Machine Learning

September 2021

BUSINESS REPORT

Athisya Nadar

Post Graduate Program in Data Science and Business Analytics [Online]

athisya@gmail.com

CONTENTS

Problem 1

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

1.1 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.

In this problem, the goal is 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. Let us begin by looking at the data dictionary:

Data Dictionary

1. vote: Party choice: Conservative or Labour
2. age: in years
3. economic.cond.national: Assessment of current national economic conditions, 1 to 5.
4. economic.cond.household: Assessment of current household economic conditions, 1 to 5.
5. Blair: Assessment of the Labour leader, 1 to 5.
6. Hague: Assessment of the Conservative leader, 1 to 5.
7. Europe: an 11-point scale that measures respondents' attitudes toward European integration. High scores represent 'Eurosceptic' sentiment.
8. political.knowledge: Knowledge of parties' positions on European integration, 0 to 3.
9. gender: female or male.

The variables include age, gender, assessment of current national and household economic conditions, assessment of the labour and conservative leaders, political knowledge and attitude toward European Integration, in total, we have 9 variables. Let us now look at the head of the data:

	Unnamed: 0	vote	age	economic.cond.national	economic.cond.household	Blair	Hague	Europe	political.knowledge	gender
0	1	Labour	43	3	3	4	1	2	2	female
1	2	Labour	36	4	4	4	4	5	2	male
2	3	Labour	35	4	4	5	2	3	2	male
3	4	Labour	24	4	2	2	1	4	0	female
4	5	Labour	41	2	2	1	1	6	2	male

We can see that apart from age all other variables are categorical:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1525 entries, 0 to 1524
Data columns (total 10 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   Unnamed: 0               1525 non-null   int64
 1   vote                     1525 non-null   object
 2   age                      1525 non-null   int64
 3   economic.cond.national   1525 non-null   int64
 4   economic.cond.household  1525 non-null   int64
 5   Blair                    1525 non-null   int64
 6   Hague                    1525 non-null   int64
 7   Europe                   1525 non-null   int64
 8   political.knowledge      1525 non-null   int64
 9   gender                   1525 non-null   object
dtypes: int64(8), object(2)
memory usage: 119.3+ KB
```

	count	mean	std	min	25%	50%	75%	max
vote	1525.0	0.697049	0.459685	0.0	0.0	1.0	1.0	1.0
age	1525.0	54.182295	15.711209	24.0	41.0	53.0	67.0	93.0
economic.cond.national	1525.0	3.245902	0.880969	1.0	3.0	3.0	4.0	5.0
economic.cond.household	1525.0	3.140328	0.929951	1.0	3.0	3.0	4.0	5.0
Blair	1525.0	3.334426	1.174824	1.0	2.0	4.0	4.0	5.0
Hague	1525.0	2.746885	1.230703	1.0	2.0	2.0	4.0	5.0
Europe	1525.0	6.728525	3.297538	1.0	4.0	6.0	10.0	11.0
political.knowledge	1525.0	1.542295	1.083315	0.0	0.0	2.0	2.0	3.0
gender	1525.0	0.467541	0.499109	0.0	0.0	0.0	1.0	1.0

From the description, we can see that :

- Age of the voters ranges from 24 to 93
- assessment of the labour and conservative leaders ranges from 0 to 5
- assessment of current national and household economic conditions ranges from 0 to 5

```
Election_Data.isnull().sum()
```

```
vote                       0
age                        0
economic.cond.national     0
economic.cond.household    0
Blair                      0
Hague                      0
Europe                     0
political.knowledge        0
gender                     0
dtype: int64
```

There are no null values in the dataset

```
Election_Data.skew()
```

```
age                        0.144621
economic.cond.national    -0.240453
economic.cond.household   -0.149552
Blair                     -0.535419
Hague                      0.152100
Europe                    -0.135947
political.knowledge       -0.426838
dtype: float64
```

If the skewness is between -0.5 and 0.5, the data are fairly symmetrical. If the skewness is between -1 and – 0.5 or between 0.5 and 1, the data are moderately skewed. If the skewness is less than -1 or greater than 1, the data are highly skewed.

Age , assessment of current national and household economic conditions ,Blair, Hague, Europe, political knowledge are fairly skewed.

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.

We can clearly see that there are no null values in any of the columns:

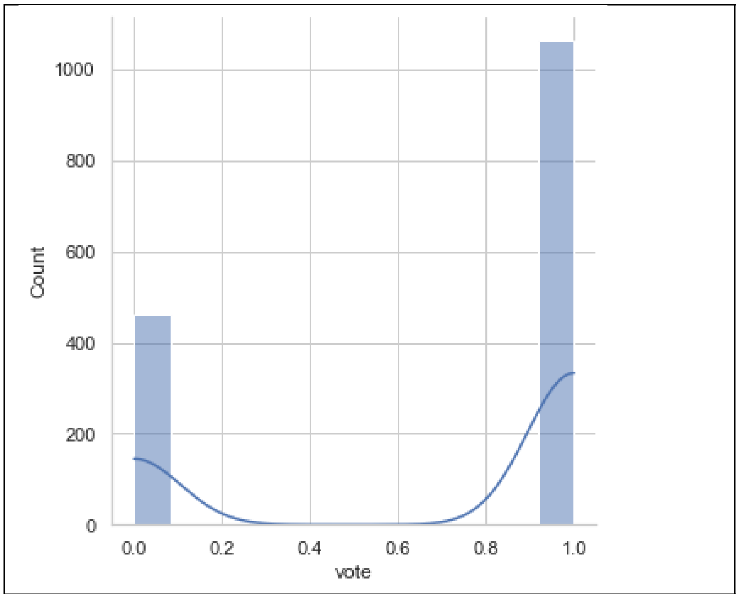
```
Election_Data.isnull().sum()
```

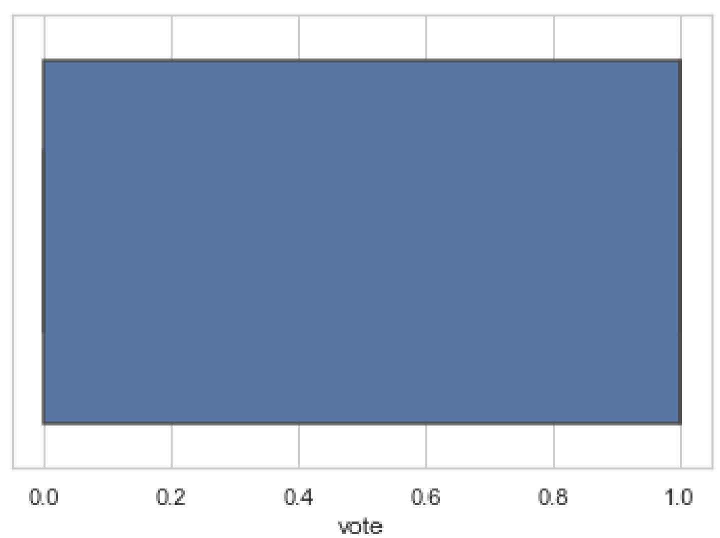
```
vote                       0
age                        0
economic.cond.national     0
economic.cond.household    0
Blair                      0
Hague                      0
Europe                     0
political.knowledge        0
gender                     0
dtype: int64
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1525 entries, 0 to 1524
Data columns (total 10 columns):
 #   Column                   Non-Null Count  Dtype 
---  ------                   --------------  ----- 
 0   Unnamed: 0               1525 non-null   int64 
 1   vote                     1525 non-null   object
 2   age                      1525 non-null   int64 
 3   economic.cond.national   1525 non-null   int64 
 4   economic.cond.household  1525 non-null   int64 
 5   Blair                    1525 non-null   int64 
 6   Hague                    1525 non-null   int64 
 7   Europe                   1525 non-null   int64 
 8   political.knowledge      1525 non-null   int64 
 9   gender                   1525 non-null   object
dtypes: int64(8), object(2)
memory usage: 119.3+ KB
```

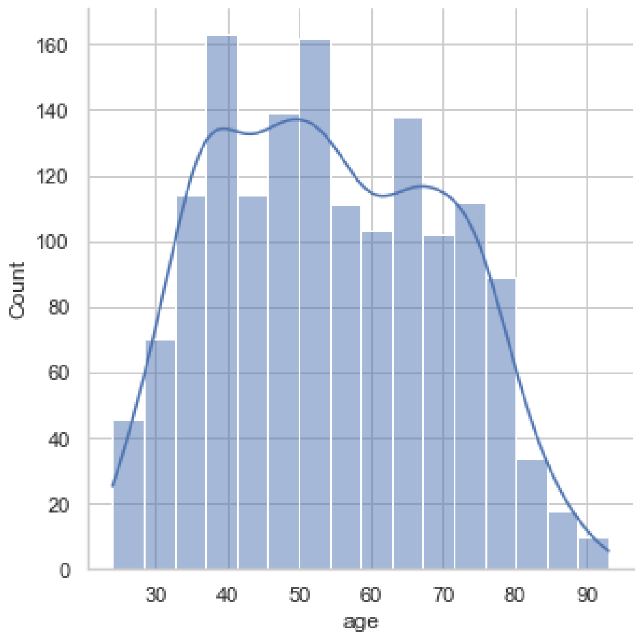
We are going to encode the vote and gender column to numerical values before performing univariate analysis. There are 9 columns and 1525 rows in the dataset.

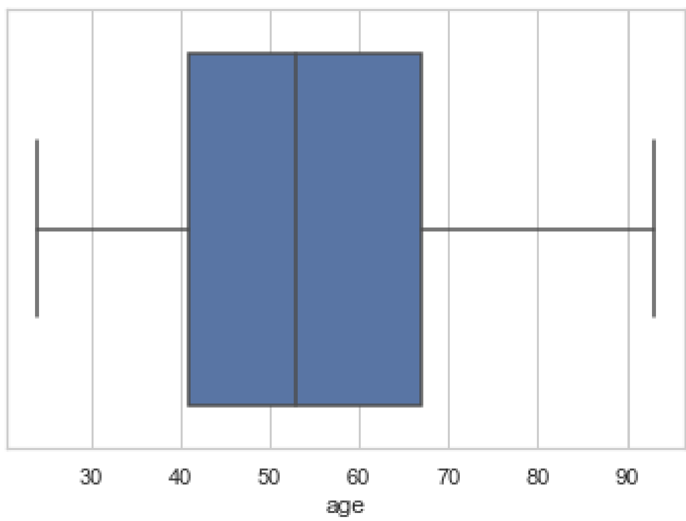
UNIVERIATE ANALYSIS:



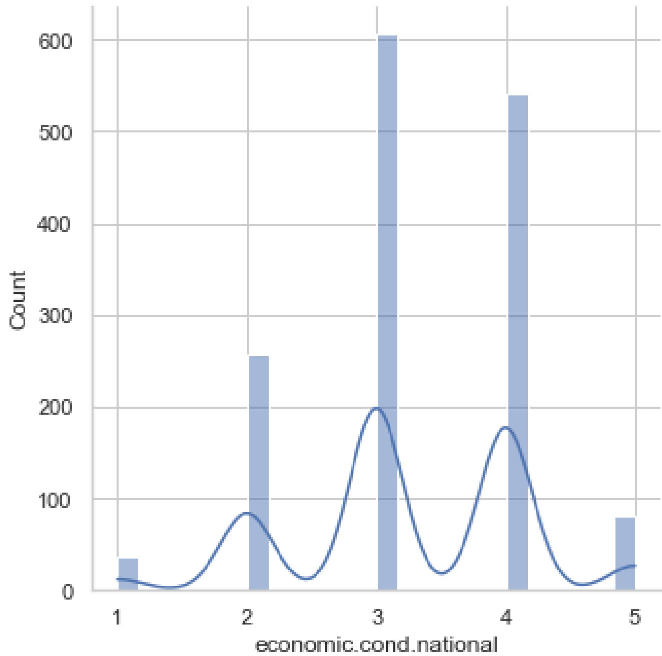


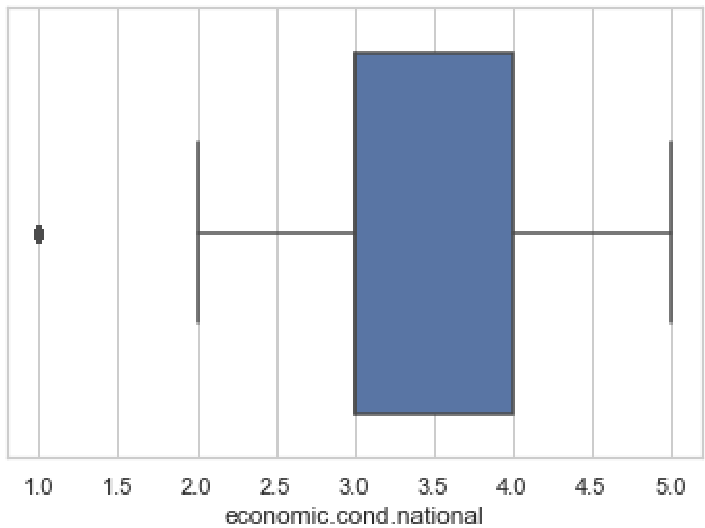
```
Description of vote
---------------------------------------------------
count    1525.000000
mean        0.697049
std         0.459685
min         0.000000
25%         0.000000
50%         1.000000
75%         1.000000
max         1.000000
Name: vote, dtype: float64 Distribution of vote
---------------------------------------------------
```



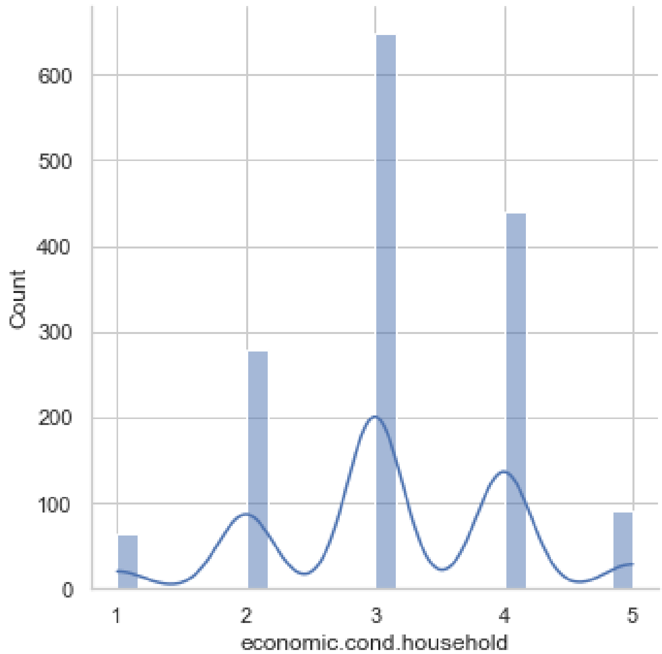


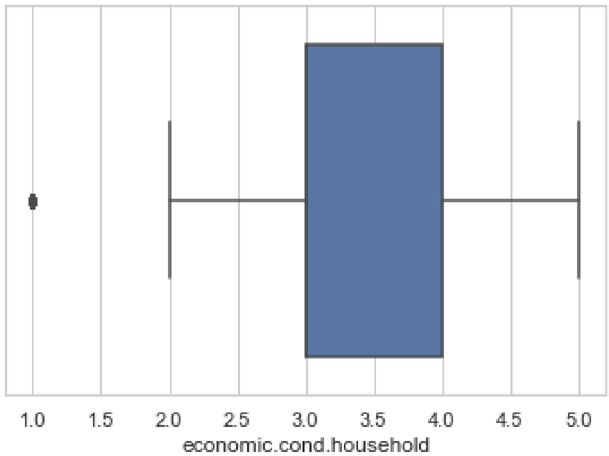

```
Description of age
-----------------------------------------------
count    1525.000000
mean       54.182295
std        15.711209
min        24.000000
25%        41.000000
50%        53.000000
75%        67.000000
max        93.000000
Name: age, dtype: float64 Distribution of age
-----------------------------------------------
```



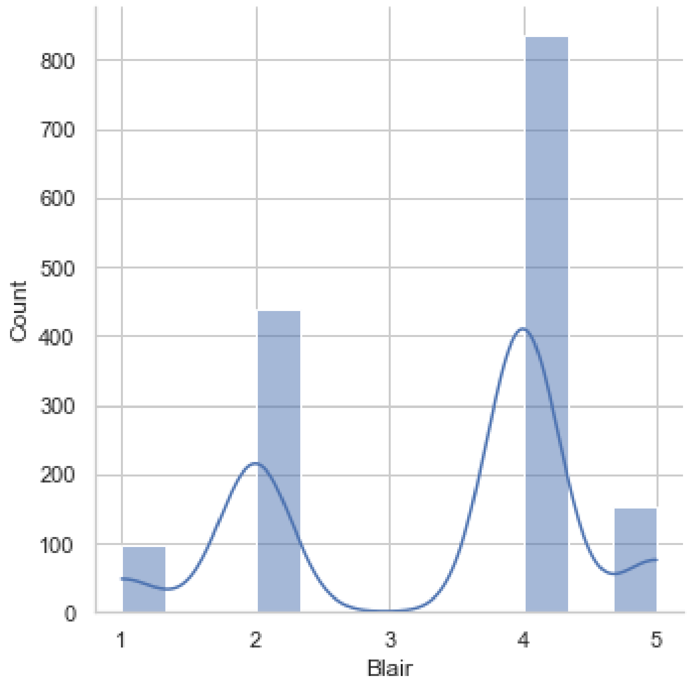


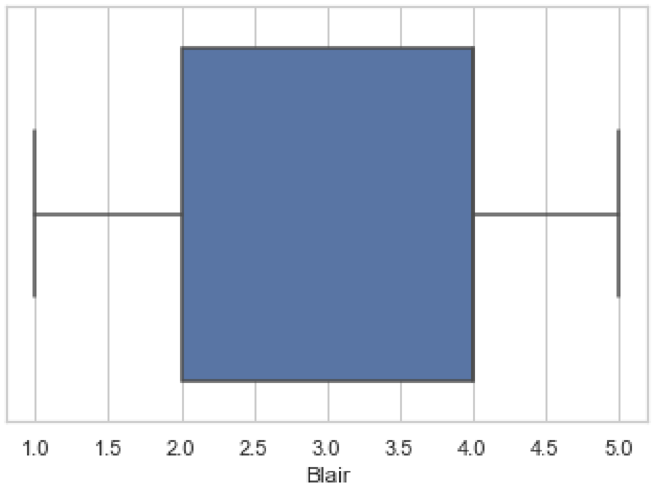

```
Description of economic.cond.national
----------------------------------------------------------------------------
count    1525.000000
mean        3.245902
std         0.880969
min         1.000000
25%         3.000000
50%         3.000000
75%         4.000000
max         5.000000
Name: economic.cond.national, dtype: float64 Distribution of economic.cond.national
----------------------------------------------------------------------------
```



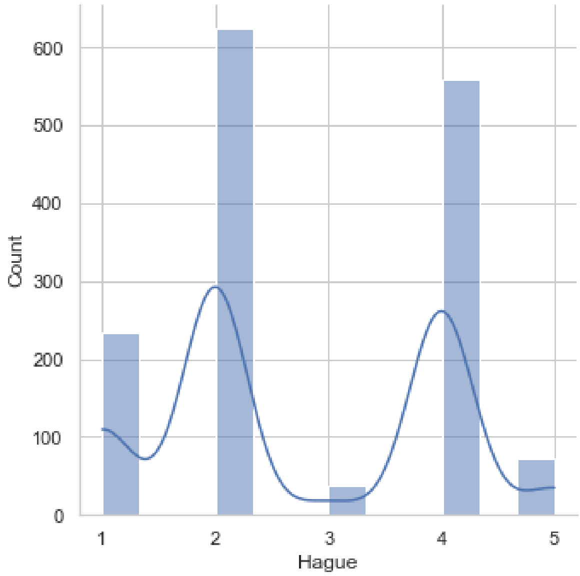


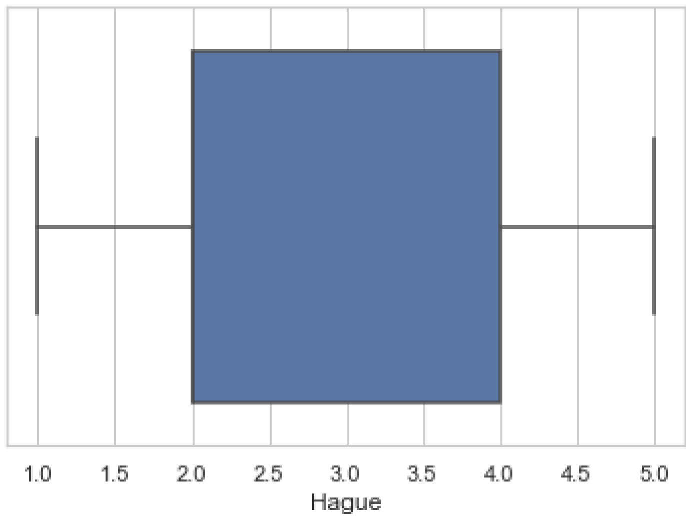

```
Description of economic.cond.household
----------------------------------------------------------------------------
count    1525.000000
mean        3.140328
std         0.929951
min         1.000000
25%         3.000000
50%         3.000000
75%         4.000000
max         5.000000
Name: economic.cond.household, dtype: float64 Distribution of economic.cond.household
----------------------------------------------------------------------------
```



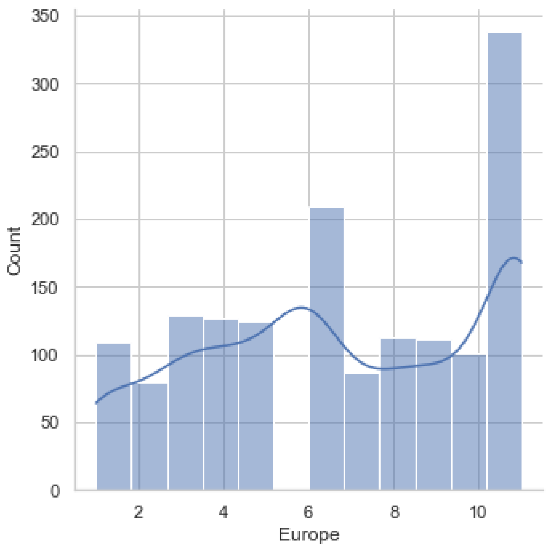


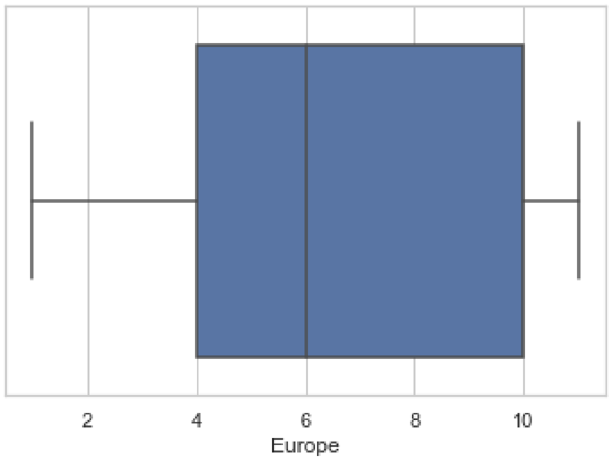

```
Description of Blair
----------------------------------------------------------------------------
count    1525.000000
mean        3.334426
std         1.174824
min         1.000000
25%         2.000000
50%         4.000000
75%         4.000000
max         5.000000
Name: Blair, dtype: float64 Distribution of Blair
----------------------------------------------------------------------------
```



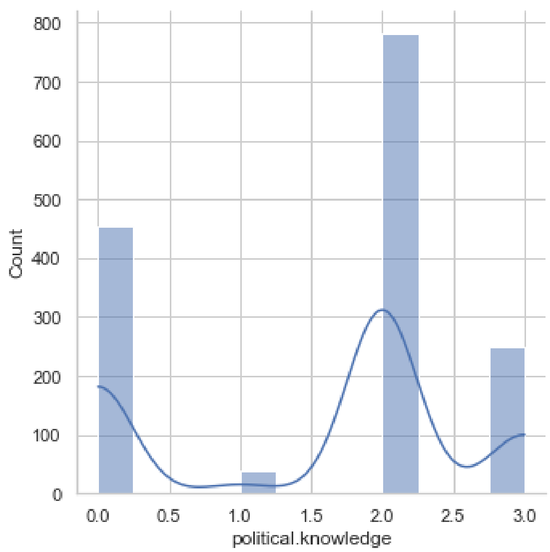


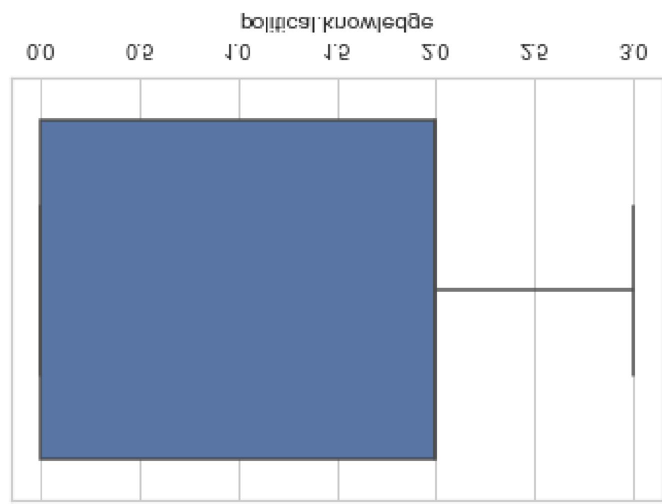

```
Description of Hague
--------------------------------------------------
count    1525.000000
mean        2.746885
std         1.230703
min         1.000000
25%         2.000000
50%         2.000000
75%         4.000000
max         5.000000
Name: Hague, dtype: float64 Distribution of Hague
--------------------------------------------------
```



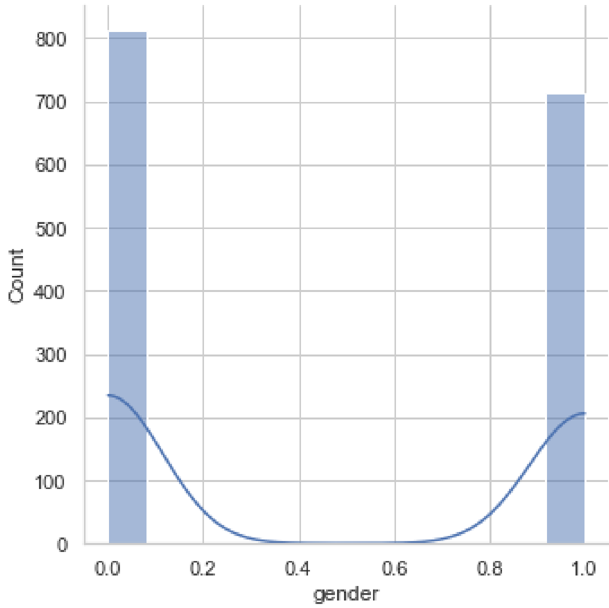


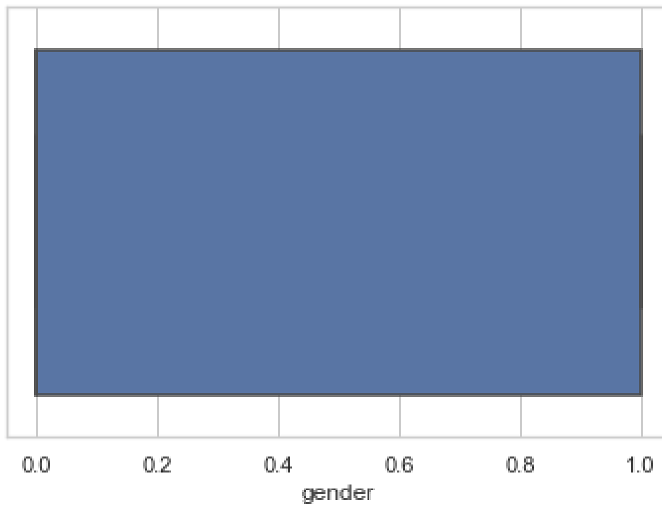
```
Description of Europe
------------------------------------------------------
count    1525.000000
mean        6.728525
std         3.297538
min         1.000000
25%         4.000000
50%         6.000000
75%        10.000000
max        11.000000
Name: Europe, dtype: float64 Distribution of Europe
------------------------------------------------------
```






```
Description of political.knowledge
------------------------------------------------------------------------------
count    1525.000000
mean        1.542295
std         1.083315
min         0.000000
25%         0.000000
50%         2.000000
75%         2.000000
max         3.000000
Name: political.knowledge, dtype: float64 Distribution of political.knowledge
------------------------------------------------------------------------------
```

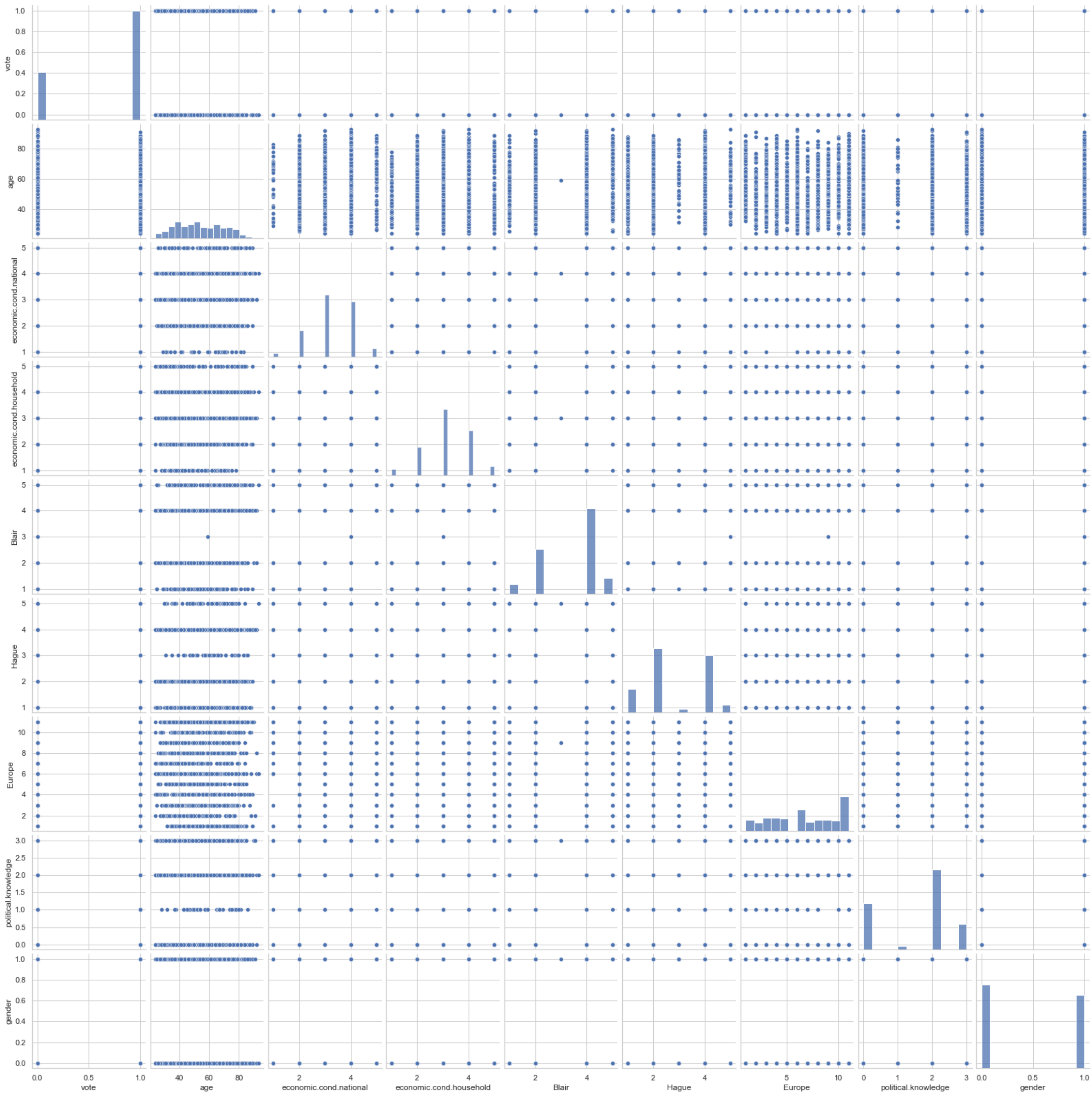




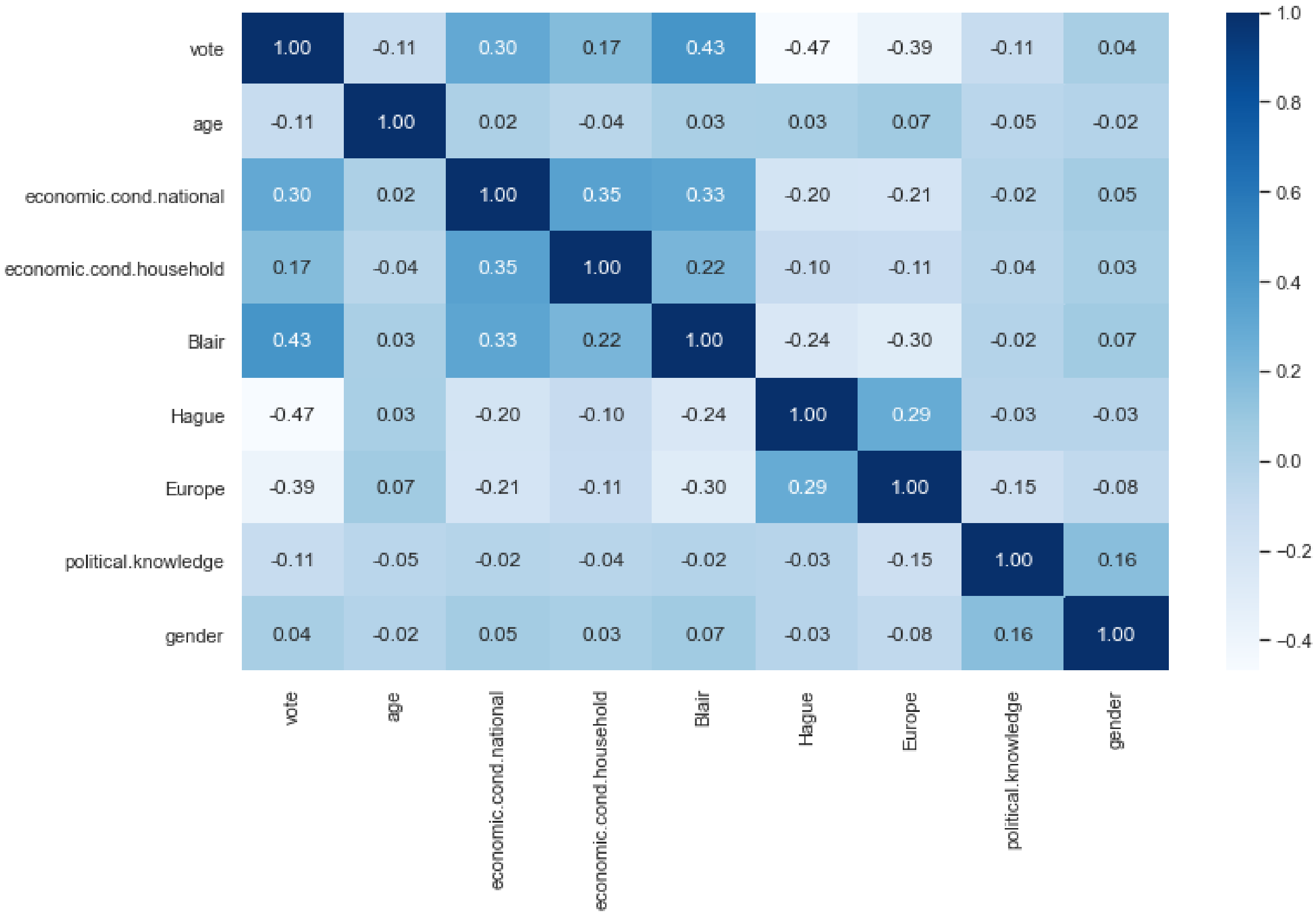

```
Description of gender
----------------------------------------------------------------------------
count    1525.000000
mean        0.467541
std         0.499109
min         0.000000
25%         0.000000
50%         0.000000
75%         1.000000
max         1.000000
Name: gender, dtype: float64 Distribution of gender
----------------------------------------------------------------------------
```

After performing univariate analysis, we can see that:

- 1063 voters voted for labour party and 462 voters voted for conservative party
- Voters age ranges from 24 to 93 with mean of 54.182295
- Most voters got 3 in national economic condition assessment
- Most voters got 3 in household economic condition assessment
- Blair assessment has majority of 4
- Hague assessment has majority of 2
- Most voters got high score of 11 which represent 'Eurosceptic' sentiment.
- Voters political knowledge has majority of 2
- There are 713 male voters and 812 female voters



From the pairplot and heatmap we can see that mostly all the variables are less correlated, blair and vote is moderately correlated(0.43), assessment of economic condition national and vote are moderately correlated(0.30), blair and assessment of economic condition national are moderately correlated(0.33), assessment of national and household economic condition are moderately correlated(0.35), Europe and hague are moderately correlated(0.29)



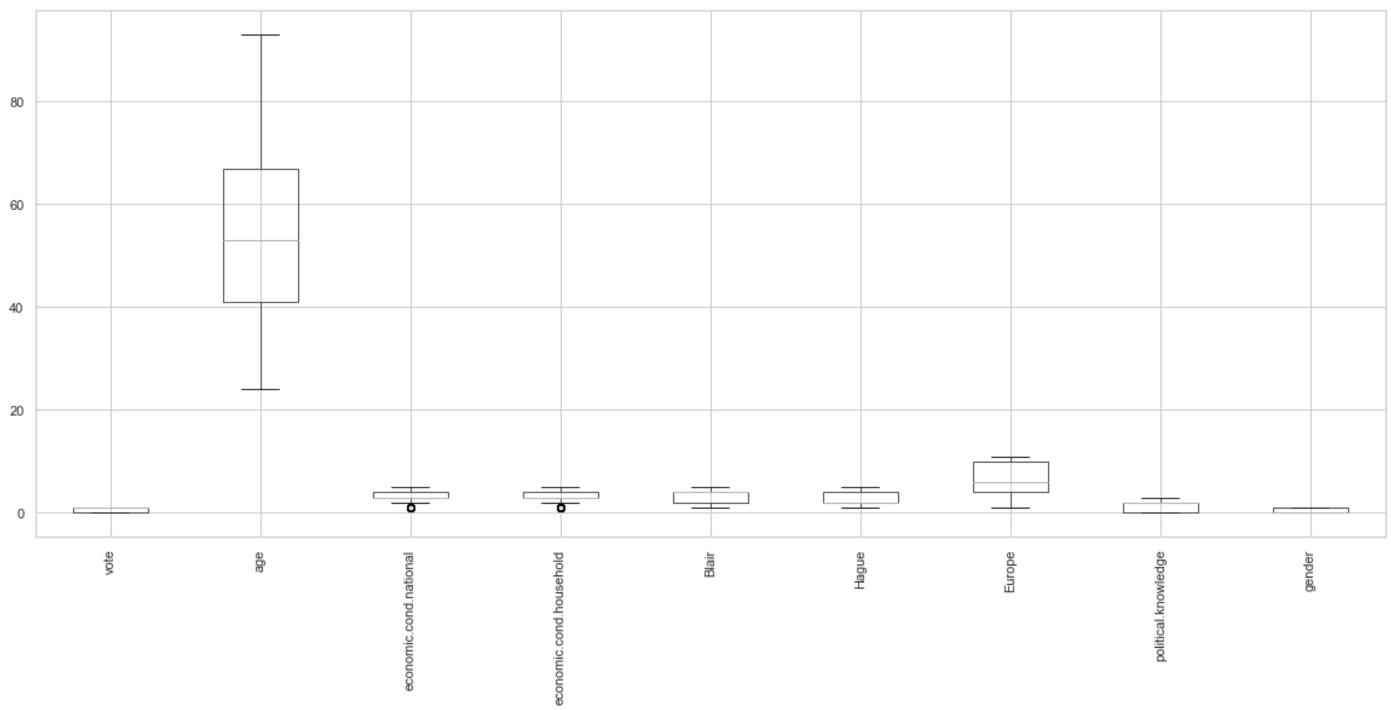
1.3) Encode the data (having string values) for Modelling. Is Scaling necessary here or not?(2 pts), Data Split: Split the data into train and test (70:30) (2 pts). The learner is expected to check and comment about the difference in scale of different features on the bases of appropriate measure for 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.

We have used 1 hot encoding to code the data. One-hot encoding is an important step for preparing the dataset for use in machine learning. One-hot encoding turns your categorical data into a binary vector representation.

Standardization (also called z-score normalization) transforms your data such that the resulting distribution has a mean of 0 and a standard deviation of 1.

Encoded categorical variables contain values on 0 and 1. Therefore, there is even no need to scale them. it makes no sense to scale and center binary (or categorical) variables so you should only center and scale continuous variables.

We can see there is no outliers in age continuous column:



We have split the data into 30:70 ratio for test and train respectively with random state 1.

1.4) Apply Logistic Regression and LDA (Linear Discriminant Analysis) (2 pts). Interpret the inferences of both model s (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)

After applying Linear Discriminant Analysis on the train data we get the following classification report:

	precision	recall	f1-score	support
0	0.77	0.71	0.74	332
1	0.88	0.90	0.89	735
accuracy			0.85	1067
macro avg	0.82	0.81	0.82	1067
weighted avg	0.84	0.85	0.84	1067

After applying Linear Discriminant Analysis on the test data we get the following classification report:

	precision	recall	f1-score	support
0	0.68	0.68	0.68	130
1	0.87	0.88	0.87	328
accuracy			0.82	458
macro avg	0.78	0.78	0.78	458
weighted avg	0.82	0.82	0.82	458

In terms of accuracy and recall LDA has performed good with 85% accuracy and 90% recall. Overall this is a robust model.

After applying Logistic Regression on the train data we get the following classification report:

	precision	recall	f1-score	support
0	0.77	0.70	0.74	332
1	0.87	0.91	0.89	735
accuracy			0.84	1067
macro avg	0.82	0.80	0.81	1067
weighted avg	0.84	0.84	0.84	1067

After applying Logistic Regression on the test data we get the following classification report:

	precision	recall	f1-score	support
0	0.67	0.65	0.66	130
1	0.86	0.88	0.87	328
accuracy			0.81	458
macro avg	0.77	0.76	0.76	458
weighted avg	0.81	0.81	0.81	458

In terms of accuracy and recall, Logistic regression has achieved 84% accuracy and 91% recall . overall this is a robust 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)

After applying KNN on the train data we get the following classification report:

	precision	recall	f1-score	support
0	0.75	0.61	0.67	332
1	0.84	0.91	0.87	735
accuracy			0.82	1067
macro avg	0.79	0.76	0.77	1067
weighted avg	0.81	0.82	0.81	1067

After applying KNN on the test data we get the following classification report:

	precision	recall	f1-score	support
0	0.54	0.55	0.55	130
1	0.82	0.81	0.82	328
accuracy			0.74	458
macro avg	0.68	0.68	0.68	458
weighted avg	0.74	0.74	0.74	458

Overfitting happens when a model learns the detail and noise in the training data to the extent that it negatively impacts the performance of the model on new data. This means that the noise or random fluctuations in the training data is picked up and learned as concepts by the model. The problem is that these concepts do not apply to new data and negatively impact the models ability to generalize.

If the performance of the model on the training dataset is significantly better than the performance on the test dataset, then the model may have overfit the training dataset.

We have got 82% accuracy on train and 74% accuracy on test. This is overfitting.

After applying Naïve Bayes Model on the train data we get the following classification report:

	precision	recall	f1-score	support
0	0.79	0.42	0.55	332
1	0.78	0.95	0.86	735
accuracy			0.79	1067
macro avg	0.79	0.69	0.71	1067
weighted avg	0.79	0.79	0.76	1067

After applying Naïve Bayes Model on the test data we get the following classification report:

	precision	recall	f1-score	support
0	0.75	0.48	0.58	130
1	0.82	0.94	0.87	328
accuracy			0.81	458
macro avg	0.78	0.71	0.73	458
weighted avg	0.80	0.81	0.79	458

With 79% accuracy and 95% recall , we can say this is a good 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.

After applying Random Forest on the train data we get the following classification report:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	332
1	1.00	1.00	1.00	735
accuracy			1.00	1067
macro avg	1.00	1.00	1.00	1067
weighted avg	1.00	1.00	1.00	1067

After applying Random Forest on the test data we get the following classification report:

	precision	recall	f1-score	support
0	0.71	0.69	0.70	130
1	0.88	0.89	0.88	328
accuracy			0.83	458
macro avg	0.79	0.79	0.79	458
weighted avg	0.83	0.83	0.83	458

After applying Bagging on the train data we get the following classification report:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	332
1	1.00	1.00	1.00	735
accuracy			1.00	1067
macro avg	1.00	1.00	1.00	1067
weighted avg	1.00	1.00	1.00	1067

After applying Bagging on the test data we get the following classification report:

	precision	recall	f1-score	support
0	0.67	0.72	0.70	130
1	0.89	0.86	0.87	328
accuracy			0.82	458
macro avg	0.78	0.79	0.78	458
weighted avg	0.83	0.82	0.82	458

Bagging has overfit the model with 100% accuracy on train and only 82% accuracy on test data.

After applying Gradient Boosting on the train data we get the following classification report:

	precision	recall	f1-score	support
0	0.84	0.78	0.81	332
1	0.90	0.93	0.92	735
accuracy			0.88	1067
macro avg	0.87	0.86	0.86	1067
weighted avg	0.88	0.88	0.88	1067

After applying Gradient Boosting on the test data we get the following classification report:

	precision	recall	f1-score	support
0	0.66	0.72	0.69	130
1	0.89	0.85	0.87	328
accuracy			0.82	458
macro avg	0.77	0.79	0.78	458
weighted avg	0.82	0.82	0.82	458

With 88% accuracy on train and 82% accuracy on test, gradient boosting is a robust model.

After applying Ada Boosting on the train data we get the following classification report:

	precision	recall	f1-score	support
0	0.77	0.71	0.74	332
1	0.87	0.90	0.89	735
accuracy			0.84	1067
macro avg	0.82	0.81	0.81	1067
weighted avg	0.84	0.84	0.84	1067

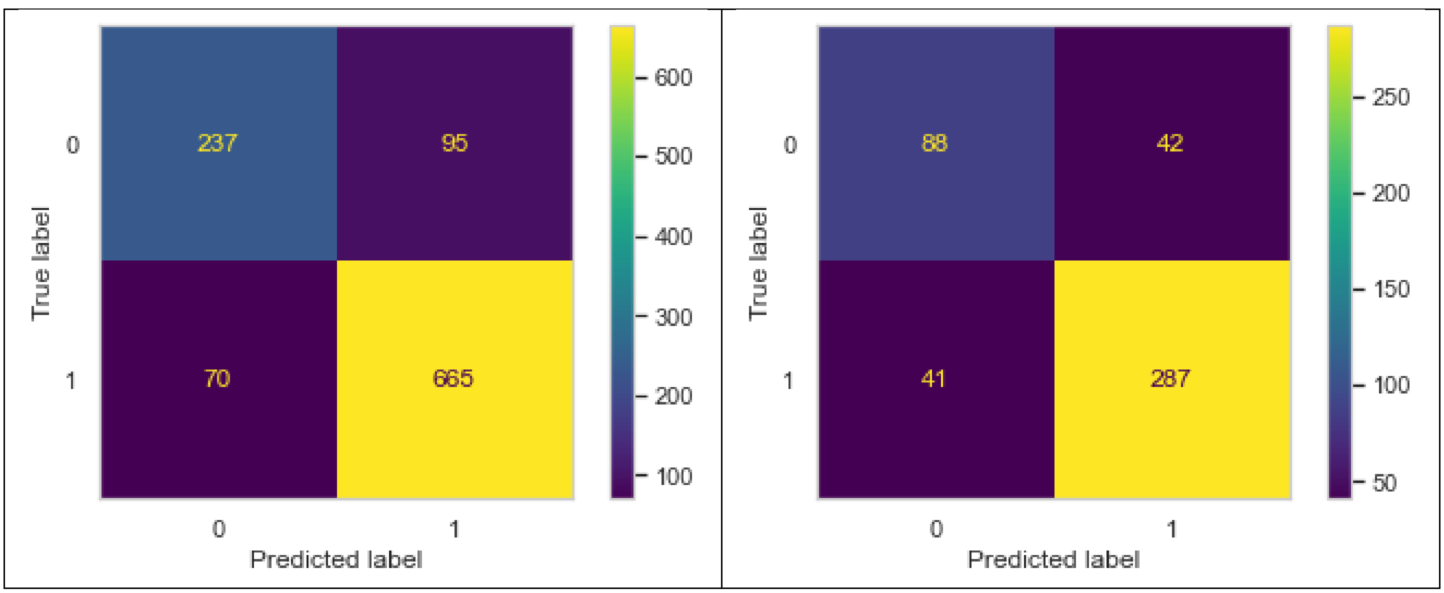
After applying Ada Boosting on the test data we get the following classification report:

	precision	recall	f1-score	support
0	0.71	0.68	0.70	130
1	0.88	0.89	0.88	328
accuracy			0.83	458
macro avg	0.79	0.79	0.79	458
weighted avg	0.83	0.83	0.83	458

With 84% accuracy on train and 83% accuracy on test, ada boosting is a robust model.

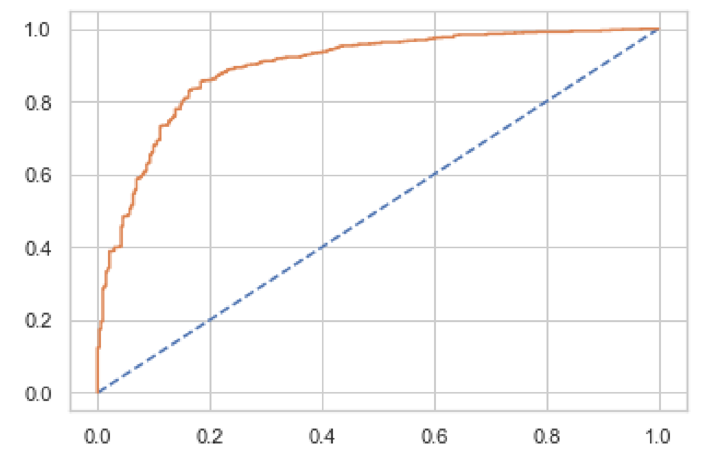
1.7 Performance Metrics: Check the performance of Predictions on Train and Test sets using Accuracy, Confusion Matrix, Plot ROC curve and get ROC_AUC score for each model, classification report (4 pts) Final Model - Compare and comment on all models on the basis of the performance metrics in a structured 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)

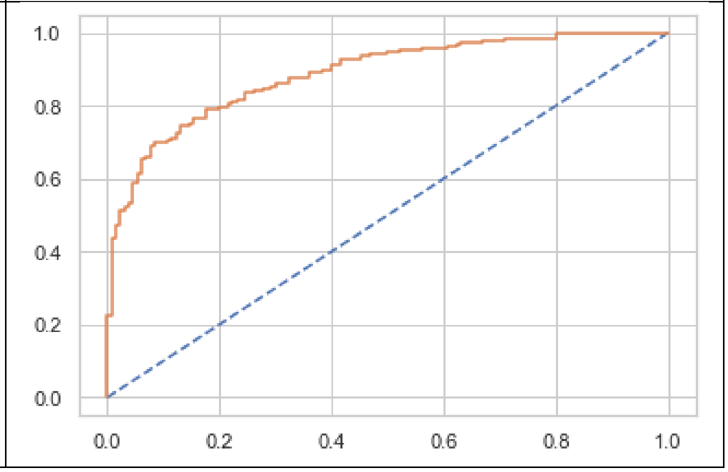
Linear Discriminant analysis



	precision	recall	f1-score	support
0	0.77	0.71	0.74	332
1	0.88	0.90	0.89	735
accuracy			0.85	1067
macro avg	0.82	0.81	0.82	1067
weighted avg	0.84	0.85	0.84	1067

	precision	recall	f1-score	support
0	0.68	0.68	0.68	130
1	0.87	0.88	0.87	328
accuracy			0.82	458
macro avg	0.78	0.78	0.78	458
weighted avg	0.82	0.82	0.82	458

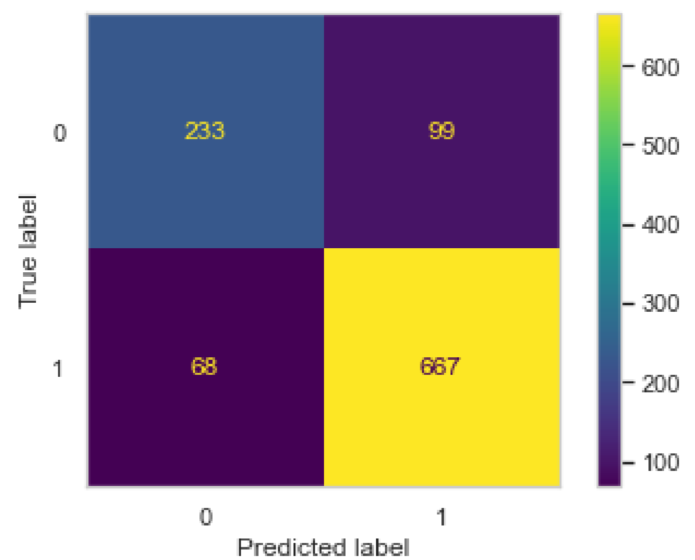


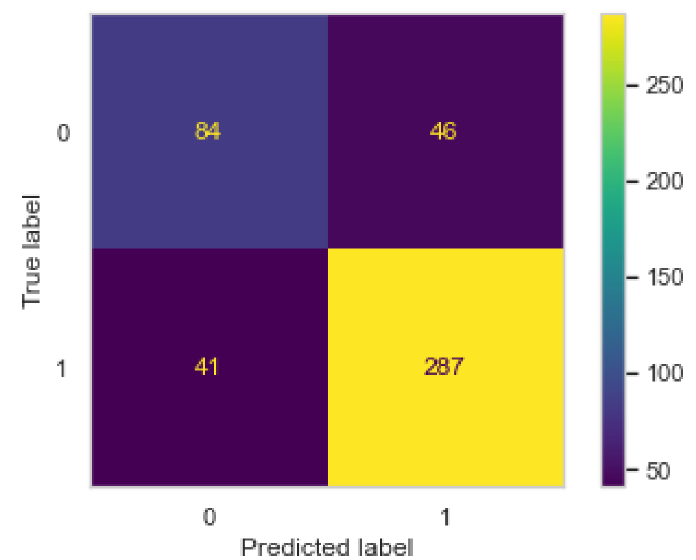


	train	test
accuracy	0.85	0.82
recall	0.90	0.88
auc	0.894	0.894
F1 score	0.89	0.87
precision	0.88	0.87

Based on the accuracy, recall, auc,f1-score we can say it is a robust model

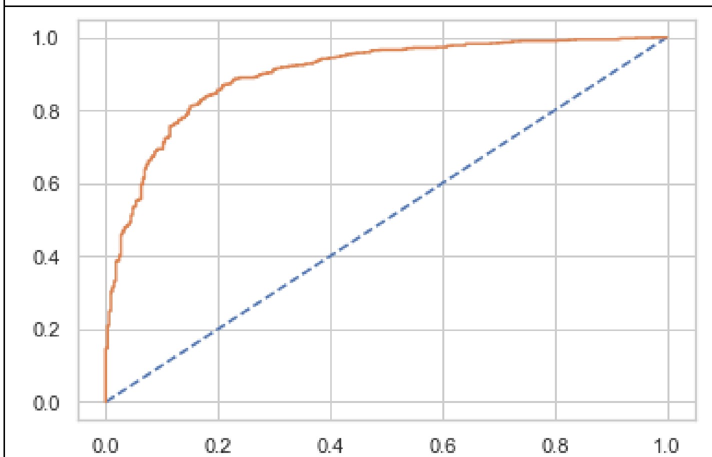
Logistic Regression

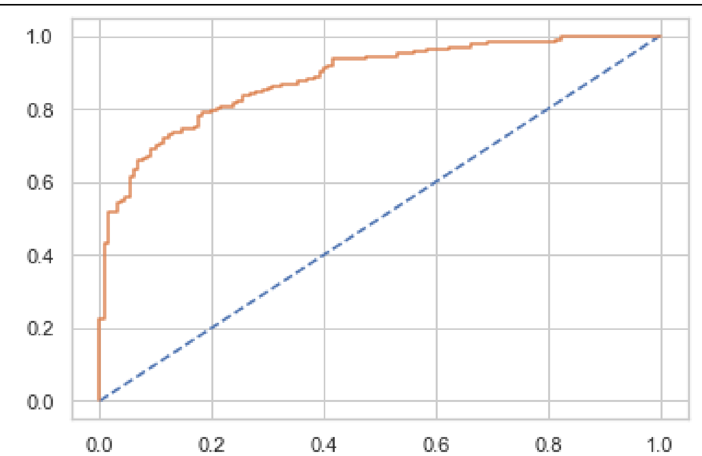





```
              precision    recall  f1-score   support

           0       0.77      0.70      0.74       332
           1       0.87      0.91      0.89       735

    accuracy                           0.84      1067
   macro avg       0.82      0.80      0.81      1067
weighted avg       0.84      0.84      0.84      1067
```

```
              precision    recall  f1-score   support

           0       0.67      0.65      0.66       130
           1       0.86      0.88      0.87       328

    accuracy                           0.81       458
   macro avg       0.77      0.76      0.76       458
weighted avg       0.81      0.81      0.81       458
```



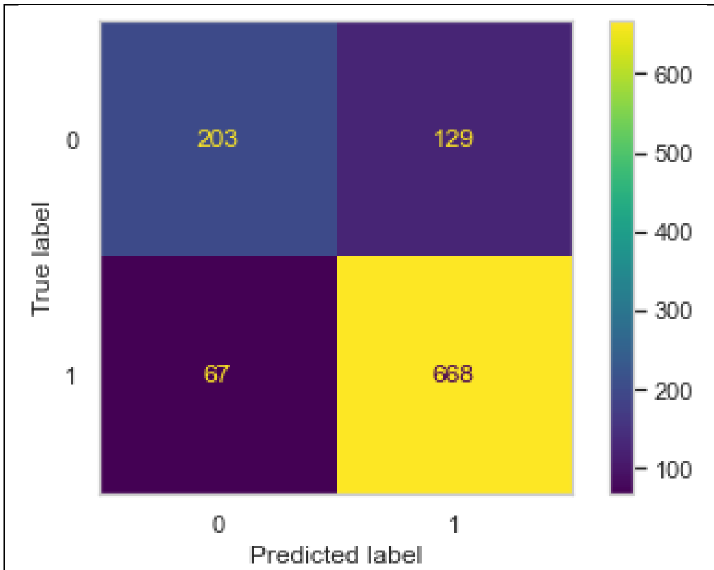


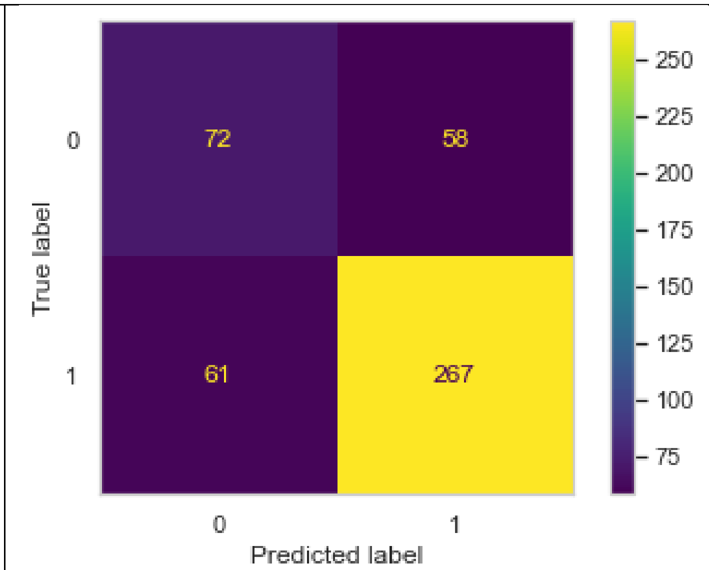
	train	test
accuracy	0.84	0.81
recall	0.91	0.88
precision	0.87	0.86
f-score	0.89	0.87

auc	0.90	0.90

Based on accuracy, recall and excellent AUC we can say that Logistic Regression has yielded a robust model

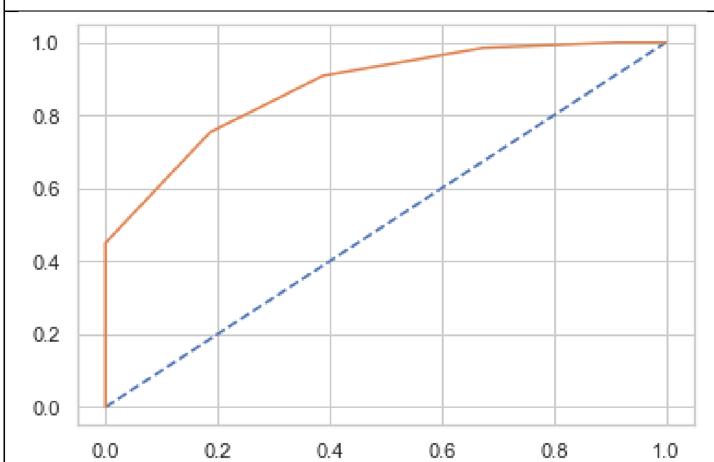
KNN

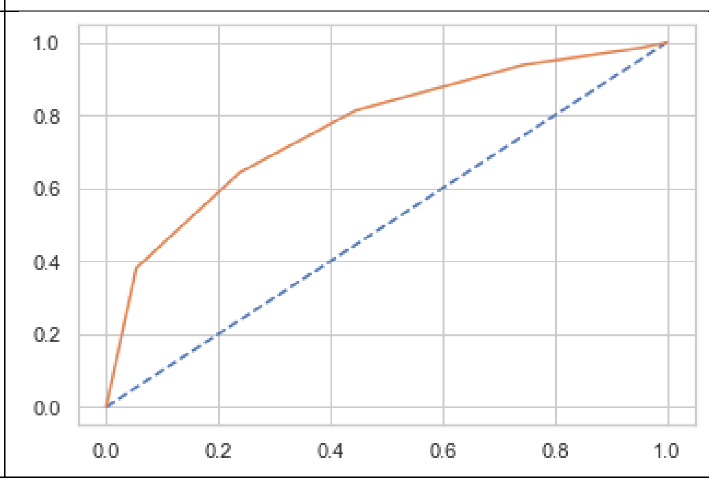





```
              precision    recall  f1-score   support

           0       0.75      0.61      0.67       332
           1       0.84      0.91      0.87       735

    accuracy                           0.82      1067
   macro avg       0.79      0.76      0.77      1067
weighted avg       0.81      0.82      0.81      1067
```

```
              precision    recall  f1-score   support

           0       0.54      0.55      0.55       130
           1       0.82      0.81      0.82       328

    accuracy                           0.74       458
   macro avg       0.68      0.68      0.68       458
weighted avg       0.74      0.74      0.74       458
```

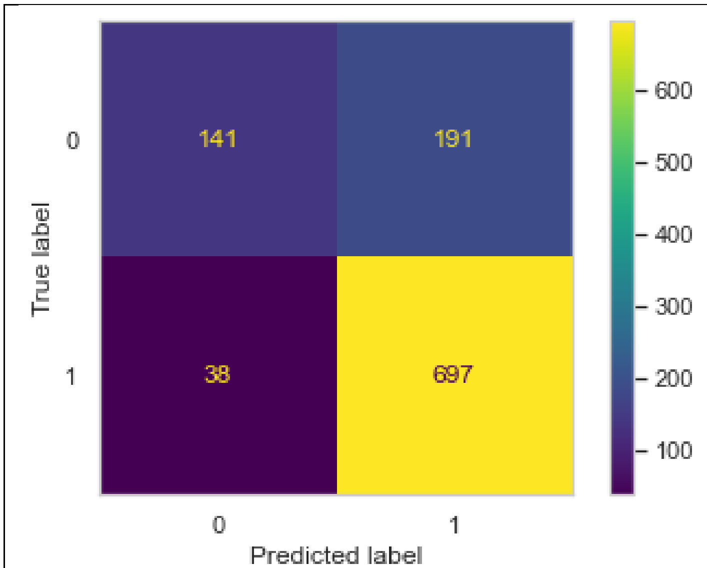




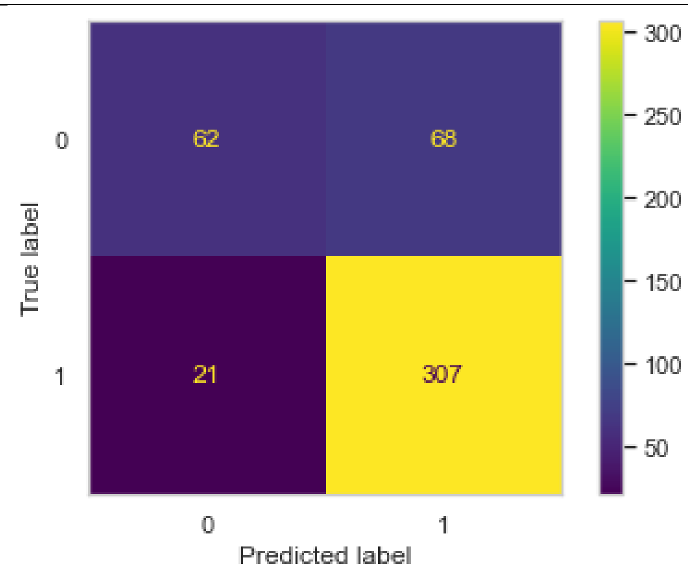
	train	test
accuracy	0.82	0.74
recall	0.91	0.81
precision	0.84	0.82
F1-score	0.87	0.82
auc	0.875	0.875

Based on the accuracy, recall and good AUC we can say KNN is a good model, However the test data is not so good.

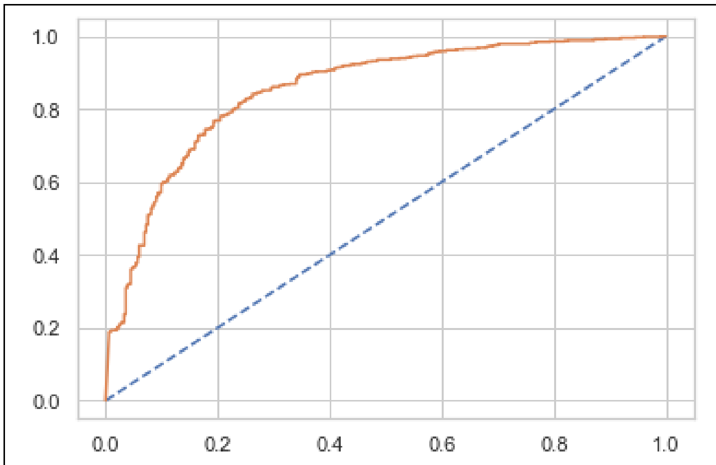
Naïve Bayes

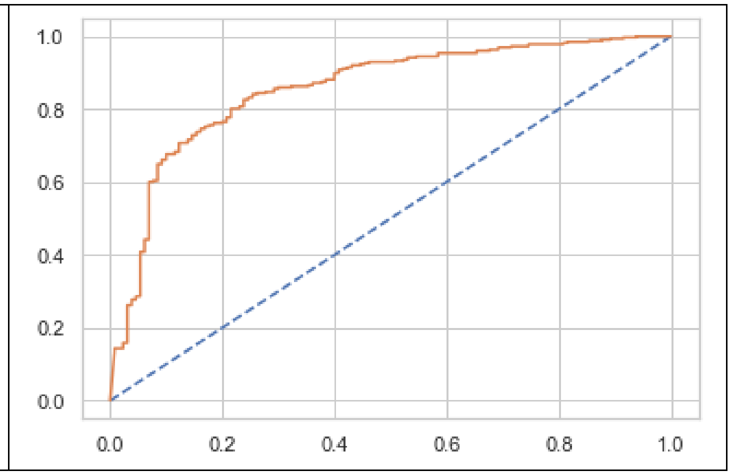


	precision	recall	f1-score	support
0	0.79	0.42	0.55	332
1	0.78	0.95	0.86	735
accuracy			0.79	1067
macro avg	0.79	0.69	0.71	1067
weighted avg	0.79	0.79	0.76	1067



	precision	recall	f1-score	support
0	0.75	0.48	0.58	130
1	0.82	0.94	0.87	328
accuracy			0.81	458
macro avg	0.78	0.71	0.73	458
weighted avg	0.80	0.81	0.79	458

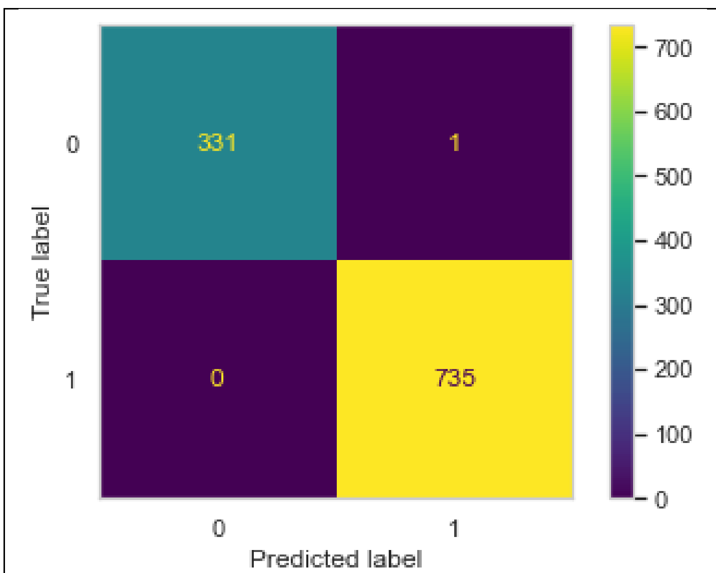


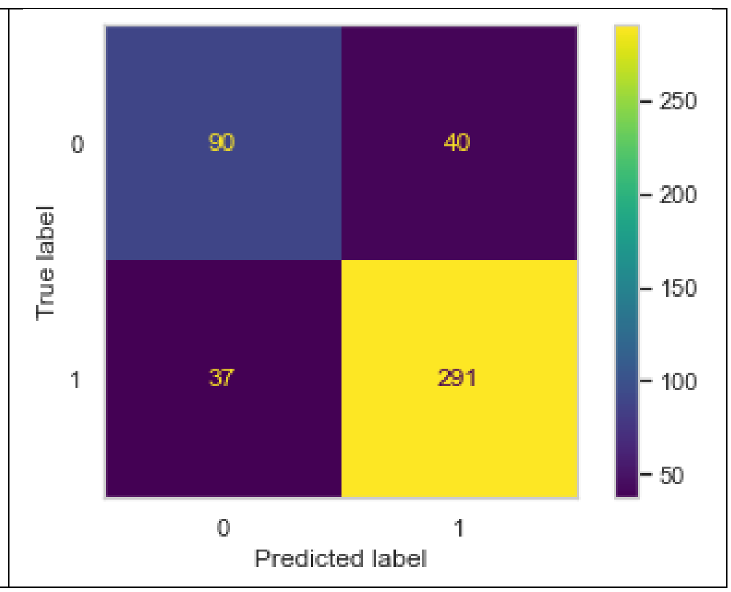


	train	test
accuracy	0.79	0.81
recall	0.95	0.94
precision	0.78	0.82
F1 score	0.86	0.87
auc	0.854	0.854

Based on the accuracy, recall and AUC we can say this is a good model

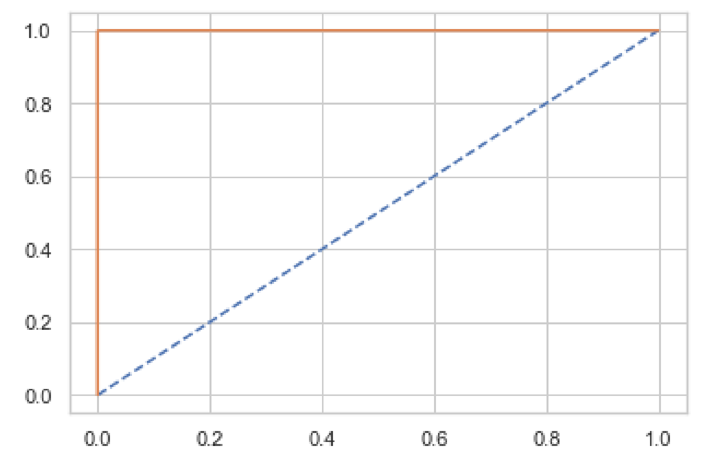
Random Forest

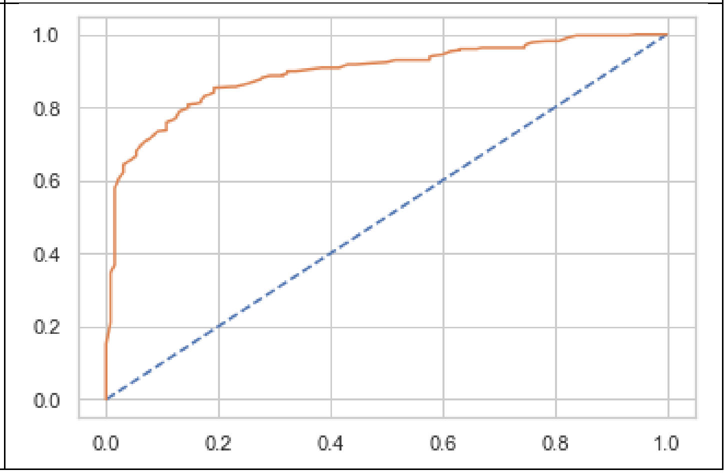




```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       332
           1       1.00      1.00      1.00       735

    accuracy                           1.00      1067
   macro avg       1.00      1.00      1.00      1067
weighted avg       1.00      1.00      1.00      1067
```

```
              precision    recall  f1-score   support

           0       0.71      0.69      0.70       130
           1       0.88      0.89      0.88       328

    accuracy                           0.83       458
   macro avg       0.79      0.79      0.79       458
weighted avg       0.83      0.83      0.83       458
```

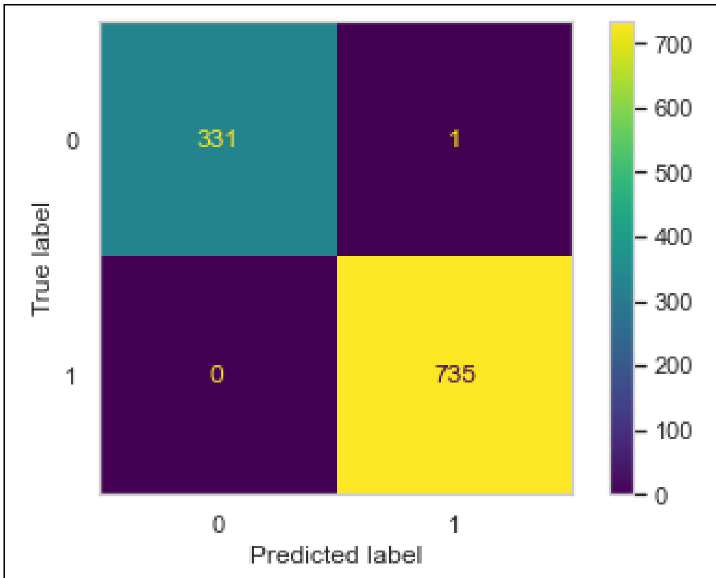


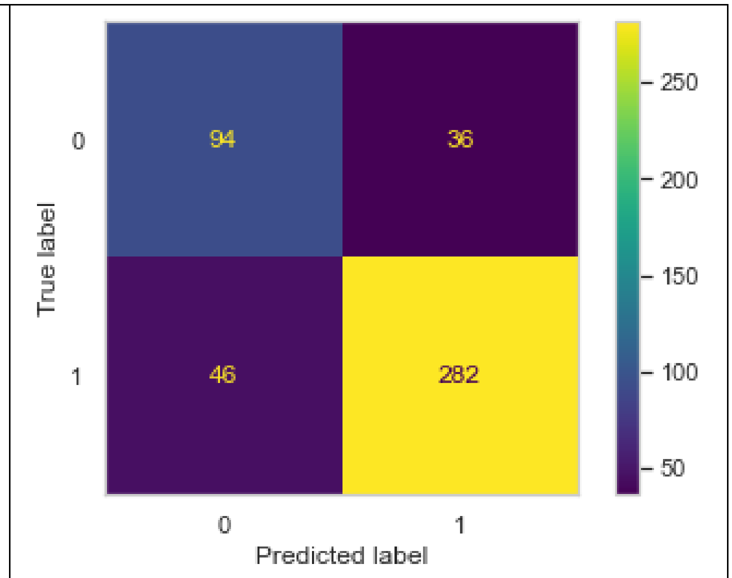


	train	test
accuracy	1.00	0.83
recall	1.00	0.88
precision	1.00	0.89
F1 score	1.00	0.88
auc	1.00	1.00

This is a clear case of overfit model because train data is performing better than test in terms of accuracy, recall, auc.

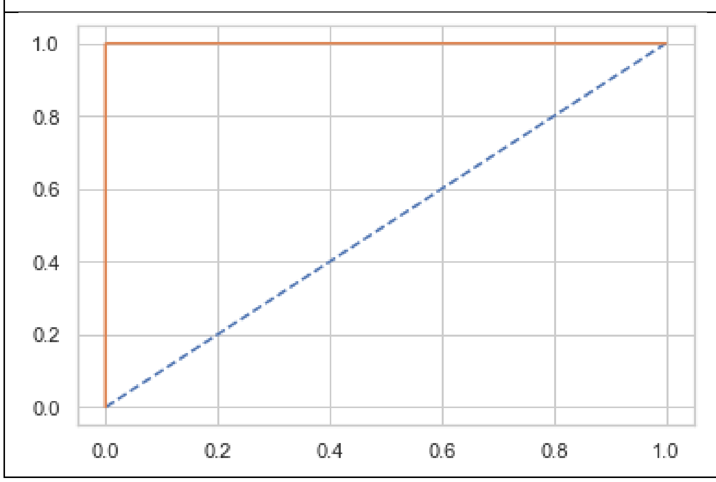
Bagging

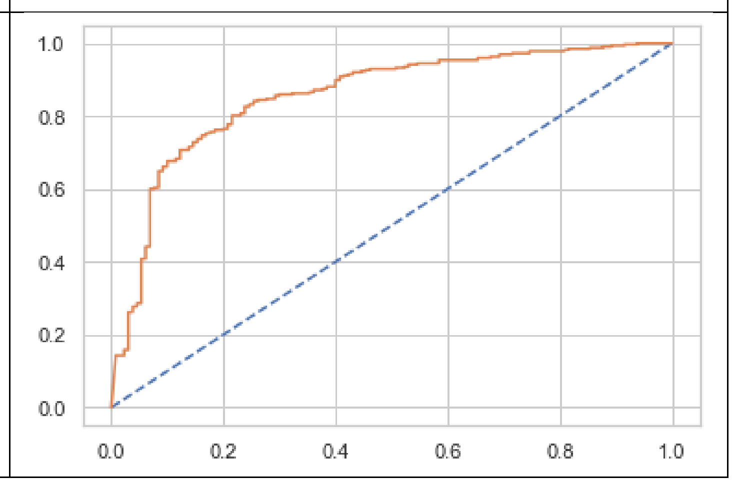





```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       332
           1       1.00      1.00      1.00       735

    accuracy                           1.00      1067
   macro avg       1.00      1.00      1.00      1067
weighted avg       1.00      1.00      1.00      1067
```

```
              precision    recall  f1-score   support

           0       0.67      0.72      0.70       130
           1       0.89      0.86      0.87       328

    accuracy                           0.82       458
   macro avg       0.78      0.79      0.78       458
weighted avg       0.83      0.82      0.82       458
```



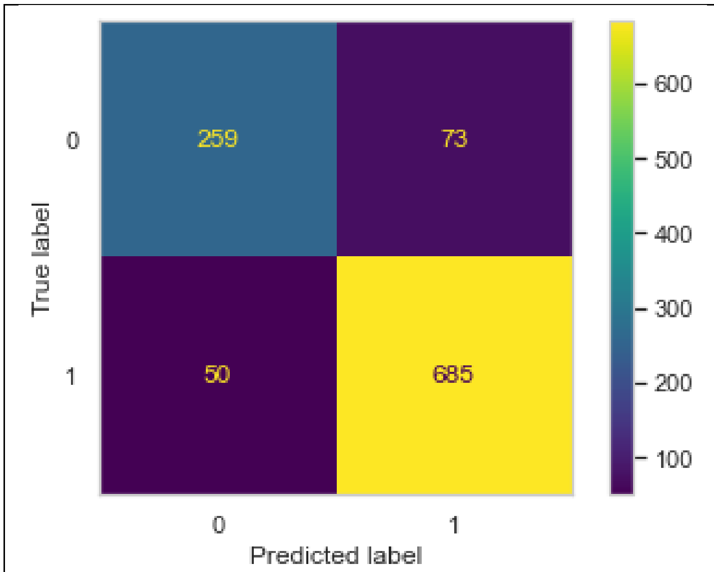


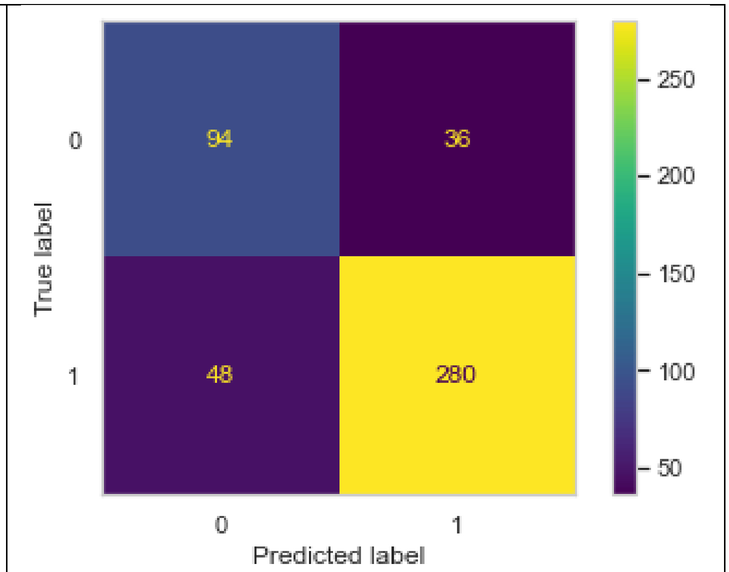
	train	test
accuracy	1.00	0.82
precision	1.00	0.89
recall	1.00	0.86

F1-score	1.00	0.87
auc	1.00	1.00

On the training data, we have got 100% everything however its performance not the same on test data…..this model is a overfit model.

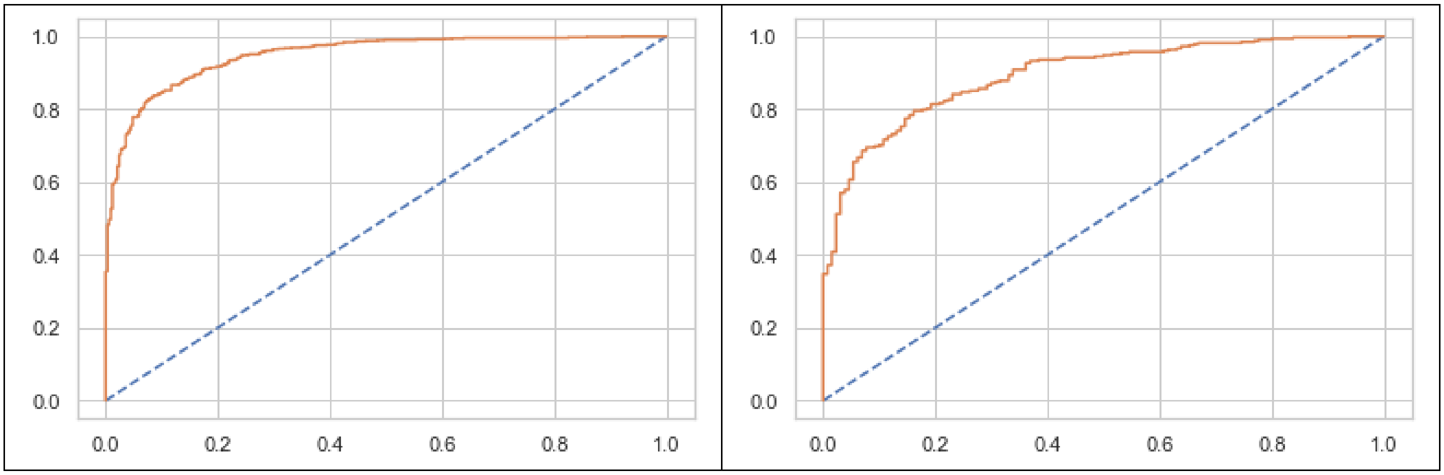
Gradient Boosting





	precision	recall	f1-score	support
0	0.84	0.78	0.81	332
1	0.90	0.93	0.92	735
accuracy			0.88	1067
macro avg	0.87	0.86	0.86	1067
weighted avg	0.88	0.88	0.88	1067

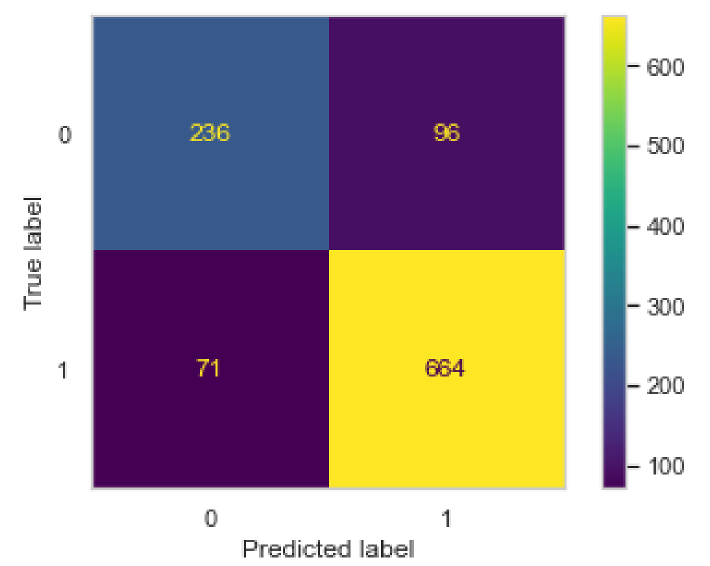
	precision	recall	f1-score	support
0	0.66	0.72	0.69	130
1	0.89	0.85	0.87	328
accuracy			0.82	458
macro avg	0.77	0.79	0.78	458
weighted avg	0.82	0.82	0.82	458

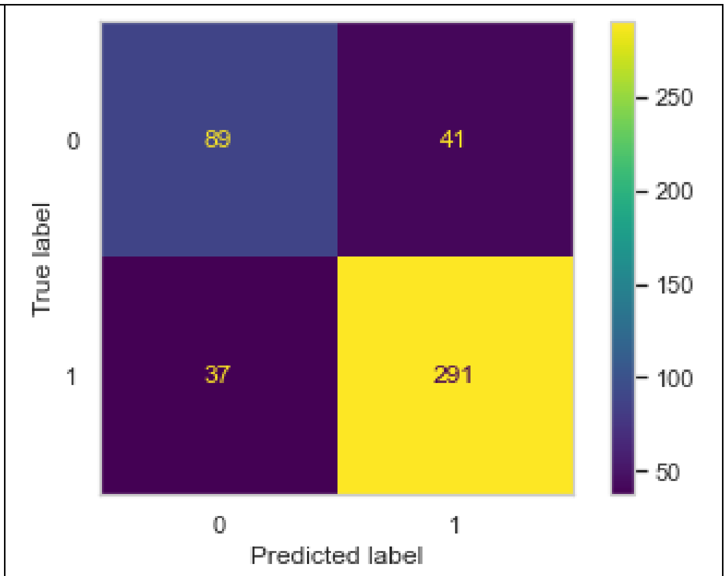


	train	test
accuracy	0.88	0.82
recall	0.93	0.85
precision	0.90	0.89
F1 score	0.92	0.87
AUC	0.949	0.949

Again this model has got outstanding AUC, with 88% accuracy and 93% recall…..again we have got robust model and the best so far.

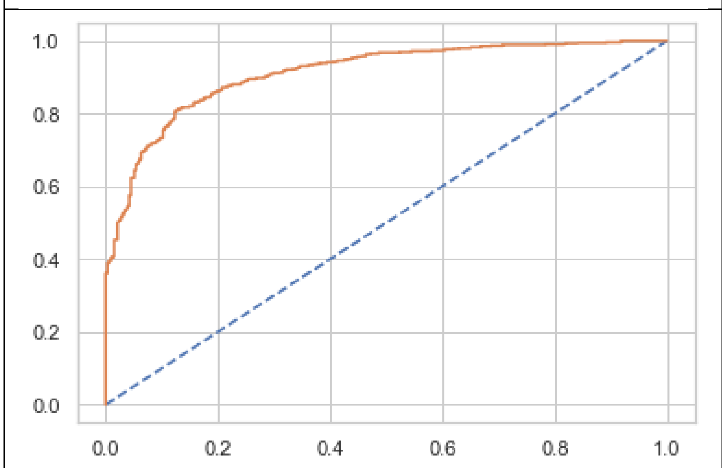
Ada boosting

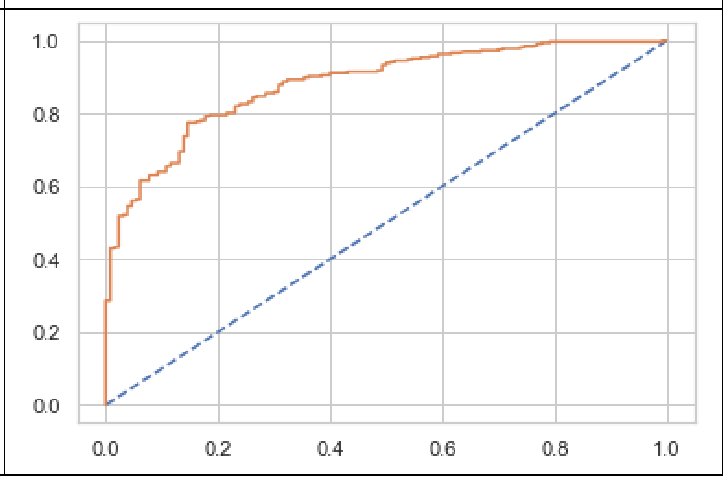





```
              precision    recall  f1-score   support

           0       0.77      0.71      0.74       332
           1       0.87      0.90      0.89       735

    accuracy                           0.84      1067
   macro avg       0.82      0.81      0.81      1067
weighted avg       0.84      0.84      0.84      1067
```

```
              precision    recall  f1-score   support

           0       0.71      0.68      0.70       130
           1       0.88      0.89      0.88       328

    accuracy                           0.83       458
   macro avg       0.79      0.79      0.79       458
weighted avg       0.83      0.83      0.83       458
```





	train	test
accuracy	0.84	0.83
recall	0.90	0.89
precision	0.87	0.88
F1-score	0.89	0.88

auc	0.911	0.911

The AUC is outstanding for this model, with 84% accuracy and 90% recall, this is again one of the best model .

1. Gradient Boosting model has performed the best even without any hyperparameter tuning for both the classes
2. Test and Train performance is within the accepted limited of +/- 10%

Out of all the models given above, considering the difference between train and test dataset performance parameters, Gradient Boosting has performed the best.

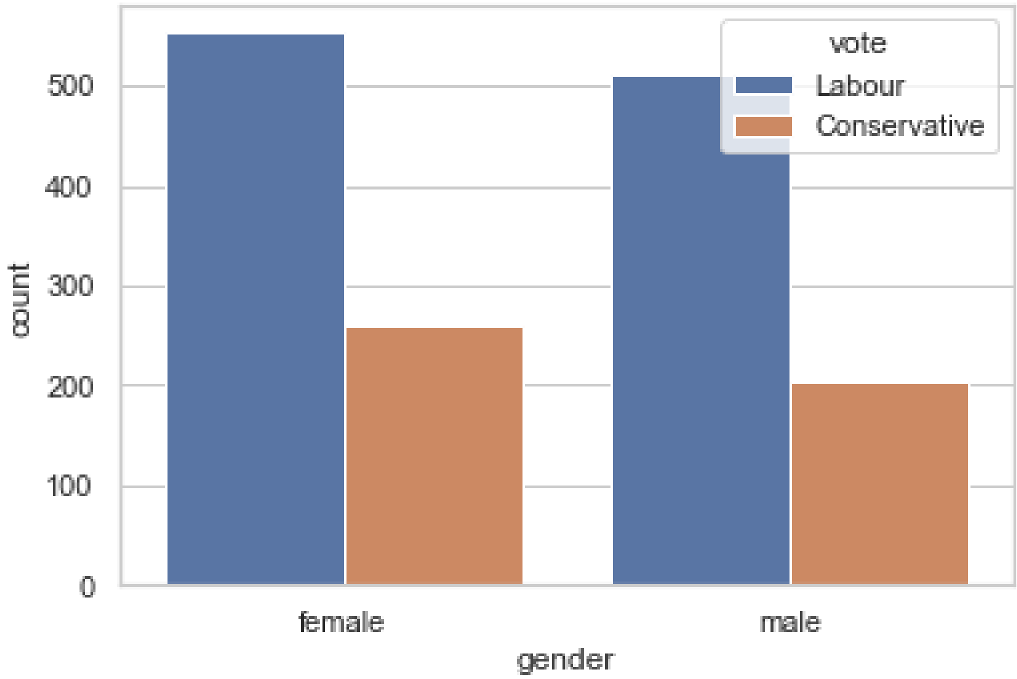
One important observation

> Its not necessary that parameter tuning will always result in a better model. As we have observed in Logistic Regression when Grid Search was used there was no improvement in the recall value of both the classes and in case of Decision Tree, recall for non-negative has gone down , post regularization.

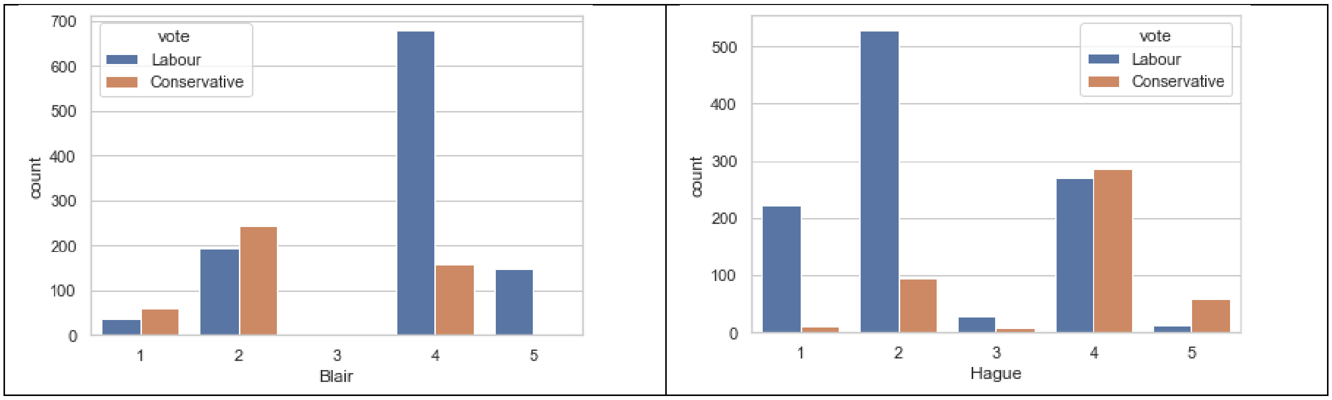
1.8) Based on your analysis and working on the business problem, detail out appropriate insights and recommendations to help the management solve the business objective. There should be at least 3-4 Recommendations and insights in total. Recommendations should be easily understandable and business specific,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

Based on the predictions, we have got following insights in the business problem:

- **Since the female voters are more than male voters the news channel CNBE need to interact with women voters to predict the win.**

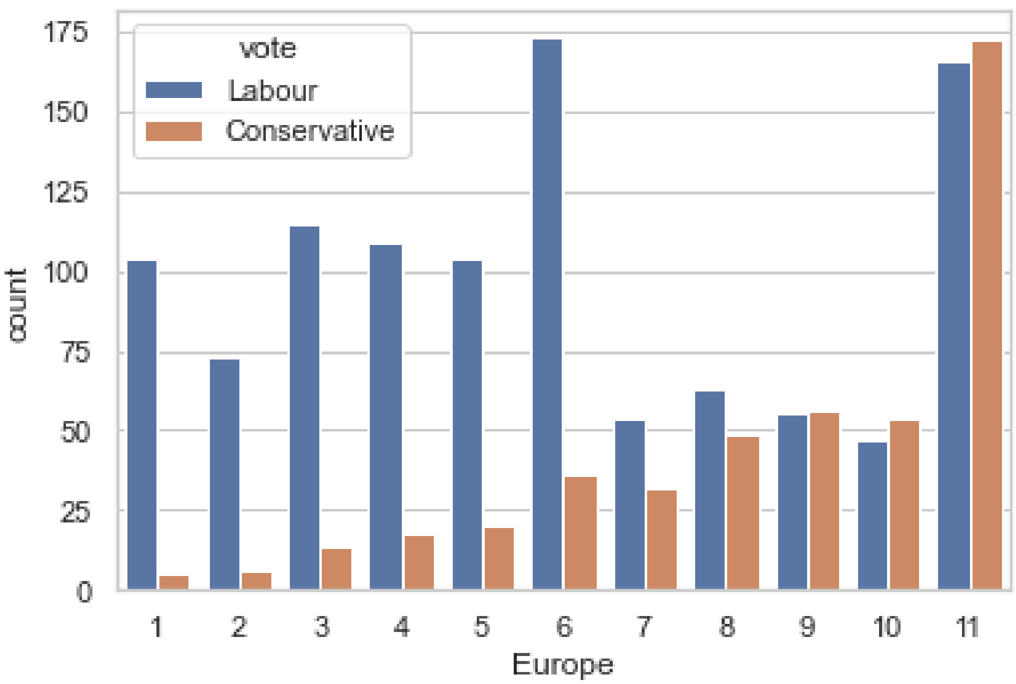


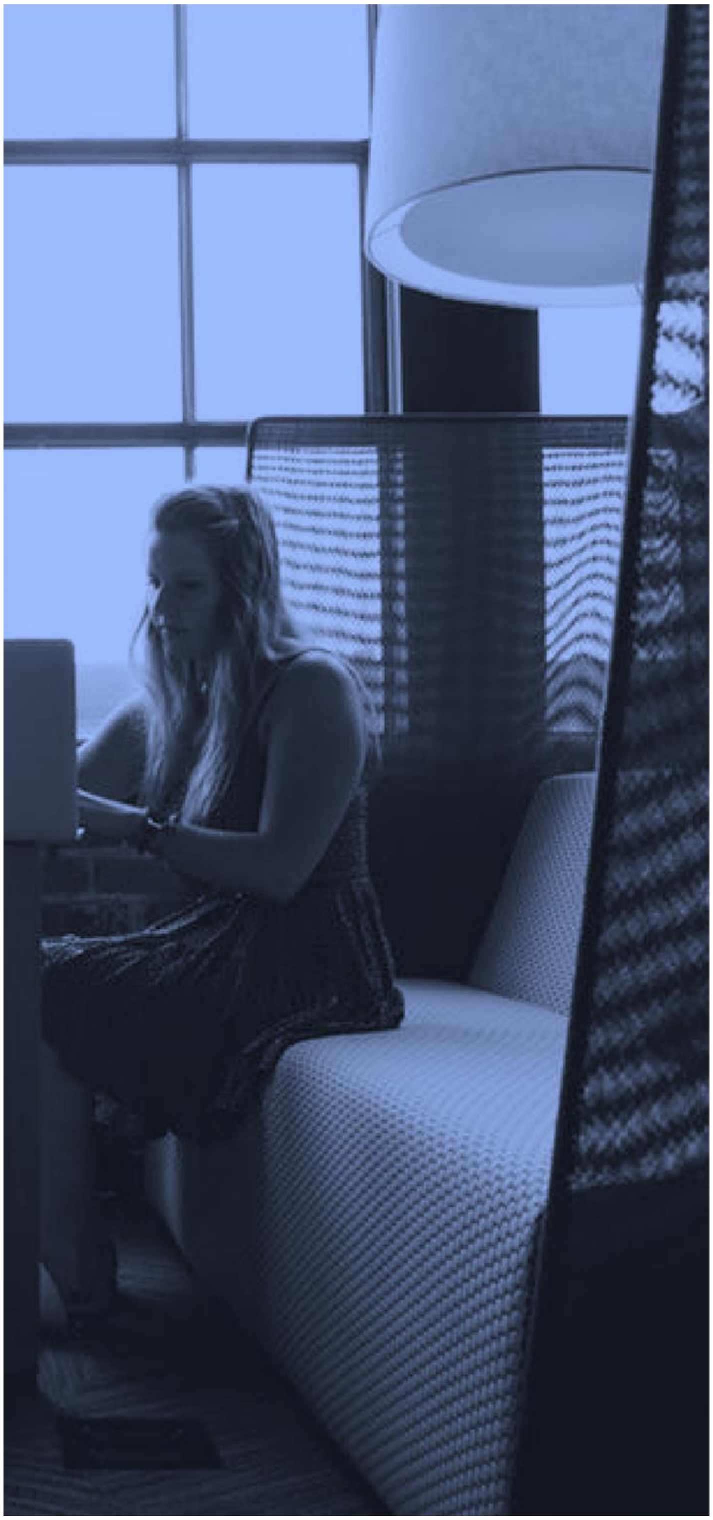
- **When we compare the assessment for Blair (Labour Leader) and Hague(the conservative leader) majority voters have assessed Blair 4 comparing with that of Hague which is only 2 which is a clear indicator for News channel CNBE.**



- **When we look at the age of the voters, most voters range from 30 to 75, so news channels CNBE needs to target these age group to get more accurate results.**

- When we look at the Europe: an 11-point scale that measures respondents' attitudes toward European integration. High scores represent 'Eurosceptic' sentiment most of the conservative voters are Eurosceptic while labour voters are not Eurosceptic which will help in identifying the voting pattern





Problem 2

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

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

(Hint: use .words(), .raw(), .sent() for extracting counts)

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

President Franklin D. Roosevelt inaugural speech in 1941 contains 1360 words, 7571 characters and 37 sentences.

President John F. Kennedy inaugural speech in 1961 contains 1390 words, 7618 characters and 25 sentences.

President Richard Nixon inaugural speech in 1973 contains 1819 words, 9991 characters and 47 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.

Number of words in President Roosevelt inaugural speech **before** Removing stop words: **1526**

Number of words in President Roosevelt inaugural speech **after** Removing stop words: **871**

['On', 'national', 'day', 'inauguration', 'since', '1789', ',', 'people', 'renewed', 'sense', 'dedication', 'United', 'States', '.']

Number of words in President Kennedy inaugural speech **before** Removing stop words: **1543**

Number of words in President Kennedy inaugural speech **after** Removing stop words: **904**

['Vice', 'President', 'Johnson', ',', 'Mr.', 'Speaker', ',', 'Mr.', 'Chief', 'Justice', ',', 'President', 'Eisenhower', ',', 'Vice', 'President', 'Nixon', ',', 'President', 'Truman', ',', 'reverend', 'clergy', ',', 'fellow', 'citizens', ',', 'observe', 'today', 'victory', 'party', ',', 'celebration', 'freedom', '--', 'symbolizing', 'end', ',', 'well', 'beginning', '--', 'signifying', 'renewal', ',', 'well', 'change', '.']

Number of words in President Nixon inaugural speech **before** Removing stop words: **2006**

Number of words in President Nixon inaugural speech **after** Removing stop words: **1094**

['Mr.', 'Vice', 'President', ',', 'Mr.', 'Speaker', ',', 'Mr.', 'Chief', 'Justice', ',', 'Senator', 'Cook', ',', 'Mrs.', 'Eisenhower', ',', 'fellow', 'citizens', 'great', 'good', 'country', 'share', 'together', ':', 'When', 'met', 'four', 'years', 'ago', ',', 'America', 'bleak', 'spirit', ',', 'depressed', 'prospect', 'seemingly', 'endless', 'war', 'abroad', 'destructive', 'conflict', 'home', '.']

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)

The top 3 words in President Roosevelt inaugural speech are **'nation', 'know', 'spirit'**

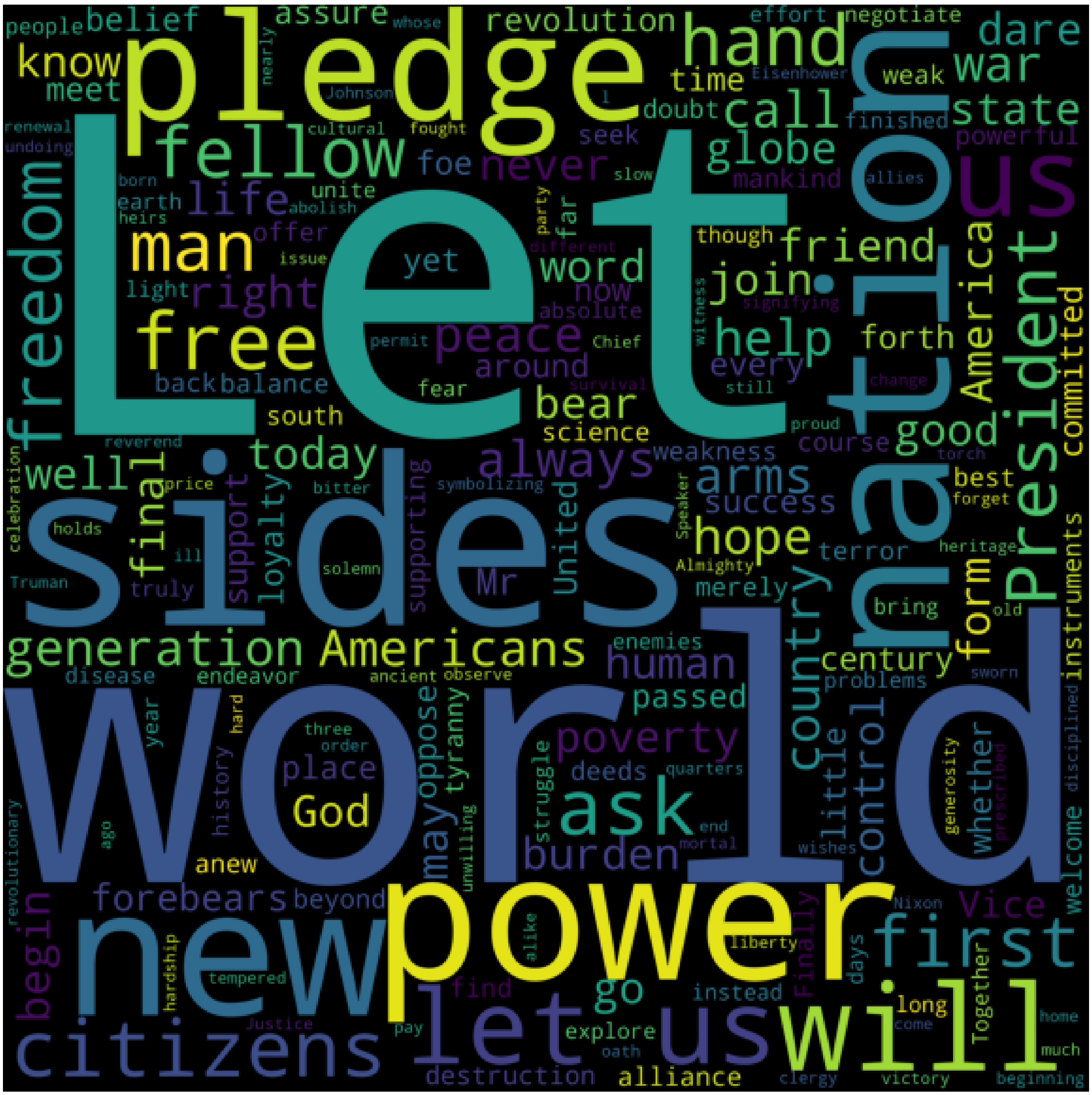
The top 3 words in President Kennedy inaugural speech are **'let', 'us', 'world'**

The top 3 words in President Nixon inaugural speech are **'us', 'let', 'america'**

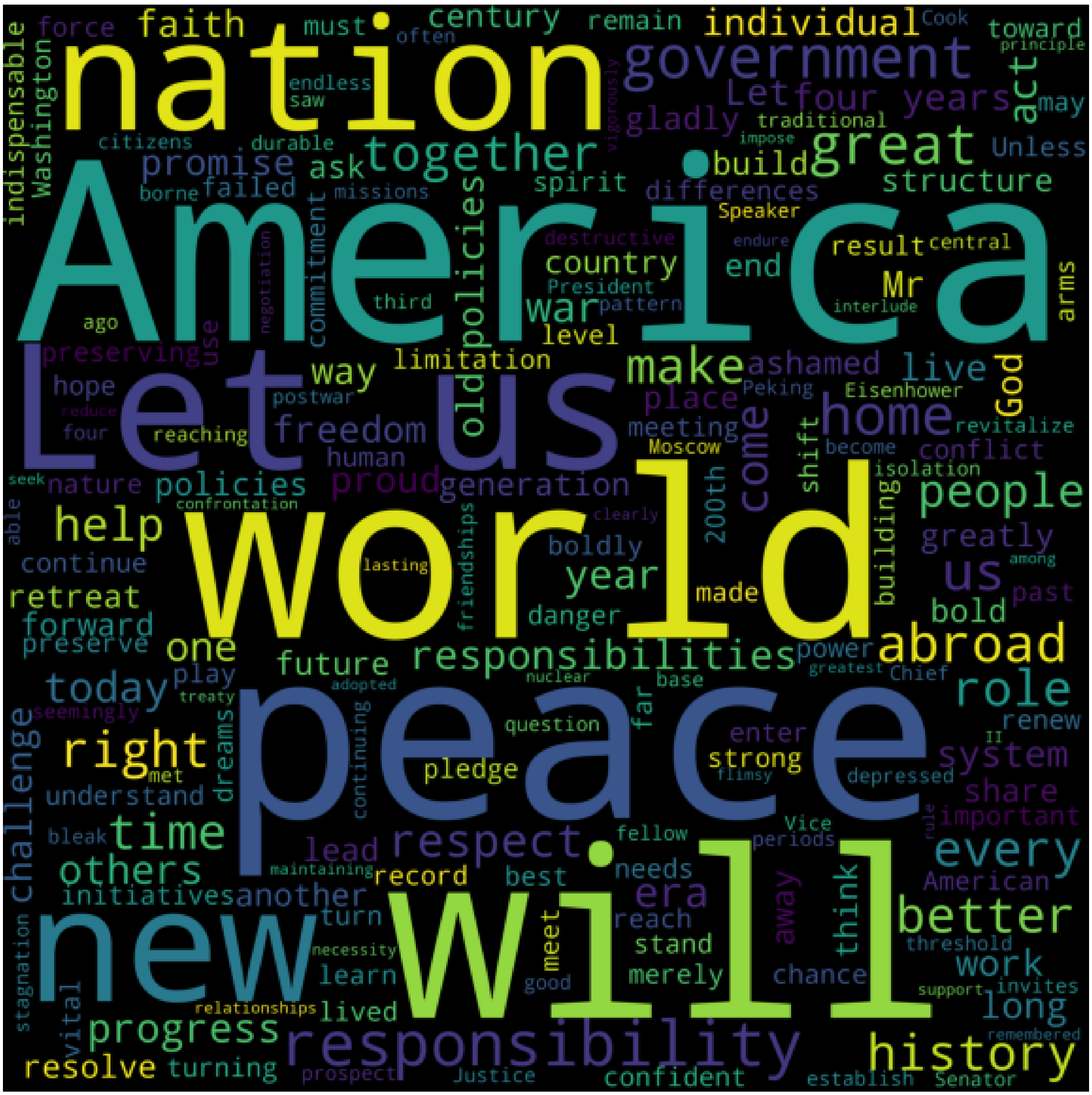
2.4) Plot the word cloud of each of the three speeches. (After removing the stopwords)

1. **word cloud for President Franklin D. Roosevelt 1941 inaugural speech**

2. **word cloud for President John F. Kennedy 1961 inaugural speech**



3. **word cloud for President Richard Nixon 1973 inaugural speech**



THANK YOU !