# Regression Analysis:
# Real-time Portugal 2019 Election Results


| Members | PSID |
|---|---|
| Sai Krishna Are | 2310421 |
| Vamsi Athota | 2310423 |
| Saran Teja Mallela | 2315340 |
| Likhitha Reddy Kesara | 2293835 |
| Nidadavolu Aditya | 2293835 |
| Preethi Kakarla | 2307250 |


## 1. Abstract


Real-time Portugal 2019 Election Results to investigate and evaluate the variables impacting the 2019 Portuguese election results. In order to comprehend the connections between different independent factors and the ultimate electoral outcomes, the study explores the field of political data, concentrating on the actual election results. The report starts out by outlining the background of the 2019 elections in Portugal as well as the importance of using regression analysis in the field of political science. It lists the study's goals, highlighting the need to find important factors that might have affected the election outcome.

Demonstrating the regression models that were employed and emphasizing the independent variables that were chosen—such as demographics, economics, voting history, or any other pertinent characteristics—make up the bulk of the report. The magnitude and trajectory of the correlations between these factors and the election results can be inferred through the analysis of regression coefficients as well as statistical significance tests. The section on findings and results presents the empirical data obtained based on the regression models. It throws light on the variables that were highly influential in determining the outcome of the election by identifying important predictors and discussing their implications. Regression and residual plots are two examples of visualizations that can be used to improve comprehension of the model's performance.

The study's limitations, including data constraints and predictions made during the regression modeling process, are addressed to recognize the extent of the investigation, and identify possible avenues for further research. A summary of the major conclusions, suggestions for future research in the area of political regression analysis, and consequences for political analysts as well as policymakers round out the report. Regression analysis's usefulness in comprehending

intricate political events occurring in real time is emphasized throughout the report, which offers a thorough and data-driven analysis of the 2019 elections in Portugal.

## 2. Introduction

Statistical techniques have been instrumental in political analysis in helping to understand the complex dynamics that influence electoral outcomes. The focus of this project's investigation is the 2019 elections in Portugal, a significant occasion in the country's democratic history. Our goal is to use regression analysis to identify the complex relationships between different independent variables as well as the electoral outcomes as we dive deeper into the world of real-time election results. This will help us gain an improved comprehension of the factors that shaped the political scene during this pivotal time.

**2.1 About dataset:** The selected dataset is used to perform regression analysis. The results of the October 6, 2019, Portuguese parliamentary elections are shown in this dataset over time. The data, which pertains to the outcomes of the 27 parties participating in the election, covers a period of 4 hours and 25 minutes at intervals of 5 minutes.

The target variable in the dataset's 28 columns, **"FinalMandates"** indicates the total number of MPs elected.

**2.2. Goal:**

The project's goal is to forecast the number of Members of Parliament that will be elected in Portugal in 2019 at the district and national levels.

## 3. Pre-Processing of the Data

The preliminary process on a dataset for any machine learning project is the Data cleaning/Preparation step. There will be null values and unwanted noise in the dataset that is redundant and needs to be puzzled out.

**3.1. Importing Libraries and Dataset**

Importing the required libraries, such as NumPy, pandas, matplotlib, and seaborn, into our notebook is the first and most important step.After that, we load the dataset in CSV format, transform it to a Pandas DataFrame, and examine the top five rows for data analysis.

**3.2. Cleaning Dataset**

1. **Checking Null Values:** Use the isnull() dataset.sum(), we verify that the dataset is free of missing values.

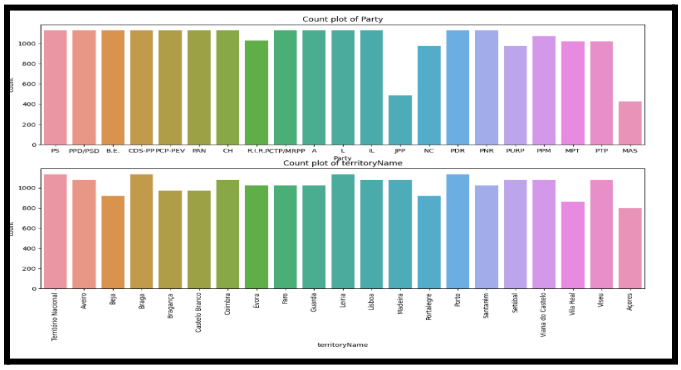
2. **Checking Datatypes:** To look for any discrepancies in the data, we verified the datatypes of every column.

3. **Converting Format:** For improved analysis, we've also switched the datatype in the time column from object to datetime format.

**3.3. Exploratory Data Analysis (EDA) for Data Quality Assessment**

**Correlation:** Using sns.heatmap() to check correlation, it was discovered that certain columns had a strong association.
There are 21 unique values in territory Name and Party columns respectively.

**3.4 DATA VISUALIZATION**: We performed Univariate analysis. We are plotting a bar graph of total count for each party and the territory names from the dataset.




From the above bar graphs, we can see that the minimum counts in the party are JPP and MAS. Most of the other parties have a count of voters ranging between 800 to 1000. There are no duplicate rows in the dataset. The shape of the dataset is (21643, 29). Next, After making a correlation graph,We may infer that a large number of components have correlations greater than 0.9 and can be further lowered to lower the data's dimensionality.Two feature variables, TerritoryName and Party, are subjected to label encoding in order to be converted to numerical values. We have checked for outliers using box plots. Removed outliers using z score. From the above results, we can see that the dataframe shape has changed from (21643, 29) to (18333, 29). As there are around 3300 rows from the dataset removed.

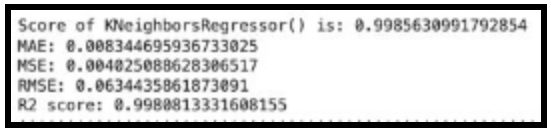
## 4.Building Machine Learning Model

We performed different Machine learning models on our dataset, our approach is that first we split our data into training, validation, and testing datasets. Later, we implemented regression models on the training and validation set and calculated the performances of each model.

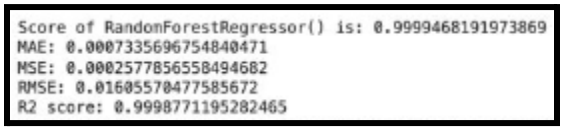
**4.1 Linear Regression:** A scalar response and one or more explanatory factors can be modelled using the linear regression technique. Using the validation set, we forecasted the result after fitting the model with the training data. We calculated the performance using MSE, MAE and RMSE metrics.

```
Score of LinearRegression() is: 0.9860639862141112
MAE: 0.0502111323942291
MSE: 0.03133534918466712
RMSE: 0.17701793464128746
R2 score: 0.9850631623482582
```

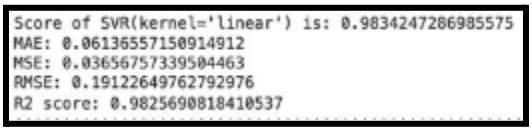
**4.2 K-Nearest Neighbors:** A machine learning approach called KNN (K-Nearest Neighbours) is utilised for regression and classification problems. By locating the K closest data points in the training set and averaging, or using the majority vote, the class or value of a given data point is predicted.



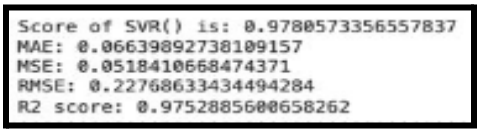

**4.3 Random Forest:** An ensemble model comprising numerous decision trees and is one of most powerful algorithms currently available.




**4.4 Support Vector Regression – Linear:** Support Vector Regression – Linear, Data is classified with the help of a hyperplane. It can be easily separated with a linear line, SVMs are machine learning models which are very popular for classification problems but are also adept when it comes to regression problems.




**4.5 Support Vector Regression - Non-Linear - Radial Basis Function:** Regression and classification problems can be effectively handled by the potent machine learning method known as Radial Basis Function Support Vector Machine (RBF SVM). This non-parametric model performs effectively when dealing with high-dimensional, non-linear data.



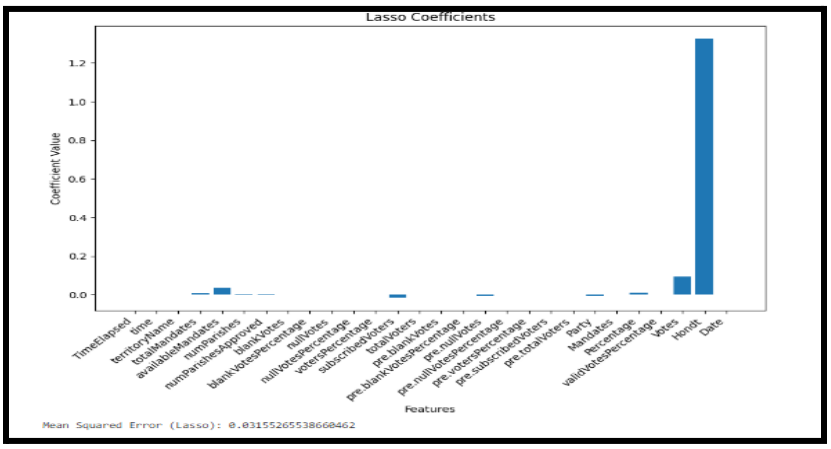

**4.6 Decision trees (Chosen Model):** Decision trees Make predictions by recursively partitioning the input data into subsets based on the values of input features.


We can conclude that Random Forest performs the best among the five algorithms, with Decision Tree closely trailing behind. The slight difference in performance suggests that both models are competitive, but Random Forest edges out as the top performer in this comparison.

**5. First Variable Selection:** When picking between Lasso, KNN, and correlation for choosing which variables to use, it depends on our data, the problem we're solving, and the characteristics of the variables we have. If we want to automatically pick the important variables, especially for linear models, we go for Lasso. It's good at finding a balance between making the model not too complicated and using the right variables. If we're interested in looking at Pairs of variables, we go for Correlation. If the relationships between nearby data points really matter, we go for KNN. We have data where, 1.No pairing is considered: Therefore, we eliminate Correlation 2. Relationship between nearby datapoints doesn't really matter, we eliminate KNN 3. Lasso is the

best option because we need to pick the important variables of too many variables ( which is present in our dataset).

**5.1. Lasso Regression:** It's a method of regularization. For a more accurate forecast, it is preferred to regression approaches. Lasso Regression eliminates the weights of least important predictors. We used grid search and got the best parameters. We have trained the model and got nine best features. Below is the MSE value and Lasso coefficients graph after applying lasso regression for the model: **`Mean Squared Error (Lasso): 0.0315`**




The Selected Variables are : 1.totalMandates, 2.availableMandates, 3.numParishes, 4.numParishesApproved, 5.blankVotesPercentage, 6.votersPercentage, 7.subscribedVoters, 8.pre.nullVotes, 9.pre.nullVotesPercentage, 10.Party, 11.Percentage, 12.Votes, 13.Hondt, 14.Date

**6. Hyperparameter Tuning for all models:**

**6.1 Hyperparameter Tuning:**

- Machine learning models are organized using configuration parameters called hyperparameters.
- Finding the best values for these parameters is known as hyperparameter tuning, and it helps the model perform better.

**6.2 Bias-Variance Tradeoff:**

- A fundamental concept in supervised learning is the bias-variance tradeoff.
- It implies the balance of the two kinds of error affecting model performance: bias & variance.

**6.3 Cross-Validation:**

- A model evaluation method called cross-validation is used to gauge the efficacy of a model on a separate dataset.
- The process entails dividing the dataset into several folds, training the algorithm on some of the folds, and testing the model on the other folds.
- Performance metrics are computed as an average following this procedure is repeated. We performed GridSearchCV to find the best parameters. Now, we tuned the model with the best parameters and performed the regression. The performance of the models after hyperparameter tuning is shown below.

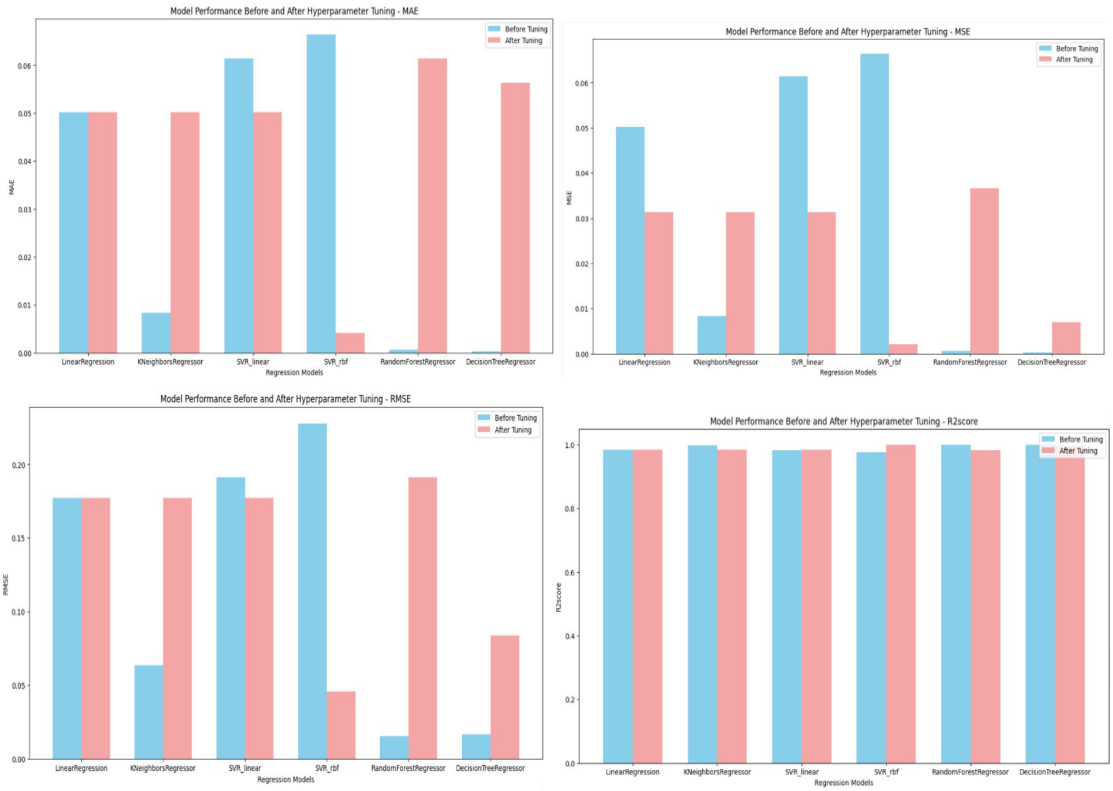
```
Best parameters for LinearRegression: {}
MAE: 0.0502111323942291
MSE: 0.03133534918466712
RMSE: 0.17701793464128746
R2 score: 0.9850631623482582
```

```
Best parameters for KNeighborsRegressor: {'n_neighbors': 3}
MAE: 0.0040905372238887365
MSE: 0.0020907190255431327
RMSE: 0.0457243810843092
R2 score: 0.9990034024999719
```

```
Best parameters for RandomForestRegressor: {'max_depth': 10, 'min_samples_split': 2, 'n_estimators': 100}
MAE: 0.0007727137594051281
MSE: 0.00025668798838313463
RMSE: 0.016021485211525636
R2 score: 0.9998776427609906
```

```
Best parameters for SVR_linear: {'C': 1}
MAE: 0.06136557150914912
MSE: 0.03656757339504463
RMSE: 0.19122649762792976
R2 score: 0.9825690818410537
```

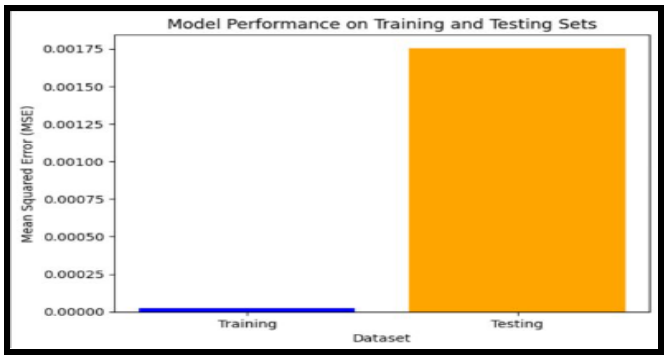
```
Best parameters for SVR_rbf: {'C': 10, 'gamma': 0.1}
MAE: 0.05636795423945647
MSE: 0.006982094420646815
RMSE: 0.08355892783327712
R2 score: 0.9966717967552963
```




Random Forest outperformed both before and after the hyper parameter tuning. Hence, we are performing a variable selection method of Bi-directional elimination as a wrapper method for random Forest.

**7. Bidirectional Elimination:**

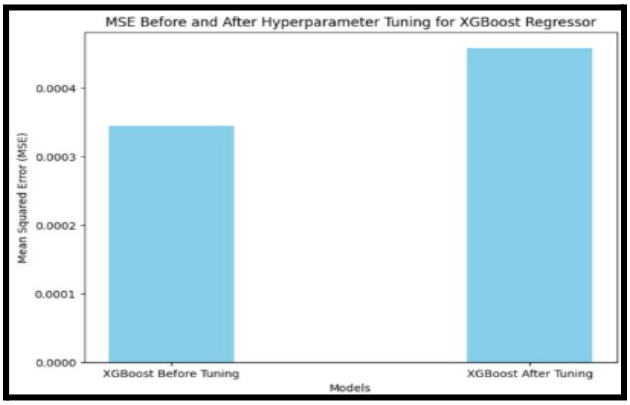
Bidirectional Wrapper Method (BWM) is a feature selection algorithm that combines both forward and backward selection methods to identify the best subset of features for a machine learning model. We have done variable selection using the best models from both step 1 and step 3 (Random Forest). We then again trained this model using only these features as input. We got below MSE values before and after applying Bi-directional elimination for each model:




```
Mean Squared Error (Training): 2.575344333833357e-05
Mean Squared Error (Testing): 0.0017546768475593128
```

**8.1. XGB Regressor:**

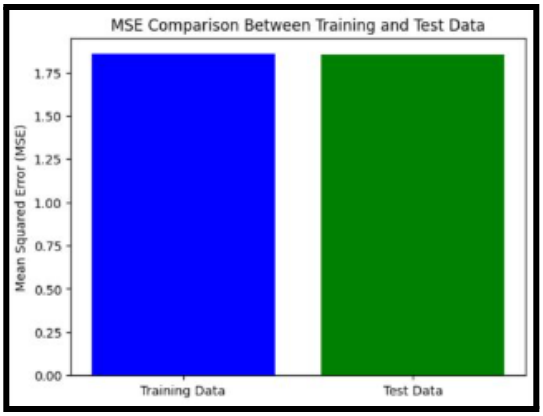
XGBoost Regressor is a specific implementation of the XGBoost algorithm for regression problems. Regression is a type of supervised learning where the goal is to predict a continuous output variable (also called the dependent variable) based on one or more input features (independent variables).

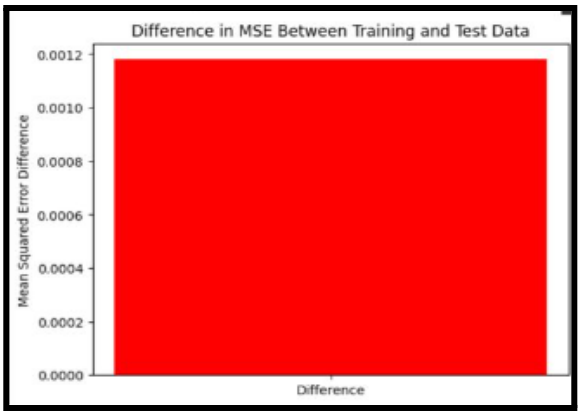



The observed values for MSE before and after hypertuning is 0.000344 and 0.000458 respectively.

**8.2: Extreme Machine Learning Model:**

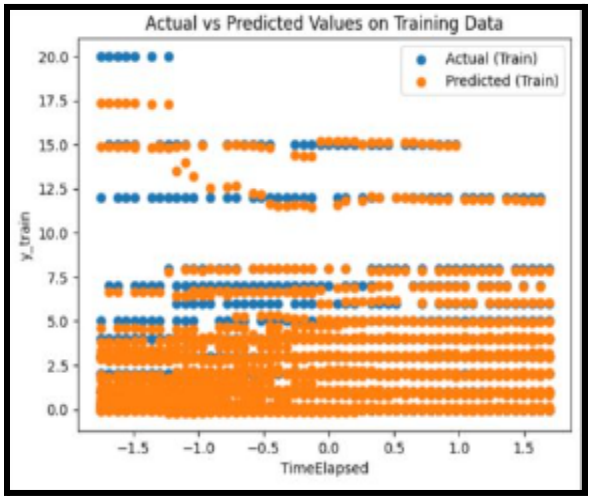
One hidden layer makes up the feedforward neural network type known as Extreme Learning Machine. ELM randomly initializes and fixes the weights linking the input and hidden layer neurons, in contrast to typical neural networks that change these weights repeatedly throughout training. Training just the weights that connect the hidden layer to the output layer is the main goal of the ELM learning process. MSE Train and MSE Test values are 1.856 and 1.855.

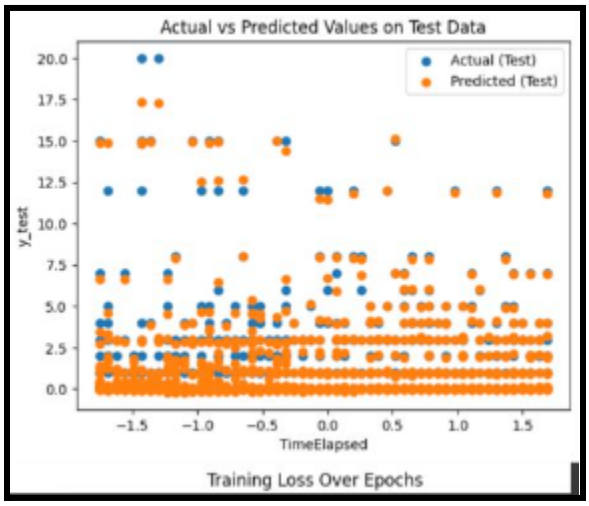


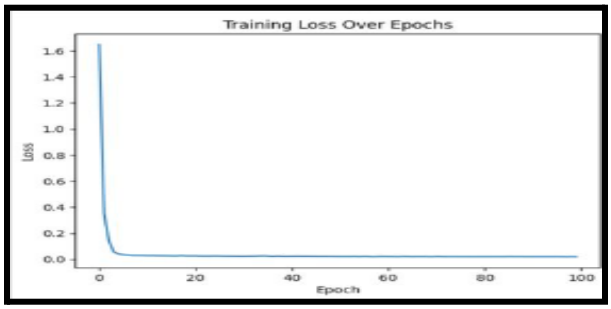

**8.3: Basic Deep learning model with 2 layers:**

Artificial Neural Network (ANN) with two or more hidden layers is known as a **Deep Neural Network**. The process of training deep neural networks is called ***deep learning***. The term **"deep"** in deep learning refers to the number of hidden layers (also called *depth*) of a neural network. MSE train and mse test values are 4.0197 and 4.156



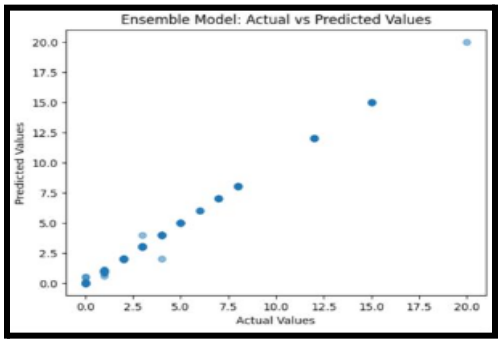



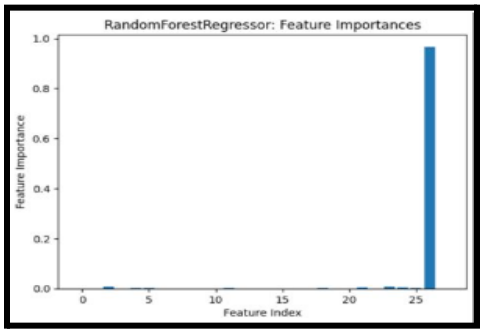

We have also plotted training loss over epochs graph.

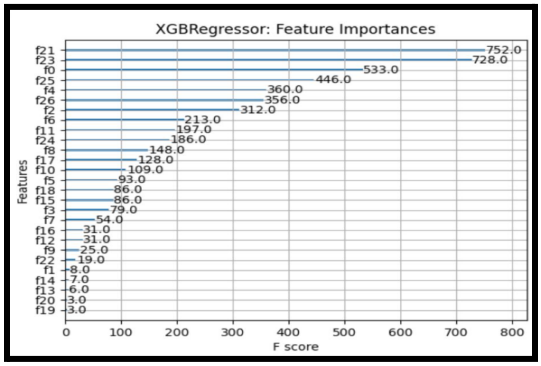



**8.4. ENSEMBLE:** This machine learning method combines several models to increase the predictability and accuracy of the results. We have used the models of RandomForest, DecisionTree and XGBRegressor and combined them. We have plotted MSE values for the

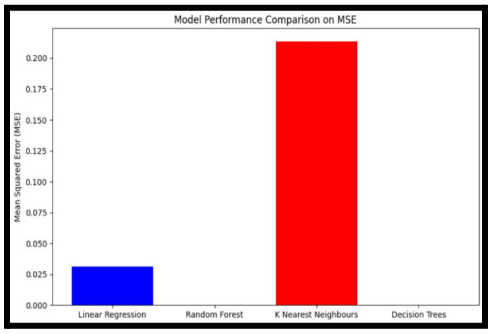
ensemble model. MSE value for the ensemble model is 0.00155. We have plotted actual values vs predicted values. Also, plotted feature importances for Randomforest and XGBRegressor.

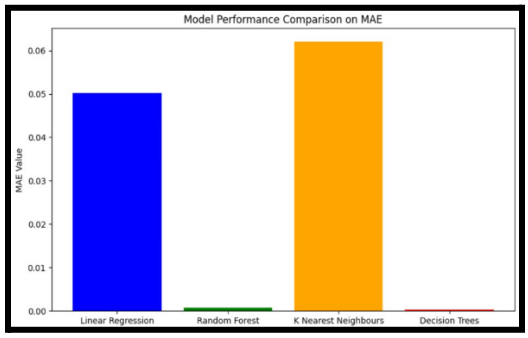


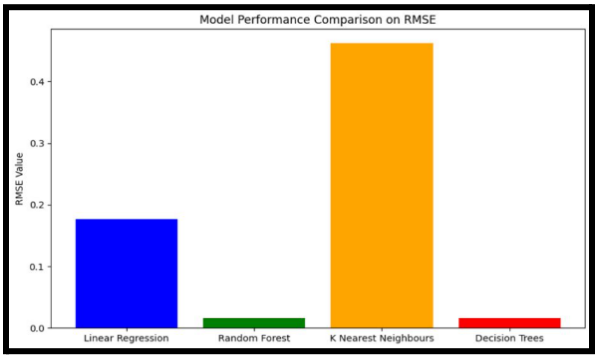




**9. Visualizations:** Visualizing real-time Portugal 2019 election results through Mean Squared Error (MSE), Mean Relative Error (MRE), and Root Mean Squared Error (RMSE) plots offers dynamic insights into the evolving accuracy and precision of the regression analysis, capturing performance variations throughout different stages of the election data analysis.



**10. Conclusion:**

In this project, we initiated the data analysis process by employing various preprocessing techniques to clean the dataset. Subsequently, we constructed multiple regression models on the refined dataset, optimizing their performance by identifying optimal hyperparameters through cross-validation. Variable selection was carried out using Lasso regression(because we need to select important variables from a large variable set and Bidirectional Elimination on the most promising model. Each step in the process, including Regression Models, Hyperparameter Tuning, First Variable Selection, and Bidirectional Elimination as a wrapper method, was systematically visualized utilizing Mean Squared Error(MSE).

Furthermore, we conducted a thorough examination of MSE, MAE, RMSE to gain insights into model performance. An ensemble model was developed, incorporating the top 3 models which include Random Forest, Decision Tree and XGBoost. Among all these models, the Random Forest model emerged as the most effective, boasting an MSE value of 0.000257. Notably, the random Forest model with hyperparameter tuning exhibited a marginal improvement, achieving an MSE of 0.000256. In conclusion, based on our comprehensive approach involving various preprocessing techniques, feature selection, and ensemble modeling, Random Forest with hyper parameters(‘max_depth’ : 10, ‘ min_samples_split’: 2, ‘n_estimators’: 100) stands out as the optimal model for predicting the target variable with given dataset, demonstrating superior accuracy.

**11. References:**


- https://scikit-learn.org/stable/modules/linear_model.html#generalized-linear-models
- Datasetlink: https://archive.ics.uci.edu/dataset/513/real+time+election+results+portugal+2019=
- C. Titus Brown and Harry W. Bullen and Sean P. Kelly and Robert K. Xiao and Steven G. Satterfield and John G. Hagedorn and Judith E. Devaney. Visualization and Data Mining in a 3D Immersive Environment.
- Enas, G.G. & Choi, S.C. (1986) "Choice the smoothing parameter and efficiency of K-Nearest Neighbor classification", Comp & Maths with Apps, 12(2): 235-244.
- .G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine," in Technical Report ICIS/03/2004, (School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore), Jan. 2004.
- D. Amaratunga, J. Cabrera, and Y.S. Lee. Enriched random forests. Bioinformatics, 24:2010–2014,2008.
- Lee, John A., and Michel Verleysen. Nonlinear dimensionality reduction. Vol. 1. New York: Springer, 2007.


**12. Video link:**

https://uofh-my.sharepoint.com/:v:/g/personal/pkakarla_cougarnet_uh_edu/EX2kLtDWil9Di8NnxZIucJcB8zhjOhEYb9lUQLPcmZfGDA?referrer=Teams.TEAMS-ELECTRON&referrerScenario=MeetingChicletGetLink.view.view=