**Machine Learning**

# Project Report

**Problem 1**

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

#### Data Ingestion: 12 mark

**Q1.1 Read the dataset. Do the descriptive statistics and do null value condition check. Write an inference on it. (5 Marks)**

We import the necessary libraries to read the data and perform other basic analysis on the data such as univariate, bivariate analysis along with graphical representation of the data.

We also import all the necessary libraries to fit the various models

To import the data, we use the pd.read function. Since our excel sheet has multiple sheets, we specify the sheet name from which we want to read the data.

We drop the unnamed:0 column as it does not add any value to our analysis of the data.

Further we use df.head function to view the first 5 rows of the data

A screenshot of a cell phone

Description automatically generated

The data has 9 variables, Vote, age, economic.cond.national,economic,cond.household, Blair, Hague, Europe, political.knowledge and gender. The data type for Vote and Gender variables in object, whereas the remaining the variables are all in integer datatype. The shape of the data is 1525 rows and 9 columns

Descriptive Statistics for the data set-

A screenshot of a cell phone

Description automatically generated

We use the df.describe function to get the 5 point summary for all the variables.

We check for null values using the df.isnull().sum() function

A picture containing bird

Description automatically generated

We can see that there are no null values present in the data.

We also check for duplicate rows in the data using the df.duplicated function. We see there are 8 duplicate rows. On further analysis we note that these values are not actually duplicate hence we don’t drop them

**Q1.2 Perform Univariate and Bivariate Analysis. Do exploratory data analysis. Check for Outliers. (7 Marks)**

**Univariate analysis**

Using the df.hist function we see the distribution of each of the variables. This helps us to identify if any variables are skewed or not

A picture containing crossword

Description automatically generated

From the histograms we can see that variable Europe is highly left skewed. Europe is also an ordinal variable however, the values on the scale seem to continuous

The variables Blair, Hague, Economic.cond.household , economic.cond.national and political.knowledge are ordinal in nature hence the histrogram has breaks in it. We can observe that both economic.cond variables are more normally distributed than the rest.

For Blair, Hague and Political.knowledge the variables are unevenly distributed across all the values.

Age seems to be slightly right skewed, indicating the sample has a majority of younger voters.

Europe is also an ordinal variable however, the values on the scale seem to continuous

Since our objective is to classify which party will a voter vote for, the variable vote is our target variable. We will check the proportion of the values in the variable.

In the dataset we can see that there is an imbalance. Out of 1525 votes, 70% votes are for the Labour party whereas 30% votes are for Conservative party

**Bivariate Analysis**

Further for bivariate analysis, we compare our target variable with other variables to gains some insights –

We use the countplot function to plot the bar charts. With the help of the hue variable we can add an additional dimension to the graphs

A screenshot of a cell phone

Description automatically generated

No. of votes by women are higher than that of men for both parties.

Using the bins function, we can also divide the age in different brackets to see if there is any trend in the voting behaviour with respect to age groups –

A picture containing drawing

Description automatically generated

From this we can see that more people in the age group 34-54 have voted for the labour party whereas for the conservative party, higher share of the vote is from the age group 64-73.

A screenshot of a video game

Description automatically generated

We also plotted the age bracket against the candidate. We can see that Blair, the labour party candidate has received a rating of 4 across all the age groups.

For the conservative party candidate, the responses received were mixed with rating of 2 as well as 4.

We can plot more permutations and combinations of the variables to see the relationships

**Multivariate analysis**

We use the pairplot function and heatmap for bivariate analysis

**A screenshot of a cell phone

Description automatically generated**

We only look at one side of the diagonal in order to spot any relationships between the variables. Since most of the variables are ordinal, there is no clear relationship between variables outline in the pair plot

Further to quantify the correlation, we plot the heatmap using the sns.heatmap function

A picture containing display

Description automatically generated

From the heat map we can observe there is no high correlation between any of the variables

**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)**

Since we have categorical data in the data frame, we will first convert these values to numerical data and then proceed to scale the data. This picks up the unique the values in respective columns and assigns a code to them, which is essentially replacing them with a numerical value. In the given data set, we have vote, and gender columns as object data type. After applying the function the datatype of these columns changes from object to integer

Further we scale the continuous variables in the data. In this case our variables here are ordinal therefore in they would not require scaling. However in my opinion since there are multiple variable with different scales, it is better to scale the variable in order to avoid any error in the analysis data. The scaling only affects distance based models in the first place and the other models would be independent to scaling.

Here we use the min max scaler to scale the variables. The encoded variables are excluded from scaling.

Further we proceed to fit the model on the data set.

We first need to split the data into training and testing data in the proportion of 70% train and 30% test data. We use the sklearn.train\_test\_split function to split the data into train and test. The training set contains a known output and the model learns on this data in order to be generalized to other data later on. We have the test dataset (or subset) in order to test our model's prediction on this subset

Once we do all this, we define our X and y data frames by dropping and populating the target variable in X and y respectively

We then split the data into test and train

To fit any model, overall these steps are followed

1.We split the data into test and train

2. We assign a variable with the respective classifier

3. We fit the model on the train data – This helps the model learn in order to make predictions for the test data

4.We then fit the model on the test data

5. We then get the predicted classes and probabilities on the test data –

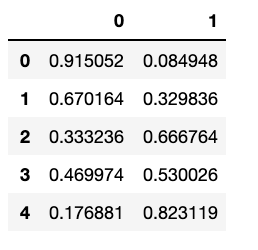
6. We then perform the model evaluation using the ROC and AUC for the training and test data respectively – This gives us the area under the curve

7. We then calculate the confusion matrix for train and test data

8. Finally we execute the command for the classification report and accuracy for both train and test data

**1.4**Apply Logistic Regression and LDA (Linear Discriminant Analysis).

We fit the logistic regression on the train data. Further we predict the target variables using the model for both train and test data. We then calculate the predicted classed and their respective probabilities



**Linear Discriminant Analysis**

For linear discriminant analysis we fit the model on X and y without splitting intro train and test data

We then calculate the predicted values and attach it to the X dataframe

Please refer the appendix for code for each of the steps for respective classification models

**Q 1.5 Apply KNN Model, Naïve Bayes Model and Support Vector Machine (SVM) model.**

**KNN**

We import the KneighborsClassifier and fit the model on the train and test data.

**Naïve Bayes**

We use the GuassianNB classifier to fit the naïve bayes model on the train and test data

**SVM**

We use the svm.SVC classifier to fit the SVM model on the train and test data

Kindly refer to the appendix for the code for each model

**Q1.6 Model Tuning, Bagging and Boosting.**

**For model tuning we follow the below steps –**

We define the parameter grid and assign a variable with the respective classifier (Decision Tree, Random Forest, or Artificial Neural Networks)

2. Perform grid search cross validation, using the classification model as the estimator - – This helps us identify the best parameters to be used for model classification

3. We then fit the grid search variable on the train data, and find the best parameters basis the values mentioned by us in the parameter grid – This helps the train model learn,in order to make predictions for the test data

4. Using the best parameter values we use it to predict on the test and train data

Note: Here model tuning is not performed for Naïve Bayes model as there are no hyper parameters to be tuned.

For each of the models above, we try to tune the models using Gridsearch cross validation in order to identify the best parameters.

**Bagging**

For bagging we use the Random Forest Classifier. Bootstrap aggregating or bagging is an ensemble technique used to improve the stability and accuracy of machine learning algorithm used for classification by reducing the variance.

With the help for Random Forest classifier and bootstrap =True as default, we fit the model on the train and test data

**Boosting**

Boosting is also an ensemble technique used for reducing bias in the data. It helps to convert weak learners into strong ones.

For Boosting we use the Xg boost of Extreme Gradient boosting algorithm.

We fit the Xg boost classifier on the train and test data.

**Q1.7 Performance Metrics: Check the performance of Predictions on Train and Test sets using Accuracy, Confusion Matrix, Plot ROC curve and get ROC\_AUC score for each model. Final Model: Compare the models and write inference which model is best/optimized.**

**For performance metrics the following steps are followed-**

1. We get the predicted classes and probabilities on the test data for each respective model

2. We then perform the model evaluation using the ROC and AUC for the training and test data respectively – This gives us the area under the curve

3. We then calculate the confusion matrix for train and test data

4. Finally we execute the command for the classification report and accuracy for both train and test data

Based on the AUC, Accuracy, sensitivity, precision and f1-score values, of the train and test data, we draw our conclusions about the model

Please refer the appendix for the code and model wise performance metrics

Please find below a summary of all the models used and their performance metrics

A screenshot of a cell phone

Description automatically generated

In the given problem, we are looking to classify that given the variables, which party would a voter vote for. There is no specific class of interest for us. We are more interested in correctly predicting the outcome.

We have encoded our target variable ‘Vote” wherein the Labour party is represented by 1 and conservative party by 0.

We should also note that our data is an imbalanced dataset. Therefore looking only at accuracy may not the best choice as it may not show the correct picture.

All the performance metric documented here pertain to class 0, i.e. Conservative party. This would give a better in highlighting the effectiveness of the model as it would show how correctly was a lesser represented class in the data predicted

Looking at the AUC scores, the highest is for KNN Train model. Even the other parameters for KNN model are high however, when we compare the test scores, the precision value reduces significantly as opposed to the other models.

Looking at all the parameters, The KNN tuned model seems to be the best optimized model with a high AUC score, relatively high precision, recall and f-1 score values with not much difference in train and test scores

**Q1.8 Based on these predictions, what are the insights?**

Based on the data available the following insights would be made –

1. Since the dataset is imbalanced, the model is more likely to predict that a voter would have voted for the Labour party. Basis that the labour party would be the predicted winner as per the exit poll with a majority of seats.
2. However If we are using tuned KNN model, with a higher precision value of 0.78 (The proportion of people we predicted as having voted for Labour actually voted for Labour) and higher recall value of 0.7 (Proportion of people that had voted for Labour party and had been correctly identified as that) this would give a clearer picture
3. Since the survey was only carried out for 1525 voters, and the dataset is an imbalanced one, this may not be the correct dataset to predict the party a voter is going to vote for. This is also visible from the Random forest algorithm wherein we are getting 100% accuracy for train data.

**Problem 2:**

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

We import nltk library which is the natural language toolkit used for symbolic and statistic natural language processing. We import the text from inaugural library from nltk.corpus. We also import all the other necessary libraries such as tokenize, stop words FreqDist and Wordcloud which will help us in processing the text.

We then import the texts for the 3 president’s speeches and store them in variables, namely Roosevelt, Kennedy and Nixon

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

To find the no. of characters, we use the len function for inaugural.raw. The functions returns the no. of characters in a string as we apply it to the raw data.

A picture containing knife, table

Description automatically generated

As we can see, Roosevelt’s speech contains, 7571 characters, Kennedy’s – 7618 and Nixon’s 9991. The number of characters include the spaces and special characters as well as punctuations

To find the no. of words we use the len function for inaugural.words . The .words function separated the words in the text and the len functions gives us a count of the words in the entire text.

A picture containing knife, table

Description automatically generated

From the result we can observe that, Roosevelt has 1536 words, Kennedy has 1546 and Nixon has 2028

Similarly, to find the no. of sentences we use the len function for inaugural.sents. The .sents functions separates the sentences in the whole text and the len function gives us a count of the sentences.

A picture containing knife, table

Description automatically generated

Nixon has the most no. of sentences, followed by Roosevelt and then Kennedy.

One interpretation that can be made here is that, even though Kennedy had more number of words than Roosevelt, he has lesser sentences which may mean that his sentences were much longer – which may not be as engaging to the crowd.

**Q 2.2 Remove all the stop words from the three speeches**

As part of pre-processing text data, we must remove the following from raw text in order to gain more insight

1. Special characters – The presence of special characters in the text do not add any insight while processing the test
2. Single characters (a, i) – Similar to special characters, single characters in text do not bring out any specific sentiment or insight
3. Multiple spaces – Multiple spaces may be counted as independent character, or may cause more problems down the line. It may lead to the same word being counted as 2 due to the presence of a space while calculating unique word count.
4. Prefix /n – For this text we have the prefix n in the beginning of some sentences. Hence we need to remove those as they are not part of the original speech

Further, we also need to convert all the text to lower case. This may be looked at as scaling the data in order to standardize it.

Now we move to removing the stop words. Stop words refers to commonly used words such as the, an, in etc.

We initialize stop words from the English language and assign it to a variable stop.

We then convert each word to a token. The tokenizer function helps to split the larger body of text into smaller lines. In our case, we are splitting it into words.

We then run a for and if function to remove the stop words from each token and then append the non-stop words to the filtered sentence list. We then join these words with spaces, using the .join function to form sentences

**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)**

In order to identify which words occur the most no. of times we use the FreqDist function. This function is used to find the frequency of words within a text.

We run the FreqDist function on the filtered sentences for each speech respectively.

To find the top 3 words we use the most\_common function and pass the argument as 3 in order for it to return the top 3 most common words for each speech.

A screenshot of a tree

Description automatically generated

Basis the result we can the top 3 words used by each president in their address

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

A word cloud is an image composed of words in used in a particular text or subject in which the size of each word represents its frequency

We plotted the word cloud for each president’s address, and following are the results –

**A screenshot of a cell phone

Description automatically generatedA close up of a sign

Description automatically generatedA close up of a sign

Description automatically generated**

As per word cloud, top 3 words for each president are as follows –

1. Roosevelt – Nation, know, spirit
2. Kennedy – let,us, world
3. Nixon – Us, let, america