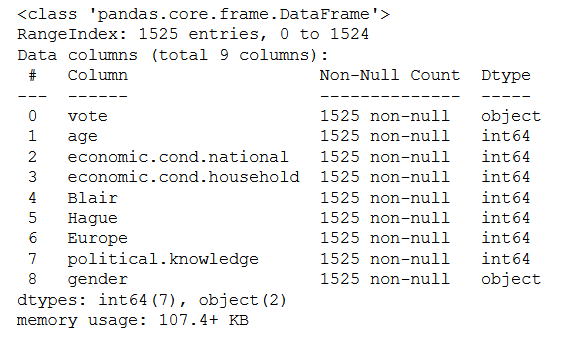
You are hired by one of the leading news channels CNBE who wants to analyse recent elections. This survey was conducted on 1525 voters with 9 variables. You have to build a model, to predict which party a voter will vote for on the basis of the given information, to create an exit poll that will help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

|  |  |
| --- | --- |
| **Variable Name** | **Description** |
| Vote | Party choice: Conservative or Labour |
| Age | Age of the individual voters in Years |
| Economic Condition  National | Assessment of current national economic conditions, 1 to 5. |
| Economic Condition  Household | Assessment of current household economic conditions, 1 to 5. |
| Blair | Assessment of the Labour leader, 1 to 5. |
| Hague | Assessment of the Conservative leader, 1 to 5. |
| Europe | An 11-point scale that measures respondents' attitudes toward European integration. High scores represent ‘Eurosceptic’ sentiment. |
| Political Knowledge | Knowledge of parties' positions on European integration, 0 to 3. |
| Gender | Female or Male. |

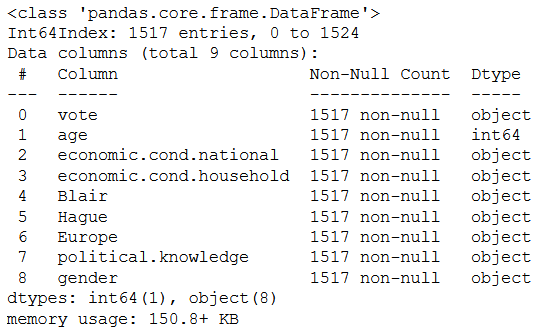
**1.1. Read the dataset. Describe the data briefly. Interpret the inferences for each. Initial steps like head() .info(), Data Types, etc . Null value check, Summary stats, Skewness must be discussed.**

*#Check for Head/Sample of the dataset – Table 1*

The above figure shows the output of the head function, showing top 5 records with the total of 9 variables or attributes

*****#Check for info – Table 2.1*

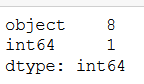
The above figure shows the different data types using the info function and here in this case we only have two data types, i.e int64 (7 variables) and object (2 variables)

*****#Check for info – Table 2.2*

From the data description, we have realised that actually all the int64 datatypes (numerical variables), except the ‘Age’, are actually categorical in nature with different levels which are numbered. For this reason, we have converted them into object datatypes

The above figure shows the different data types using the info function and here in this case we only have three data types, i.e int64(1 variables) and object(8 variables)

*#Check the datatypes of the dataset- Table 3*



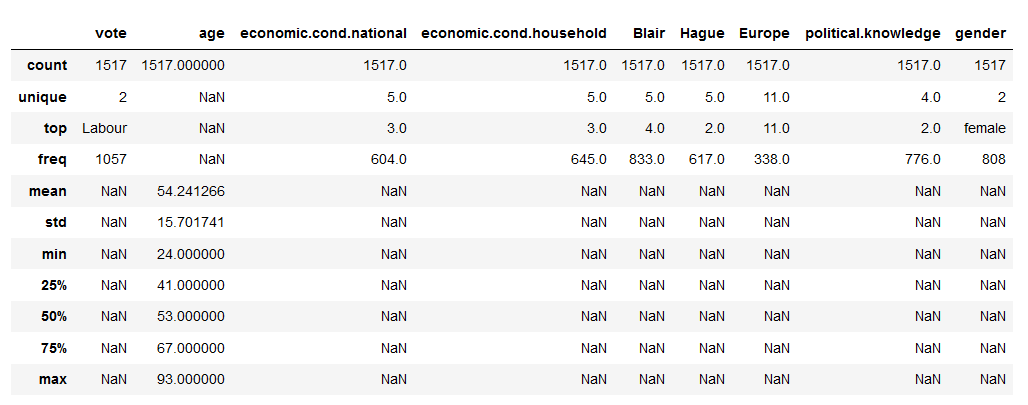
The different datatypes are shown in the table on the left.

Object type – 8

Numerical type - 1

*#Check the shape of the dataset*

The dimensions of the whole dataset are shown in the above figure

*#Check for Summary stats- Table 4*

The above figure shows the statistical measures associated with the continuous and categorical variables.

The description of the same is shown below-

***Age***

#### HereAgeis actually*Age of the individual voters.*

**Average** Ageis **54.182295 years** with **standard deviation** of **15.711209 years**

The Agerange is between **24 years (Minimum)** and **93 years (Maximum)**

The **median** of the Ageis **53 years**

Total count is **1525.**

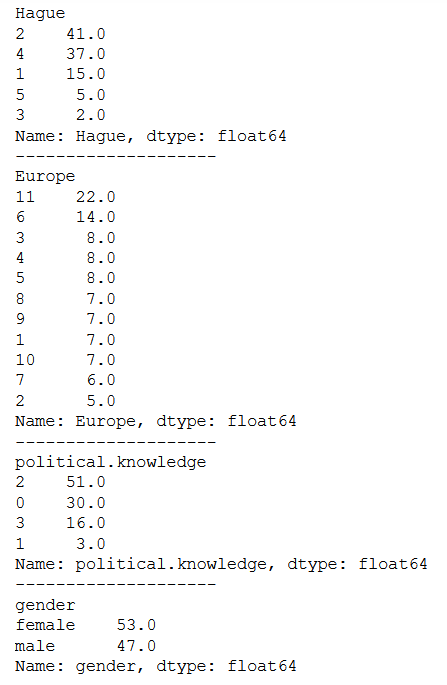
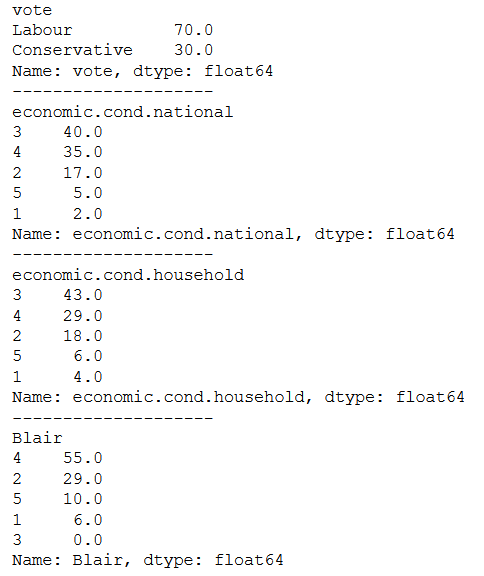
------------------------------------------------------------------------------------------------------------------------

The data description also shows the description of the categorical variables.

The detailed categories are shown below-

|  |
| --- |
| ***Economic Condition- National*** **Here,** theEconomic Conditionis **actually *Assessment of current national economic conditions, 1 to 5.***The levels are ordered: 1-lowest and 5-highest There are **5 unique** class for this variable.  **Highest count(40%) –3**  **Lowest count(2%)- 1** |
| ***Economic Condition- Household*** **Here,** theEconomic Conditionis **actually *Assessment of current household economic conditions, 1 to 5.*** The levels are ordered: 1-lowest and 5-highest  There are **5 unique** class for this variable.  **Highest count(43%) – 3**  **Lowest count(4%)- 1**  **Best color type has the lowest count.** |
| ***Blair*** **Here,** theBlairis **actually *Assessment of the Labour leader, 1 to 5.***The levels are ordered from low to high There are **5 unique** class for this variable.  **Highest count (55%) –** 4  **Lowest count (0%)- 3** |
| ***Hague*** **Here,** TheHagueis **actually *Assessment of the Conservative leader, 1 to 5.***The levels are ordered from low to high There are **5 unique** class for this variable.  **Highest count (41%) – 2**  **Lowest count (2%) - 3** |
| ***Europe*****Here,** theEurope is **actually *an 11-point scale that measures respondents' attitudes toward European integration. High scores represent ‘Eurosceptic’ sentiment.***The levels are ordered from low to high There are **11 unique** class for this variable.  **Highest count (22%) –** **11**  **Lowest count (5%)- 2** |
| ***Political Knowledge*****Here,** thePolitical Knowledge is **actually *Knowledge of parties' positions on European integration, 0 to 3.***The levels are ordered from low to high There are **4 unique** class for this variable.  **Highest count (51%) –** **2** **Lowest count (3%) - 1** |
| ***Vote*****Here,** thePolitical Knowledge is **actually *Party choice: Conservative or Labour*** There are **2 unique** class for this variable.  **Highest count (70%) –** **Labour** **Lowest count (30%)- Conservative** |
| ***Gender*****Here,** theGender is **actually *the two types of gender of the individual voters*** There are **2 unique** class for this variable.  **Highest count (53%) –** **Female** **Lowest count (47%) - Male** |

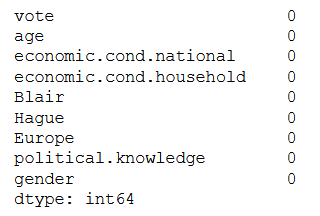
*Check for Summary stats- Table 5*



------------------------------------------------------------------------------------------------------------------------

***Null Values***

*Check for the Null Values- Table 6*

****

The missing values or ‘NaN’ or ‘NA’ values are in general to be cleaned during the data cleaning process.

They are basically –

1. Removed
2. Replaced with mode function (Categorical variables)
3. Replaced with Median function (Continuous variables)
4. Replace with other processes

In this dataset, there is **no missing values** present

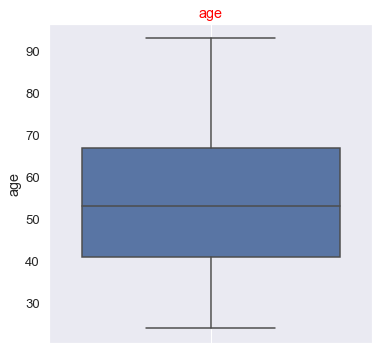
***Duplicated Values***

****

***Outliers Values-***

**Outliers are basically the extreme values in the dataset.** Outliers increase the variability in your data, which decreases statistical power.

**This will be explained in the Boxplot section after, where no outliers can be identified.**



**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**1.2) Perform EDA (Check the null values, Data types, shape, Univariate, bivariate analysis). Also check for outliers (4 pts). Interpret the inferences for each (3 pts) Distribution plots (histogram) or similar plots for the continuous columns. Box plots, Correlation plots. Appropriate plots for categorical variables. Inferences on each plot. Outliers’ proportion should be discussed, and inferences from above used plots should be there. There is no restriction on how the learner wishes to implement this but the code should be able to represent the correct output and inferences should be logical and correct.**

**Null Values**

This has already been mentioned in the **last question 1.1 (Refer page 9, Table 6)**

There has been **no null or missing values** present in the dataset

**Data Types**

The details about the DataTypes in this dataset is being explained in the **last question 1.1(Refer – [page 6, Table 3 ] and [pages 7-9 ] and [page-9, Table5])**

**Shape**

This has already been mentioned in the **last question 1.1 (Refer page 7)**

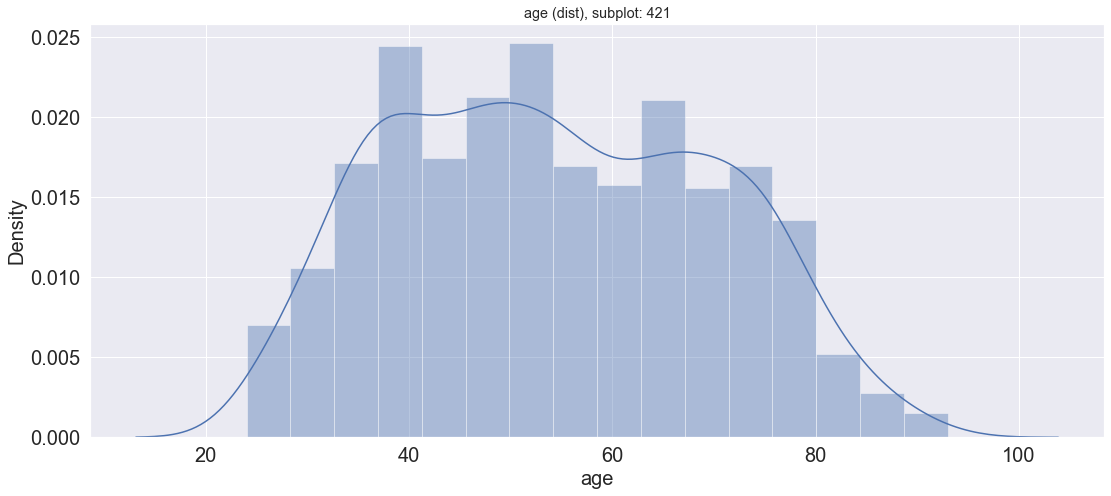
There has been **1525 rows and 9 columns** present in the dataset

--------------------------------------------------------------------------------------------------------------------------

**Univariate Analysis**

The term **univariate analysis**refers to the analysis of one variable. You can remember this because the prefix “uni” means “one.”

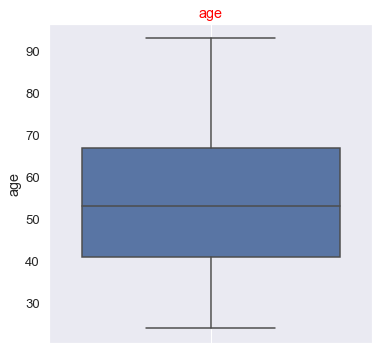
The purpose of univariate analysis is to understand the distribution of values for a single variable.

**#Check for*Distribution plots (histogram) - Figure 1*

**After plotting the histogram plots for the numerical variable ie., Age, the following can be concluded-**

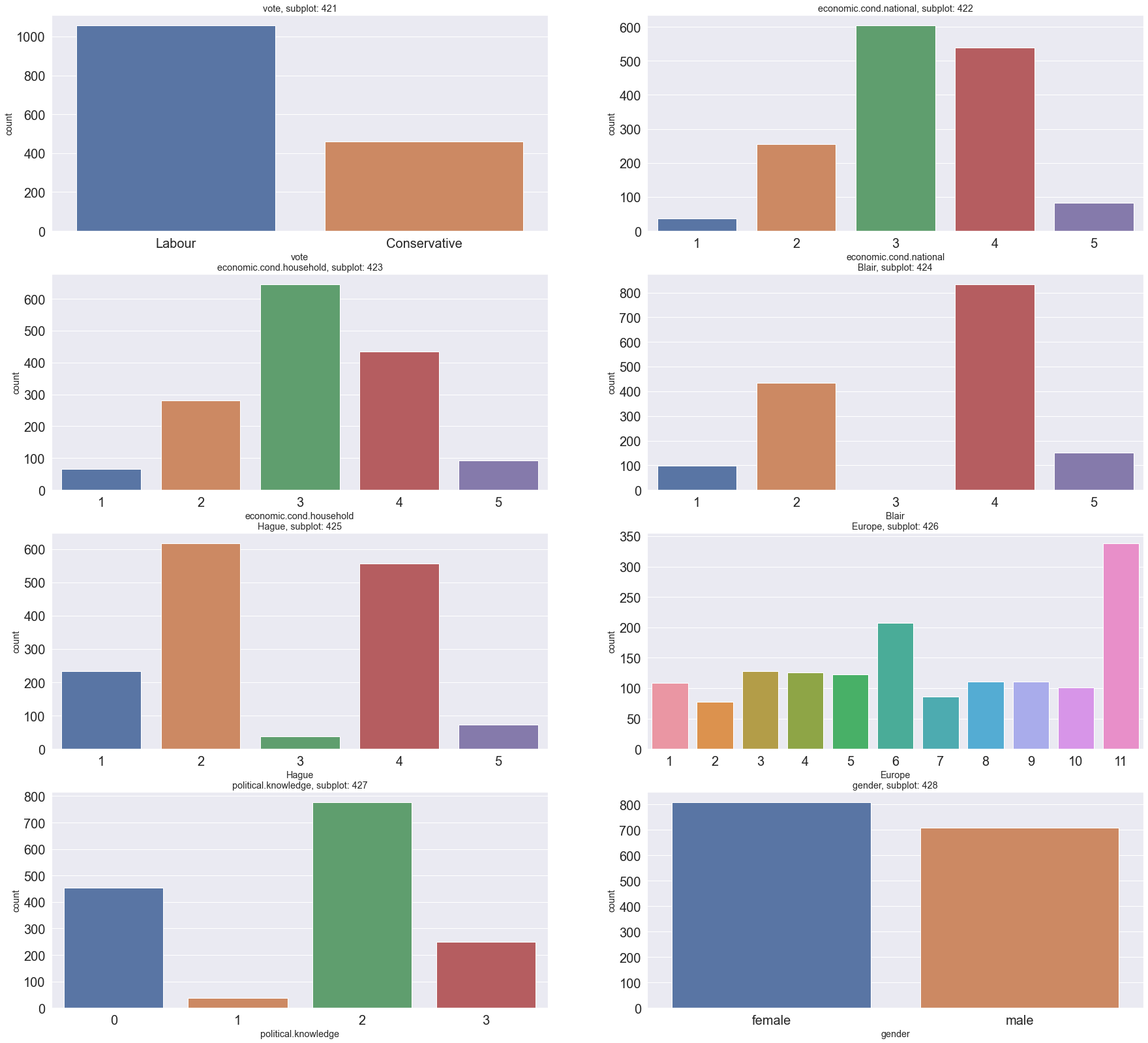
Age is not fully bell-shaped or normal in nature but shows multiple modes in the histogram plots. This is because, the range in terms of value is not too big, neither is the variance.

------------------------------------------------------------------------------------------------------------------------

*#Check for Box plots (without outlier treatment) for the continuous columns- Figure 2*

* Age dataset doesn’t contain any outlier, as the range is not too large considering the min and max value
* Age-Here Age has been a continuous variable type. But we could have used binning in order to convert this variable into a categorical datatype.
* This has been avoided, because of the process of scaling to be used in here.

--------------------------------------------------------------------------------------------------------------------------

*#Check for Bar plots for the categorical columns- Figure 3*

**The conclusions that can be drawn from the above bar plots are-**

* In the first bar graph, we can observe different categories of ‘Vote’ feature. Most no of Votes belonged to the Labour Party(70%) as compared to Conservative ones(30%)[It can be referenced from page 9)
* In the second bar graph, we can observe different categories of ‘Economies Conditions- National’ feature. Most frequent is the type 3, followed by type 4, with least frequency in the type 1.
* In the third bar graph, we can observe different categories of ‘Economies Conditions- Household’ feature. Most frequent is the type 3, followed by type 4, with least frequency in the type 1.
* In the fourth bar graph, we can observe different categories of ‘Blair’ feature which assess the Labour party Leader. Most frequent rating is the 4, followed by 2, with least frequency in the 3.
* In the fifth bar graph, we can observe different categories of ‘Hague’ feature which assess the Conservative party Leader. Most frequent rating is the 2, followed by 4, with least frequency in the 3.
* In the sixth bar graph, we can observe different categories of ‘Europe’ feature which measures respondents' attitudes toward European integration. Most frequent rating is the 11, followed by 6, with least frequency in the 2.

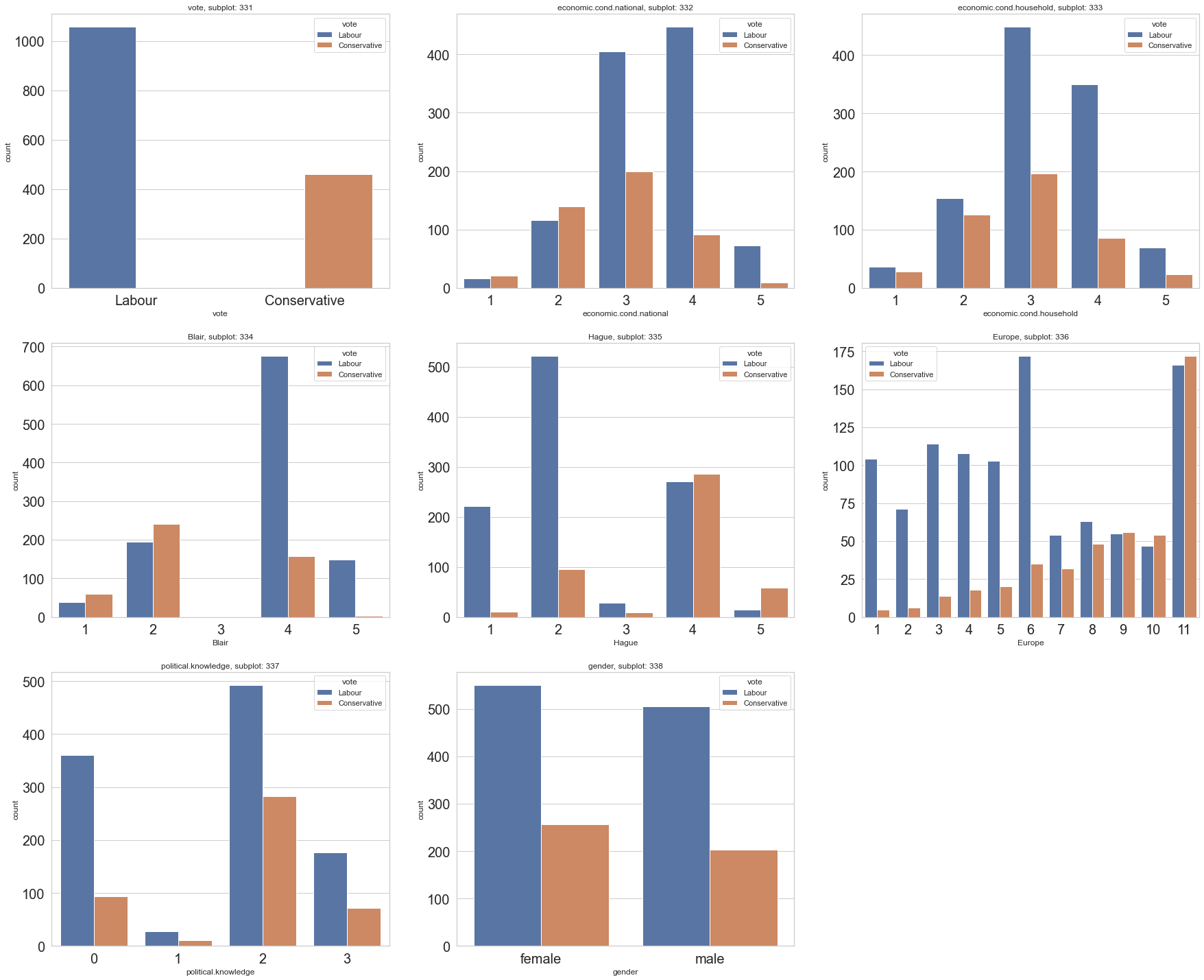
Thus, it shows that majority of the voters represent ‘Eurosceptic’ sentiment.

* In the seventh bar graph, we can observe different categories of ‘Political knowledge’ feature which assess the Knowledge of parties' positions on European integration. Most frequent rating is the 2, followed by 0, with least frequency in the 1.
* In the eighth bar graph, we can observe different categories of ‘Gender’ feature. Most no of Voters were Female(53%) as compared to Male(47%)[It can be referenced from page 9)

---------------------------------------------------------------------------------------------------------------------

**Bivariate Analysis**

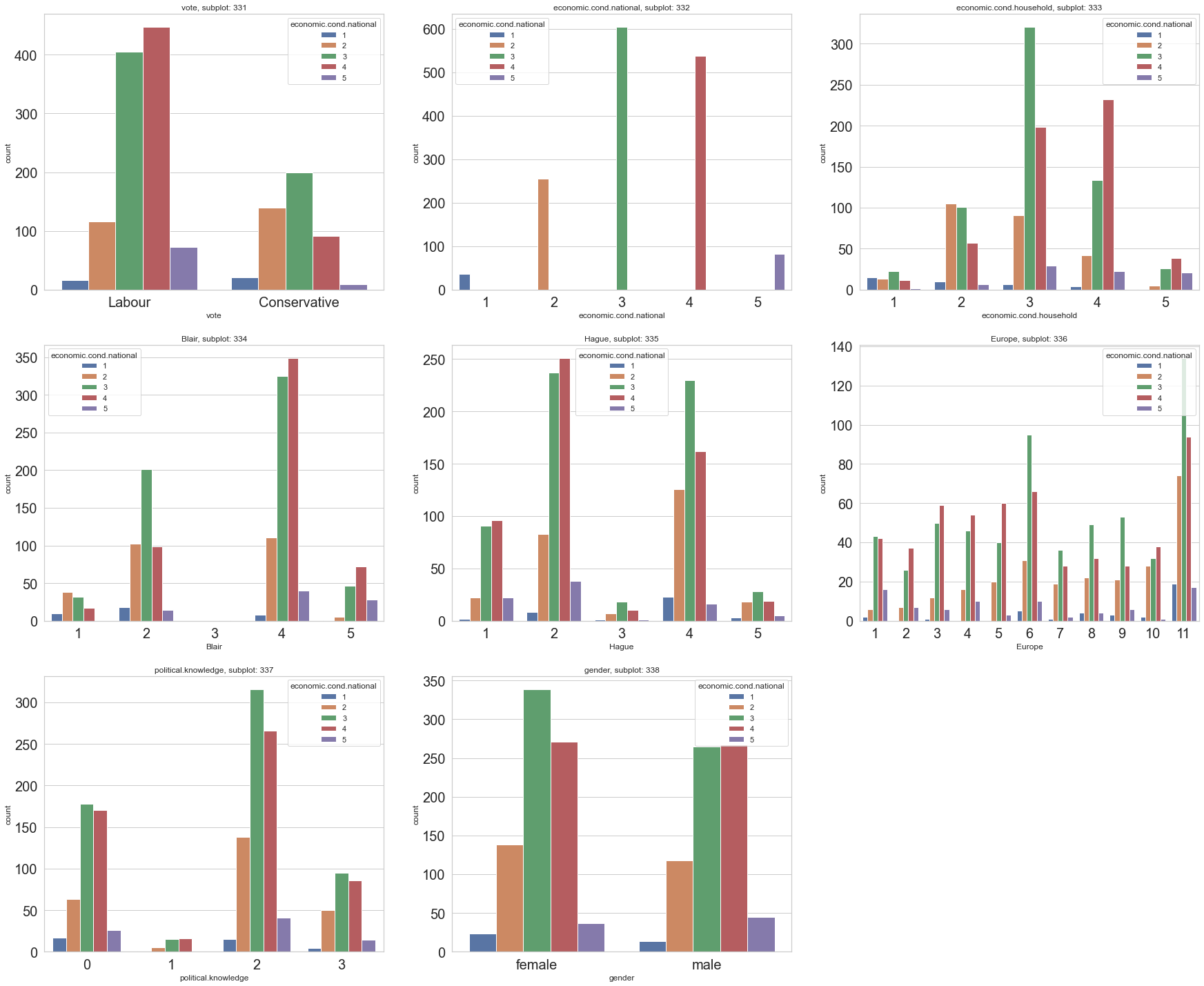
The term **bivariate analysis**involves the analysis of two variables, for the purpose of determining the empirical relationship between them. Bivariate analysis can be helpful in testing simple hypotheses of association.

*# Check Bar plots for the categorical columns w.r.t. Vote- Figure 4*

**The conclusions that can be drawn from the above bar plots are-**

* In the first bar graph, we can observe two types of Parties involved- Labour and Conservative.
* In the second graph and third graphs, it shows that for the rating 3,4 of national economic condition as well as household economic condition, labour party has most of the votes belonged to these categories.
* For the fourth and fifth graph, for Blair and Hague, votes are more for Labour party with 4th rating and labour party with 2nd rating
* In the sixth graph, for the Europe category, for 6th and 11th rating, most no of votes are for the Labour party, while in case of 11th rating, the votes have been the highest for Conservative party. So, we can observe an Eurosceptic sentiment among respondents from both the parties.
* In the seventh graph, we can see that highest no. of votes casted for Labour party for a Rating of 2 in terms of political knowledge.
* In the eighth graph we can observe for female and male- majority votes- Labour party

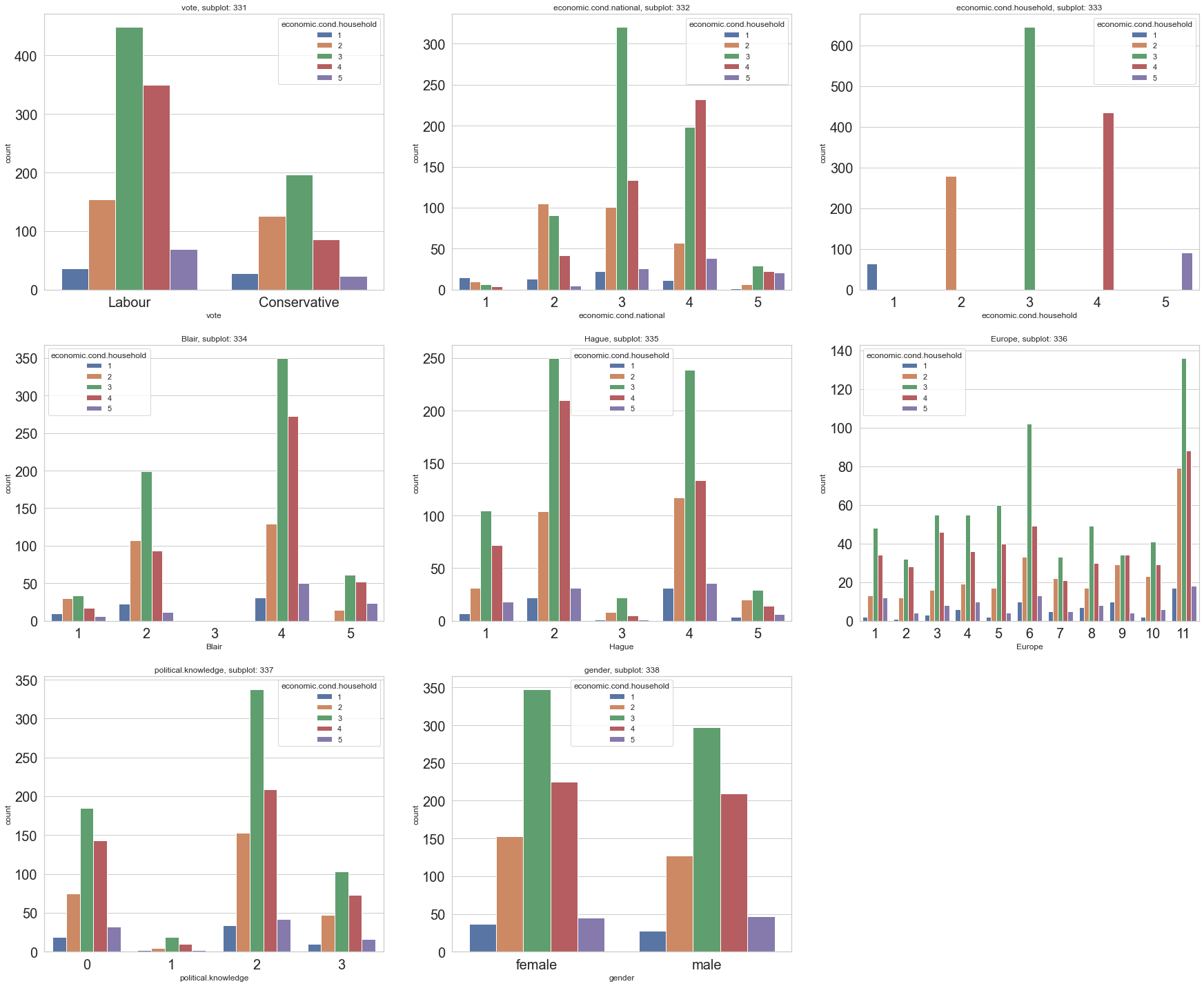
--------------------------------------------------------------------------------------------------------------------------

*# Check Bar plots for the categorical columns w.r.t. economic.cond.national – Figure 5*

**The conclusions that can be drawn from the above bar plots are-**

* In the first bar graph, we can observe that for Labour party, rating of economic national is 4th, while for Conservative it is 3rd.
* In the second graph and third graphs, it shows that the national economic condition shows a mediocre rating of 3, and also for economic condition of the households, most frequent is the Rating of 3
* For the fourth and fifth graph, for Blair and Hague, that for Blair, the majority in terms of the rating has been 4th, while for Hague, it is for the 2nd Rating.
* In the sixth graph, for the Europe category, for 11th rating, most no of votes are generated. So, we can observe a Eurosceptic sentiment among respondents from both the parties.
* In the seventh graph, we can see that highest no. of votes for Rating of 2 in terms of political knowledge.
* In the eighth graph we can observe for female and male- majority votes- 3rd rating

-------------------------------------------------------------------------------------------------------------------------

*# Check Bar plots for the categorical columns w.r.t. economic.cond.Household– Figure 6*

**The conclusions that can be drawn from the above bar plots are-**

* In the first bar graph, we can observe that for Labour party as well as for Conservative, rating of household economic condition is 3.
* In the second graph and third graphs, it shows that the national economic condition has a mediocre rating of 3, and also for economic condition of the households, most frequent Rating is also 3
* For the fourth and fifth graph, ie., for Blair, the majority in terms of the household economic condition rating has been 4 with economic cond. of 3, while for Hague, it is for the 2nd Rating for the same economic cond.
* In the sixth graph, for the Europe category, for 11th rating, most no of votes are generated. So, we can observe a Eurosceptic sentiment among respondents from both the parties.
* In the seventh graph, we can see that highest no. of votes for Rating of 2 in terms of political knowledge and within that 3rd rating in terms of household economic condition
* In the eighth graph we can observe for female and male- majority votes- 3rd rating for household economic cond.

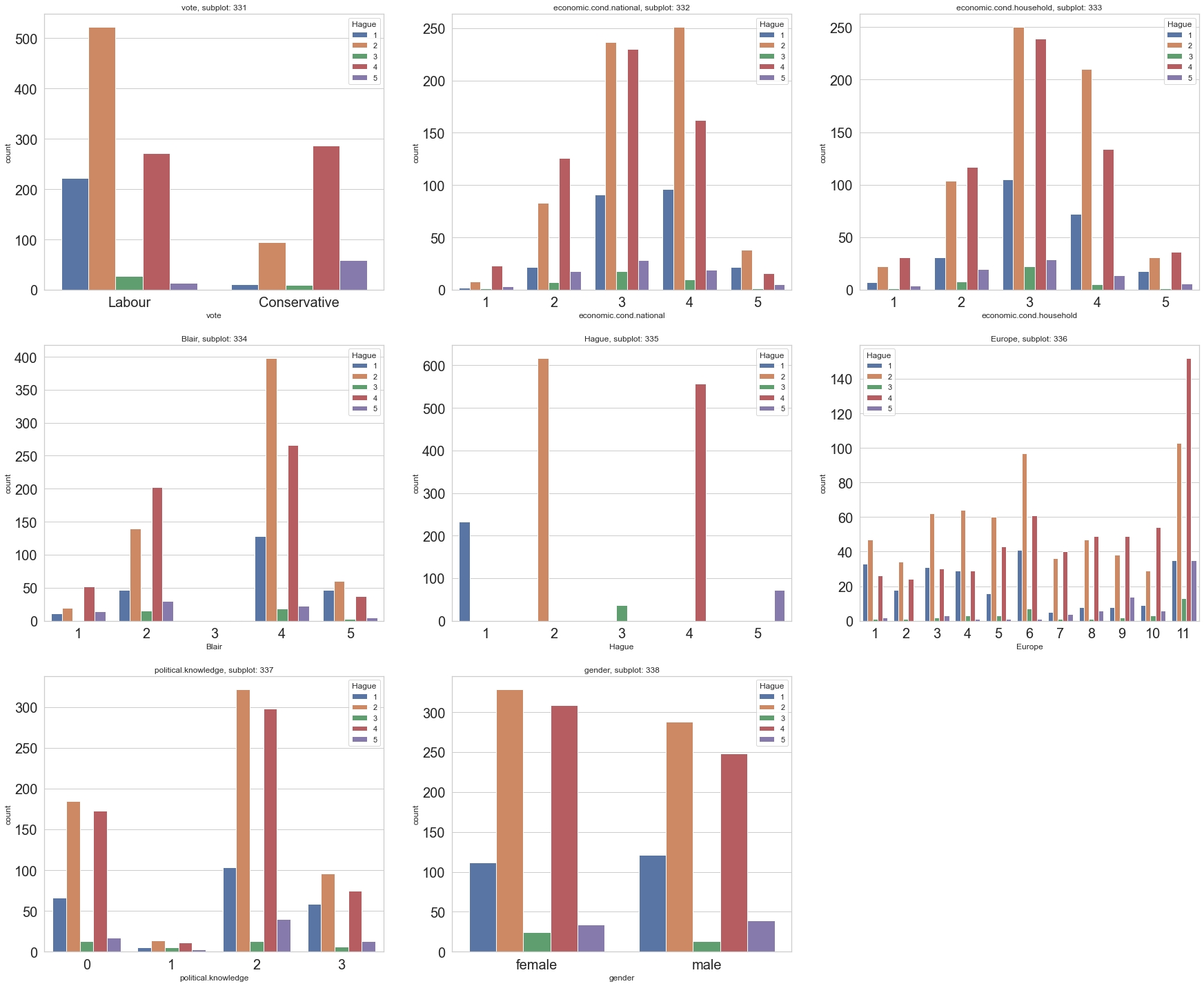
-------------------------------------------------------------------------------------------------------------------------

*# Check Bar plots for the categorical columns w.r.t. Blair– Figure 7*

**The conclusions that can be drawn from the above bar plots are-**

* In the first bar graph, we can observe that for Labour party majority rating is 4,while for Conservative one, the majority rating is 2
* In the second graph and third graphs, it shows most of the votes are for 3rd and 4th rating for either of the cases.
* For the fourth and fifth graph, ie., for Blair, the majority of the ratings has been 4 , while for Hague, it is for the 2nd Rating .
* In the sixth graph, for the Europe category, for 11th rating, most no of votes are generated. So, we can observe a Eurosceptic sentiment among respondents from both the parties.
* In the seventh graph, we can see that highest no. of votes for Rating of 2 in terms of political knowledge and within that 4th rating in terms of Blair
* In the eighth graph we can observe for female and male- majority votes- 4th rating for Blair

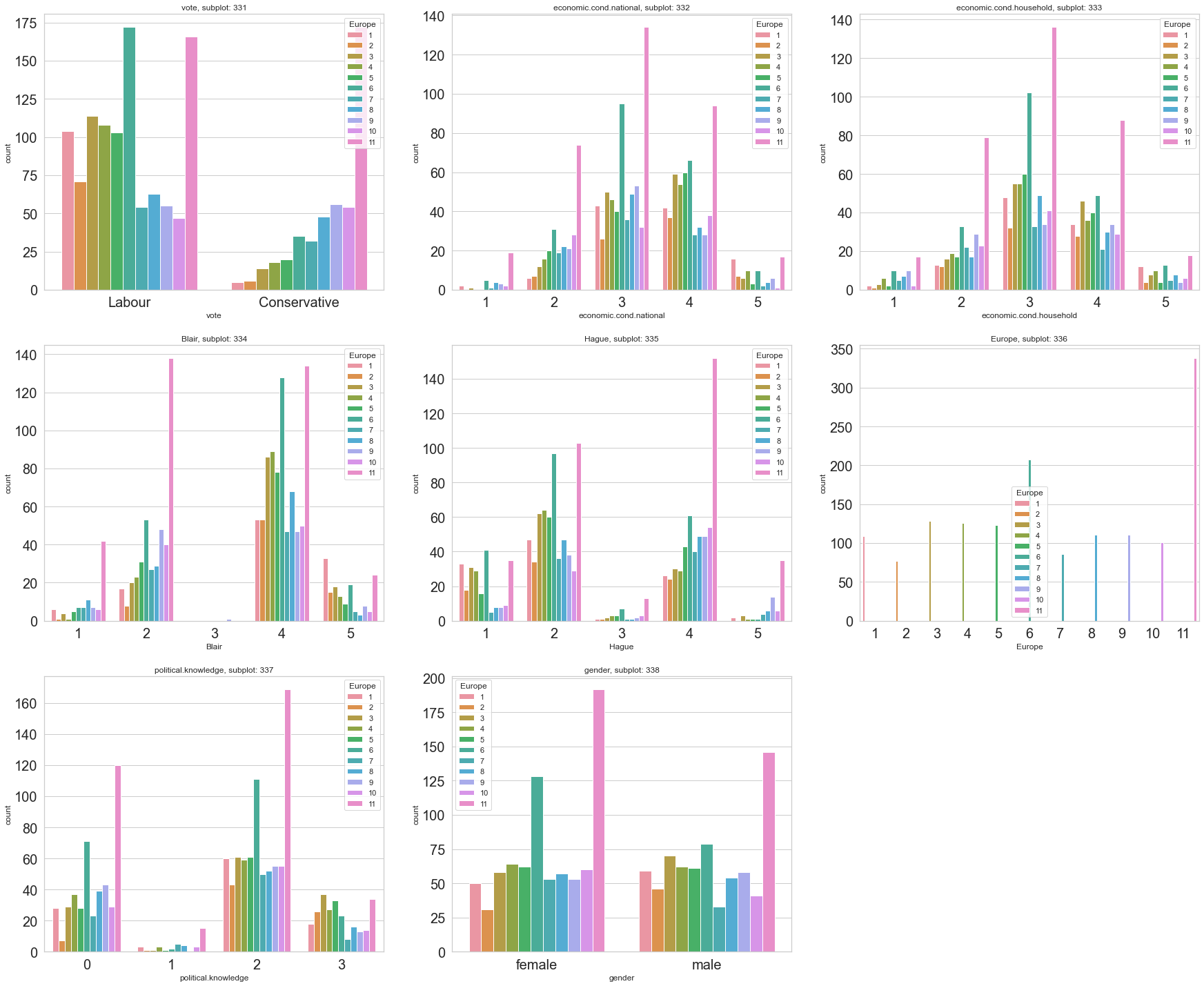
-------------------------------------------------------------------------------------------------------------------------

*# Check Bar plots for the categorical columns w.r.t. Hague– Figure 8*

**The conclusions that can be drawn from the above bar plots are-**

* In the first bar graph, we can observe that for Labour party majority rating is 2,while for Conservative one, the majority rating is 4
* In the second graph and third graphs, it shows most of the votes are for 3rd and 4th rating for either of the cases.
* For the fourth and fifth graph, ie, for Blair, the majority of the ratings has been 4 , while for Hague, it is for the 2nd Rating .
* In the sixth graph, for the Europe category, for 11th rating, most no of votes are generated. So, we can observe a Eurosceptic sentiment among respondents from both the parties.
* In the seventh graph, we can see that highest no. of votes for Rating of 2 in terms of political knowledge and within that, it is 2nd rating in terms of Hague
* In the eighth graph we can observe for female and male- majority votes- 2nd rating for Hague.

-------------------------------------------------------------------------------------------------------------------------

*# Check Bar plots for the categorical columns w.r.t. Europe– Figure 9*

**The conclusions that can be drawn from the above bar plots are-**

* The above subplots are too hard to explain as Europe has 11 level ratings associated.

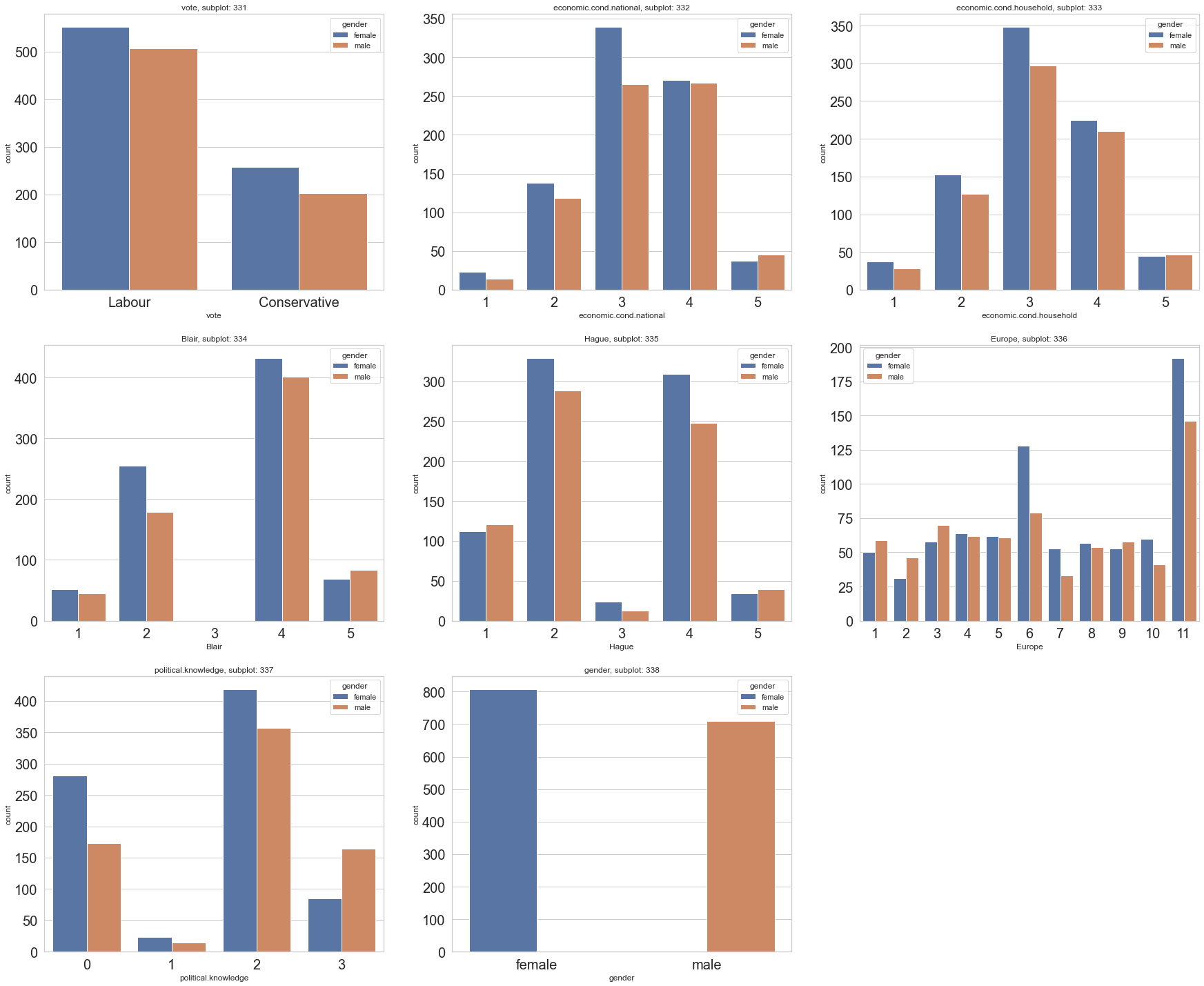
-------------------------------------------------------------------------------------------------------------------------

*# Check Bar plots for the categorical columns w.r.t. political.knowledge– Figure 10*

**The conclusions that can be drawn from the above bar plots are-**

* In the first bar graph, we can observe that for Labour party as well as Conservative party majority rating is 2.
* In the second graph and third graphs, it shows most of the votes are for 3rd and 4th rating for either of the cases, with sub rating of 2 for political knowledge in both.
* For the fourth and fifth graph, ie, for Blair, the majority of the ratings has been 4, while for Hague, it is for the 2nd Rating, while the sub rating of 2 for political knowledge in both.
* In the sixth graph, for the Europe category, for 11th rating, most no of votes are generated. So, we can observe a Eurosceptic sentiment among respondents from both the parties.
* In the seventh graph, we can see that highest no. of votes for Rating of 2 in terms of political knowledge
* In the eighth graph we can observe for female and male- majority votes- 2nd rating in terms of political knowledge.

-------------------------------------------------------------------------------------------------------------------------

*# Check Bar plots for the categorical columns w.r.t. Gender– Figure 11*

**The conclusions that can be drawn from the above bar plots are-**

* In the first bar graph, we can observe for two types of Parties involved- Labour and Conservative- majority are female.
* In the second graph and third graphs, it shows that for the rating 3,4 of national economic condition as well as household economic condition, majority are female
* For the fourth and fifth graph, for Blair and Hague as well, the majority are contributed by female.
* In the sixth graph, for the Europe category, for 11th rating, the contribution has been the highest for female category. So, we can observe a Eurosceptic sentiment among respondents from both the types.
* In the seventh graph, we can see that highest no. of votes by female category for a Rating of 2 in terms of political knowledge.
* In the eighth graph we can observe for female category has more individuals w.r.t. males.

--------------------------------------------------------------------------------------------------------------------------

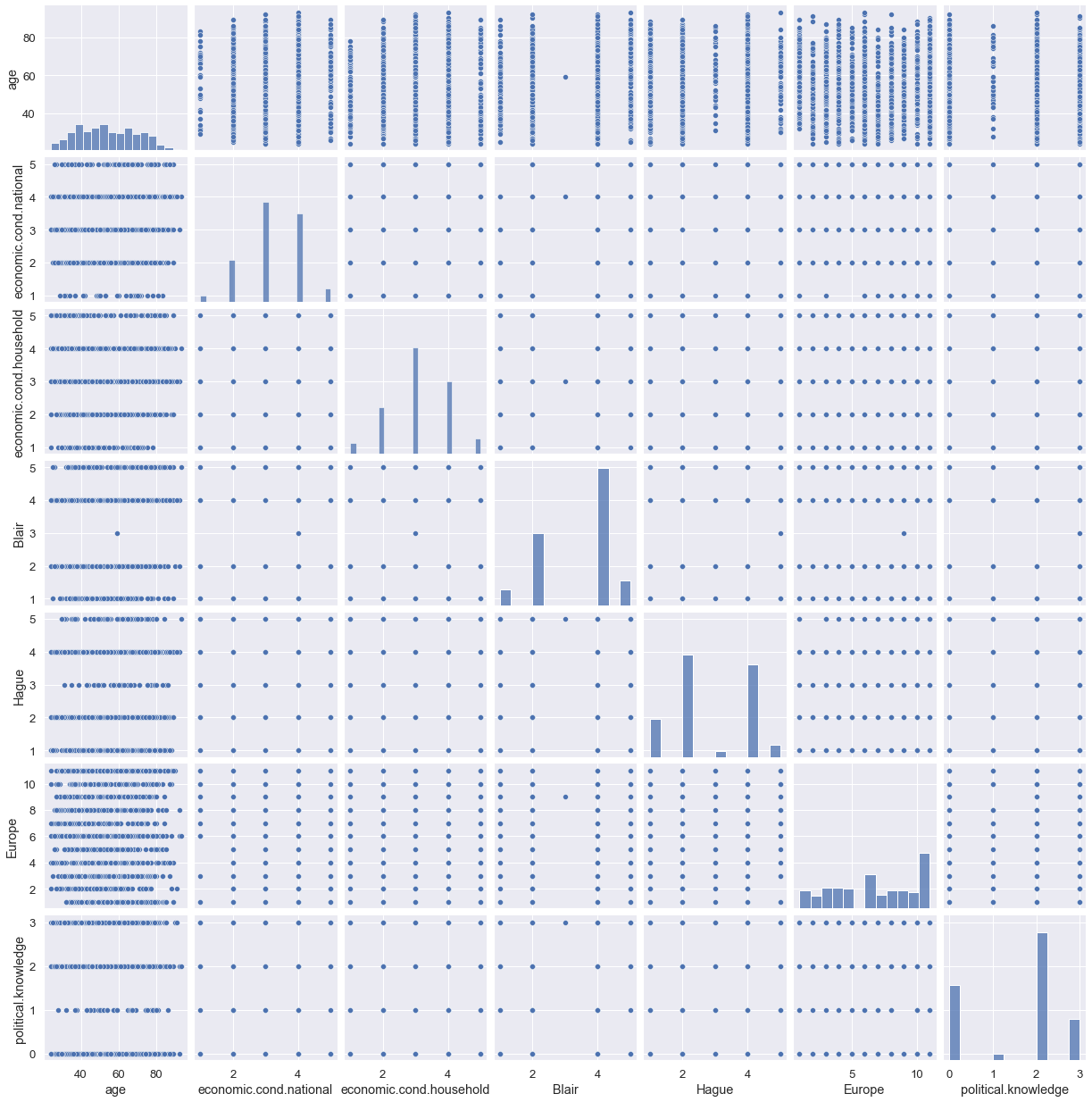
*# Check the Heat map with only continuous variables- Figure 12*

**Heatmap or Corrleation plot** is basically being used to evaluate the **relationship** between the **different numeric variables within a dataset**

**From the above Correlation plots using Heatmap, the following facts can be concluded-**

* We can observe very little correlation between the independent variables.
* We can observe moderate correlation between economic.cond.national w.r.t. economic.cond.household and Blair.

**-------------------------------------------------------------------------------------------------------------------------**

*****# Check with the pair plots for Bi variate data- Figure 13*

**With the help of the pairplot , we can understand all the univariate and bivariate trend of the datapoints/ variables in the dataset**

**Outlier**

This has already been mentioned in this **question (Refer page 11, Figure 2)**

There has been no outlier in the dataset.

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**1.3) Encode the data (having string values) for Modelling. Is Scaling necessary here or not?( 2 pts), Data Split: Split the data into train and test (70:30) (2 pts). The learner is expected to check and comment about the difference in scale of different features on the bases of appropriate measure for example std dev, variance, etc. Should justify whether there is a necessity for scaling. Object data should be converted into categorical/numerical data to fit in the models. (pd.categorical().codes(), pd.get\_dummies(drop\_first=True)) Data split, ratio defined for the split, train-test split should be discussed.**

**Encoding**

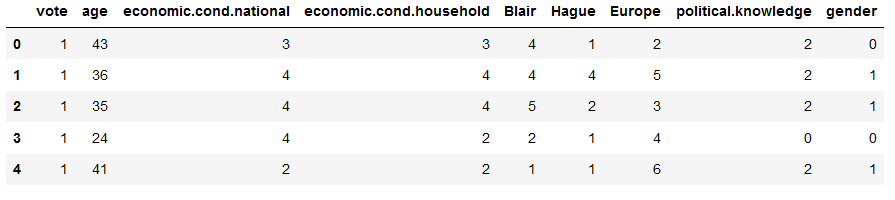
**Here in this dataset, we have come across two different types of categorical data**

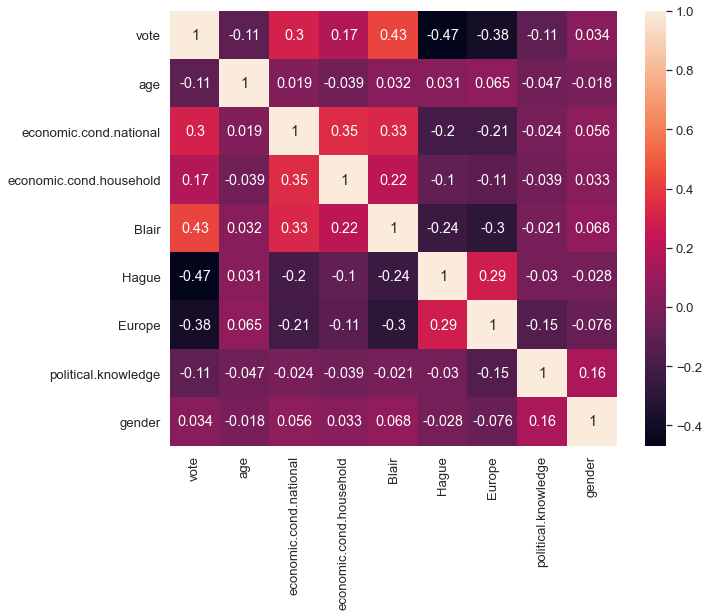
**First type**

* **The columns such as** economic.cond.national, economic.cond.household, Blair, Hague, Europe, political.knowledge- have discrete type of labelled data in numerical terms
* These are the **ordinal data** in this dataset.
* We first converted them from Numerical type (int64) to Object type(object) before carrying out with the EDA, as they were not actually numerical in literal terms but are basically the Rating categories
* Before splitting the data for model creation, we have further converted those object datatypes into int64 format for the ease in the process as required.

**Second Type**

* **The columns include Vote and Gender**
* **We have used pd.get\_dummies(drop\_first=True))** to convert them into **numerical types (int8)**
* These two datatypes are basically binary classified.

*Check for the head Values after Label Encoding- Table 7*

*****# Check the heatmap with only numerical variables after encoding- Figure 14*

**Conclusions**

**The Heatmap of the correlation for the dataset after manual encoding of the categorical values is shown in the figure**

**The target variable “Vote”, shows low to moderate correlation with political.knowledge, age, economic.cond.household, economic.cond.national, Europe, Blair, Hague and least with Gender**

**It also shows Positive correlation with economic.cond.household, economic.cond.national, Blair, Gender.**

**It also shows Negative correlation with political.knowledge, age, Hague, Europe.**

**The correlation between the independent variables are low to very moderate in nature and hence not that much significant.**

**------------------------------------------------------------------------------------------------------------------------**

**Shape of the dataset w.r.t. feature and target**

**Before splitting into train and test, we have first separated the X- feature based dataset and Y-target based dataset**



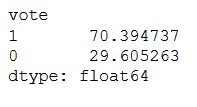
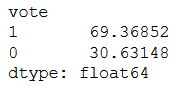
**After this the model is being split into train and test dataset.**

After the above steps, the data is being split into train and test using the train\_test\_split function by a ratio of 70:30



Train Data Class

Test Data Class



**Scaling**

Feature scaling in machine learning is one of the most critical steps during the pre-processing of data before creating a machine learning model. Scaling can make a difference between a weak machine learning model and a better one.

Machine learning algorithm just sees number — if there is a vast difference in the range say few ranging in thousands and few ranging in the tens, and it makes the underlying assumption that higher ranging numbers have superiority of some sort. So these more significant number starts playing a more decisive role while training the model.

### **Gradient Descent Based Algorithms**

**Machine learning algorithms like linear regression, logistic regression, neural network, etc. that use gradient descent as an optimization technique require data to be scaled.**

### **Distance-Based Algorithms**

Distance algorithms like **KNN, K-means, and SVM** are **most affected** by the range of features. This is because behind the scenes **they are using distances between data points to determine their similarity.**

### **Tree-Based Algorithms**

Tree-based algorithms, on the other hand, are fairly **insensitive** to the scale of the features.

* Algorithms like **Linear Discriminant Analysis (LDA), Naive Bayes is** by design equipped to handle this and give weights to the features accordingly. Performing features scaling in these algorithms may not have much effect.
* Scaling is critical while performing**Principal Component Analysis (PCA)**. PCA tries to get the features with maximum variance, and the variance is high for high magnitude features and skews the PCA towards high magnitude features.

**Type of Scaling Technique used** -

We will scale the data based on Z-Score method or Standard Scalar in SkLearn

• Standard Scalar standardizes a feature by subtracting the mean and then scaling to unit variance. Unit variance means dividing all the values by the standard deviation.

• Standard Scalar results in a distribution with a standard deviation equal to 1. The variance is equal to 1 also, because variance = standard deviation squared. And 1 squared = 1.

• Standard Scalar makes the mean of the distribution 0. About 68% of the values will lie be between -1 and 1.

Standard Scalar normalizes the data using the formula (x-mean)/standard deviation.

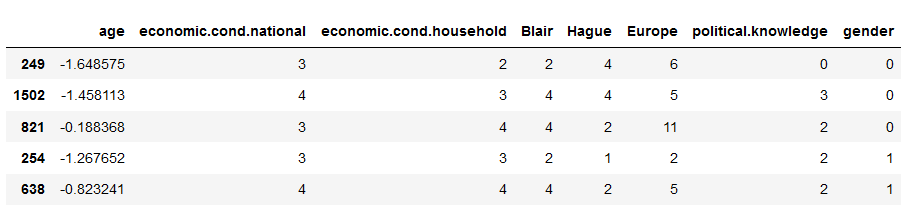
**Z = (value -mean)/standard deviation**

**Why it is to be used?**

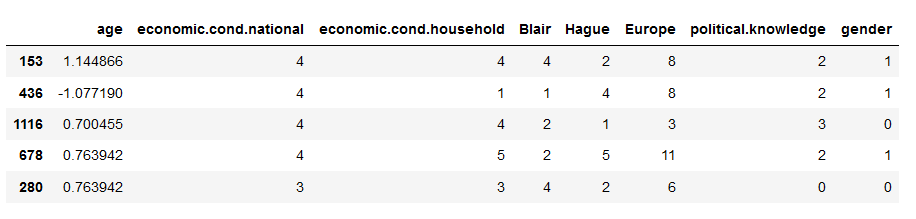
### If your variables are of incomparable units (e.g. height in cm and weight in kg) then you should standardize variables, of course. Even if variables are of the same units but show quite different variances it is still a good idea to standardize **Gradient Descent Based Algorithms** and **Distance-Based Algorithms.**

**Explanation with respect to the Variables given**

As per the data provided, standard scaling is required as all the different variables are provided in different units, for example, Age is a continuous variable in years. Rest all are categorical in nature. As there are differences in units, hence other values expressed in higher units will outweigh the variables in lower units and can give varied results. This is why scaling is important, and Standard scalar, as mentioned above, normalise the data points with mean 0 and standard deviation 1.

*#Check with the output of scaled data-Trained data- Table 8*

The above table shows the columns with scaled value for the Continuous data types in the trained dataset

*#Check with the output of scaled data-Test data- Table 9*

The above table shows the columns with scaled value for the Continuous data types in the test dataset

In this particular dataset, as can be observed from the above tables, we have used Standard scaler to scale the ‘Age’ column as it is the only continuous variable present with higher range in values.

--------------------------------------------------------------------------------------------------------------------------

**DATA CLASS IMBALANCE**

We can observe from Page 27, where the Train and Test data containing the Target Variable –Vote is shown with the Class percentage.

It can be concluded, that there has been high percentage of difference in data between the two classes in Votes, which has separated the dataset into Major and Minor classes based on the weightage.

**What is Class Imbalance?**

When observation in one class is higher than the observation in other classes then there exists a class imbalance.

Class Imbalance is a common problem in machine learning, especially in classification problems. Imbalance data can hamper our model accuracy big time.

## **The Problem with Class Imbalance**

Most machine learning algorithms work best when the number of samples in each class are about equal. This is because most algorithms are designed to maximize accuracy and reduce errors.

However, if the data set in imbalance then In such cases, you get a pretty high accuracy just by predicting the **majority class**, but you fail to capture the **minority class**, which is most often the point of creating the model in the first place

**Type of Ways to Handle Class Imbalance**

## Random Under-Sampling

## Random Over-Sampling

## Synthetic Minority Oversampling Technique (SMOTE)

Here in this problem we have used SMOTE to reduce Class Imbalance

## **SYNTHETIC MINORITY OVERSAMPLING TECHNIQUE (SMOTE)**

This technique generates synthetic data for the minority class.

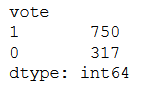
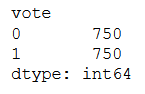
SMOTE (Synthetic Minority Oversampling Technique) works by randomly picking a point from the minority class and computing the k-nearest neighbours for this point.

The **synthetic points are added** between the chosen point and its neighbours.

After applying SMOTE, the train data set is now balanced.

Before SMOTE

After SMOTE



\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**1.4) Apply Logistic Regression and LDA (Linear Discriminant Analysis) (2 pts). Interpret the inferences of both models (2 pts). Successful implementation of each model. Logical reason behind the selection of different values for the parameters involved in each model. Calculate Train and Test Accuracies for each model. Comment on the validness of models (over fitting or under fitting)**

**LOGISTIC REGRESSION**

Logistic regression is a classification algorithm used to assign observations to a discrete set of classes. Some of the examples of classification problems are Email spam or not spam, online transactions Fraud or not Fraud

Logistic regression transforms its output using the logistic sigmoid function to return a probability value.

Logistic Regression is thus a Machine Learning algorithm which is used for the classification problems, it is a predictive analysis algorithm and is based on the concept of probability.

## **What are the types of logistic regression**

* Binary (eg. Tumour Malignant or Benign)
* Multi-linear functions fails Class (eg. Cats, dogs or Sheep's)

**Pros of Logistic Regression**

* Logistic regression is easier to implement, interpret, and very efficient to train.
* It makes no assumptions about distributions of classes in feature space.
* It can easily extend to multiple classes (multinomial regression) and a natural probabilistic view of class predictions.
* It not only provides a measure of how appropriate a predictor (coefficient size)is, but also its direction of association (positive or negative).
* It is very fast at classifying unknown records.
* Good accuracy for many simple data sets and it performs well when the dataset is linearly separable.
* It can interpret model coefficients as indicators of feature importance.
* Logistic regression is less inclined to over-fitting but it can overfit in high dimensional datasets. One may consider Regularization (L1 and L2) techniques to avoid over-fitting in these scenarios.

**Cons of Logistic Regression**

* If the number of observations is lesser than the number of features, Logistic Regression should not be used, otherwise, it may lead to overfitting.
* It constructs linear boundaries.
* The major limitation of Logistic Regression is the assumption of linearity between the dependent variable and the independent variables.
* It can only be used to predict discrete functions. Hence, the dependent variable of Logistic Regression is bound to the discrete number set.
* Non-linear problems can’t be solved with logistic regression because it has a linear decision surface. Linearly separable data is rarely found in real-world scenarios.
* Logistic Regression requires average or no multicollinearity between independent variables.
* It is tough to obtain complex relationships using logistic regression. More powerful and compact algorithms such as Neural Networks can easily outperform this algorithm.
* In Linear Regression independent and dependent variables are related linearly. But Logistic Regression needs that independent variables are linearly related to the log odds

(log(p/(1-p)).

*#Steps involved*

Using the train dataset, we have created Logistic model and then further testing the same on the test dataset.

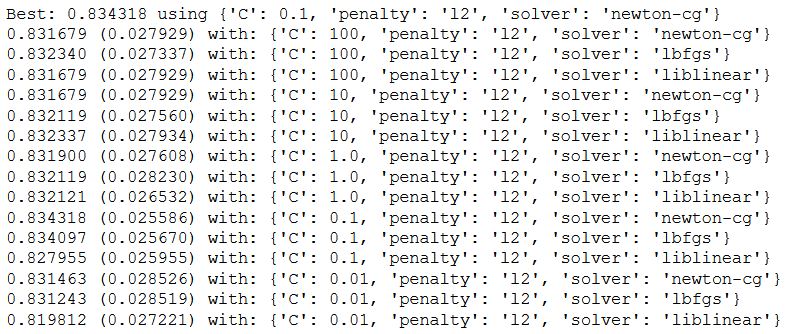
For the Sigmoid formation, we have imported the LogisticRegression module from the ‘sklearn’ package

With the help of afore mentioned package, Logistic regression model is created, in order to fit the training data into this model.

In the following step, we have used Grid Search CV method to evaluate the best parameters for the model in order to get a better performed Logistic Regression.

***Using GRID Search CV-***

GridSearchCV, can be briefly defined as a library function that is a member of sklearn's model\_selection package. **It helps to loop through predefined hyper parameters and fit your estimator (model) on your training set**. So, in the end, one can select the best parameters from the listed hyper parameters.



Using the above mentioned GridSearchCV package, we have identified the best parameters and an Optimised Logistic Regression model is being built after doing few iterations with the values we have received in each step. But even with this the model couldn’t generate better accuracy, precision and recall. We will discuss about the same in the next questions

**Note – Kindly refer the code file for the steps involved in the CART formation.**

***Logistic Regression with SMOTE data***

We have also tried to run the logistic regression with the balanced dataset (SMOTE based).

There has not been a change in hyper-parameters for the Grid Search CV when applied on the balanced data.

But the key metrics based performance changed.

We will compare the performances of the metrics’ later in question 1.7.

**Overfitting-** This are most prominent while using smote, although, overfitting issues are more or less there even in the base and tuned model.

--------------------------------------------------------------------------------------------------------------------------

**LDA**

**Linear Discriminant Analysis** or **Normal Discriminant Analysis** or **Discriminant Function Analysis** is a dimensionality reduction technique that is commonly used for supervised classification problems. It is used for modelling differences in groups i.e. separating two or more classes. It is used to project the features in higher dimension space into a lower dimension space.

For example, we have two classes and we need to separate them efficiently. Classes can have multiple features. Using only a single feature to classify them may result in some overlapping as shown in the below figure. So, we will keep on increasing the number of features for proper classification.

Two criteria are used by LDA to create a new axis:

* Maximize the distance between means of the two classes.
* Minimize the variation within each class.

**Extensions to LDA:**

1. **Quadratic Discriminant Analysis (QDA):** Each class uses its own estimate of variance (or covariance when there are multiple input variables).
2. **Flexible Discriminant Analysis (FDA):** Where non-linear combinations of inputs are used such as splines.
3. **Regularized Discriminant Analysis (RDA):** Introduces regularization into the estimate of the variance (actually covariance), moderating the influence of different variables on LDA.

**Pros of LDA**

* It is simple, fast and portable algorithm. It still beats some algorithms (logistic regression) when its assumptions are met.

**Cons of LDA**

* It requires normal distribution assumption on features/predictors.
* Sometimes not good for few categories variables.

*#Steps involved*

First, we have used the train dataset to create the Linear Discriminants and to test on the test dataset.

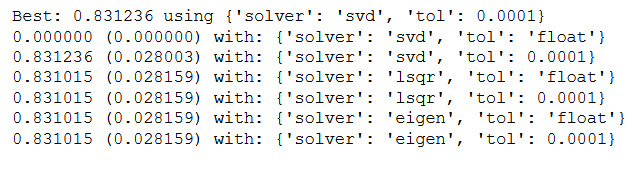
For the LDA formation, we have imported the LinearDiscriminantAnalysis module from the ‘sklearn’ package

With the help of afore mentioned package, LDA model is created, in order to fit the training data into this model.

A LDA model is thus being created which is being further used for model performance evaluation.

**Using *GRID Search CV-***

We know that by using GRID Search CV, one can select the best parameters from the listed hyper parameters.



Using the above mentioned GridSearchCV package, we have identified the best parameters and an Optimised Linear Discriminant Analysis model is being built after doing few iterations with the values we have received in each step. But even with this the model couldn’t generate better accuracy, precision and recall. We will discuss about the same in the next questions

**Note – Kindly refer the code file for the steps involved in the LDA formation.**

***LDA with SMOTE data***

We have also tried to run the LDA with the balanced dataset (SMOTE based).

We have also used Grid Search CV here.

But the key metrics based performance changed.

We will compare the performances of the metrics’ later in question 1.7.

The codes for the same is being attached with the Jupyter notebook (Machine Learning)

For the model evaluation and comparison, we have stated the same in the later half of the question, where model performances are being compared.

**Overfitting-** This are most prominent while using smote in most of the metrices, although, overfitting issues are more or less there even in the base and tuned model.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**1.5) Apply KNN Model and Naïve Bayes Model (2pts). Interpret the inferences of each model (2 pts). Successful implementation of each model. Logical reason behind the selection of different values for the parameters involved in each model. Calculate Train and Test Accuracies for each model. Comment on the validness of models (over fitting or under fitting)**

**KNN**

K Nearest Neighbour is a simple algorithm that stores all the available cases and classifies the new data or case based on a similarity measure. It is mostly used to classify a data point based on how its neighbours are classified.

‘k’ in KNN algorithm is based on feature similarity choosing the right value of K is a process called parameter tuning and is important for better accuracy. Finding the value of k is not easy.

Scaling is one of the important steps to be applied before approaching to create KNN models. The reason being **they are using distances between data points to determine their similarity, which is impacted if not scaling is being done**

# Few ideas on picking a value for ‘K’

1. There is no structured method to find the best value for “K”. We need to find out with various values by trial and error and assuming that training data is unknown.
2. Choosing smaller values for K can be noisy and will have a higher influence on the result.
3. Larger values of K will have smoother decision boundaries which mean lower variance but increased bias. Also, computationally expensive.
4. Another way to choose K is though cross-validation. One way to select the cross-validation dataset from the training dataset. Take the small portion from the training dataset and call it a validation dataset, and then use the same to evaluate different possible values of K. This way we are going to predict the label for every instance in the validation set using with K equals to 1, K equals to 2, K equals to 3 and then we look at what value of K gives us the best performance on the validation set and then we can take that value and use that as the final setting of our algorithm so we are minimizing the validation error.
5. In general, practice, choosing the value of **k** is **k = sqrt(N)** where **N** stands for the **number of samples in your training dataset**.
6. Try and keep the value of k odd in order to avoid confusion between two classes of data

# **Pros of KNN**

1. Simple to implement
2. Flexible to feature/distance choices
3. Naturally handles multi-class cases
4. Can do well in practice with enough representative data

# **Cons of KNN**

1. Need to determine the value of parameter K (number of nearest neighbours)
2. Computation cost is quite high because we need to compute the distance of each query instance to all training samples.
3. Storage of data
4. Must know we have a meaningful distance function.

*#Steps involved*

First, we have used the train dataset to create the KNN Classifier and to test on the test dataset.

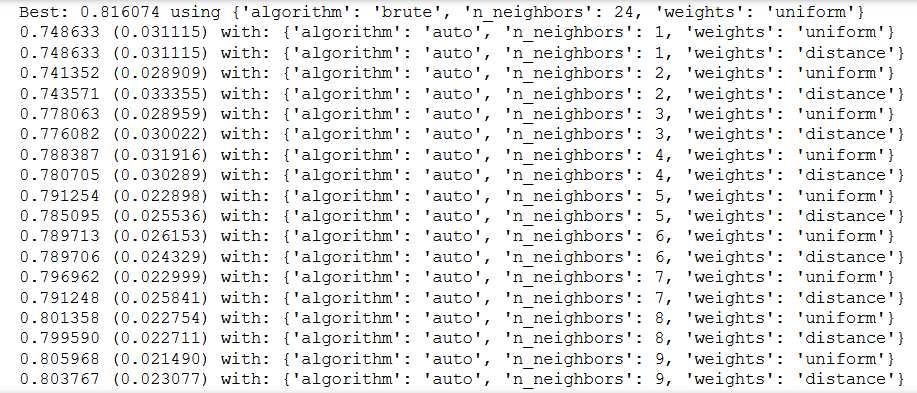
For the KNN formation, we have imported the KNeighborsClassifier module from the ‘sklearn’ package

With the help of afore mentioned package, KNN model is created, in order to fit the training data into this model.

A KNN model is thus being created which is being further used for model performance evaluation.

**Using *GRID Search CV-***

We know that by using GRID Search CV, one can select the best parameters from the listed hyper parameters.



Using the above mentioned GridSearchCV package, we have identified the best parameters and an Optimised Linear Discriminant Analysis model is being built after doing few iterations with the values we have received in each step. But even with this the model couldn’t generate best accuracy, precision and recall. We will discuss about the same in the next questions

**Note – Kindly refer the code file for the steps involved in the KNN formation.**

***KNN with SMOTE data***

We have also tried to run the KNN with the balanced dataset (SMOTE based).

We have also used Grid Search CV here.

But the key metrics based performance changed.

We will compare the performances of the metrics’ later in question 1.7.

The codes for the same is being attached with the Jupyter notebook (Machine Learning)

For the model evaluation and comparison, we have stated the same in the later half of the question, where model performances are being compared.

**Overfitting-** This are most prominent while using base model followed by smote in most of the metrices, although, overfitting issues is also marginally present in tuned model.

--------------------------------------------------------------------------------------------------------------------------

**Naïve Bayes**

Naive Bayes is a classification technique based on Bayes’ Theorem with an assumption of independence among predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature.

For example, a fruit may be considered to be an apple if it is red, round, and about 3 inches in diameter. Even if these features depend on each other or upon the existence of the other features, all of these properties independently contribute to the probability that this fruit is an apple and that is why it is known as ‘Naive’.

Naive Bayes model is easy to build and particularly useful for very large data sets. Along with simplicity, Naive Bayes is known to outperform even highly sophisticated classification methods.

# **Pros of Naïve Bayes**

* It is easy and fast to predict class of test data set. It also perform well in multi class prediction
* When assumption of independence holds, a Naive Bayes classifier performs better compare to other models like logistic regression and you need less training data.
* It perform well in case of categorical input variables compared to numerical variable(s). For numerical variable, normal distribution is assumed (bell curve, which is a strong assumption).

**Cons of Naïve Bayes:**

* If categorical variable has a category (in test data set), which was not observed in training data set, then model will assign a 0 (zero) probability and will be unable to make a prediction. This is often known as “Zero Frequency”. To solve this, we can use the smoothing technique. One of the simplest smoothing techniques is called Laplace estimation.
* On the other side Naive Bayes is also known as a bad estimator, so the probability outputs from predict\_proba are not to be taken too seriously.
* Another limitation of Naive Bayes is the assumption of independent predictors. In real life, it is almost impossible that we get a set of predictors which are completely independent.

*#Steps involved*

First, we have used the train dataset to create the Naïve Bayes Classifier and to test on the test dataset.

For the Naïve Bayes formation, we have imported the GaussianNB module from the ‘sklearn’ package

With the help of afore mentioned package, Naïve Bayes model is created, in order to fit the training data into this model.

A Naïve Bayes model is thus being created which is being further used for model performance evaluation.

**Using *GRID Search CV-***

We know that by using GRID Search CV, one can select the best parameters from the listed hyper parameters.

But due to very few no of parameters Grid Search CV is not being used in Naïve Bayes to tune the model

**Note – Kindly refer the code file for the steps involved in the Naïve Bayes formation.**

***Naïve Bayes with SMOTE data***

We have also tried to run the Naïve Bayes with the balanced dataset (SMOTE based).

But the key metrics based performance changed.

We will compare the performances of the metrics’ later in question 1.7.

The codes for the same is being attached with the Jupyter notebook (Machines Learning)

For the model evaluation and comparison, we have stated the same in the later half of the question, where model performances are being compared.

**Overfitting-** This are most prominent while using base model followed by smote in most of the metrices, although, overfitting issues is also marginally present in tuned model.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**1.6) Model Tuning (4 pts), Bagging (1.5 pts) and Boosting (1.5 pts). Apply grid search on each model (include all models) and make models on best\_params. Define a logic behind choosing particular values for different hyper-parameters for grid search. Compare and comment on performances of all. Comment on feature importance if applicable. Successful implementation of both algorithms along with inferences and comments on the model performances.**

**Model Tuning**

Tuning is usually a trial-and-error process by which you change some hyper parameters (for example, the number of trees in a tree-based algorithm or the value of alpha in a linear algorithm), run the algorithm on the data again, then compare its performance on your validation set in order to determine which set of hyper parameters results in the most accurate model.

All machine learning algorithms have a “default” set of hyperparameters, which Machine Learning Mastery defines as “a configuration that is external to the model and whose value cannot be estimated from data.” Different algorithms consist of different hyperparameters. For example, regularized regression models have coefficients penalties, decision trees have a set number of branches, and neural networks have a set number of layers. When building models, analysts and data scientists choose the default configuration of these hyperparameters after running the model on several datasets.

While the generic set of hyperparameters for each algorithm provides a starting point for analysis and will generally result in a well-performing model, it may not have the optimal configurations for your particular dataset and business problem. In order to find the best hyperparameters for your data, you need to tune them.

## **Why is Model Tuning Important?**

Model tuning allows you to customize your models so they generate the most accurate outcomes and give you highly valuable insights into your data, enabling you to make the most effective business decisions.

Model tuning means finding better parameters for the model and not just use the default values. For what parameters can be changed refer to the algorithm documentation.

Each ML algorithm has its own parameters to tune the model. Please check the grid search implementation in mentoring session notebooks. Ex: In random forest, two important hyperparameters are n\_estimators, max\_features.

**We have used GRID Search CV in order to find the best hyperparameters to tune the model**

We can observe changes in terms of model metrics performances with respect to the base model.

**We will also observe that the base models have fared well as compared to the tuned models**

**In few of the algorithms.**

**Bagging**

A Bagging classifier is an ensemble meta-estimator that fits base classifiers each on random subsets of the original dataset and then aggregate their individual predictions (either by voting or by averaging) to form a final prediction. Such a meta-estimator can typically be used as a way to reduce the variance of a black-box estimator (e.g., a decision tree), by introducing randomization into its construction procedure and then making an ensemble out of it.

Each base classifier is trained in parallel with a training set which is generated by randomly drawing, with replacement, N examples (or data) from the original training dataset – where N is the size of the original training set. Training set for each of the base classifiers is independent of each other. Many of the original data may be repeated in the resulting training set while others may be left out.

Bagging reduces overfitting (variance) by averaging or voting, however, this leads to an increase in bias, which is compensated by the reduction in variance though.

# **Pros of Bagging**

* Bagging method helps when we face variance or overfitting in the model. It provides an environment to deal with variance by using N learners of same size on same algorithm.
* During the sampling of train data, there are many observations which overlaps. So, the combination of these learners helps in overcoming the high variance.
* Bagging uses Bootstrap sampling method.

**Cons of Bagging**

* Bagging is not helpful in case of bias or under fitting in the data.
* Bagging ignores the value with the highest and the lowest result which may have a wide difference and provides an average result.

*#Steps involved*

First, we have used the train dataset to create the Bagging Classifier and to test on the test dataset.

For the Bagging formation, we have imported the BaggingClassifier module from the ‘sklearn’ package

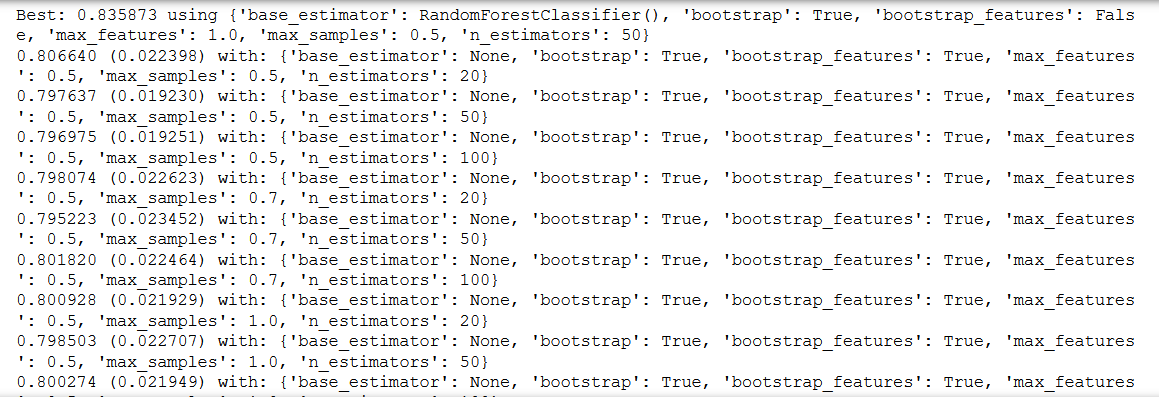
With the help of afore mentioned package, Bagging model is created, in order to fit the training data into this model.

A Bagging model is thus being created which is being further used for model performance evaluation.

**Using *GRID Search CV-***

For Bagging, base estimator would be required and bagging classifier should also be used here along with a random forest.

We know that by using GRID Search CV, one can select the best parameters from the listed hyper parameters.

****Note – Kindly refer the code file for the steps involved in the Bagging formation.**

***Bagging with SMOTE data***

We have also tried to run the Bagging with the balanced dataset (SMOTE based).

We also applied Grid Search on the model

But the key metrics based performance changed.

We will compare the performances of the metrics’ later in question 1.7.

The codes for the same is being attached with the Jupyter notebook (Machines Learning)

For the model evaluation and comparison, we have stated the same in the later half of the question, where model performances are being compared

**Overfitting-** This are most prominent while using base model followed by smote in most of the metrices, although, overfitting issues is also marginally present in tuned model.

--------------------------------------------------------------------------------------------------------------------------

**Random Forest**

Random forest is a **Supervised Machine Learning Algorithm** that is **used widely in Classification and Regression problems**. It builds decision trees on different samples and takes their majority vote for classification and average in case of regression.

One of the most important features of the Random Forest Algorithm is that it can handle the data set containing **continuous variables** as in the case of regression and **categorical variables** as in the case of classification. It performs better results for classification problems.

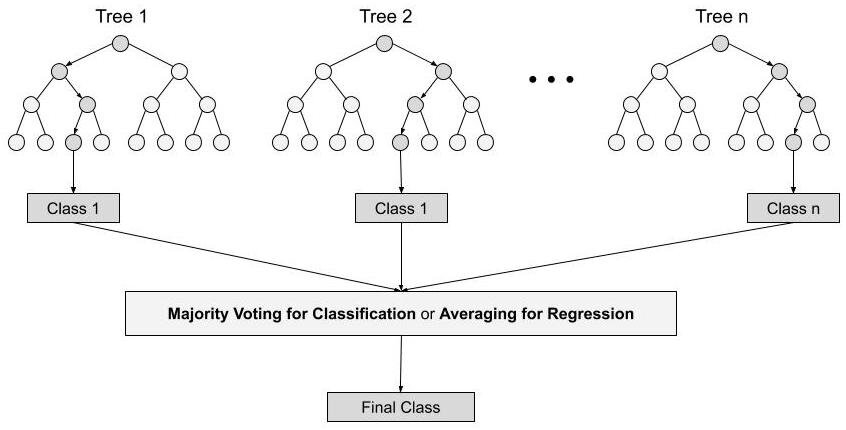
**Steps involved in random forest algorithm:**

Step 1: In Random forest n number of random records are taken from the data set having k number of records.

Step 2: Individual decision trees are constructed for each sample.

Step 3: Each decision tree will generate an output.

Step 4: Final output is considered based on ***Majority Voting or Averaging***for Classification and regression respectively.



# **Pros of Random Forest**

* Robust to outliers.
* Works well with non-linear data.
* Lower risk of overfitting.
* Runs efficiently on a large dataset.
* Better accuracy than other classification algorithms.

**Cons of Random Forest**

* Random forests are found to be biased while dealing with categorical variables.
* Slow Training.
* Not suitable for linear methods with a lot of sparse features

*#Steps involved*

First, we have used the train dataset to create the RandomForestClassifier and to test on the test dataset.

For the Random Forest formation, we have imported the RandomForestClassifier module from the ‘sklearn’ package

With the help of afore mentioned package, Random Forest model is created, in order to fit the training data into this model.

A final model is thus being created which is being further used for model performance evaluation.

**Using *GRID Search CV-***

For Random Forest, base estimator would be required to get a tuned model

We know that by using GRID Search CV, one can select the best parameters from the listed hyper parameters.

**Note – Kindly refer the code file for the steps involved in the Bagging formation.**

***Random Forest with SMOTE data***

We have also tried to run the Random Forest with the balanced dataset (SMOTE based).

We also applied Grid Search on the model

But the key metrics based performance changed.

We will compare the performances of the metrics’ later in question 1.7.

The codes for the same is being attached with the Jupyter notebook (Machines Learning)

For the model evaluation and comparison, we have stated the same in the later half of the question, where model performances are being compared

**Overfitting-** This are most prominent while using base model followed by smote in most of the metrices, although, overfitting issues is also marginally present in tuned model.

--------------------------------------------------------------------------------------------------------------------------

**Boosting**

**Boosting** is an ensemble modelling technique that attempts to build a strong classifier from the number of weak classifiers. Boosting combines the weak learners to form a strong learner, where a weak learner defines a classifier slightly correlated with the actual classification. In contrast to a weak learner, a strong learner is a classifier associated with the correct categories.

Types of Boosting to be used-

* AdaBoost
* GradientBoost

***AdaBoost Classifier***

An AdaBoost classifier is a meta-estimator that begins by fitting a classifier on the original dataset and then fits additional copies of the classifier on the same dataset but where the weights of incorrectly classified instances are adjusted such that subsequent classifiers focus more on difficult cases.

This class implements the algorithm known as AdaBoost-SAMME

**class sklearn.ensemble.AdaBoostClassifier(base\_estimator=None, \*, n\_estimators=50, learning\_rate=1.0, algorithm='SAMME.R', random\_state=None)**

***GradientBoost Classifier***

GB builds an additive model in a forward stage-wise fashion; it allows for the optimization of arbitrary differentiable loss functions. In each stage n\_classes regression trees are fit on the negative gradient of the binomial or multinomial deviance loss function. Binary classification is a special case where only a single regression tree is induced.

***class* *sklearn.ensemble.GradientBoostingClassifier(\*, loss='deviance', learning\_rate=0.1, n\_estimators=100, subsample=1.0, criterion='friedman\_mse', min\_samples\_split=2, min\_samples\_leaf=1, min\_weight\_fraction\_leaf=0.0, max\_depth=3, min\_impurity\_decrease=0.0, init=None, random\_state=None, max\_features=None, verbose=0, max\_leaf\_nodes=None, warm\_start=False, validation\_fraction=0.1, n\_iter\_no\_change=None, tol=0.0001, ccp\_alpha=0.0)***

# **Pros of Boosting**

* Boosting technique takes care of the weightage of the higher accuracy sample and lower accuracy sample and then gives the combined results.
* Net error is evaluated in each learning steps. It works well with interactions.
* Boosting technique helps when we are dealing with bias or under fitting in the data set.
* Multiple boosting techniques are available. For example: AdaBoost, LPBoost, XGBoost, GradientBoost, BrownBoost

**Cons of Boosting**

* Boosting technique often ignores overfitting or variance issues in the data set.
* It increases the complexity of the classification.
* Time and computation can be a bit expensive.

*#Steps involved*

First, we have used the train dataset to create the Boosting Classifier (ADA Boost, Gradient Boost) and to test on the test dataset.

For the Boosting formation, we have imported the AdaBoostClassifier and GradientBoostingClassifier modules from the ‘sklearn’ package

With the help of afore mentioned package, Boosting models are created, in order to fit the training data into the models.

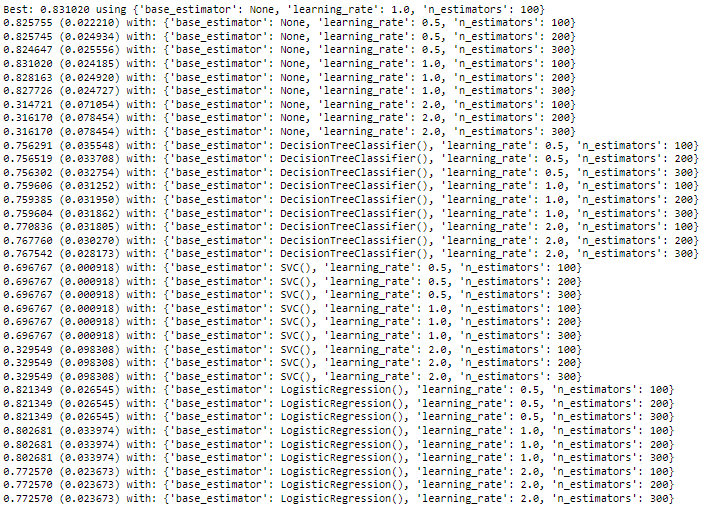
A Boosting models are thus being created which being further used for model performance evaluation.

We first applied ADABoost Classifier, followed by Gradient Boost Classifier

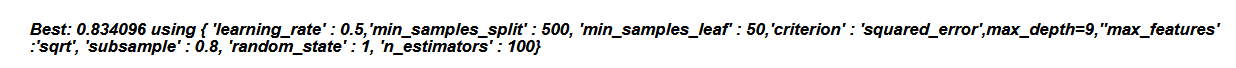
**Using *GRID Search CV-***

We know that by using GRID Search CV, one can select the best parameters from the listed hyper parameters.

**Note – Kindly refer the code file for the steps involved in the Boosting formation.**

***AdaBoost Classifier Grid Search Output***

***GradientBoost Classifier Grid Search Output***

****

***Boosting with SMOTE data***

We have also tried to run the both types of Boosting methods with the balanced dataset (SMOTE based).

We also applied Grid Search on the models

But the key metrics based performance changed.

We will compare the performances of the metrics’ later in question 1.7.

The codes for the same is being attached with the Jupyter notebook (Machines Learning)

For the model evaluation and comparison, we have stated the same in the later half of the question, where model performances are being compared.

**Overfitting-** This are most prominent while using base model followed by smote in most of the metrices, although, overfitting issues is also marginally present in tuned model.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**1.7 Performance Metrics: Check the performance of Predictions on Train and Test sets using Accuracy, Confusion Matrix, Plot ROC curve and get ROC\_AUC score for each model, classification report (4 pts) Final Model - Compare and comment on all models on the basis of the performance metrics in a structured tabular manner. Describe on which model is best/optimized, After comparison which model suits the best for the problem in hand on the basis of different measures. Comment on the final model.(3 pts)**

After doing all the necessary steps for model fitting and generating predictions, we came to the very outcome i.e, the part of model evaluation.

***ACCURACY***

It is a part of metrices derived from confusion matrix which is basically a **NxN** matrix, where **N** is the **number of classes to be predicted**

It is the **proportion** of the total number of **predictions** that were **correct**.

It is easily suited for **binary** as well as a **multiclass** **classification** **problem** which are **well** **balanced** and **not skewed** or **No class imbalance**.



|  |  |  |  |
| --- | --- | --- | --- |
| **Model Types** | **Categories** | **Train Data Accuracy** | **Test Data Accuracy** |
| **Logistic Regression(Grid Search)** | **Base Model** | **84%** | **82%** |
| **Grid Search CV** | **85%** | **81%** |
| **Smote** | **85%** | **80%** |
| **LDA** | **Base Model** | **84%** | **81%** |
| **Grid Search CV** | **84%** | **81%** |
| **Smote** | **84%** | **79%** |
| **KNN** | **Base Model** | **86%** | **80%** |
| **Grid Search CV** | **85%** | **80%** |
| **Smote** | **85%** | **78%** |
| **Naïve Bayes** | **Base Model** | **85%** | **80%** |
| **Grid Search CV** |  |  |
| **Smote** | **83%** | **80%** |

*#Train and Test accuracies for each model – Table - 10*

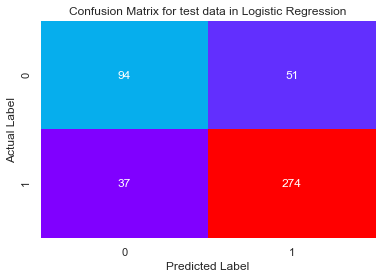
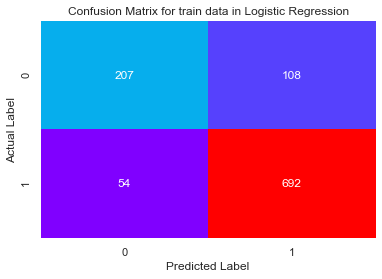
|  |  |  |  |
| --- | --- | --- | --- |
| **Model Types** | **Categories** | **Train Data Accuracy** | **Test Data Accuracy** |
| **Bagging** | **Base Model** | **100%** | **81%** |
| **Grid Search CV** | **92%** | **83%** |
| **Smote** | **94%** | **83%** |
| **Random Forest** | **Base Model** | **100%** | **83%** |
| **Grid Search CV** | **88%** | **84%** |
| **Smote** | **92%** | **84%** |
| **ADABoost** | **Base Model** | **86%** | **79%** |
| **Grid Search CV** | **85%** | **82%** |
| **Smote** | **88%** | **80%** |
| **Gradient Boost** | **Base Model** | **89%** | **83%** |
| **Grid Search CV** | **85%** | **83%** |
| **Smote** | **91%** | **84%** |

As per the model validation is concerned, we can observe that once after using the Hyper Tuning Parameters, through Grid Search Cross Validation, the models show generalisation, which means they are no longer Under fitted or Over fitted.

*#Confusion Matrix for Each Model- Figure 15*

**Logistic Reg. Confusion Matrix- Train Data**

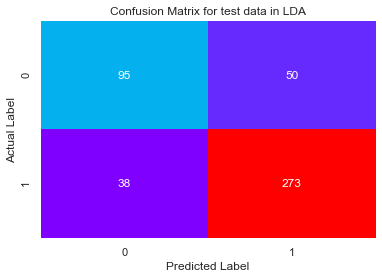
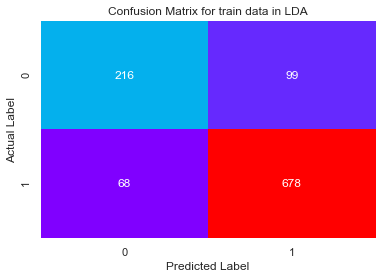
**Logistic Reg. Confusion Matrix - Test Data**



* **Confusion Matrix from Train data**- Majority of the data belong to True positives and True Negatives as compared to the False ones, which are responsible for considerably high Accuracy. But overall, False positives and False negatives are not very high, which are the causes for the metrics doing well.
* **Confusion Matrix from Test data-** Even though the majority of the data shows the reasons of considerably high accuracy, but there has been considerable decrease in False Negatives and Positives, which causes the accuracy score, precision and recall.

**LDA Confusion Matrix- Train Data**

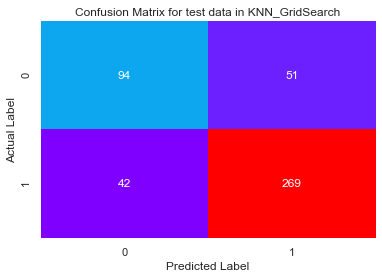
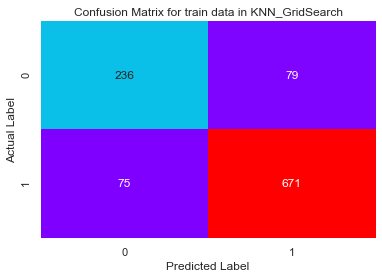
**LDA Confusion Matrix - Test Data**



* **Confusion Matrix from Train data**- Similar to logistic regression, majority of the data belong to True positives and True Negatives as compared to the False ones, which are responsible for Moderate to high Accuracy. But overall, False positives and False negatives are very less, which are the causes high performance of metrices.
* **Confusion Matrix from Test data-** Even though the majority of the data shows the reasons of moderate to high accuracy, but there has been considerable decrease in False Negatives and Positives, which brings up the accuracy score, precision and recall.

**KNN Confusion Matrix- Train Data**

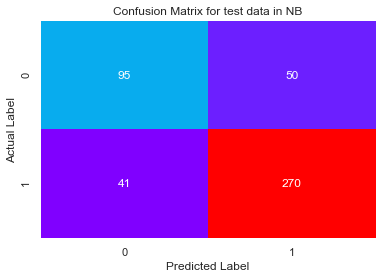
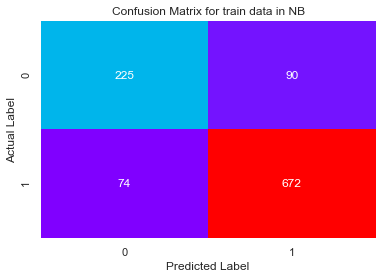
**KNN Confusion Matrix - Test Data**



* **Confusion Matrix from Train data**- Similar to logistic regression and LDA, majority of the data belong to True positives and True Negatives as compared to the False ones, which are responsible for Moderate to high Accuracy. But overall, False positives and False negatives are very less, which are the causes high performance of metrices.
* **Confusion Matrix from Test data-** Even though the majority of the data shows the reasons of moderate to high accuracy, but there has been considerable decrease in False Negatives and Positives, which brings up the accuracy score, precision and recall.

**NB Confusion Matrix- Train Data**

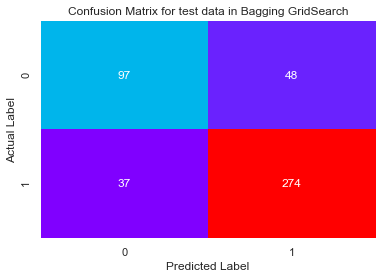
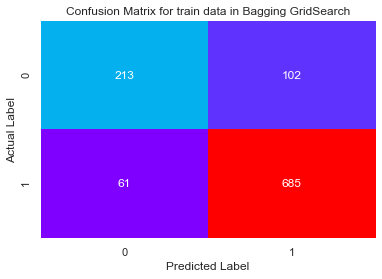
**NB Confusion Matrix - Test Data**



* **Confusion Matrix from Train data**- Similar to other Algorithms, majority of the data belong to True positives and True Negatives as compared to the False ones, which are responsible for Moderate to high Accuracy. But overall, False positives and False negatives are very less, which are the causes high performance of metrices.
* **Confusion Matrix from Test data-** Even though the majority of the data shows the reasons of moderate to high accuracy, but there has been considerable decrease in False Negatives and Positives, which brings up the accuracy score, precision and recall.

**Bagging Confusion Matrix- Train Data**

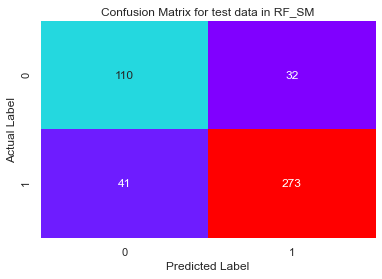
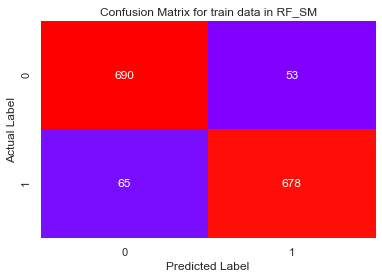
**Bagging Confusion Matrix - Test Data**



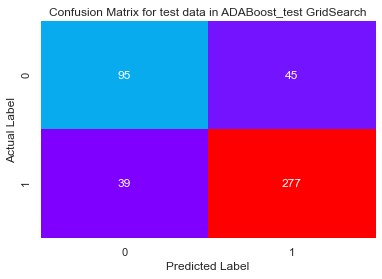
* **Confusion Matrix from Train data**- Similar to other Algorithms, majority of the data belong to True positives and True Negatives as compared to the False ones, which are responsible for Moderate to high Accuracy. But overall, False positives and False negatives are very less, which are the causes high performance of metrices.
* **Confusion Matrix from Test data-** Even though the majority of the data shows the reasons of moderate to high accuracy, but there has been considerable decrease in False Negatives and Positives, which brings up the accuracy score, precision and recall.

**RF Confusion Matrix- Train Data**

**RF Confusion Matrix - Test Data**

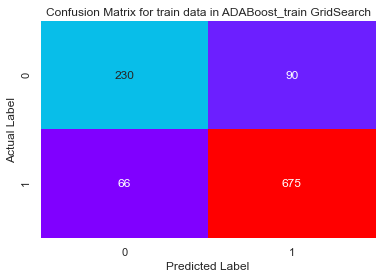


* **Confusion Matrix from Train data**- Similar to other Algorithms, majority of the data belong to True positives and True Negatives as compared to the False ones, which are responsible for Moderate to high Accuracy. But overall, False positives and False negatives are very less, which are the causes high performance of metrices.
* **Confusion Matrix from Test data-** Even though the majority of the data shows the reasons of moderate to high accuracy, but there has been considerable decrease in False Negatives and Positives, which brings up the accuracy score, precision and recall.

**

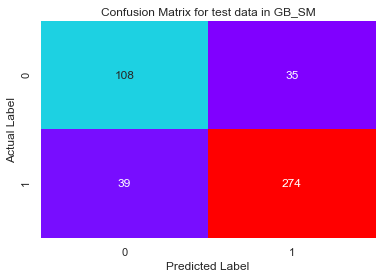
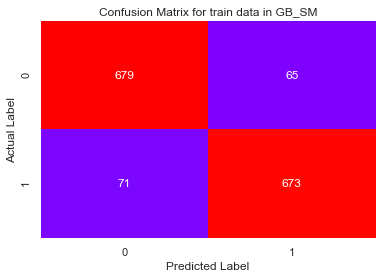
**ADABoost Confusion Matrix- Train Data**

**ADABoost Confusion Matrix - Test Data**

* ****Confusion Matrix from Train data**- Similar to other Algorithms, majority of the data belong to True positives and True Negatives as compared to the False ones, which are responsible for Moderate to high Accuracy. But overall, False positives and False negatives are very less, which are the causes high performance of metrices.
* **Confusion Matrix from Test data-** Even though the majority of the data shows the reasons of moderate to high accuracy, but there has been considerable decrease in False Negatives and Positives, which brings up the accuracy score, precision and recall.

**GBoost Confusion Matrix- Train Data**

**GBoost Confusion Matrix - Test Data**



* **Confusion Matrix from Train data**- Similar to other Algorithms, majority of the data belong to True positives and True Negatives as compared to the False ones, which are responsible for Moderate to high Accuracy. But overall, False positives and False negatives are very less, which are the causes high performance of metrices.
* **Confusion Matrix from Test data-** Even though the majority of the data shows the reasons of moderate to high accuracy, but there has been considerable decrease in False Negatives and Positives, which brings up the accuracy score, precision and recall

***ROC AUC Score***

This is again one of the popular metrics used in the industry.

The ROC (**Receiver operating characteristic**) curve is the plot between **sensitivity** and (**1- specificity**). (**1- specificity**) is also known as **false positive rate** and **sensitivity** is also known as **True Positive rate**.

The biggest advantage of using ROC curve is that it is **independent** of the **change in proportion of responders.**

*#ROC -AUC Score for each model- Table -11*

|  |  |  |  |
| --- | --- | --- | --- |
| **Model Types** | **Categories** | **Train Data ROC AUC Score** | **Test Data ROC AUC Score** |
| **Logistic Regression(Grid Search)** | **Base Model** | **89.86%** | **86.49%** |
| **Grid Search CV** | **89.86%** | **86.55%** |
| **Smote** | **90.21%** | **86.43%** |
| **LDA** | **Base Model** | **89.86%** | **86.31%** |
| **Grid Search CV** | **89.86%** | **86.31%** |
| **Smote** | **90.20%** | **86.39%** |
| **KNN** | **Base Model** | **92.97%** | **85.05%** |
| **Grid Search CV** | **91.25%** | **85.96%** |
| **Smote** | **92.78%** | **85.80%** |
| **Naïve Bayes** | **Base Model** | **89.45%** | **86.79%** |
| **Grid Search CV** |  |  |
| **Smote** | **89.97%** | **86.78%** |

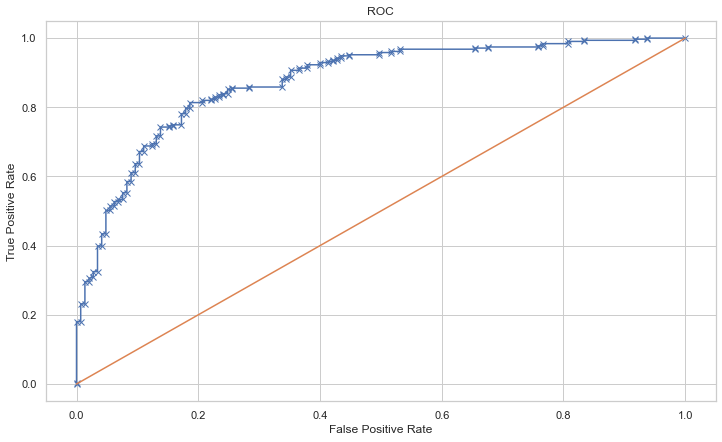
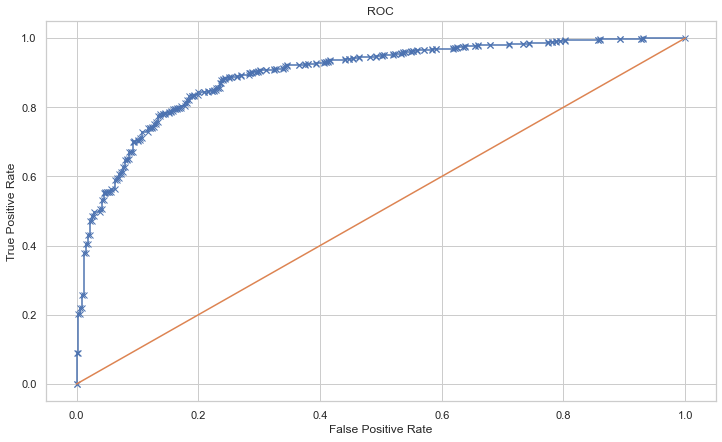
|  |  |  |  |
| --- | --- | --- | --- |
| **Model Types** | **Categories** | **Train Data ROC AUC Score** | **Test Data ROC AUC Score** |
| **Bagging** | **Base Model** | **100%** | **85.98%** |
| **Grid Search CV** | **97.96%** | **89.15%** |
| **Smote** | **98.91%** | **89.00%** |
| **Random Forest** | **Base Model** | **99.99%** | **87.15%** |
| **Grid Search CV** | **89.86%** | **86.46%** |
| **Smote** | **97.90%** | **89.12%** |
| **ADABoost** | **Base Model** | **92.21%** | **86.40%** |
| **Grid Search CV** | **91.78%** | **87.04%** |
| **Smote** | **94.74%** | **87.59%** |
| **Gradient Boost** | **Base Model** | **95.71%** | **89.97%** |
| **Grid Search CV** | **91.78%** | **88.75%** |
| **Smote** | **96.96%** | **88.94%** |

***ROC AUC Curve***

*#ROC -AUC Curve for each model – Figure 16*

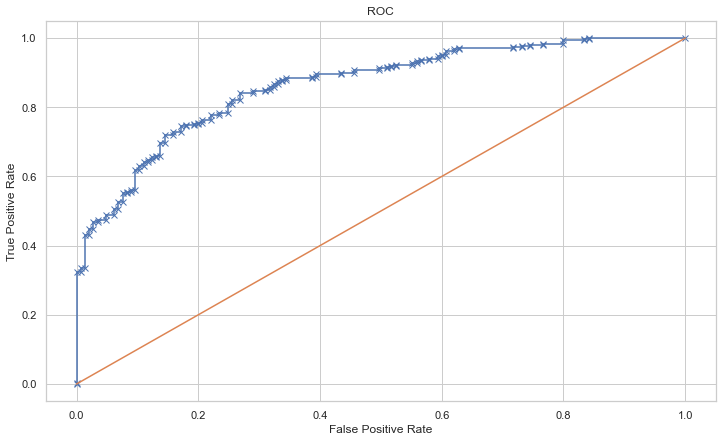
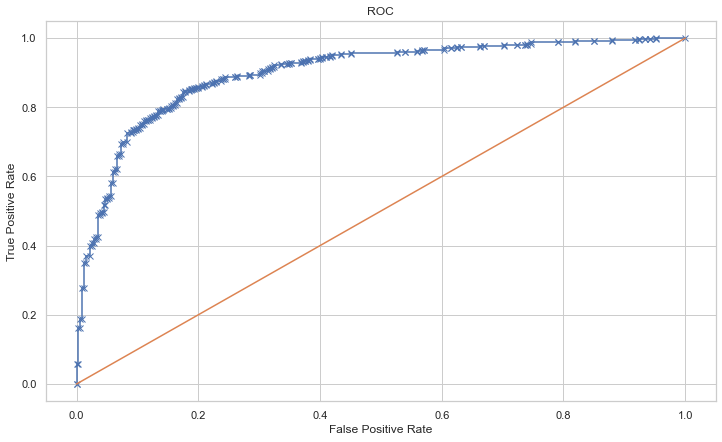
**Logistic Reg. AUC Curve- Train Data**

**Logistic Reg. AUC Curve- Test Data**



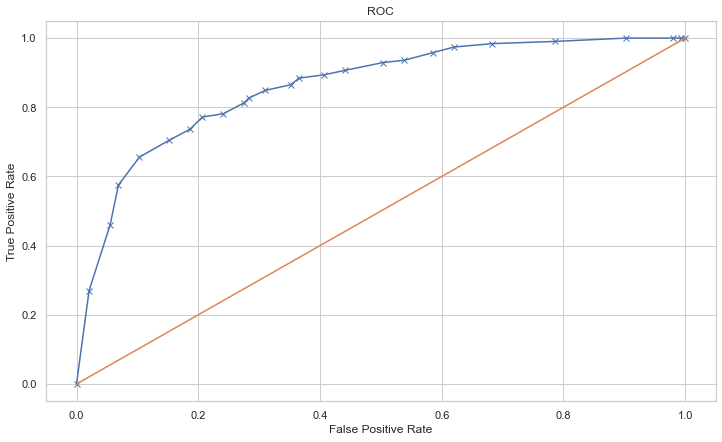
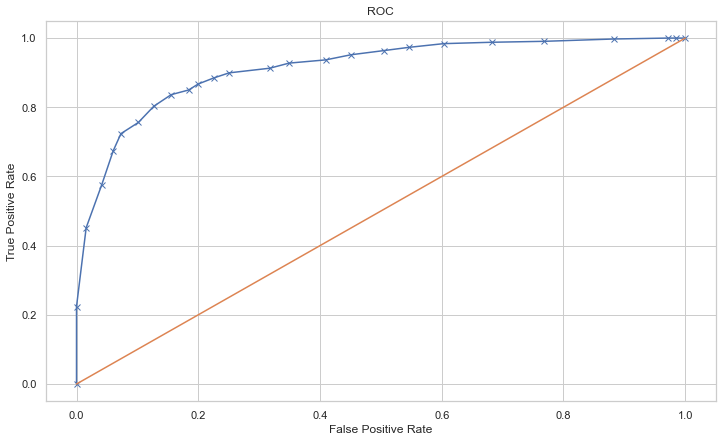
**LDA ROC AUC Curve- Train Data**

**LDA ROC AUC Curve- Test Data**



**KNN ROC AUC Curve- Train Data**

**KNN ROC AUC Curve- Test Data**

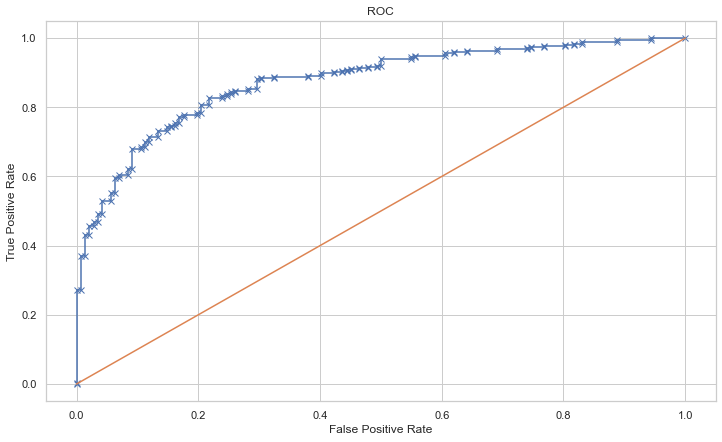
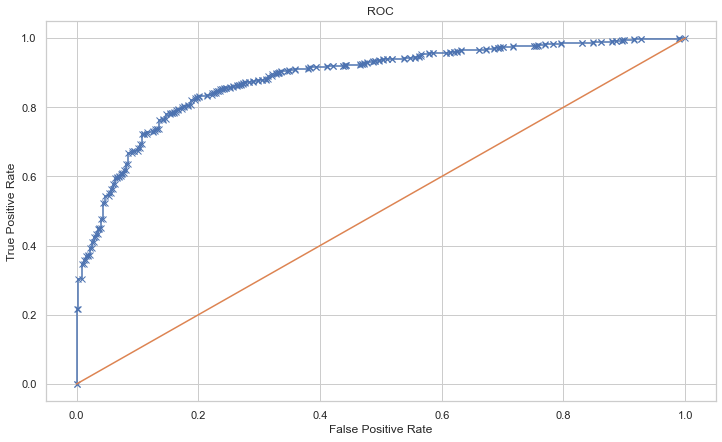


-------------------------------------------------------------------------------------------------------------------------

*#ROC -AUC Curve for each model – Figure 8*

**Naïve Bayes AUC Curve- Train Data**

**Naïve Bayes AUC Curve- Test Data**

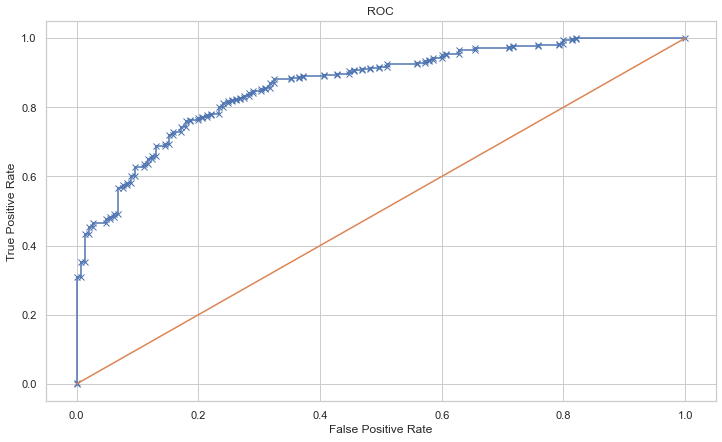
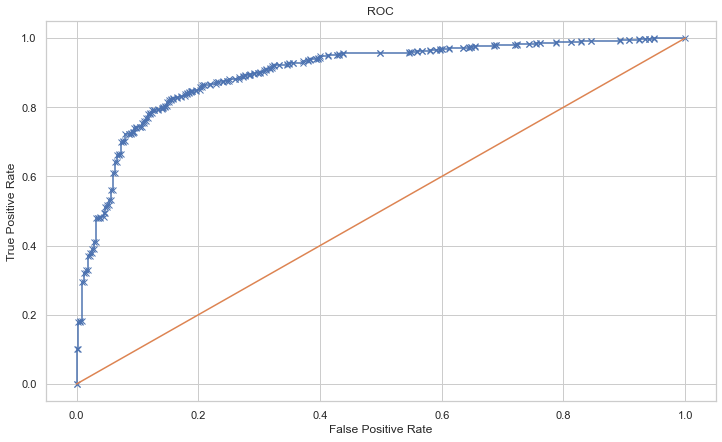
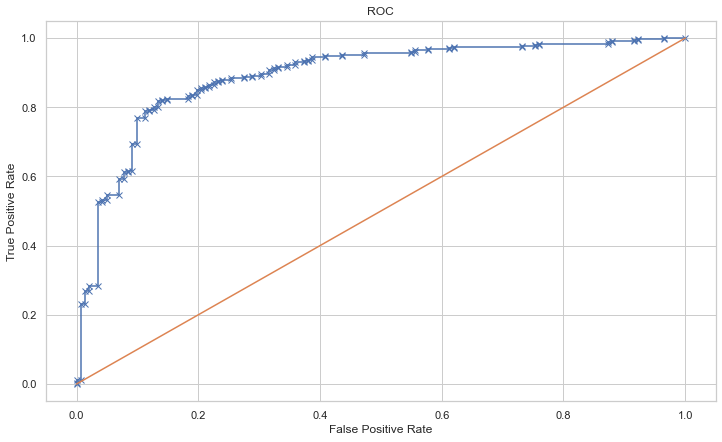
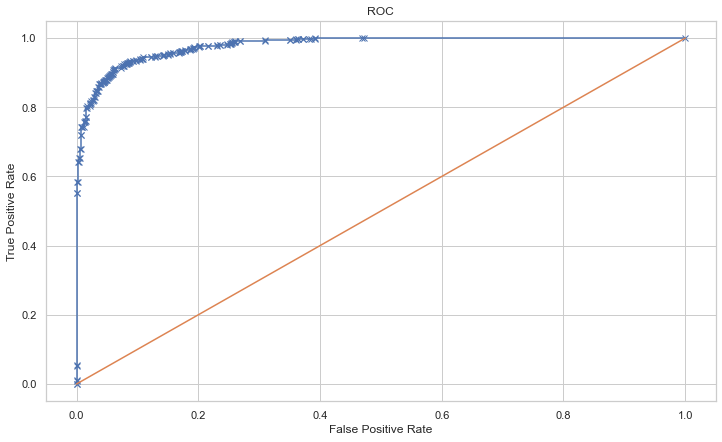


**Bagging ROC AUC Curve- Train Data**

**Bagging ROC AUC Curve- Test Data**

**Random Forest AUC Curve- Train Data**

**Random Forest ROC AUC Curve- Test Data**

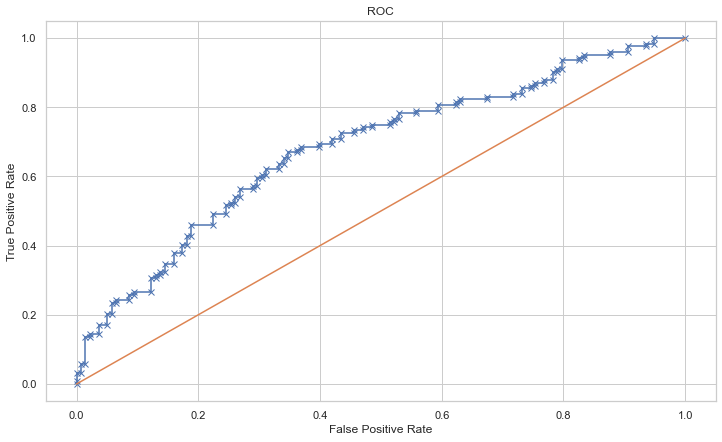
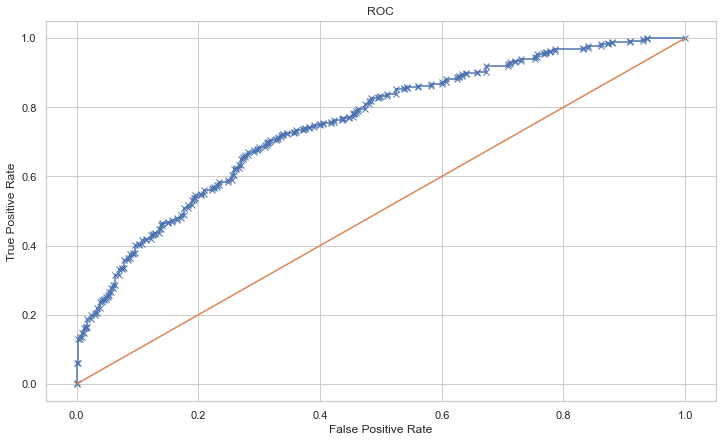


--------------------------------------------------------------------------------------------------------------------------

*#ROC -AUC Curve for each model – Figure 8*

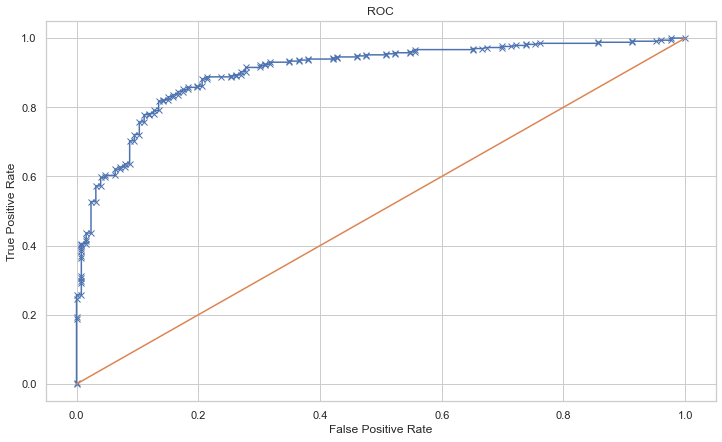
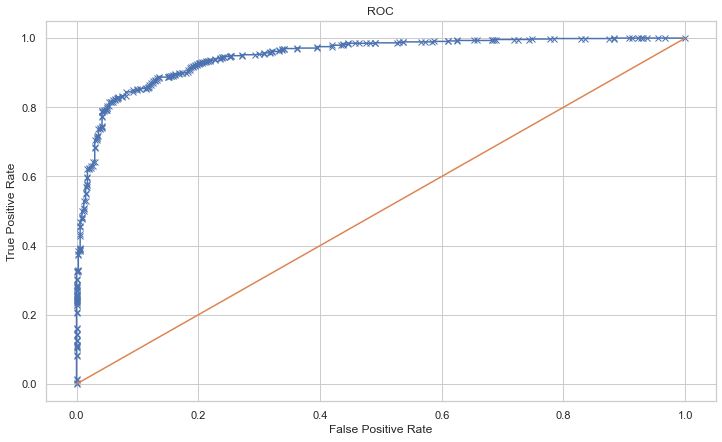
**ADA Boost ROC AUC Curve- Train Data**

**ADA Boost ROC AUC Curve- Test Data**



**Gradient Boost ROC AUC Curve- Train Data**

**Gradient Boost ROC AUC Curve- Test Data**



--------------------------------------------------------------------------------------------------------------------------

***PRECISION***

Similar to the Accuracy, it is a metric derived from confusion matrix

**Positive Predictive Value or Precision** is also defined as the proportion of positive cases that were correctly identified.

In other words, it determines the proportion of **predicted Positives**which is truly Positive

Precision is a valid choice of evaluation metric when we want to be very sure of our prediction.



*#Train and Test Precision for each model- Table -12*

|  |  |  |  |
| --- | --- | --- | --- |
| **Model Types** | **Categories** | **Train Data Precision** | **Test Data Precision** |
| **Logistic Regression(Grid Search)** | **Base Model** | **87%** | **85%** |
| **Grid Search CV** | **86%** | **84%** |
| **Smote** | **85%** | **88%** |
| **LDA** | **Base Model** | **87%** | **85%** |
| **Grid Search CV** | **87%** | **85%** |
| **Smote** | **85%** | **88%** |
| **KNN** | **Base Model** | **90%** | **85%** |
| **Grid Search CV** | **89%** | **84%** |
| **Smote** | **88%** | **89%** |
| **Naïve Bayes** | **Base Model** | **88%** | **84%** |
| **Grid Search CV** |  |  |
| **Smote** | **84%** | **88%** |

|  |  |  |  |
| --- | --- | --- | --- |
| **Model Types** | **Categories** | **Train Data Precision** | **Test Data Precision** |
| **Bagging** | **Base Model** | **100%** | **86%** |
| **Grid Search CV** | **93%** | **85%** |
| **Smote** | **95%** | **88%** |
| **Random Forest** | **Base Model** | **100%** | **86%** |
| **Grid Search CV** | **90%** | **85%** |
| **Smote** | **93%** | **90%** |
| **ADABoost** | **Base Model** | **88%** | **83%** |
| **Grid Search CV** | **88%** | **86%** |
| **Smote** | **88%** | **88%** |
| **Gradient Boost** | **Base Model** | **91%** | **86%** |
| **Grid Search CV** | **88%** | **86%** |
| **Smote** | **91%** | **89%** |

-------------------------------------------------------------------------------------------------------------------------

***RECALL***

Similar to the Precision, it is a metric derived from confusion matrix.

**Sensitivity or Recall** is also defined as the proportion of actual positive cases which are correctly identified.

In other words, it determines the proportion of **actual Positives**is correctly classified

Recall is a valid choice of evaluation metric when we want to capture as many positives as possible.



*#Train and Test Recall for each model- Table -13*

|  |  |  |  |
| --- | --- | --- | --- |
| **Model Types** | **Categories** | **Train Data Recall** | **Test Data Recall** |
| **Logistic Regression(Grid Search)** | **Base Model** | **92%** | **88%** |
| **Grid Search CV** | **93%** | **88%** |
| **Smote** | **84%** | **81%** |
| **LDA** | **Base Model** | **91%** | **88%** |
| **Grid Search CV** | **91%** | **88%** |
| **Smote** | **84%** | **81%** |
| **KNN** | **Base Model** | **91%** | **85%** |
| **Grid Search CV** | **90%** | **86%** |
| **Smote** | **80%** | **77%** |
| **Naïve Bayes** | **Base Model** | **90%** | **87%** |
| **Grid Search CV** |  |  |
| **Smote** | **83%** | **81%** |

|  |  |  |  |
| --- | --- | --- | --- |
| **Model Types** | **Categories** | **Train Data Recall** | **Test Data Recall** |
| **Bagging** | **Base Model** | **100%** | **86%** |
| **Grid Search CV** | **96%** | **92%** |
| **Smote** | **93%** | **87%** |
| **Random Forest** | **Base Model** | **100%** | **90%** |
| **Grid Search CV** | **94%** | **93%** |
| **Smote** | **91%** | **87%** |
| **ADABoost** | **Base Model** | **92%** | **87%** |
| **Grid Search CV** | **91%** | **88%** |
| **Smote** | **87%** | **83%** |
| **Gradient Boost** | **Base Model** | **93%** | **89%** |
| **Grid Search CV** | **91%** | **90%** |
| **Smote** | **90%** | **88%** |

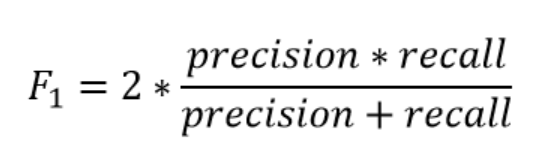
--------------------------------------------------------------------------------------------------------------------------

***F1 SCORE***

Similar to the Precision and Recall, it is a metric derived from confusion matrix.

The **F1 score** is a **number between 0** and **1** and is the **harmonic mean** of **precision** and **recall** values for a **classification** problem.

F1 score sort of maintains a balance between the precision and recall for your classifier. If your precision is low, the F1 is low and if the recall is low again your F1 score is low.



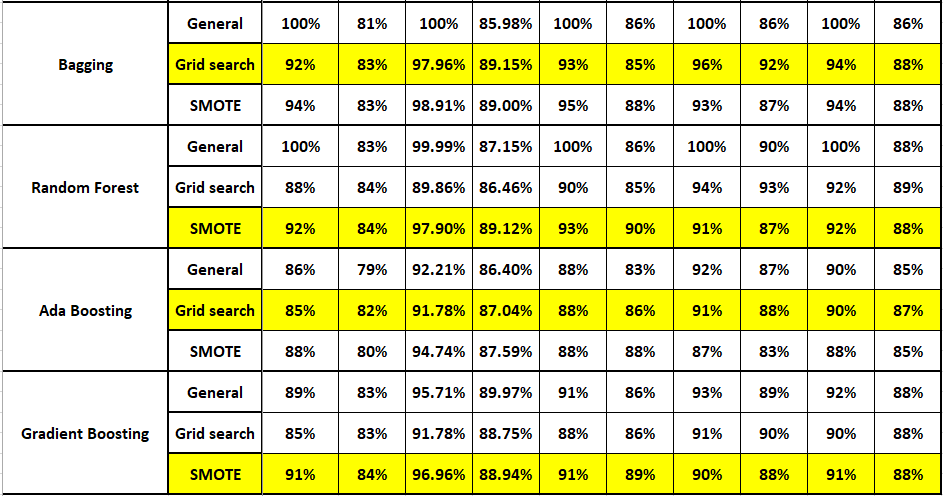
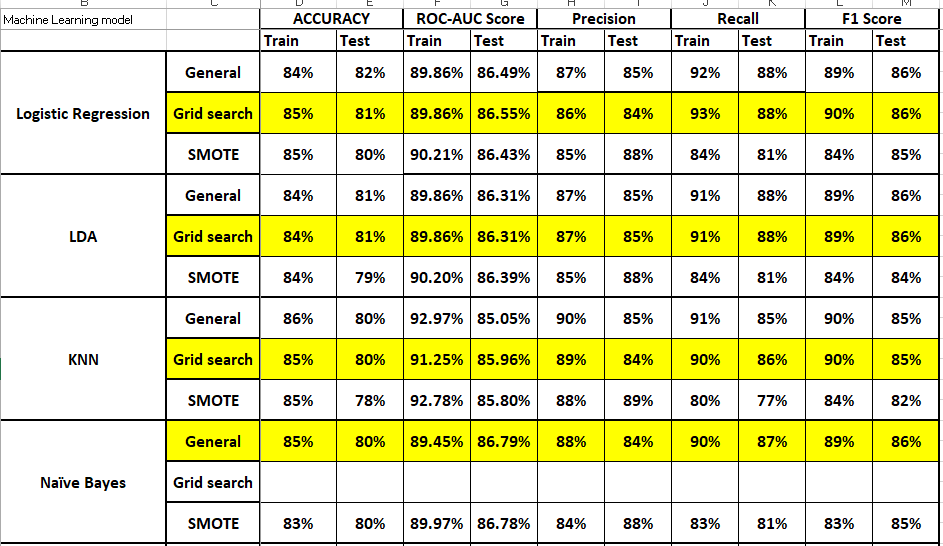
*#Train and Test F1 score for each model- Table -14*

|  |  |  |  |
| --- | --- | --- | --- |
| **Model Types** | **Categories** | **Train Data F1Score** | **Test Data F1Score** |
| **Logistic Regression(Grid Search)** | **Base Model** | **89%** | **86%** |
| **Grid Search CV** | **90%** | **86%** |
| **Smote** | **84%** | **85%** |
| **LDA** | **Base Model** | **89%** | **86%** |
| **Grid Search CV** | **89%** | **86%** |
| **Smote** | **84%** | **84%** |
| **KNN** | **Base Model** | **90%** | **85%** |
| **Grid Search CV** | **90%** | **85%** |
| **Smote** | **84%** | **82%** |
| **Naïve Bayes** | **Base Model** | **89%** | **86%** |
| **Grid Search CV** |  |  |
| **Smote** | **83%** | **85%** |

|  |  |  |  |
| --- | --- | --- | --- |
| **Model Types** | **Categories** | **Train Data F1Score** | **Test Data F1Score** |
| **Bagging** | **Base Model** | **100%** | **86%** |
| **Grid Search CV** | **94%** | **88%** |
| **Smote** | **94%** | **88%** |
| **Random Forest** | **Base Model** | **100%** | **88%** |
| **Grid Search CV** | **92%** | **89%** |
| **Smote** | **92%** | **88%** |
| **ADABoost** | **Base Model** | **90%** | **85%** |
| **Grid Search CV** | **90%** | **87%** |
| **Smote** | **88%** | **85%** |
| **Gradient Boost** | **Base Model** | **92%** | **88%** |
| **Grid Search CV** | **90%** | **88%** |
| **Smote** | **91%** | **88%** |

**--------------------------------------------------------------------------------------------------------------------------**

*#Train and Test scores for each model- Table -15*



**1.8) Based on your analysis and working on the business problem, detail out appropriate insights and recommendations to help the management solve the business objective. There should be at least 3-4 Recommendations and insights in total. Recommendations should be easily understandable and business specific, students should not give any technical suggestions. Full marks should only be allotted if the recommendations are correct and business specific.**

***INSIGHTS***

From the above comparable performance metrics or evaluators we can observe, Random Forest as the Machine Learning Technique is the best model , followed by Bagging and Gradient Boosting

In few of the cases we have seen, the base models have performed better than the tuned ones, but there we have opted for the best in terms of most no. of metrices concerned.

***Reasons***

For both the Train and Test datasets-

* The Precision is better in few
* Accuracy is better than other models
* AUC-ROC score is better than other models
* Improved Recall and F1 Scores

***Recommendations***

All the three models are more or less similar in terms or performance, barring a few metrices.

But overall, most classes are considered to be rightly classified which sets the difference in performance of the metrics in train and test sets.

We have also observed that by improving class balance between minority and majority ones using the SMOTE have caused an impact in the performances.

Which is why, it is recommended to cross-check the dataset thoroughly.

Look for any data entry error, which have given rise to such insufficiency, inaccuracy, etc.

It is necessary to make use of proper resources, to look for misinterpretation of classes or overlapping of data values in classes.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Topic: Text Analysis**

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:

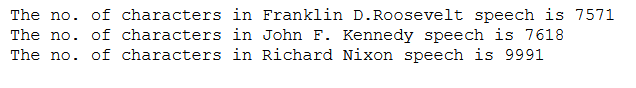
1. President Franklin D. Roosevelt in 1941
2. President John F. Kennedy in 1961
3. President Richard Nixon in 1973

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

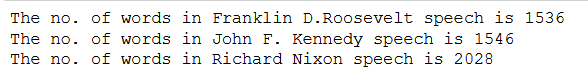
The required no of Characters, words and sentences from the document could be extracted using .raw(), .words(), and .sent() respectively.

The outputs are as follows-

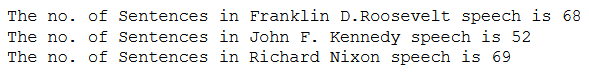
**No. of Characters**



**No. of Words**



**No. of Sentences**



\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

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

In order to remove all the stopwords, the package named ‘stopwords’ is being used which is imported from nltk.corpus library.

The libraries imported are basically-

* from nltk.corpus import stopwords
* from nltk.stem.porter import PorterStemmer

The stopwords library contains all type of stop words such as ‘and’, ‘a’, ‘is’, ‘.’, ‘to’, of’, etc, which usually don’t have any importance in undertaking the sentiments or usefulness in Machine Learning algorithm.

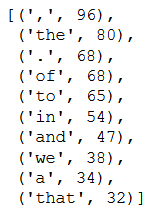
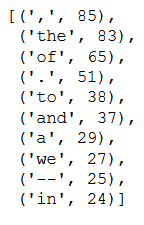
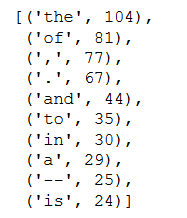
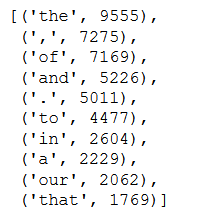
These stopwords present in the package are the universally accepted stopwords, and we can add and remove them as per the requirements.

One important step is to specify the language we are working with before defining the functions, as there are multiple language packages. Here we will be using English.

Stemming is one of the process which helps the model understand the words that have similar meaning. Through this process, the words are brought down to their base or root level by removing the affixes. These are basically being used frequently in Search Engines.

Example- **Running, Ran, Run** – all of these will be reduced to **Run** after Stemming

Before approaching for Step words Removal, we have find out the most common words in each speeches and the Total Speeches together



***Roosevelt***

***Kennedy***

***Nixon***

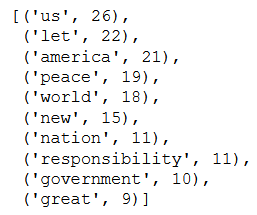
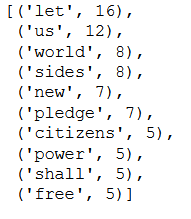
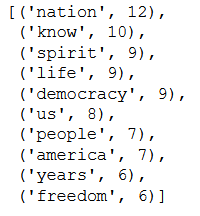
The most common stopwords are –

Most common words after removal of stop words-

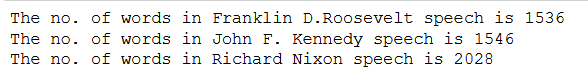
***Roosevelt***

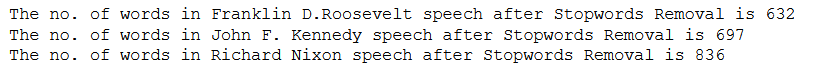
***Kennedy***

***Nixon***



***The no. of words before the Stopwords Removal -***



***The no. of words after the Stopwords Removal -***

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

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

After removing stop words and stemming, the most frequent words are as follows.

For President Franklin D. Roosevelt’s Speech in 1941, the Top three words are –



Most occurring is **‘nation’**

For President John F. Kennedy’s Speech in 1961 , the Top three words are –



Most occurring is **‘let’**

For President Richard Nixon’ Speech in 1973, the Top three words are –



Most occurring is **‘us’**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**2.4) Plot the word cloud of each of the three speeches. (after removing the stopwords)**

We have used several packages to create wordcloud for each of the speeches

The packages and libraries being used are shared below-

* !pip install wordcloud
* from wordcloud import WordCloud, STOPWORDS
* import collections
* from collections import Counter

The code for such wordcloud creation is being mentioned in the Jupyter notebook (Machine Learning Project 2)

***Kennedy***

***Roosevelt***



***Nixon***



\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

***Reference***

* *https://towardsdatascience.com/all-about-feature-scaling-bcc0ad75cb35*
* *https://scikit-learn.org/stable/modules/preprocessing.html*
* *https://www.analyticsvidhya.com/blog/2020/04/feature-scaling-machine-learning-normalization-standardization/*
* *https://www.analyticsvidhya.com/blog/2020/07/10-techniques-to-deal-with-class-imbalance-in-machine-learning/*
* *https://machinelearningmastery.com/linear-discriminant-analysis-with-python/*
* *https://www.analyticsvidhya.com/blog/2017/03/imbalanced-data-classification/*
* *https://towardsdatascience.com/a-simple-introduction-to-k-nearest-neighbors-algorithm-b3519ed98e*
* *https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.AdaBoostClassifier.html*
* *https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.BaggingClassifier.html?highlight=bagging#sklearn.ensemble.BaggingClassifier*
* *https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.AdaBoostClassifier.html*
* *https://www.geeksforgeeks.org/ml-bagging-classifier/*
* *https://medium.com/swlh/my-first-data-science-project-a-word-cloud-comparing-the-2020-presidential-debates-c4a2c705dae2*
* *https://pythoncircle.com/post/689/python-script-16-generating-word-cloud-image-of-a-text-using-python/*
* *https://stackoverflow.com/questions/15228054/how-to-count-the-amount-of-sentences-in-a-paragraph-in-python*
* *https://training-course-material.com/training/Natural\_Language\_Processing\_in\_Python*
* *https://towardsai.net/p/machine-learning/why-choose-random-forest-and-not-decision-trees*
* *https://www.analyticsvidhya.com/blog/2021/06/understanding-random-forest/*
* *https://towardsdatascience.com/optimizing-hyperparameters-in-random-forest-classification-ec7741f9d3f6*