UCS2612 - Machine Learning Lab
Mini Project

Project Title : Predicting the Candidates vote in Indian general Election

Team Members

Anand K 3122 21 5001 0

Tejaswi Kakarla 3122 21 5001 0

Karthikeyan A 3122 21 5001 041

Mega V 3122 21 5001 051

Class: CSE - A

Problem Statement

Now a days opinion polls often employ random sampling techniques to gather data from a representative sample of the population. In the context of telephone-based polling, researchers choose only limited number of phone numbers randomly . This ensures that every phone number in the target population has an equal chance of being selected for the survey. Once a phone number is dialed, interviewers conduct the survey by asking questions about voter preferences, opinions on political issues, and other relevant topics . But that count is nearly less than one percent in our population. But this itself is a very complicated process for our massive population. Reduce the man power and improve the accuracy of opinion poll results we are trying to build a machine learning models.

Domain

- Domain : Indian Politics - Indian General Election
- Data Type : Text Data
- Model Type : Regression

Objectives

Efficiency and Scalability

One objective is to streamline the polling process by reducing the reliance on manual labor and human interviewers.

Cost Reduction

By reducing the need for a large number of human interviewers, the project aims to lower the overall cost of conducting opinion polls.

Improved Accuracy and Representativeness

The project seeks to enhance the accuracy and representativeness of opinion poll results by leveraging advanced statistical techniques and machine learning algorithms.

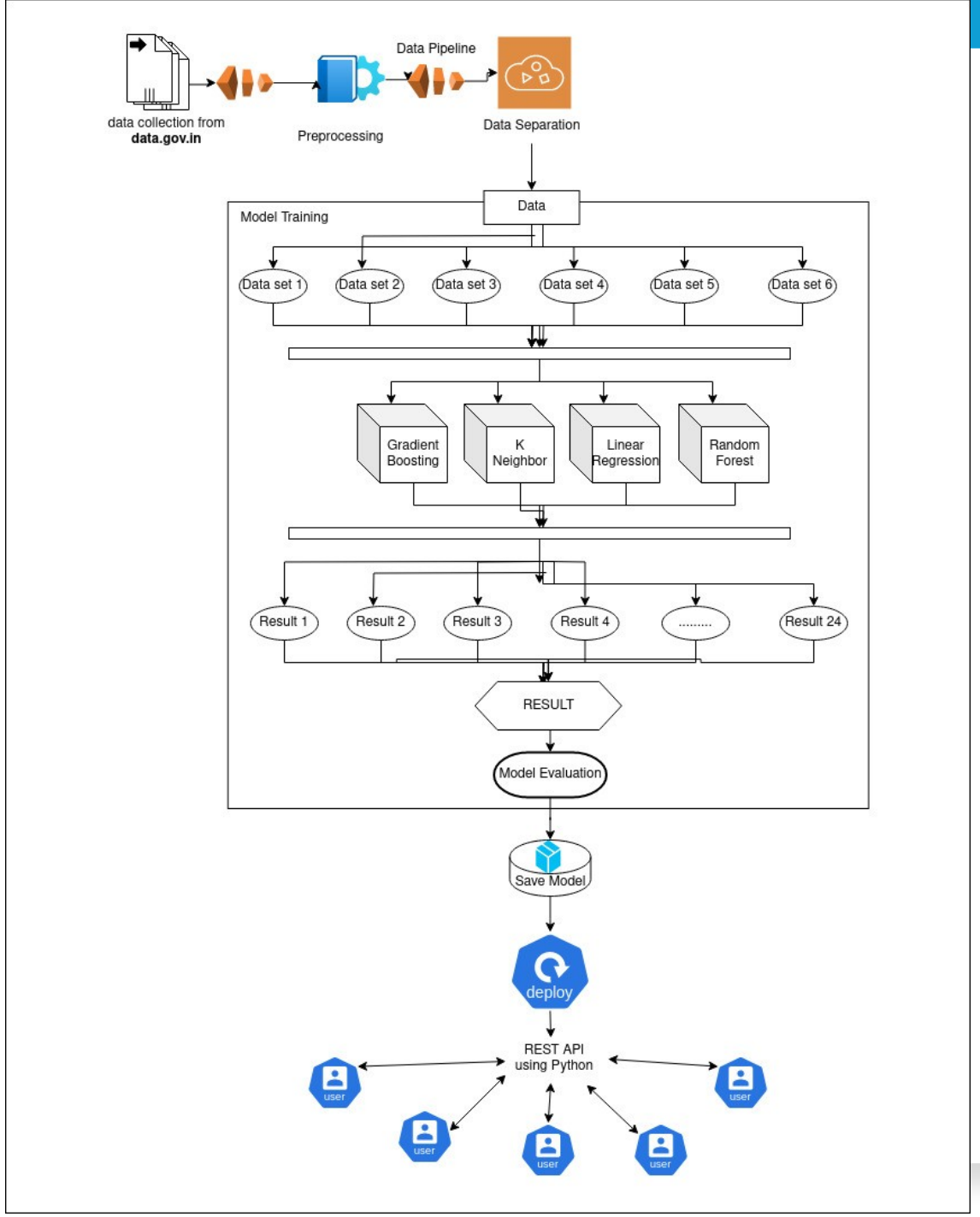
Real-time Analysis and Insight

Machine learning models can enable real-time analysis of polling data, allowing for quicker turnaround times and more timely insights into shifting voter sentiments and trends.

Development Environment

IDE / Editor	Jupyter Notebook, VS Code
Programming Language	Python
Libraries/Frameworks	scikit-learn
Data Exploration Tools	Pandas, Matplotlib , Numpy
Model Evaluation Tools	scikit-learn metrics
Deployment Tool	Rapid Minor

System Architecture



Steps In Building Model

1) Loading dataset

Using Pandas

2) Exploratory Data Analysis

Details Of Dataset

Display The Each Parliament Constitution victory Person in Each election

Display Election Results in Each Year

Display Correlation between each attributes using heatmap

Display Correlation Graph

3) Data Preprocessing

Handling Missing Values

Label encoder

Normalization

4) Feature Engineering

Select K best K=3

Select K best K=4

Select K best K=5

Select K best K=6

PCA

LDA

Steps In Building Model

4) Model Building

Random Forest Regression

Linear Regression

K Neighbour Regression

Gradient Boosting Regression

5) Model Training and testing

6) Model Deployment in Rapid Minor

Dataset Upload

Model Select

Processing Models

Obtain results

Comparison of Four Models

Results For Four Models

MODEL	FEATURE ENGINEERING	MEAN SQUARED ERROR	R SQUARED SCORE	ACCURACY
Gradient Boosting Regression Model	K = 3	4689954366.56	0.572450	57.25 %
	K = 4	4691351252.52	0.572323	57.23 %
	K = 5	4651621664.73	0.5759450	57.59 %
	K = 6	4651487603.76	0.575957	57.6 %
	PCA	4468413418.89	0.592646	59.26 %
	LDA	8283375210.57	0.244864	24.49 %
K Neighbor Regression Model	K = 3	4358090945.67	0.602704	60.27 %
	K = 4	4620176737.24	0.578811	57.88 %
	K = 5	4353191882.72	0.603150	60.32 %
	K = 6	4353191882.72	0.603150	60.32 %
	PCA	2828647999.62	0.742132	74.21 %
	LDA	6685534694.59	0.390527	39.05 %
Linear Regression Model	K = 3	10869082765.09	0.0091438	0.91 %
	K = 4	10857343649.72	0.010213	1.02 %
	K = 5	10857354163.93	0.010213	1.02 %
	K = 6	10857354163.93	0.010213	1.02 %
	PCA	10857162242.02	0.010230	1.02 %
	LDA	10857162242.02	0.010230	1.02 %
Random Forest Regression Model	K = 3	3751661185.88	0.657988	65.8 %
	K = 4	4116258138.62	0.624750	62.48 %
	K = 5	3362184729.95	0.693493	69.35 %
	K = 6	1504170913.24	0.862875	86.29 %
	PCA	2055588563.91	0.812606	81.26 %
	LDA	3516979397.51	0.679382	67.94 %

Based On Accuracy

Based on accuracy,

- **Random Forest Regression model with K = 6** achieves the highest accuracy of **86.29%,** followed by the **PCA- based Random Forest model** with an accuracy of **81.26%.**
- The Random Forest regression models with different data provides the higher Accuracy compare than the other three models.
- The K Neighbor Regression model comparatively provides the higher accuracy than Linear Regression models and Gradient Boosting Regression models
- Linear Regression models, along with Gradient Boosting Regression models, generally perform poorly compared to Random Forest and K Nearest Neighbors models.
- Best Models For The dataset based on Accuracy,

    1) Random Forest Regression
    2) K Neighbor Regression Model

Comparison of Four Models

Computational Time For Models

Model	Computational Time
Random Forest Regression	26.039166927337646 seconds
Linear Regression	0.10006356239318848 seconds
K neighbor Regression	0.15254878997802734 seconds

Based On Accuracy

- By seeing the above result the Linear regression model takes the least computational time but provides the very low accuracy which is less than 1 %.So this is worst model for the our prediction
- The K Neighbor regression takes the second least computational time among four and also provide the best accuracy during the testing data . K Neighbor regression is the best model for the prediction
- The Gradient Boosting Regression takes more than 10 sec computational time and also provide the average accuracy during the testing data . So Gradient Boosting regression is the worst model for the prediction.
- The Random Forest Regression takes more than 20 seconds computational time which is the highest among four and also provide the best accuracy during the testing data . So Random Forest Regression is the best model for the prediction based on the Accuracy not for the computational time.
- Best Models For The dataset based on Computational Time ,

1) K Neighbor Regression Model
2) Random Forest Regression

Based On Fitting

In Random Forest Regression

while increasing the dataset size the validation accuracy goes nearly to the training accuracy. So the Random Forest regression fits a Good fit . So Linear Regression is a Good fit model for this dataset

In Linear Regression

while increasing the dataset size the training accuracy and validation accuracy both goes nearly to 0. So the linear regression fits underfitting . So Linear Regression is a underfit model for this dataset

In K neighbor Regression

There is a huge difference between the training accuracy and validation accuracy. So the K neighbor fits a over fit . So K neighbor is a over fit model for this dataset

In Gradient Boosting Regression

while increasing the dataset size the validation accuracy goes nearly to the training accuracy. So the Gradient Boosting regression fits a Good fit. So Gradient Boosting regression is a Good fit model for this dataset

Comparison of Four Models

Fitting Results

Model	Fitting
Random Forest Regression	Good fit
Linear Regression	Under Fit
K neighbor Regression	Over fit
Gradient Boosting Regression	Good fit

Comparison of Four Models

Comparison Results

Model	Accuracy	Computational Time	Fitting
Random Forest Regression	High Accuracy	High	Good fit
Linear Regression	Worst Accuracy	Low	Under Fit
K Neighbor Regression	High Accuracy	Low	Over fit
Gradient Boosting Regression	Worst Accuracy	High	Good fit

Inferences

Random Forest Regression

Achieves high accuracy.

Requires a relatively high computational time.

Provides a good fit to the data.

Inference: Random Forest Regression is suitable for tasks where accuracy is crucial and computational resources are available.

Linear Regression

Yields the worst accuracy.

Requires low computational time.

Tends to underfit the data.

Inference: Linear Regression is efficient but may not capture the complexity of the data well, making it suitable for simpler problems with fewer features

Inferences

K Nearest Neighbors Regression

Achieves high accuracy.

Requires low computational time.

Tends to overfit the data.

Inference: K Nearest Neighbors Regression is efficient and effective for smaller datasets but may not generalize well to unseen data due to overfitting.

Gradient Boosting Regression

Yields the worst accuracy.

Requires a relatively high computational time.

Provides a good fit to the data.

Inference: Gradient Boosting Regression provides a good fit but may require more computational resources and tuning compared to other models.

Impact of Project on Society

Political parties and candidates could use the predictions to optimize their campaign strategies.

Resource Allocation: Parties could use the predictions to strategically allocate campaign resources such as funding, manpower, and time.

Voter Targeting: ML models could help parties identify potential swing voters or undecided voters more accurately.

ML-based predictions could challenge traditional political strategies and the role of intuition and experience in decision-making. Parties and candidates may rely more on data-driven approaches, potentially leading to a shift in the political landscape.

Voters may benefit from a better understanding of electoral dynamics and candidate performance.

Predictive models could stimulate interest and engagement in the political process.

There's a risk that predictive models could be used to manipulate or influence voters.

Conclusion

In Conclusion,

The choice of the best model depends on the specific requirements of the problem, such as the importance of accuracy, computational resources available, and the trade-off between overfitting and underfitting. Random Forest Regression and K Nearest Neighbors Regression are suitable for tasks where accuracy is crucial and computational resources are limited, while Linear Regression may be preferred for simpler problems with low computational requirements. Gradient Boosting Regression can be effective but may require more computational resources and tuning to achieve optimal performance.

By considering the results The best Model for predicting the Voters in Indian general election is **Random Forest Regression**

Future Work

In the current election voting prediction models, only the previous year's election results are considered. This approach relies solely on historical data to make predictions about future elections. However, it doesn't take into account the current situation or any new factors that may influence voter behavior, such as prevailing emotions or emerging issues.

In future work, it is proposed to include these additional factors, such as emotions and sentiments (like “pity" votes in the election), to enhance the accuracy and relevance of the prediction models. Here's a brief explanation of how this inclusion could improve the models:

Emotions and Sentiments

By incorporating emotions and sentiments prevalent among voters during the current election cycle, the models can better capture the mood of the electorate. Analyzing sentiment from social media, news articles, or surveys can provide valuable insights into the prevailing mood and sentiment of voters.

Future Work

Improved Predictive Power

Incorporating current situation factors enhances the predictive power of the models by providing a more comprehensive understanding of voter behavior. By considering both historical trends and present-day dynamics, the models can better anticipate shifts in voter sentiment and behavior, leading to more accurate predictions of election outcomes.

Overall, By integrating emotions, sentiments, and other current situation factors into election voting prediction models, we can create more robust and insightful tools for understanding and forecasting electoral dynamics. This approach enables us to capture the complexities of voter behavior more accurately and adaptively, thereby enhancing the effectiveness of election prediction efforts.

Learning Outcome

- Better understanding about the various machine learning regression model
- We learnt about the strengths, weaknesses, and suitability of ML model
- The Machine learning models are evaluated by several evaluation metrics
- We learnt about the importance of feature engineering in improving the machine model performance
- Better understanding about how to select and preprocess features to enhance the predictive power of machine learning models.
- learnt how to compare different machine learning models based on performance metrics and make inferences about their suitability for specific tasks
- Gain knowledge about the trade-offs between model accuracy, computational efficiency, and fitting.
- We understand that the additional information may increase the performance of the machine learning models.
- We identifying limitations in existing models and proposing future work to address them

References

https://en.wikipedia.org/wiki/Linear_regression

https://www.geeksforgeeks.org/random-forest-regression-in-python/

https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsRegressor.html

https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsRegressor.html

https://lms.ssn.edu.in/course/view.php?id=2231