MACHINE LEARNING PROJECT

Aniket Ganguly

PGPDSBA – JULY 2023

Problem - 1

Context

CNBE, a prominent news channel, is gearing up to provide insightful coverage of recent elections, recognizing the importance of data-driven analysis. A comprehensive survey has been conducted, capturing the perspectives of 1525 voters across various demographic and socio-economic factors. This dataset encompasses 9 variables, offering a rich source of information regarding voters' characteristics and preferences.

Objective

The primary objective is to leverage machine learning to build a predictive model capable of forecasting which political party a voter is likely to support. This predictive model, developed based on the provided information, will serve as the foundation for creating an exit poll. The exit poll aims to contribute to the accurate prediction of the overall election outcomes, including determining which party is likely to secure the majority of seats.

Define the problem and perform Exploratory Data Analysis

- Problem definition - Check shape, Data types, and statistical summary - Univariate analysis - Multivariate analysis - Use appropriate visualizations to identify the patterns and insights - Key meaningful observations on individual variables and the relationship between variables

Data Pre-processing

Prepare the data for modelling: - Outlier Detection(treat, if needed)) - Encode the data - Data split - Scale the data (and state your reasons for scaling the features)

Model Building

- Metrics of Choice (Justify the evaluation metrics) - Model Building (KNN, Naive bayes, Bagging, Boosting) - Metrics of Choice (Justify the evaluation metrics) - Model Building (KNN, Naive bayes, Bagging, Boosting)

Model Performance evaluation

- Check the confusion matrix and classification metrics for all the models (for both train and test dataset) - ROC-AUC score and plot the curve - Comment on all the model performance

Model Performance improvement

- Improve the model performance of bagging and boosting models by tuning the model - Comment on the model performance improvement on training and test data

Final Model Selection

- Compare all the model built so far - Select the final model with the proper justification - Check the most important features in the final model and draw inferences.

Actionable Insights & Recommendations

- Compare all four models - Conclude with the key takeaways for the business

Solution - 1

Define the problem and perform Exploratory Data Analysis

- Problem definition - Check shape, Data types, and statistical summary - Univariate analysis - Multivariate analysis - Use appropriate visualizations to identify the patterns and insights - Key meaningful observations on individual variables and the relationship between variables

- Shape of the Data

```
(1525, 10)
```

- Data types in the Dataset

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

- Statistical Summary of the data

	count	unique	top	freq	mean	std	min	25%	50%	75%	max
vote	1517	2	Labour	1057	NaN	NaN	NaN	NaN	NaN	NaN	NaN
age	1517.0	NaN	NaN	NaN	54.241266	15.701741	24.0	41.0	53.0	67.0	93.0
economic.cond.national	1517.0	NaN	NaN	NaN	3.245221	0.881792	1.0	3.0	3.0	4.0	5.0
economic.cond.household	1517.0	NaN	NaN	NaN	3.137772	0.931069	1.0	3.0	3.0	4.0	5.0
Blair	1517.0	NaN	NaN	NaN	3.335531	1.174772	1.0	2.0	4.0	4.0	5.0
Hague	1517.0	NaN	NaN	NaN	2.749506	1.232479	1.0	2.0	2.0	4.0	5.0
Europe	1517.0	NaN	NaN	NaN	6.740277	3.299043	1.0	4.0	6.0	10.0	11.0
political.knowledge	1517.0	NaN	NaN	NaN	1.540541	1.084417	0.0	0.0	2.0	2.0	3.0
gender	1517	2	female	808	NaN	NaN	NaN	NaN	NaN	NaN	NaN

- Removed Duplicate values

```
8
```

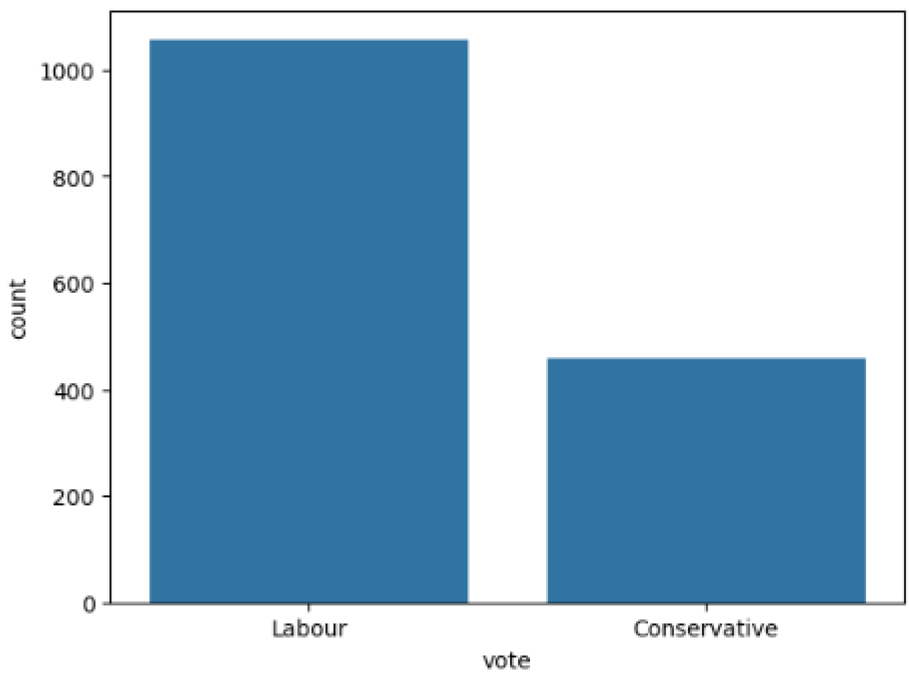
- Statistical summary of numerical columns

	Ad - Length	Ad- Width	Ad Size	Available_Impressions	Matched_Queries	Impressions	Clicks	Spend	Fee	Revenue	CTR	CPM	CPC
count	23066.000000	23066.000000	23066.000000	2.306600e+04	2.306600e+04	2.306600e+04	23066.000000	23066.000000	23066.000000	23066.000000	23066.000000	23066.000000	23066.000000
mean	385.163097	337.896037	96674.468048	2.131361e+06	1.147036e+06	1.096652e+06	9470.897945	2490.930382	0.336056	1745.232210	7.990410	8.046289	0.320174
std	233.651434	203.092885	61538.329557	3.592680e+06	1.956591e+06	1.887081e+06	12831.144277	3300.194828	0.028942	2448.207098	7.684490	6.419515	0.289672
min	120.000000	70.000000	33600.000000	4.862500e+02	1.602500e+02	1.492500e+02	13.000000	1.030000	0.250000	0.669500	0.183992	1.194793	0.056989
25%	120.000000	250.000000	72000.000000	3.367225e+04	1.828250e+04	7.990500e+03	710.000000	85.180000	0.330000	55.365375	0.265107	1.749084	0.089736
50%	300.000000	300.000000	72000.000000	4.837710e+05	2.580875e+05	2.252900e+05	4425.000000	1425.125000	0.350000	926.335000	9.391248	8.371566	0.139347
75%	720.000000	600.000000	84000.000000	2.527712e+06	1.180700e+06	1.112428e+06	12793.750000	3121.400000	0.350000	2091.338150	13.470571	13.042018	0.546242
max	728.000000	600.000000	216000.000000	1.436391e+07	7.803449e+06	7.473380e+06	50662.000000	12899.765000	0.350000	9674.825000	23.782197	20.378835	0.925477

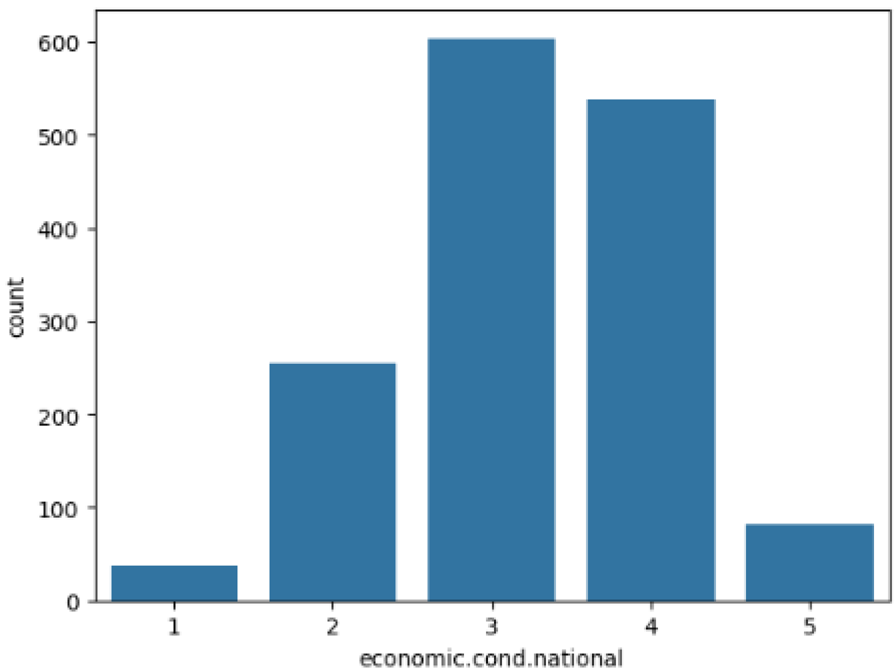
- Shape of the data

```
(23066, 19)
```

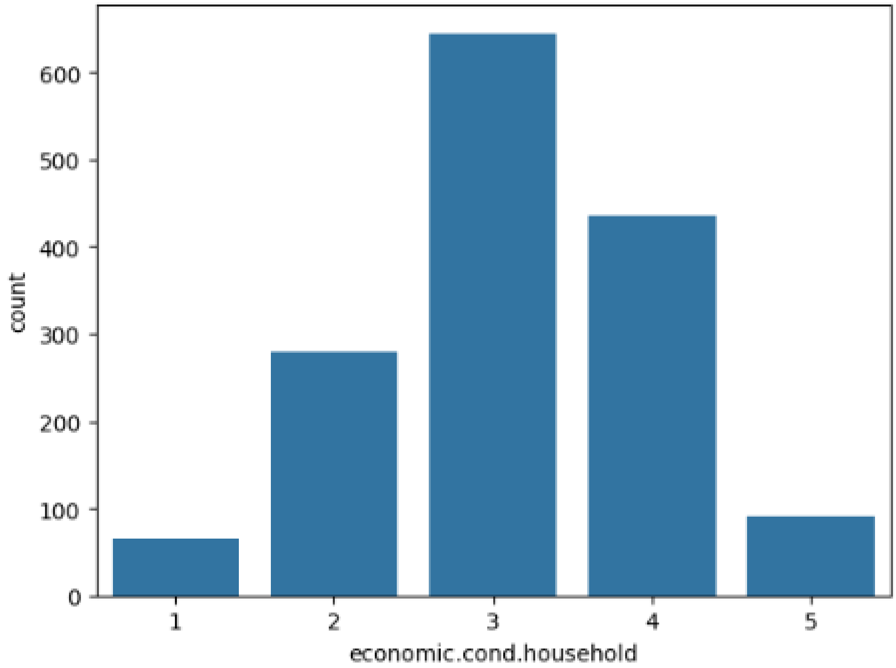
- Univariate Analysis

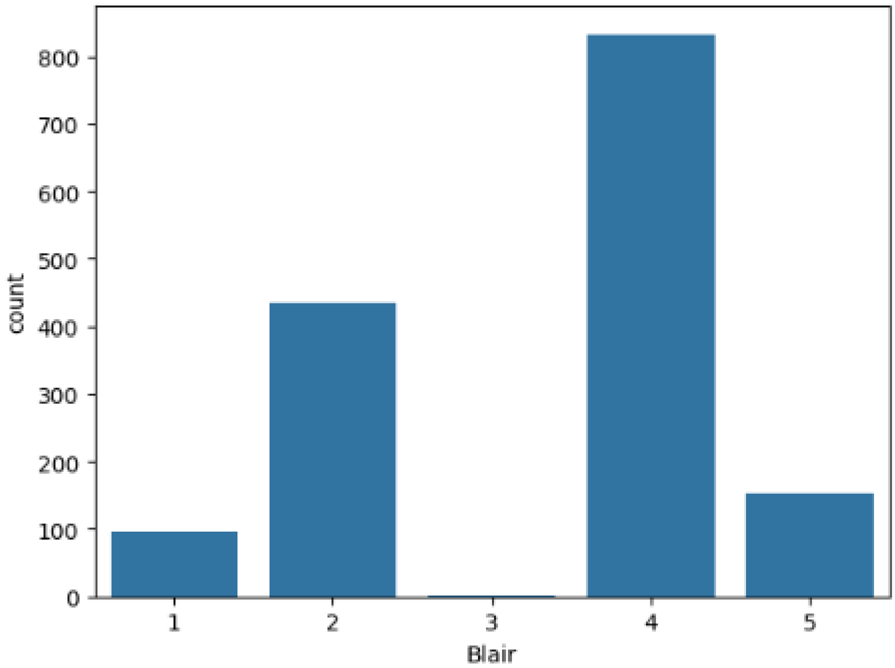


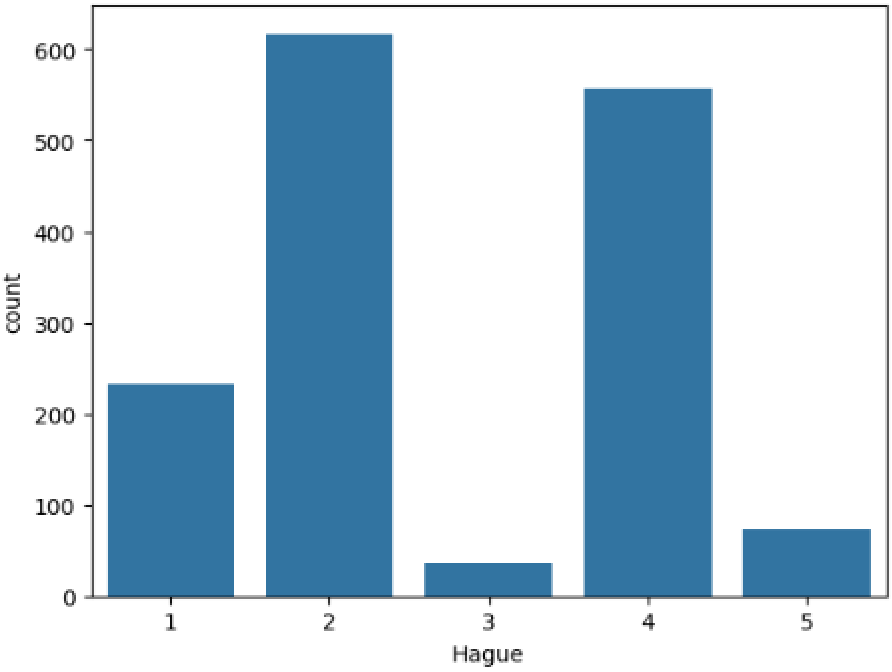
1000
800
600
400
200
0
count
Labour
Conservative
vote

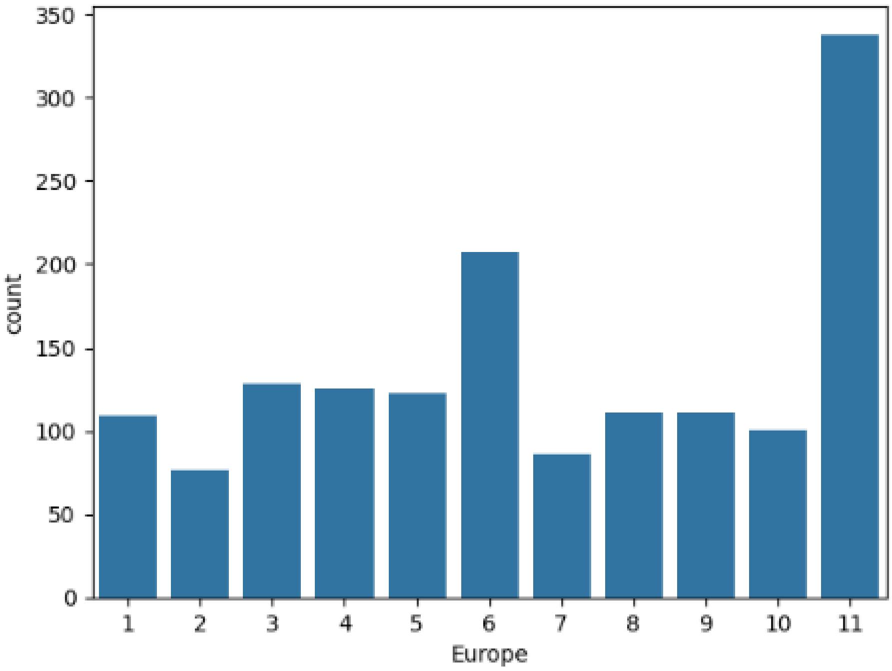


600
500
400
300
200
100
0
count
1
2
3
4
5
economic.cond.national

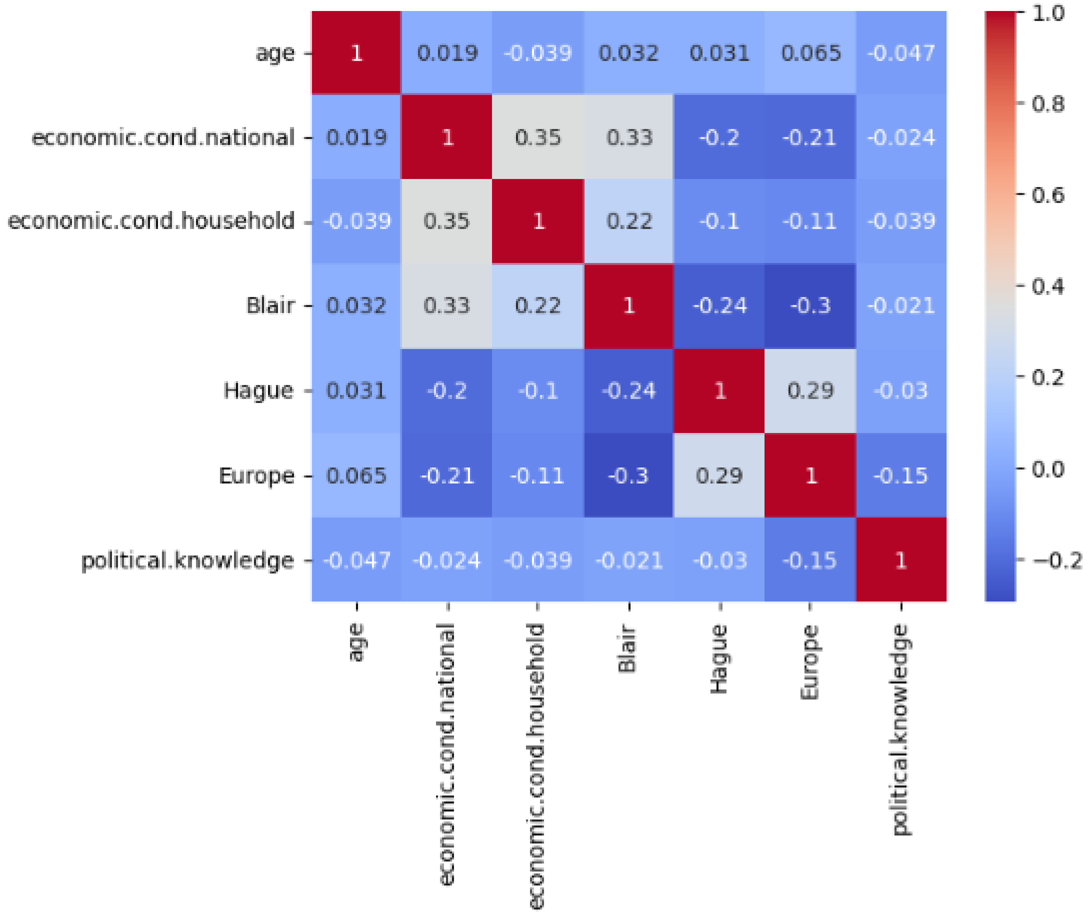

600
500
400
300
200
100
0
count
1
2
3
4
5
economic.cond.household


800
700
600
500
400
300
200
100
0
count
1
2
3
4
5
Blair





- Multivariate Analysis

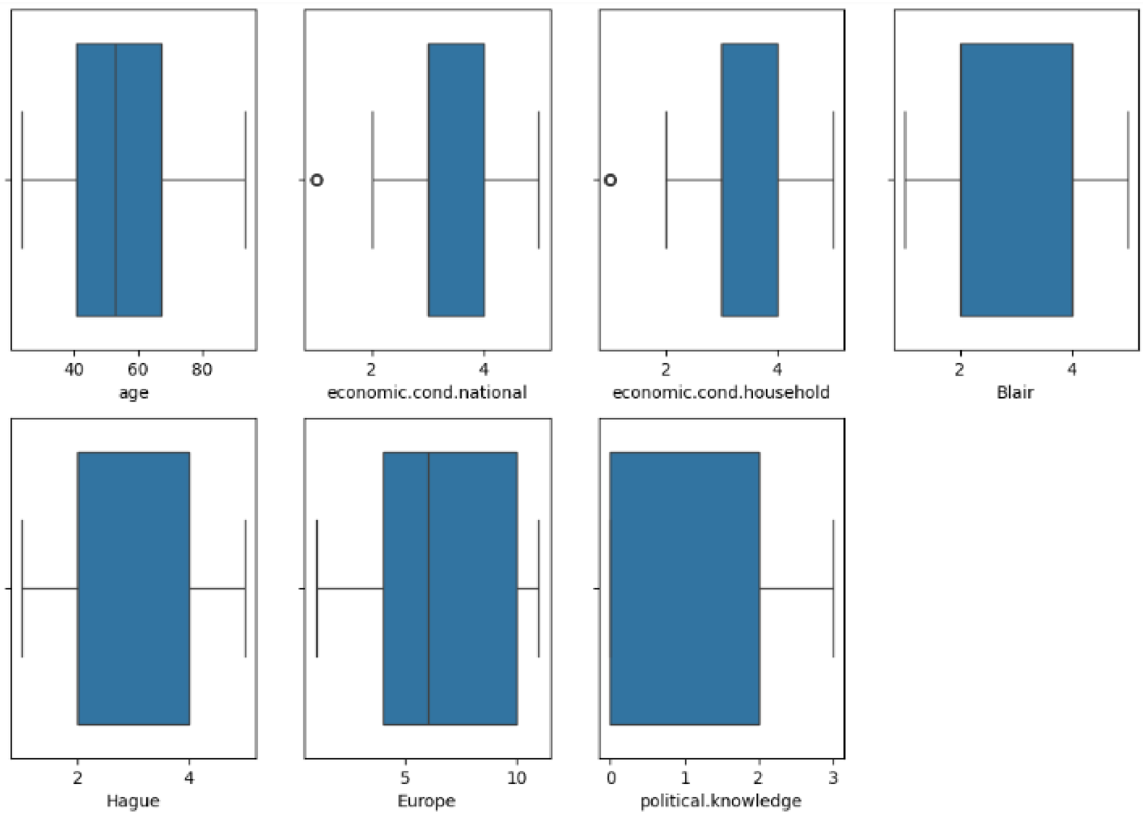


- Key Points
  - "votes" are larger in numbers in Labour.
  - More female voters than male.
  - There is no strong correlation between the variables.

Data Pre-processing

Prepare the data for modelling: - Outlier Detection(treat, if needed)) - Encode the data - Data split - Scale the data (and state your reasons for scaling the features)

- Outlier Detection



- Encode the Data

[ ]

	age	economic.cond.national	economic.cond.household	Blair	Hague	Europe	political.knowledge	gender_male	vote_Labour
1505	75	3	2	4	4	6	2	1	1
1506	62	3	3	4	1	6	2	1	1
1507	52	2	1	1	4	8	2	0	0
1508	70	2	2	4	2	11	2	1	0
1509	67	3	3	5	2	4	2	0	1
1510	73	4	3	4	2	11	0	1	1
1511	63	3	3	4	2	8	2	0	1
1512	75	3	3	2	4	7	1	1	0
1513	46	3	3	4	2	4	2	1	1
1514	74	3	3	5	2	11	0	0	1
1515	82	2	2	2	1	11	2	0	0
1516	30	3	4	4	2	4	2	1	1
1517	76	4	3	2	2	11	2	1	1
1518	50	3	4	4	2	5	2	1	1
1519	35	3	4	4	2	8	2	1	0
1520	67	5	3	2	4	11	3	1	0
1521	73	2	2	4	4	8	2	1	0
1522	37	3	3	5	4	2	2	1	1
1523	61	3	3	1	4	11	2	1	0
1524	74	2	3	2	4	11	0	0	0

- “vote” and “gender” needed to be encoded as they are categorical columns, converting them to numerical columns will help in fitting the data in ML models.

- Scale the data
- **K-Nearest Neighbors (KNN):**

KNN is a distance-based algorithm that classifies a data point based on the majority class of its k-nearest neighbors. The algorithm's performance can be sensitive to the scale of features. If features have different scales, those with larger magnitudes may dominate the distance calculation.

- **Gaussian Naive Bayes (GaussianNB):**

GaussianNB assumes that the features follow a Gaussian (normal) distribution. While the algorithm is not inherently sensitive to feature scales, scaling can still be beneficial in cases where the input features have significantly different magnitudes. It might help in achieving better convergence during training.

- **BaggingClassifier:**

Bagging, short for Bootstrap Aggregating, is an ensemble method that combines multiple base models, often decision trees. While decision trees are not typically sensitive to feature scales, the BaggingClassifier as a whole may benefit from scaling, especially if the base models are sensitive to feature scales.

- **AdaBoostClassifier:**

AdaBoost combines multiple weak classifiers to create a strong classifier. Weak classifiers are typically decision trees with limited depth. While decision trees themselves may not be sensitive to feature scales, AdaBoost may benefit from feature scaling as it adjusts the weights of misclassified samples at each iteration.

Benefits of Scaling:

- **Improved Convergence:** Gradient-based optimization algorithms, which are often used in training machine learning models, can converge faster when features are on a similar scale.
- **Equal Weight to Features:** Scaling ensures that all features contribute equally to the model, preventing features with larger scales from dominating the learning process.

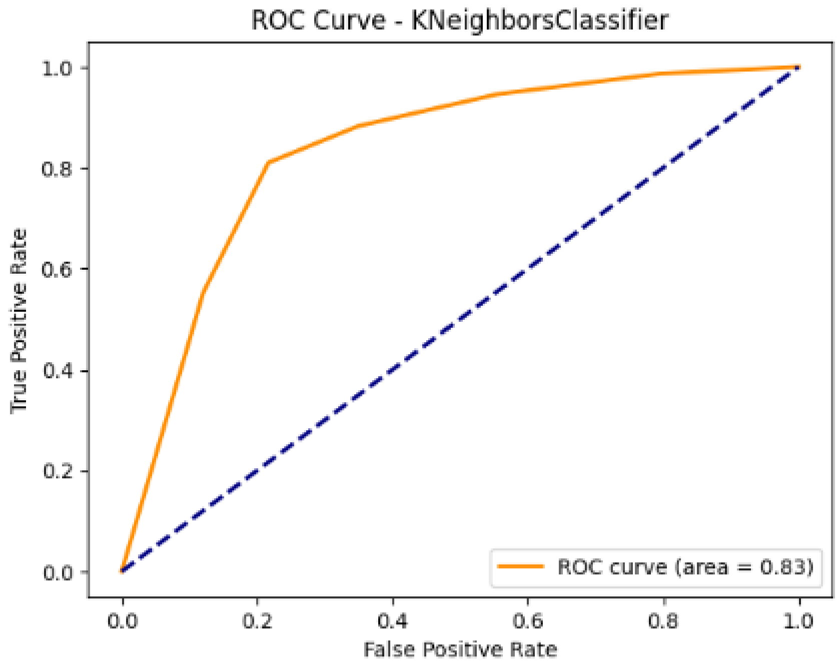
- **Distance-Based Algorithms:** Algorithms like KNN rely on distance metrics, and scaling ensures that all features contribute proportionally to the distance calculation.

Model Building

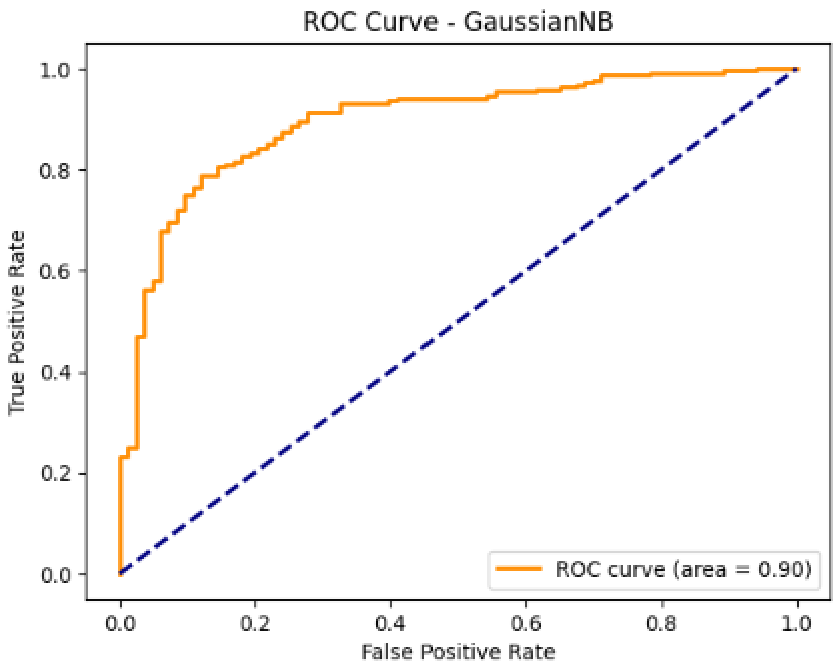
- Metrics of Choice (Justify the evaluation metrics) - Model Building (KNN, Naive bayes, Bagging, Boosting) - Metrics of Choice (Justify the evaluation metrics) - Model Building (KNN, Naive bayes, Bagging, Boosting)

Model Performance evaluation

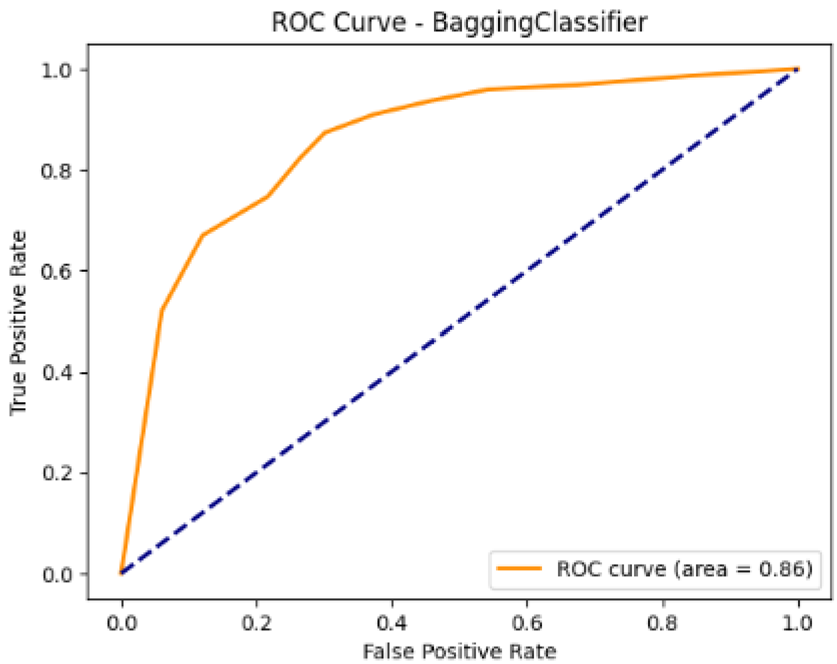
- Check the confusion matrix and classification metrics for all the models (for both train and test dataset) - ROC-AUC score and plot the curve - Comment on all the model performance



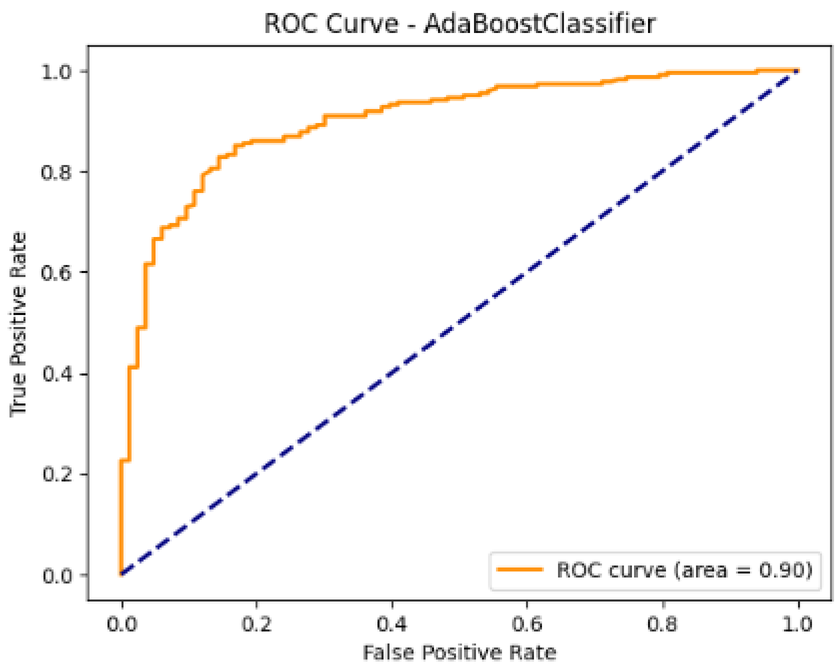
```
Model: KNeighborsClassifier
Train Data Confusion Matrix:
[[287  90]
 [ 79 757]]
Train Classification Report:
              precision    recall  f1-score   support

           0       0.78      0.76      0.77       377
           1       0.89      0.91      0.90       836

    accuracy                           0.86      1213
   macro avg       0.84      0.83      0.84      1213
weighted avg       0.86      0.86      0.86      1213

Test Data Confusion Matrix:
[[ 54  29]
 [ 26 195]]
Test Classification Report:
              precision    recall  f1-score   support

           0       0.68      0.65      0.66        83
           1       0.87      0.88      0.88       221

    accuracy                           0.82       304
   macro avg       0.77      0.77      0.77       304
weighted avg       0.82      0.82      0.82       304
```



```
Model: GaussianNB
Train Data Confusion Matrix:
[[263 114]
 [ 94 742]]
Train Classification Report:
              precision    recall  f1-score   support

           0       0.74      0.70      0.72       377
           1       0.87      0.89      0.88       836

    accuracy                           0.83      1213
   macro avg       0.80      0.79      0.80      1213
weighted avg       0.83      0.83      0.83      1213

Test Data Confusion Matrix:
[[ 60  23]
 [ 23 198]]
Test Classification Report:
              precision    recall  f1-score   support

           0       0.72      0.72      0.72        83
           1       0.90      0.90      0.90       221

    accuracy                           0.85       304
   macro avg       0.81      0.81      0.81       304
weighted avg       0.85      0.85      0.85       304
```



```
Model: BaggingClassifier
Train Data Confusion Matrix:
[[368   9]
 [ 14 822]]
Train Classification Report:
              precision    recall  f1-score   support

           0       0.96      0.98      0.97       377
           1       0.99      0.98      0.99       836

    accuracy                           0.98      1213
   macro avg       0.98      0.98      0.98      1213
weighted avg       0.98      0.98      0.98      1213

Test Data Confusion Matrix:
[[ 58  25]
 [ 28 193]]
Test Classification Report:
              precision    recall  f1-score   support

           0       0.67      0.70      0.69        83
           1       0.89      0.87      0.88       221

    accuracy                           0.83       304
   macro avg       0.78      0.79      0.78       304
weighted avg       0.83      0.83      0.83       304
```



```
Model: AdaBoostClassifier
Train Data Confusion Matrix:
[[269 108]
 [ 87 749]]
Train Classification Report:
              precision    recall  f1-score   support

           0       0.76      0.71      0.73       377
           1       0.87      0.90      0.88       836

    accuracy                           0.84      1213
   macro avg       0.81      0.80      0.81      1213
weighted avg       0.84      0.84      0.84      1213

Test Data Confusion Matrix:
[[ 58  25]
 [ 20 201]]
Test Classification Report:
              precision    recall  f1-score   support

           0       0.74      0.70      0.72        83
           1       0.89      0.91      0.90       221

    accuracy                           0.85       304
   macro avg       0.82      0.80      0.81       304
weighted avg       0.85      0.85      0.85       304
```

Model Performance improvement

- Improve the model performance of bagging and boosting models by tuning the model - Comment on the model performance improvement on training and test data

```
Best Parameters for Bagging: {'n_estimators': 10}
Best Parameters for Boosting: {'learning_rate': 0.1, 'n_estimators': 200}
Tuned Bagging Model Classification Report:
              precision    recall  f1-score   support

           0       0.64      0.70      0.67        83
           1       0.88      0.86      0.87       221

    accuracy                           0.81       304
   macro avg       0.76      0.78      0.77       304
weighted avg       0.82      0.81      0.81       304

Tuned Boosting Model Classification Report:
              precision    recall  f1-score   support

           0       0.74      0.61      0.67        83
           1       0.86      0.92      0.89       221

    accuracy                           0.84       304
   macro avg       0.80      0.77      0.78       304
weighted avg       0.83      0.84      0.83       304
```

1. **Training Data:**

- **Improved Convergence**: Scaling can lead to faster convergence during the training process, especially for optimization algorithms that involve gradient descent. This is because features on a similar scale allow the optimization algorithm to find the optimal solution more efficiently.
- **Equal Contribution**: Scaling ensures that all features contribute more equally to the model during training, preventing features with larger scales from dominating the learning process.

2. **Test Data:**

- **Consistent Generalization**: Scaling helps ensure that the model generalizes well to unseen data (test data). When the model encounters new data during testing, having consistent feature scales ensures that the model makes predictions based on the patterns it learned during training.

3. **Impact on Different Algorithms:**

- **K-Nearest Neighbors (KNN)**: KNN is sensitive to the scale of features because it relies on distance metrics. Scaling can lead to more accurate and reliable predictions, especially when features have different magnitudes.
- **Gaussian Naive Bayes (GaussianNB)**: While GaussianNB is not as sensitive to feature scales, scaling may still contribute to better convergence during training, resulting in more robust models.
- **BaggingClassifier and AdaBoostClassifier**: Decision trees, often used in these ensemble methods, are not as sensitive to feature scales. However, the overall ensemble may still benefit from scaling, particularly if base models are sensitive to scales.

4. **Considerations:**

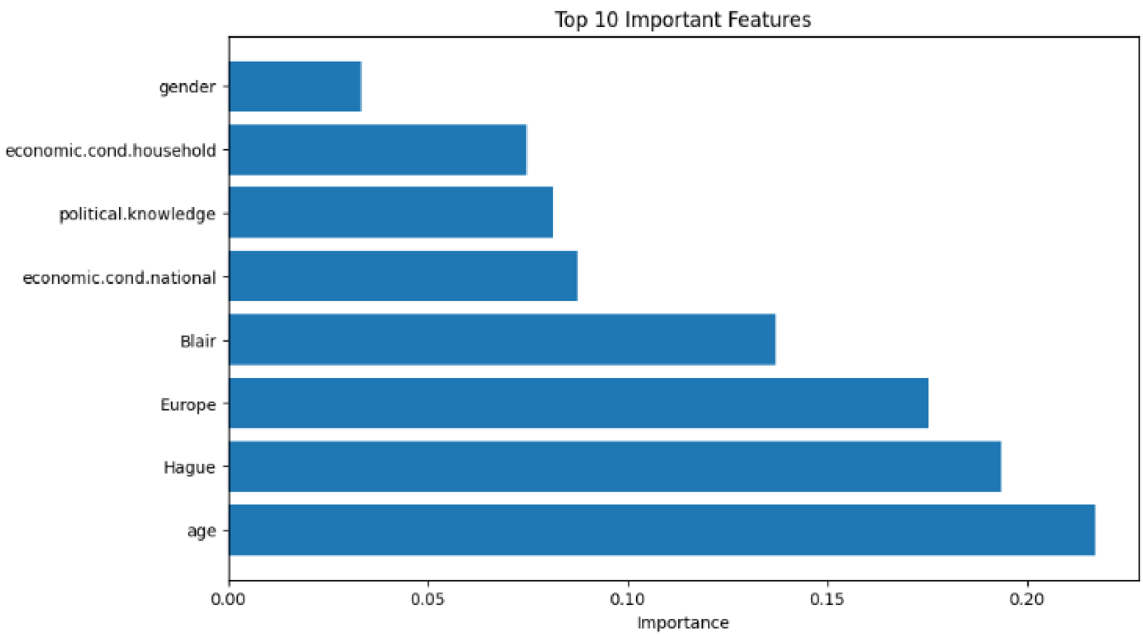
- **Algorithm Robustness**: Some algorithms, such as decision tree-based models, may be inherently robust to feature scales. In such cases, the impact of scaling may be minimal, and the choice of scaling may depend on the specific characteristics of the dataset.

- **Experimentation**: It's essential to experiment with both scaled and unscaled data to observe the actual impact on model performance. Cross-validation can help provide a more robust assessment of model performance.

Final Model Selection

- Compare all the model built so far - Select the final model with the proper justification - Check the most important features in the final model and draw inferences.

- Among all the 4 models it seems for the current dataset the best model will be **AdaBoostClassifier.** It has got higher AUC-ROC value and higher accuracy for train and test data. (**AUC-ROC provides a single value summarizing the model's discrimination ability. A higher AUC-ROC indicates better performance.)**
- Most Important features




```
Feature Importances:
                   Feature  Importance
0                      age    0.217051
4                    Hague    0.193485
5                   Europe    0.175244
3                    Blair    0.137202
1   economic.cond.national    0.087545
6      political.knowledge    0.081259
2  economic.cond.household    0.074865
7                   gender    0.033350
```

- **Inference:**

> - The important variable in predicting the dependent variables are “Age”, “Hague”, “Europe”, “Blair”
> - As the frequency distribution suggests most of the people gave 4 starts to “Blair” and there are larger number of people gave 2 stars to “Hague” which made an impact in the dependent variable “vote”

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

Problem 2 - Define the problem and Perform Exploratory Data Analysis

-Problem Definition - Find the number of Character, words & sentences in all three speeches

Problem 2 - Text cleaning

- Stopword removal - Stemming - find the 3 most common words used in all three speeches

Problem 2 - Plot Word cloud of all three speeches

- Show the most common words used in all three speeches in the form of word clouds

Solution - 2

Problem 2 - Define the problem and Perform Exploratory Data Analysis

-Problem Definition - Find the number of Character, words & sentences in all three speeches

```
[nltk_data] Downloading package inaugural to /root/nltk_data...
[nltk_data]   Package inaugural is already up-to-date!
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
Roosevelt's Speech Analysis:
Number of Characters: 7571
Number of Words: 1526
Number of Sentences: 68

Kennedy's Speech Analysis:
Number of Characters: 7618
Number of Words: 1543
Number of Sentences: 52

Nixon's Speech Analysis:
Number of Characters: 9991
Number of Words: 2006
Number of Sentences: 68
```

Problem 2 - Text cleaning

- Stopword removal - Stemming - find the 3 most common words used in all three speeches

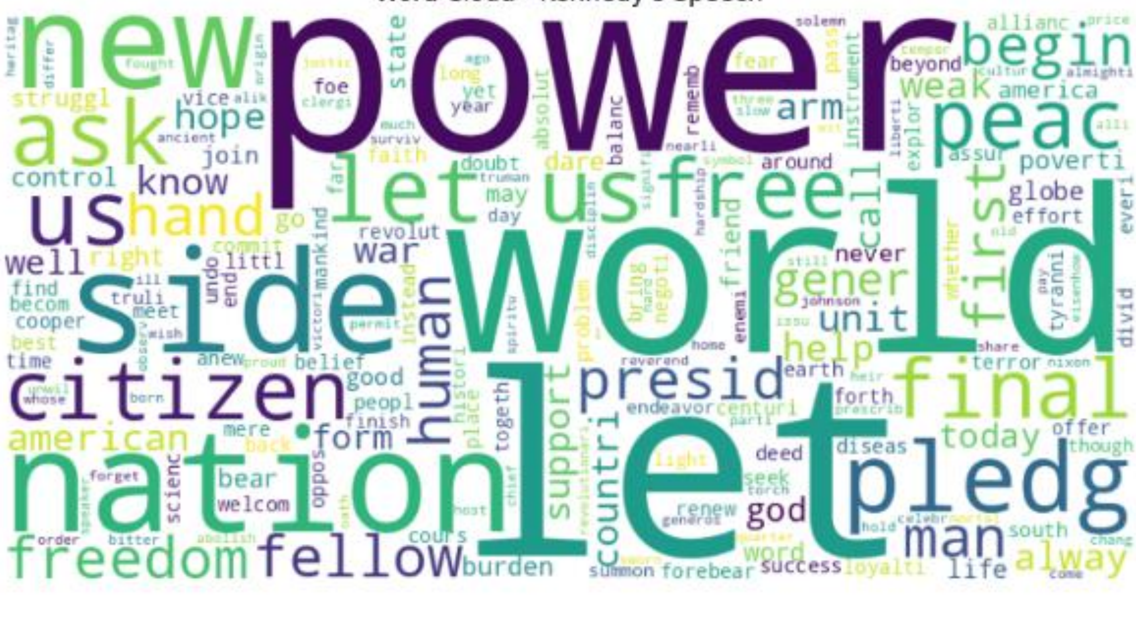
```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
Roosevelt's 3 Most Common Words:
nation: 17
know: 10
peopl: 9

Kennedy's 3 Most Common Words:
let: 16
us: 12
power: 9

Nixon's 3 Most Common Words:
us: 26
let: 22
america: 21
```

Problem 2 - Plot Word cloud of all three speeches

- Show the most common words used in all three speeches in the form of word clouds

Word Cloud - Roosevelt's Speech

Word Cloud - Kennedy's Speech



Word Cloud - Nixon's Speech