PGP - Data Science and Business Analysis - Jan’ 21

Machine Learning Project Report

*by Sarang Manohar*

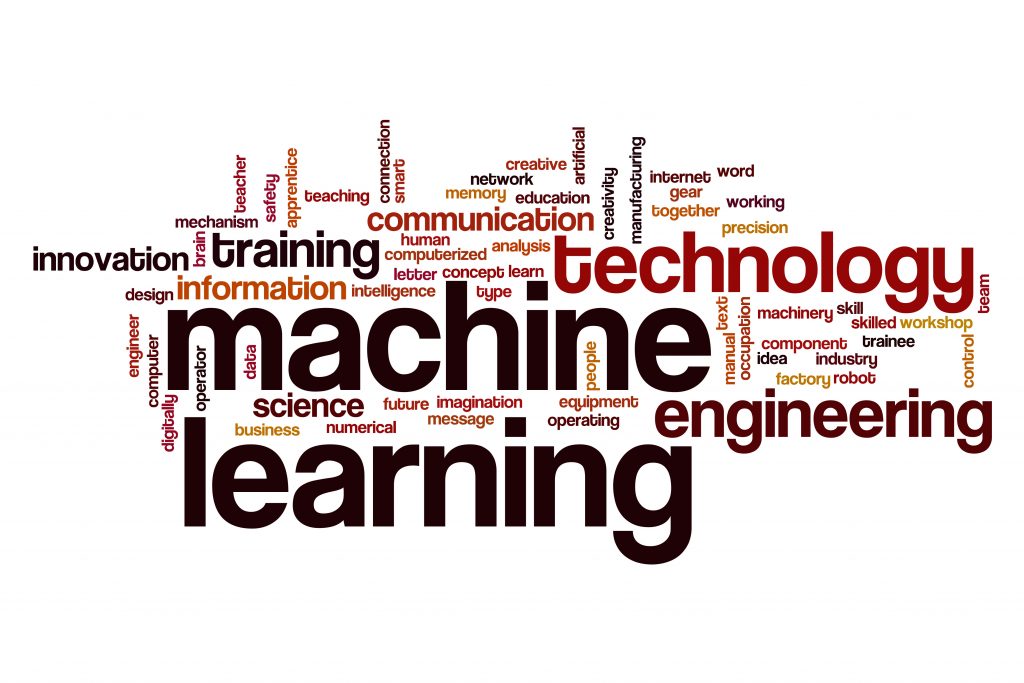
**

Table of Contents

[Table of Contents 2](#_Toc78738521)

[Introduction 3](#_Toc78738522)

[Problem 1: Machine Learning 3](#_Toc78738523)

[Data Dictionary: 3](#_Toc78738524)

[Question 1.1. 3](#_Toc78738525)

[Question 1.2 5](#_Toc78738526)

[Question 1.3 10](#_Toc78738527)

[Question 1.4 10](#_Toc78738528)

[Question 1.5 13](#_Toc78738529)

[Question 1.6 15](#_Toc78738530)

[Question 1.7 26](#_Toc78738531)

[Question 1.8 27](#_Toc78738532)

[Problem 2: Text Analytics 28](#_Toc78738533)

[Question 2.1 28](#_Toc78738534)

[Question 2.2 28](#_Toc78738535)

[Question 2.3 29](#_Toc78738536)

[Question 2.4 29](#_Toc78738537)

[End of the business report 31](#_Toc78738538)

[Illustration Index 31](#_Toc78738539)

[Table Index 32](#_Toc78738540)

Introduction

This project report is in response to the final assignment for completion of Machine Learning module. In this module, we learned different machine learning techniques including ensemble types, as well as text analytics for unstructured data sets. This module helped strengthen our learnings from past modules.

Problem 1: Machine Learning

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

Data Dictionary:

|  |  |
| --- | --- |
| **Variable Name** | **Description** |
| vote | Party choice: Conservative or Labour |
| age | In years |
| economic.cond.national | Assessment of current national economic conditions, 1 to 5. |
| economic.cond.household | Assessment of current household economic conditions, 1 to 5. |
| Blair | Assessment of the Labour leader, 1 to 5. |
| Hague | Assessment of the Conservative leader, 1 to 5. |
| Europe | An 11-point scale that measures respondents' attitudes toward European integration. High scores represent ‘Eurosceptic’ sentiment. |
| political.knowledge | Knowledge of parties' positions on European integration, 0 to 3. |
| Gender | Female or male |

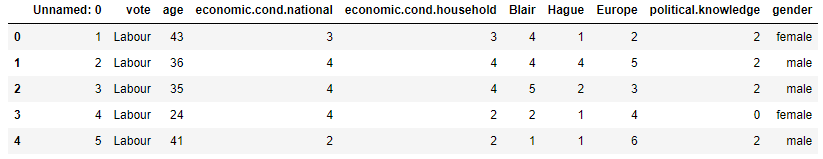
*Table 1: ProblemStatement1\_DataDictionary*

Question 1.1.

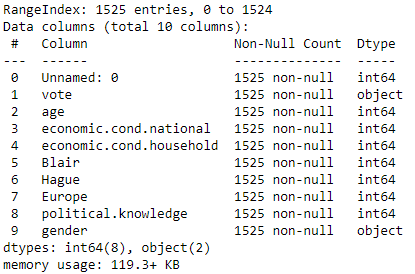
Read the dataset. Do the descriptive statistics and do the null value condition check. Write an inference on it.

We start by importing the data and checking the first few records. A snapshot of the same is below

Some quick observations on the data after looking at the sample records.

 Illustration 1: First 5 records from the data

1. There are 10 features in the data, of which the first feature 'Unnamed: 0' seemed to be the record id which should be dropped
2. Except age all the features are categorical in nature
3. The categorical features are ordinal in nature expect for gender
4. The vote is the target feature which we need to predict

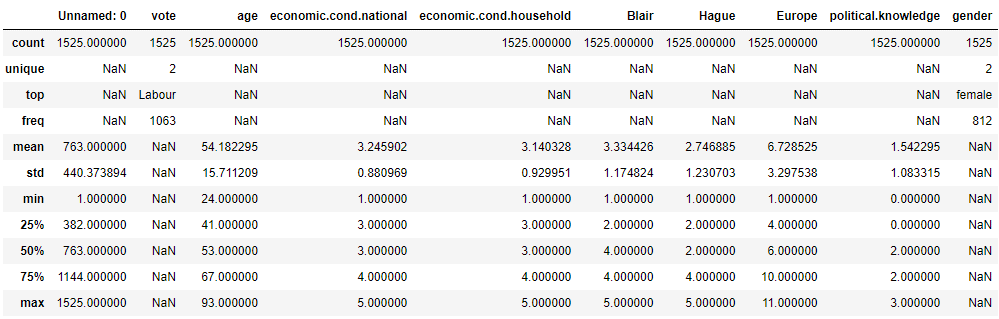
 Illustration 2: Basic information about the data frame

Moving ahead with EDA, we run a info() check on the data and compare the results with our assumptions by looking at the sample data.

1. There are 1,525 records in the data set
2. All the features have not null values across all records
3. As expected, vote and gender have object datatype which is as per the data dictionary definition
4. All the remaining features are int64 which is as per the expectations as well

Next, we run describe method on the continuous features to get am insight on how the data is distributed.

1. Age is the only continuous feature and it appears to be not normally distributed

 Illustration 3: Results of describe method on all the features

1. The range of age is between 24 and 93. Thus, there are no outliers in the data
2. All the remaining features are within expected range and in line with their definition

Question 1.2

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

**Uni-variate analysis**

Our focus in this sub-section will be to analyze each feature individually.

**Age**

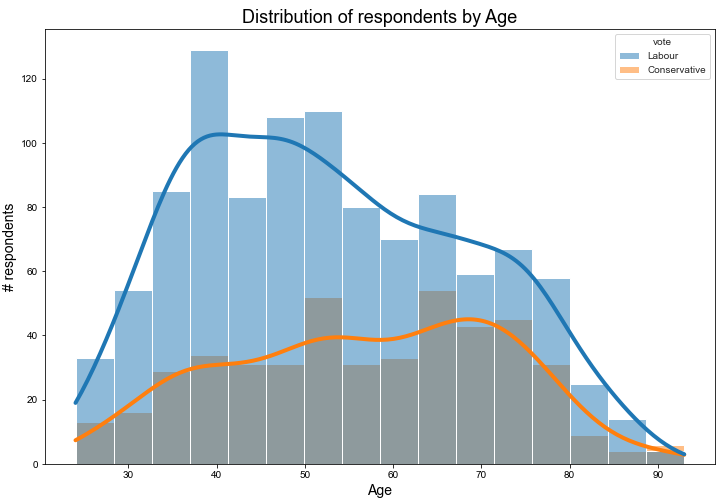


Illustration 4: Distribution of respondents by Age and Political choice

The distribution of age appears to be skewed when looked at from the filter of political choice. Conservative party appears find majority of its supports among old age where as Labour party has younger supporters.

**National Economic Rating**

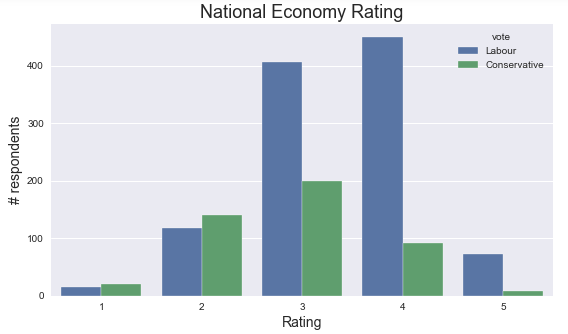


Illustration 5: Ratings of National Economic Condition

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **1** | **2** | **3** | **4** | **5** |
| **Conservative** | 4.5% | 30.3% | 43.3% | 19.9% | 1.9% |
| **Labour** | 1.5% | 11.0% | 38.3% | 42.3% | 6.9% |

The majority of conservative supporters rate the economic condition between 2-3 out of 5, whereas the labour party supporters feel that economic condition is in better shape and have given the ratings between 3-4 out of 5.

**Household economic condition**

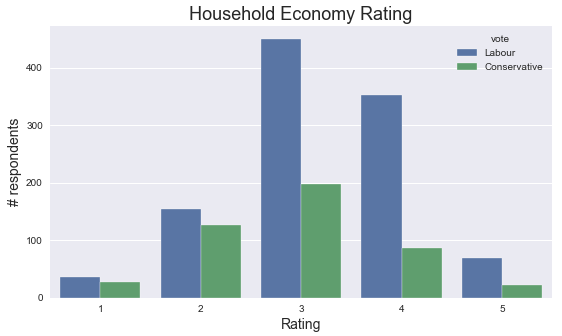


Illustration 6: Ratings of Household Economic Condition

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **1** | **2** | **3** | **4** | **5** |
| **Conservative** | 6.1% | 27.3% | 42.9% | 18.8% | 5.0% |
| **Labour** | 3.5% | 14.5% | 42.3% | 33.2% | 6.5% |

The majority of conservative supporters rate the household condition between 2-3 out of 5, whereas the labour party supporters feel that household condition is in better shape and have given the ratings between 3-4 out of 5.

**Blair’s Rating**



Illustration 7: Ratings of The Labour Party’s candidate - Blair

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **1** | **2** | **3** | **4** | **5** |
| **Conservative** | 12.8% | 52.4% | 0.2% | 34.0% | 0.6% |
| **Labour** | 3.6% | 18.4% | 0% | 63.9% | 14.1% |

As expected, the labour party supporters rate the candidate of their party very highly with ~78% of them rating him either 4 or 5 out of 5. Interestingly, ~34% of the conservative supporters also rated Blair with a rating of 4 out of 5.

**Hague’s Rating**

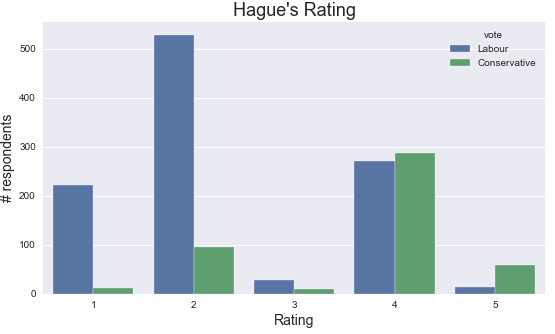


Illustration 8: Ratings of The Conservative Party’s candidate - Hague

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **1** | **2** | **3** | **4** | **5** |
| **Conservative** | 2.4% | 20.8% | 1.9% | 62.1% | 12.8% |
| **Labour** | 29.0% | 49.7% | 2.6% | 25.5% | 1.3% |

As expected, the conservative party supporters rate the candidate of their party very highly with ~75% of them rating him either 4 or 5 out of 5.

**Skepticism towards European Integration**

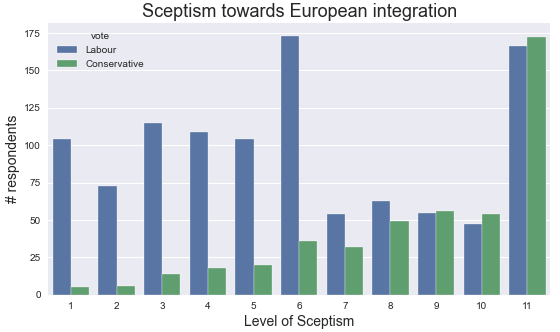


Illustration 9: Skepticism around European Integration

The conservative party supporters have usually high skepticism around European Integration when compared to the labour party supporters

**Knowledge of Party’s position among the supporters**

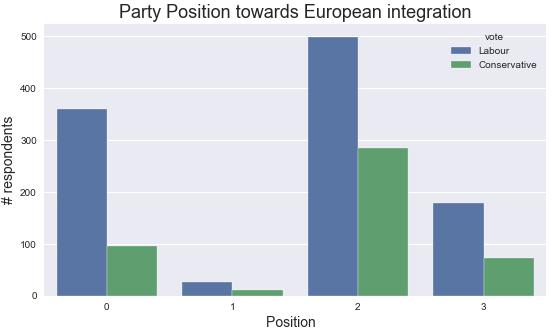


Illustration 10: Knowledge of Party’s position among the supporters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **0** | **1** | **2** | **3** |
| **Conservative** | 20.6% | 2.4% | 61.5% | 15.6% |
| **Labour** | 33.4% | 2.5% | 46.8% | 16.7% |

Around ~61.5% of the conservative party supporters seemed to have good knowledge of their party’s position around the European integration. Almost, a third of the labour party supporters seemed to have no knowledge about their party’s position around the European integration.

**Distribution of gender**

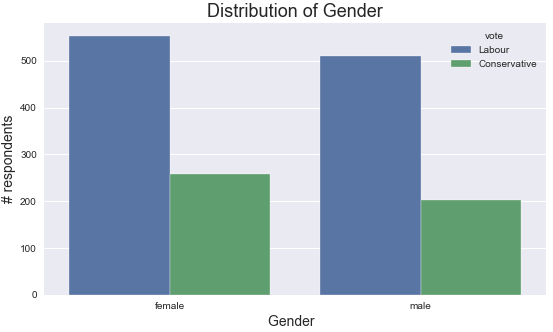


Illustration 11: Distribution of respondents by Gender

|  |  |  |
| --- | --- | --- |
|  | **female** | **male** |
| **Conservative** | 56.1% | 43.9% |
| **Labour** | 52.0% | 48.0% |

Both the parties seemed to have equal distribution of male and female distributions.

**Multi-variate analysis**

Since, all the variables are categorical in nature except for age, we need to run Chi-Squared Contingency tests between the variables to investigate the correlation between them. Below mentioned grid explains the outcome of Chi-Squared Contingency tests between some logical pairs of variables.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **#** | **Variable 1** | **Variable 2** | **Degrees of freedom** | **Test statistics score** | **p-value of Test Statistics** | **Inference** |
| 1 | National Economic Condition | Blair | 16 | 216.21 | 4.11554E-37 | Correlated |
| 2 | Household Economic Condition | Blair | 16 | 96.94 | 1.29094E-13 | Correlated |
| 3 | National Economic Condition | Hague | 16 | 70.77 | 7.30737E-09 | Correlated |
| 4 | Household Economic Condition | Hague | 16 | 38.49 | 0.001288441 | Correlated |
| 5 | Gender | Blair | 4 | 11.56 | 0.020900818 | Correlated |
| 6 | Gender | Hague | 4 | 7.02 | 0.135042493 | Independent |
| 7 | Vote | National Economic Condition | 4 | 145.48 | 1.88897E-30 | Correlated |
| 8 | Vote | Household Economic Condition | 4 | 58.01 | 7.58609E-12 | Correlated |
| 9 | Vote | Scepticism around European Integration | 10 | 247.26 | 2.04725E-47 | Correlated |
| 10 | Vote | Knowledge of Party Stance | 3 | 32.83 | 3.49788E-07 | Correlated |

*Table 2: Chi-Squared Contingency test results*

Expect for the combination of Gender and Hague all the categorical variables are correlated. It suggests that political views as biased for the most part.

Question 1.3

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

The gender has string values and is encoded via get\_dummies method to split the column in two and have binary values to represent the intended values.

The target variable also has string values and the values were replaced as Labour as 0 and Conservative as 1.

Age is continuous and ranges between 24 to 93 whereas the rest of the variables are ratings ranging between either 1-5, 1-11, 0-3, or 0-1(encoded gender variables). Since, the scales of the variables are varied, scaling of the data is required before we build models on top of it.

The data was scaled using StandardScaler method from sci-kit learn’s preprocessing library and then was further spilt into train and test data with 70:30 ratio.

Question 1.4

Apply Logistic Regression and LDA (linear discriminant analysis)

Logistic Regression

Below are the results of the base model created using the default parameters.

LogisticRegression(C=1.0, class\_weight=None, dual=False, fit\_intercept=True,

intercept\_scaling=1, l1\_ratio=None, max\_iter=100,

multi\_class='auto', n\_jobs=None, penalty='l2',

random\_state=None, solver='lbfgs', tol=0.0001, verbose=0,

warm\_start=False)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Training data | | Actuals | |  |  |  |  |  |  |
| Accuracy | 83.1% | Labour | Conservative |  |  | precision | recall | f1-score | support |
| Predicted | Labour | 686 | 68 |  | Labour | 0.86 | 0.91 | 0.88 | 754 |
| Conservative | 111 | 196 |  | Conservative | 0.74 | 0.64 | 0.69 | 307 |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Test data | | Actuals | |  |  |  |  |  |  |
| Accuracy | 83.1% | Labour | Conservative |  |  | precision | recall | f1-score | support |
| Predicted | Labour | 268 | 35 |  | Labour | 0.86 | 0.88 | 0.87 | 303 |
| Conservative | 42 | 111 |  | Conservative | 0.76 | 0.73 | 0.74 | 153 |

The base model is slightly underfit on the data as it has low recall (0.64) for Conservative class. However, the performance on test data has better recall (0.73) for Conservative class at the cost of a drop in recall for Labour class. The overall precision remains more or less same for both the classes across the two data sets. Finally, the accuracy on the both the data sets are same at 83.1%.

**Parameters of importance**: Logistic Regression works on regularization of the coefficients to make them as small as possible. The important parameters are focused to regularize the coefficients in most optimal manner and hence choosing the right parameters is important

C = the inverse of regularization strength. It should be positive. The default here is 1 which means no regularization would happen via this parameter

Penalty: It causes the coefficients to shrink to zero. The default value was l2 which is Ridge regularization

Solver: This specifies the type of algorithm used to that minimizes cost function. By default this is set to lbfgs. lbfgs saves memory and tends to be faster in small to medium size datasets. It does a decent job but may not be the optimal choice.

Tolerance: It is the limits the iteration when the minimum error is achieved. The default is 1e-04.

|  |  |
| --- | --- |
| Training Data | Test Data |
| AUC = 89.0%  Illustration 12: ROC of Logistic Regression model using training data | AUC=88.3%  Illustration 13: ROC of Logistic Regression model using test data |

The area under the ROC curve is also close for both training and test at 89.0% and 88.3% resp.

Linear Discriminant Analysis (LDA)

Below are the results of the base model created using the default parameters.

LinearDiscriminantAnalysis(n\_components=None, priors=None, shrinkage=None,

solver='svd', store\_covariance=False, tol=0.0001)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Training data | | Actuals | |  |  |  |  |  |  |
| Accuracy | 83.4% | Labour | Conservative |  |  | precision | recall | f1-score | support |
| Predicted | Labour | 685 | 69 |  | Labour | 0.86 | 0.91 | 0.89 | 754 |
| Conservative | 107 | 200 |  | Conservative | 0.74 | 0.65 | 0.69 | 307 |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Test data | | Actuals | |  |  |  |  |  |  |
| Accuracy | 83.3% | Labour | Conservative |  |  | precision | recall | f1-score | support |
| Predicted | Labour | 269 | 34 |  | Labour | 0.86 | 0.89 | 0.88 | 303 |
| Conservative | 42 | 111 |  | Conservative | 0.77 | 0.73 | 0.74 | 153 |

The base model is slightly underfit on the data as it has low recall (0.65) for Conservative class. However, the performance on test data has better recall (0.73) for Conservative class at the cost of a drop in recall for Labour class. The overall precision remains more or less same for both the classes across the two data sets. Finally, the accuracy on the training and test data sets are close at 83.4% & 83.3% respectively.

**Parameters of importance**:

Solver: This specifies the type of algorithm used to that minimizes cost function. By default this is set to svd. Svd doesn’t compute the covariance matrix.

|  |  |
| --- | --- |
| Training Data | Test Data |
| AUC = 88.9%  Illustration 14: ROC of LDA model using training data | AUC=88.8%  Illustration 15: ROC of LDA model using test data |

The area under the ROC curve is also close for both training and test at 88.9% and 88.8% resp.

Question 1.5

Apply KNN Model and Naïve Bayes Model

KNN Model

Below are the results of the base model created using the default parameters.

KNeighborsClassifier(algorithm='auto', leaf\_size=30, metric='minkowski',

metric\_params=None, n\_jobs=None, n\_neighbors=5, p=2,

weights='uniform')

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Training data | | Actuals | |  |  |  |  |  |  |
| Accuracy | 85.6% | Labour | Conservative |  |  | precision | recall | f1-score | support |
| Predicted | Labour | 690 | 64 |  | Labour | 0.89 | 0.92 | 0.90 | 754 |
| Conservative | 89 | 218 |  | Conservative | 0.77 | 0.71 | 0.74 | 307 |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Test data | | Actuals | |  |  |  |  |  |  |
| Accuracy | 82.5% | Labour | Conservative |  |  | precision | recall | f1-score | support |
| Predicted | Labour | 271 | 32 |  | Labour | 0.85 | 0.89 | 0.87 | 303 |
| Conservative | 48 | 105 |  | Conservative | 0.77 | 0.69 | 0.72 | 153 |

The base model does a good job in predicting the Labour class with high values of both precision (0.89) and recall (0.91) on the training data and with 0.85 and 0.89 on the test data. Comparatively, the prediction of Conservative class has bad results across both the training and test data. This can be attributed data imbalance where almost 2/3rds of the data are represented by Labour and remaining by Conservative class. The performance of the model on the test data is fair as the accuracy is 82.5% as compared to 85.6% on the training data.

**Parameters of importance**: KNN is a distance-based model. Hence, choosing the best distance calculation mechanism is important to achieve satisfactory performance.

N\_neighbors = Number of neighbors to use by default for kneighbors queries. A number too high will tend to have the predictions converge to similar values thus and would not be able to generalize. The default value is 5

algorithm: Algorithm used to compute the nearest neighbors. Default value is auto which will attempt to decide the most appropriate algorithm based on the values passed to fit method

leaf\_size: Leaf size passed to BallTree or KDTree. Default value is 30.

p: Default value is 2, meaning minkowski distance will be used.

|  |  |
| --- | --- |
| Training Data | Test Data |
| AUC = 92.7%  Illustration 16: ROC of KNN model using training data | AUC=87.0%  Illustration 17: ROC of KNN model using test data |

The area under the ROC curve with training data is 92.7% and test data is 87.0%.

Naïve Bayes Model

Below are the results of the base model created using the default parameters.

GaussianNB(priors=None, var\_smoothing=1e-09)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Training data | | Actuals | |  |  |  |  |  |  |
| Accuracy | 82.9% | Labour | Conservative |  |  | precision | recall | f1-score | support |
| Predicted | Labour | 668 | 86 |  | Labour | 0.88 | 0.89 | 0.88 | 754 |
| Conservative | 95 | 212 |  | Conservative | 0.71 | 0.69 | 0.70 | 307 |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Test data | | Actuals | |  |  |  |  |  |  |
| Accuracy | 82.5% | Labour | Conservative |  |  | precision | recall | f1-score | support |
| Predicted | Labour | 263 | 40 |  | Labour | 0.87 | 0.87 | 0.87 | 303 |
| Conservative | 40 | 113 |  | Conservative | 0.74 | 0.74 | 0.74 | 153 |

The base model is slightly underfit on the data as it has low recall (0.69) for Conservative class. However, the performance on test data has better recall (0.74) for Conservative class at the cost of a drop in recall for Labour class. The overall precision remains more or less same for both the classes across the two data sets. Finally, the accuracy on the training and test data sets are close at 82.9% & 82.5% respectively.

**Parameters of importance**: The Naïve Bayes is a simple ml algorithm that doesn’t need much of parameter tuning.

var\_smoothing: Portion of the largest variance of all features that is added to variances for calculation stability. The default value is 1e-09

|  |  |
| --- | --- |
| Training Data | Test Data |
| AUC = 88.9%  Illustration 18: ROC of Naïve Bayes model using training data | AUC=87.5%  Illustration 19: ROC of Naïve Bayes model using test data |

The area under the ROC curve is also close for both training and test at 88.9% and 87.5% resp.

Question 1.6

Model Tuning, Bagging (Random Forest), and Boosting

Model Tuning

Logistic Regression

Below are the results of the tuned model created using the following parameters.

LogisticRegression(C=0.08497534359086456, class\_weight=None, dual=False,

fit\_intercept=True, intercept\_scaling=1, l1\_ratio=None,

max\_iter=100, multi\_class='auto', n\_jobs=4, penalty='l1',

random\_state=1, solver='liblinear', tol=0.01, verbose=1,

warm\_start=False)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Training data | | Actuals | |  |  |  |  |  |  |
| Accuracy | 84.0% | Labour | Conservative |  |  | precision | recall | f1-score | support |
| Predicted | Labour | 696 | 58 |  | Labour | 0.86 | 0.92 | 0.89 | 754 |
| Conservative | 112 | 195 |  | Conservative | 0.77 | 0.64 | 0.70 | 307 |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Test data | | Actuals | |  |  |  |  |  |  |
| Accuracy | 82.9% | Labour | Conservative |  |  | precision | recall | f1-score | support |
| Predicted | Labour | 269 | 34 |  | Labour | 0.86 | 0.89 | 0.87 | 303 |
| Conservative | 44 | 109 |  | Conservative | 0.76 | 0.71 | 0.74 | 153 |

The tuned model is slightly underfit on the data as it has low recall (0.64) for Conservative class. However, the performance on test data has better recall (0.71) for Conservative class at the cost of a drop in recall for Labour class. The overall precision remains more or less same for both the classes across the two data sets. Finally, the accuracy on the training and test data sets are close at 84.0% & 82.9% respectively.

**Parameters of choice:**

C = The gridsearchcv was provided with 100 values rangning between 1e-06 and 100(from a very strong regularization factor to a very weak). The method suggested 0.08497 as the right factor for the model

Penalty: Among all the options provided for penalty the gridsearchcv method suggested to use l1 i.e. Lasso Regularization

Solver: liblinear. It works best for small data sets

Tolerance: 1e-04. Same as the default value. Any number larger would have made the algorithm to converge and exit soon without identifying the right coefficients. A smaller number would have caused the model to run very slow.

|  |  |
| --- | --- |
| Training Data | Test Data |
| AUC = 88.9%  Illustration 20: ROC of Logistic Regression model using training data | AUC=88.2%  Illustration 21: ROC of Logistic Regression model using test data |

The area under the ROC curve is also close for both training and test at 88.9% and 88.2% resp.

Linear Discriminant Analysis (LDA)

Below are the results of the tuned model created using the following parameters.

LinearDiscriminantAnalysis(n\_components=None, priors=None, shrinkage=None,

solver='svd', store\_covariance=False, tol=0.0001)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Training data | | Actuals | |  |  |  |  |  |  |
| Accuracy | 83.4% | Labour | Conservative |  |  | precision | recall | f1-score | support |
| Predicted | Labour | 685 | 69 |  | Labour | 0.86 | 0.91 | 0.89 | 754 |
| Conservative | 107 | 200 |  | Conservative | 0.74 | 0.65 | 0.69 | 307 |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Test data | | Actuals | |  |  |  |  |  |  |
| Accuracy | 83.3% | Labour | Conservative |  |  | precision | recall | f1-score | support |
| Predicted | Labour | 269 | 34 |  | Labour | 0.86 | 0.89 | 0.88 | 303 |
| Conservative | 42 | 111 |  | Conservative | 0.77 | 0.73 | 0.74 | 153 |

The tuned model suggested the same parameters as the default we had for the base model. The model was slightly underfit on the data as it has low recall (0.65) for Conservative class. However, the performance on test data has better recall (0.73) for Conservative class at the cost of a drop in recall for Labour class. The overall precision remains more or less same for both the classes across the two data sets. Finally, the accuracy on the training and test data sets are close at 83.4% & 83.3% respectively.

**Parameter of choice:**

Solver: This specifies the type of algorithm used to that minimizes cost function. By default this is set to svd. Svd doesn’t compute the covariance matrix.

Tolerance: 1e-04. Same as the default value. Any number larger would have made the algorithm to converge and exit soon without identifying the right coefficients. A smaller number would have caused the model to run very slow.

|  |  |
| --- | --- |
| Training Data | Test Data |
| AUC = 88.9%  Illustration 22: ROC of LDA model using training data | AUC=88.8%  Illustration 23: ROC of LDA model using test data |

The area under the ROC curve is also close for both training and test at 88.9% and 88.8% resp.

KNN Model

Below are the results of the tuned model created using the following parameters.

KNeighborsClassifier(algorithm='auto', leaf\_size=30, metric='minkowski',

metric\_params=None, n\_jobs=None,

n\_neighbors=5, p=2,

weights='uniform')

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Training data | | Actuals | |  |  |  |  |  |  |
| Accuracy | 83.8% | Labour | Conservative |  |  | precision | recall | f1-score | support |
| Predicted | Labour | 698 | 56 |  | Labour | 0.86 | 0.93 | 0.89 | 754 |
| Conservative | 116 | 191 |  | Conservative | 0.77 | 0.62 | 0.69 | 307 |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Test data | | Actuals | |  |  |  |  |  |  |
| Accuracy | 83.1% | Labour | Conservative |  |  | precision | recall | f1-score | support |
| Predicted | Labour | 278 | 25 |  | Labour | 0.84 | 0.92 | 0.88 | 303 |
| Conservative | 52 | 101 |  | Conservative | 0.80 | 0.66 | 0.72 | 153 |

The tuned model is slightly underfit on the data as it has low recall (0.62) for Conservative class. However, the performance on test data has better recall (0.66) for Conservative class at the cost of a drop in recall for Labour class. The overall precision remains more or less same for both the classes across the two data sets. Finally, the accuracy on the training and test data sets are close at 83.8% & 83.1% respectively.

**Parameters of choice**: The gridsearchcv choose to suggest the same set of parameters as default.

N\_neighbors = value was 5

algorithm: The value was ‘auto’ which will attempt to decide the most appropriate algorithm based on the values passed to fit method

leaf\_size: The value was 30.

p: The value of 2, meaning minkowski distance was used.

|  |  |
| --- | --- |
| Training Data | Test Data |
| AUC = 90.1%  Illustration 24: ROC of KNN model using training data | AUC=88.6%  Illustration 25: ROC of KNN model using test data |

The area under the ROC curve is also close for both training and test at 90.1% and 88.6% resp.

Naïve Bayes Model

Below are the results of the tuned model created using the following parameters.

GaussianNB(priors=None, var\_smoothing=1e-09)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Training data | | Actuals | |  |  |  |  |  |  |
| Accuracy | 82.9% | Labour | Conservative |  |  | precision | recall | f1-score | support |
| Predicted | Labour | 668 | 86 |  | Labour | 0.88 | 0.89 | 0.88 | 754 |
| Conservative | 95 | 212 |  | Conservative | 0.71 | 0.69 | 0.70 | 307 |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Test data | | Actuals | |  |  |  |  |  |  |
| Accuracy | 82.5% | Labour | Conservative |  |  | precision | recall | f1-score | support |
| Predicted | Labour | 263 | 40 |  | Labour | 0.87 | 0.87 | 0.87 | 303 |
| Conservative | 40 | 113 |  | Conservative | 0.74 | 0.74 | 0.74 | 153 |

The tuned model suggested the same parameters as the default we had for the base model. The model was slightly underfit on the data as it has low recall (0.69) for Conservative class. However, the performance on test data has better recall (0.74) for Conservative class at the cost of a drop in recall for Labour class. The overall precision remains more or less same for both the classes across the two data sets. Finally, the accuracy on the training and test data sets are close at 82.9% & 82.5% respectively.

**Parameter of choice:**

var\_smoothing: The gridsearchcv was provided with 100 values ranging between 1 and 1e-09. It recommended 1e-09 as the optimal value which is same as the default value.

|  |  |
| --- | --- |
| Training Data | Test Data |
| AUC = 88.8%  Illustration 26: ROC of Naïve Bayes model using training data | AUC=87.5%  Illustration 27: ROC of Naïve Bayes model using test data |

The area under the ROC curve is also close for both training and test at 88.8% and 87.5% resp.

Bagging

Base model

Below are the results of the base model created using the default parameters.

BaggingClassifier(base\_estimator=RandomForestClassifier(),

bootstrap=True, bootstrap\_features=False, max\_features=1.0,

max\_samples=1.0, n\_estimators=10, n\_jobs=None,

oob\_score=False, random\_state=None, verbose=0,

warm\_start=False)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Training data | | Actuals | |  |  |  |  |  |  |
| Accuracy | 97.1% | Labour | Conservative |  |  | precision | recall | f1-score | support |
| Predicted | Labour | 747 | 7 |  | Labour | 0.97 | 0.99 | 0.98 | 754 |
| Conservative | 24 | 283 |  | Conservative | 0.98 | 0.92 | 0.95 | 307 |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Test data | | Actuals | |  |  |  |  |  |  |
| Accuracy | 82.7% | Labour | Conservative |  |  | precision | recall | f1-score | support |
| Predicted | Labour | 275 | 28 |  | Labour | 0.84 | 0.91 | 0.87 | 303 |
| Conservative | 51 | 102 |  | Conservative | 0.78 | 0.67 | 0.72 | 153 |

The base model is overfitting on the training data as it is doing extremely well on training data but the performance is only fair on test data.

**Parameters of importance:**

Base\_estimator: This parameter allows to choose the base estimator for the bagging model. By default the DecisionTreeClassfier is choose, but for base model we specified RandomForestClassfier

N\_estimator: number of the estimators to be picked for bagging. The default value is 10

Max\_samples: maximum numbers of samples to be drawn from the training data with replacement. Default is 1.0 meaning all the records will be included in the sample

Max\_features : maximum numbers of features to be drawn from the training data with replacement. Default is 1.0 meaning all the features will be included in the sample

The parameters for the base estimator can also be specified explicitly. For the purpose of this solution we have kept them as default.

|  |  |
| --- | --- |
| Training Data | Test Data |
| AUC = 99.4%  Illustration 28: ROC of random forest based bagging model using training data | AUC=89.5%  Illustration 29: ROC of random forest based bagging model using test data |

The area under the ROC curve with training data is 99.4% and test data is 89.5%.

Model Tuning

Below are the results of the tuned model created using the following parameters.

BaggingClassifier(base\_estimator=RandomForestClassifier(), bootstrap=False,

bootstrap\_features=True, max\_features=3, max\_samples=60,

n\_jobs=-1, random\_state=1, verbose=True)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Training data | | Actuals | |  |  |  |  |  |  |
| Accuracy | 79.3% | Labour | Conservative |  |  | precision | recall | f1-score | support |
| Predicted | Labour | 719 | 35 |  | Labour | 0.80 | 0.95 | 0.87 | 754 |
| Conservative | 185 | 122 |  | Conservative | 0.78 | 0.40 | 0.53 | 307 |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Test data | | Actuals | |  |  |  |  |  |  |
| Accuracy | 75.4% | Labour | Conservative |  |  | precision | recall | f1-score | support |
| Predicted | Labour | 281 | 22 |  | Labour | 0.76 | 0.93 | 0.83 | 303 |
| Conservative | 90 | 63 |  | Conservative | 0.74 | 0.41 | 0.53 | 153 |

The tuned model appears to be underfit on the data and has very poor recall (0.40) for Conservative class, but does fairly well on the Labour class (0.95). Due to poor recall the overall accuracy of the model is also poor and is 79.3% and 75.4% resp. on training and test data.

**Parameters of choice:**

Base\_estimator: RandomForestClassfier as directed in the problem statement

N\_estimator: The gridsearchcv was provided with 8, 10, 12 options. The recommendation was to use 10 which was sample as default value.

Max\_samples: 60 was used for the model training which is a suitable for a overall sample size of 1525

Max\_features : 3 was used. The general thumb rule is to used sqrt or log of the number of features available in the data set. Since, there were 9 features we used sqrt = 3.

The parameters for the base estimator can also be specified explicitly. For the purpose of this solution we have kept them as default.

|  |  |
| --- | --- |
| Training Data | Test Data |
| AUC = 87.6%  Illustration 30: ROC of random forest based bagging model using training data | AUC=82.8%  Illustration 31: ROC of random forest based bagging model using test data |

The area under the ROC curve with training data is 87.6% and test data is 82.8%.

Gradient Boost

Base model

Below are the results of the base model created using the default parameters.

GradientBoostingClassifier(ccp\_alpha=0.0, criterion='friedman\_mse', init=None,

learning\_rate=0.1, loss='deviance', max\_depth=3,

max\_features=None, max\_leaf\_nodes=None,

min\_impurity\_decrease=0.0, min\_impurity\_split=None,

min\_samples\_leaf=1, min\_samples\_split=2,

min\_weight\_fraction\_leaf=0.0, n\_estimators=100,

n\_iter\_no\_change=None, presort='deprecated',

random\_state=None, subsample=1.0, tol=0.0001,

validation\_fraction=0.1, verbose=0,

warm\_start=False)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Training data | | Actuals | |  |  |  |  |  |  |
| Accuracy | 89.3% | Labour | Conservative |  |  | precision | recall | f1-score | support |
| Predicted | Labour | 708 | 47 |  | Labour | 0.91 | 0.94 | 0.93 | 754 |
| Conservative | 68 | 239 |  | Conservative | 0.84 | 0.78 | 0.81 | 307 |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Test data | | Actuals | |  |  |  |  |  |  |
| Accuracy | 83.6% | Labour | Conservative |  |  | precision | recall | f1-score | support |
| Predicted | Labour | 276 | 27 |  | Labour | 0.85 | 0.91 | 0.88 | 303 |
| Conservative | 48 | 105 |  | Conservative | 0.80 | 0.69 | 0.74 | 153 |

This base model is so far has performed best among all the models base or tuned. With overall accuracy of almost 88% on training data it also satisfactory recall (0.78) on Conservative class. The performance on the test data is comparable to traning data with recall (0.69) on Conservative class.

**Parameters of importance:**

loss: The loss function to be optimized. Default value is **deviance**

N\_estimator: number of the estimators. The default value is 100

Subsample: fraction of samples used to be fit on the base learners. Default is 1.0 meaning all the records will be included in the sample

criterion: The function used to measure the quality of split. Default is **freidman\_mae**

min\_samples\_split: minimum number of samples required to split an internal node. Default is 2

max\_features: The number of features to consider when looking for the best split

|  |  |
| --- | --- |
| Training Data | Test Data |
| AUC = 95.1%  Illustration 32: ROC of Gradient boost model using training data | AUC=89.9%  Illustration 33: ROC of Gradient Boost model using test data |

The area under the ROC curve with training data is 95.1% and test data is 89.9%.

Model Tuning

Below are the results of the tuned model created using the following parameters.

GradientBoostingClassifier(criterion='mse', loss='exponential',

max\_features='sqrt', min\_samples\_split=10,

n\_estimators=80, random\_state=1, subsample=1,

verbose=True)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Training data | | Actuals | |  |  |  |  |  |  |
| Accuracy | 87.9% | Labour | Conservative |  |  | precision | recall | f1-score | support |
| Predicted | Labour | 707 | 47 |  | Labour | 0.90 | 0.94 | 0.92 | 754 |
| Conservative | 81 | 226 |  | Conservative | 0.83 | 0.74 | 0.78 | 307 |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Test data | | Actuals | |  |  |  |  |  |  |
| Accuracy | 83.3% | Labour | Conservative |  |  | precision | recall | f1-score | support |
| Predicted | Labour | 273 | 30 |  | Labour | 0.86 | 0.90 | 0.88 | 303 |
| Conservative | 46 | 107 |  | Conservative | 0.78 | 0.70 | 0.74 | 153 |

The tuned model performs fairly on the training data and has satisfactory recall (0.74) for Conservative class, and does eqally well on the test data (0.70). The overall accuracy of the model is also acceptable and is 87.9% and 83.3% resp. on training and test data.

**Parameters of choice:**

loss: The value was ‘exponential’, meaning internally for loss recovery Adaboost algorithm was used

N\_estimator: The value was 80

Subsample: The value was 1.0 meaning all the records will be included in the sample

criterion: mse, mean squared error which is the most trusted approach to adjudge the performance of the models

min\_samples\_split: The value was 10, Ideally, it should have been a higher number but considering the overall sample size which is small, 10 is the recommendation from the gridsearchcv

max\_features: The value was sqrt = 3, since the we have 9 features in the training data set.

|  |  |
| --- | --- |
| Training Data | Test Data |
| AUC = 93.9%  Illustration 34: ROC of Gradient Boost model using training data | AUC=90.1%  Illustration 35: ROC of Gradient boost model using test data |

The area under the ROC curve with training data is 93.9% and test data is 90.1%.

Ada Boost

Base model

Below are the results of the base model created using the default parameters.

AdaBoostClassifier(algorithm='SAMME.R',

base\_estimator=RandomForestClassifier(),

learning\_rate=1.0, n\_estimators=50, random\_state=None)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Training data | | Actuals | |  |  |  |  |  |  |
| Accuracy | 100.0% | Labour | Conservative |  |  | precision | recall | f1-score | support |
| Predicted | Labour | 754 | 0 |  | Labour | 1 | 1 | 1 | 754 |
| Conservative | 0 | 307 |  | Conservative | 1 | 1 | 1 | 307 |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Test data | | Actuals | |  |  |  |  |  |  |
| Accuracy | 82.0% | Labour | Conservative |  |  | precision | recall | f1-score | support |
| Predicted | Labour | 272 | 31 |  | Labour | 0.84 | 0.90 | 0.87 | 303 |
| Conservative | 51 | 102 |  | Conservative | 0.77 | 0.67 | 0.71 | 153 |

The base model is overfitting on the training data as it is doing extremely well on training data but the performance is only fair on test data.

**Parameters of importance:**

Base\_estimator: This parameter allows to choose the base estimator for the bagging model. By default the DecisionTreeClassfier is choose, but for base model we specified RandomForestClassfier

N\_estimator: number of the estimators to be picked for bagging. The default value is 50

Algorithm: Type of algorithm for convergence. Default value is SAMME.R as it is fast and is able to achieve a lower test error with fewer boosting iterations.

The parameters for the base estimator can also be specified explicitly. For the purpose of this solution we have kept them as default.

|  |  |
| --- | --- |
| Training Data | Test Data |
| AUC = 100%  Illustration 36: ROC of Adaboost model using training data | AUC=88.9%  Illustration 37: ROC of Adaboost model using test data |

The area under the ROC curve with training data is 100% and test data is 88.9%.

Model Tuning

Below are the results of the tuned model created using the following parameters.

AdaBoostClassifier(algorithm='SAMME',

base\_estimator=RandomForestClassifier(),

learning\_rate=1, n\_estimators=30, random\_state=1)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Training data | | Actuals | |  |  |  |  |  |  |
| Accuracy | 100% | Labour | Conservative |  |  | precision | recall | f1-score | support |
| Predicted | Labour | 754 | 0 |  | Labour | 1 | 1 | 1 | 754 |
| Conservative | 0 | 307 |  | Conservative | 1 | 1 | 1 | 307 |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Test data | | Actuals | |  |  |  |  |  |  |
| Accuracy | 81.8% | Labour | Conservative |  |  | precision | recall | f1-score | support |
| Predicted | Labour | 271 | 32 |  | Labour | 0.84 | 0.89 | 0.87 | 303 |
| Conservative | 51 | 102 |  | Conservative | 0.76 | 0.67 | 0.71 | 153 |

Similar to the base model, the tuned model is also overfitting on the training data as it is doing extremely well on training data but the performance is only fair on test data.

Base\_estimator: RandomForestClassfier kept is same as Bagging model

N\_estimator: The value was 30

Algorithm: The value was SAMME that uses discrete boosting algorithm

The parameters for the base estimator can also be specified explicitly. For the purpose of this solution we have kept them as default.

|  |  |
| --- | --- |
| Training Data | Test Data |
| AUC = 100%  Illustration 38: ROC of Adaboost model using training data | AUC=78.1%  Illustration 39: ROC of Adaboost model using test data |

The area under the ROC curve is also close for both training and test at 100% and 78.1% resp.

Question 1.7

Performance Metrics

Based on the below grid that compares the accuracy and area under ROC, we can see that the tuned version of Gradient boost performs best on both training and test data

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Base Models |  |  |  |  |
|  | Training Data | | Test Data | |
| Model | Acc. | AUC | Acc. | AUC |
| LogReg | 83.1% | 89.0% | 83.1% | 88.3% |
| LDA | 83.4% | 88.9% | 83.3% | 88.8% |
| KNN | 85.6% | 92.7% | 82.5% | 87.0% |
| NB | 82.9% | 88.9% | 82.5% | 87.5% |
| Bagging | 97.1% | 99.4% | 82.7% | 89.5% |
| Gboost | 89.3% | 95.1% | 83.6% | 89.9% |
| Adaboost | 100.0% | 100.0% | 82.0% | 88.9% |
|  |  |  |  |  |
| Tuned Models |  |  |  |  |
|  | Training Data | | Test Data | |
| Model | Acc. | AUC | Acc. | AUC |
| LogReg | 84.0% | 88.9% | 82.9% | 88.2% |
| LDA | 83.4% | 88.9% | 83.3% | 88.8% |
| KNN | 83.8% | 90.1% | 83.1% | 88.6% |
| NB | 82.9% | 88.8% | 82.5% | 87.5% |
| Bagging | 79.3% | 87.6% | 75.4% | 82.8% |
| Gboost | 87.9% | 93.9% | 83.3% | 90.1% |
| Adaboost | 100.0% | 100.0% | 81.8% | 78.1% |

*Table 3: Accuracy and AUC comparison of base and tuned models*

Bagging did really well on the case models but the performance deteriorated in the tuned model suggesting the base version of its model wasn’t generalized and would have performed poor on the unseen data sets.

The best accuracy of Gradient boost model is also a function of it providing the best precision and recall values for both the target classes among all the models.

The final model of choice should be the tuned model of Gradient boost. The tuned version has undergone cross-validation and is much more generalized than the base model even though the base model has slightly better performance than the tuned version. Also, the tuned version has a better AUC on test data when compared to the AUC on test data of the base model which suggests the probability of it making the right prediction on the unseen data would be better.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Labour Class |  |  |  |  |  |  |
|  | Training Data | | | Test Data | | |
| Tuned Model | Precision | Recall | F1 Score | Precision | Recall | F1 Score |
| LogReg | 0.86 | 0.92 | 0.89 | 0.86 | 0.89 | 0.87 |
| LDA | 0.86 | 0.91 | 0.89 | 0.86 | 0.89 | 0.88 |
| KNN | 0.86 | 0.93 | 0.89 | 0.84 | 0.92 | 0.88 |
| NB | 0.88 | 0.89 | 0.88 | 0.87 | 0.87 | 0.87 |
| Bagging | 0.8 | 0.95 | 0.87 | 0.76 | 0.93 | 0.83 |
| Gboost | 0.91 | 0.94 | 0.93 | 0.85 | 0.91 | 0.88 |
| Adaboost | 1 | 1 | 1 | 0.84 | 0.89 | 0.87 |
|  |  |  |  |  |  |  |
| Conservative Class |  |  |  |  |  |  |
|  | Training Data | | | Test Data | | |
| Tuned Model | Precision | Recall | F1 Score | Precision | Recall | F1 Score |
| LogReg | 0.77 | 0.64 | 0.7 | 0.76 | 0.71 | 0.74 |
| LDA | 0.74 | 0.65 | 0.69 | 0.77 | 0.73 | 0.74 |
| KNN | 0.77 | 0.62 | 0.69 | 0.8 | 0.66 | 0.72 |
| NB | 0.71 | 0.69 | 0.7 | 0.74 | 0.74 | 0.74 |
| Bagging | 0.78 | 0.4 | 0.53 | 0.74 | 0.41 | 0.53 |
| Gboost | 0.84 | 0.78 | 0.81 | 0.8 | 0.69 | 0.74 |
| Adaboost | 1 | 1 | 1 | 0.76 | 0.67 | 0.71 |

This can be further asserted by looking at the individual precision and recall values of the tuned models.

*Table 4: Precision, Recall and F1 score comparison of tuned models*

It is again evident that the Gradient boost’s tuned model does consistently well in predicting both the classes on training and test data.

Question 1.8

Based on these predictions, what are the insights?

Following are the inferences that can be made based on the analysis so far.

* 1. The Labour party has a supporter population that is comparatively of younger age than the supporters of Conservation party
  2. Majority of the Conservative party supporters are cautious about the European integration
  3. Additional features like, ratings of employment generation, internal security, education system, health system, infrastructure, climate change, etc. would help us get better predictions
  4. Enough effort should be put in collecting the right mix of data. For example, ensuring both classes are equally represented, responses from all the age groups, ethnicity, origin, gender, economic class, education background, employed/unemployed, type of employment, geographical regions, urban/rural etc. are taken into consideration

With the above features and factors much more interesting insights can be derived and would make the model richer and would perform well on the unseen data.

Problem 2: Text Analytics

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

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

Question 2.1

Find the number of characters, words and sentences for the mentioned document

We start with importing the data and running EDA on the inaugural speeches of Roosevelt(1941), Kennedy(1961) & Nixon(1973). The below grid gives us a high-level overview of the stats from these speeches

|  |  |  |  |
| --- | --- | --- | --- |
| **President (Year)** | **Sentences** | **Words** | **Characters** |
| **Roosevelt (1941)** | 68 | 1536 | 6174 |
| **Kennedy (1961)** | 52 | 1546 | 6202 |
| **Nixon (1973)** | 69 | 2028 | 8122 |

*Table 5: Basic stats of speeches*

From the above grid, it appears that though Roosevelt used as many sentences in his speech as Nixon, the total number of words were much less. It is can be inferred that Roosevelt’s speech had much shorter sentences when compared to other speeches.

Question 2.2

Remove all the stopwords from all three speeches

Moving ahead with the text analysis. The first step would be to tokenize the words, then remove and punctuations and finally removing the stopwords. The below grid captures the outcome of these operations in detail

|  |  |  |  |
| --- | --- | --- | --- |
| **President (Year)** | **Tokens** | **Tokens without punctuations** | **Tokens without stopwords** |
| **Roosevelt (1941)** | 1344 | 1344 | 626 |
| **Kennedy (1961)** | 1370 | 1370 | 689 |
| **Nixon (1973)** | 1816 | 1816 | 845 |
| *Table 6: Word counts before & after removing stopwords* | | | |

It can be observed that the count of the stopwords is a little more than 50% of all the words in each of these speeches.

A sample sentence with first 20 words(after removing the stopwords) from each speech is below.

Roosevelt’s speech –

"national day inauguration since people renewed sense dedication united states washington 's day task people create weld together nation lincoln"

Kennedy’s speech –

"vice president johnson mr. speaker mr. chief justice president eisenhower vice president nixon president truman reverend clergy fellow citizens observe"

Nixon’s speech –

"mr. vice president mr. speaker mr. chief justice senator cook mrs. eisenhower fellow citizens great good country share together met"

Question 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)

On further analysis, we identify the words which are most often used in each of these speeches. The below grid shows top 3 words in these speeches made by the presidents.

|  |  |  |  |
| --- | --- | --- | --- |
| **President (Year)** | **Top 1** | **Top 2** | **Top 3** |
| **Roosevelt (1941)** | nation(12) | know(10) | life(9),  spirit(9),  democracy(9) |
| **Kennedy (1961)** | let(16) | us(12) | world(8), sides(8) |
| **Nixon (1973)** | us(26) | let(22) | america(21) |

*Table 7: Top words*

The grid has top word(times it occurs in the speech).

Question 2.4

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

Below are the word clouds of the three speeches

Word cloud of Roosevelt’s speech



*Illustration 40: Word cloud from Roosevelt’s speech*

Word cloud of Kennedy’s speech



*Illustration 41: Word cloud from Kennedy’s speech*

Word cloud of Nixon’s speech



*Illustration 42: Word cloud from Nixon’s speech*

End of the business report

Illustration Index

Illustration 1: First 5 records from the data 3

Illustration 2: Basic information about the data frame 4

Illustration 3: Results of describe method on all the features 4

Illustration 4: Distribution of respondents by Age and Political choice 5

Illustration 5: Ratings of National Economic Condition 5

Illustration 6: Ratings of Household Economic Condition 6

Illustration 7: Ratings of The Labour Party’s candidate - Blair 6

Illustration 8: Ratings of The Conservative Party’s candidate - Hague 7

Illustration 9: Skepticism around European Integration 7

Illustration 10: Knowledge of Party’s position among the supporters 8

Illustration 11: Distribution of respondents by Gender 8

Illustration 12: ROC of Logistic Regression model using training data 11

Illustration 13: ROC of Logistic Regression model using test data 11

Illustration 14: ROC of LDA model using training data 12

Illustration 15: ROC of LDA model using test data 12

Illustration 16: ROC of KNN model using training data 13

Illustration 17: ROC of KNN model using test data 13

Illustration 18: ROC of Naïve Bayes model using training data 14

Illustration 19: ROC of Naïve Bayes model using test data 14

Illustration 20: ROC of Logistic Regression model using training data 15

Illustration 21: ROC of Logistic Regression model using test data 15

Illustration 22: ROC of LDA model using training data 16

Illustration 23: ROC of LDA model using test data 16

Illustration 24: ROC of KNN model using training data 17

Illustration 25: ROC of KNN model using test data 17

Illustration 26: ROC of Naïve Bayes model using training data 18

Illustration 27: ROC of Naïve Bayes model using test data 18

Illustration 28: ROC of random forest based bagging model using training data 19

Illustration 29: ROC of random forest based bagging model using test data 19

Illustration 30: ROC of random forest based bagging model using training data 20

Illustration 31: ROC of random forest based bagging model using test data 21

Illustration 32: ROC of Gradient boost model using training data 22

Illustration 33: ROC of Gradient boost model using test data 22

Illustration 34: ROC of Gradient boost model using training data 23

Illustration 35: ROC of Gradient boost model using test data 23

Illustration 36: ROC of Adaboost model using training data 24

Illustration 37: ROC of Adaboost model using test data 24

Illustration 38: ROC of Adaboost model using training data 25

Illustration 39: ROC of Adaboost model using test data 25

Illustration 40: Word cloud from Roosevelt’s speech 28

Illustration 41: Word cloud from Kennedy’s speech 28

Illustration 42: Word cloud from Nixon’s speech 29

Table Index

Table1\_ProblemStatement1\_DataDictionary 3

Table 2: Chi-Squared Contingency test results 9

Table 3: Accuracy and AUC comparison of base and tuned models 25

Table 4: Precision, Recall and F1 score comparison of tuned models 26

Table 5: Basic stats of speeches 27

Table 6: Word counts before and after removing stopwords 27

Table 7: Top words 28