Title: Predictive Analysis of Delhi Election Results (2025)

Approach :

Objective

The objective of this analysis is to analyze the trends in the Delhi elections and predict the performance of the three major parties (BJP, AAP, and INC) in the upcoming 2025 elections. The analysis aims to identify key factors influencing election outcomes and build a predictive model to forecast the winning party in each assembly constituency.

Approach
The primary objective of this analysis is to understand the voting trends in Delhi elections over the years and to build a predictive model to determine the winning party based on various features. The analysis involves data cleaning, exploratory data analysis (EDA), feature engineering, and model building.

Methodology
1. Data Loading and Cleaning
   - Imported necessary libraries and loaded the datasets.
   - Cleaned the data by handling missing values, converting data types, and renaming columns for consistency.

2. Exploratory Data Analysis (EDA):
   - Visualized trends using bar charts, line charts, pie charts, heatmaps, and scatter plots.
   - Analyzed the distribution of votes among different parties across various assemblies and years.

3. Feature Engineering
   - Created new features such as `winner_party` to identify the winning party in each assembly.
   - Normalized numerical columns and encoded categorical variables.

4. Correlation Analysis
   - Calculated the correlation matrix to identify highly correlated features.
   - Removed features with high multicollinearity to improve model performance.

5. Model Building
   - Split the data into training, validation, and test sets.

- Trained multiple models (Logistic Regression, Decision Tree, Random Forest, AdaBoost, LGBM, Naive Bayes) using GridSearchCV for hyperparameter tuning.
- Evaluated models based on validation accuracy and selected the best-performing model.

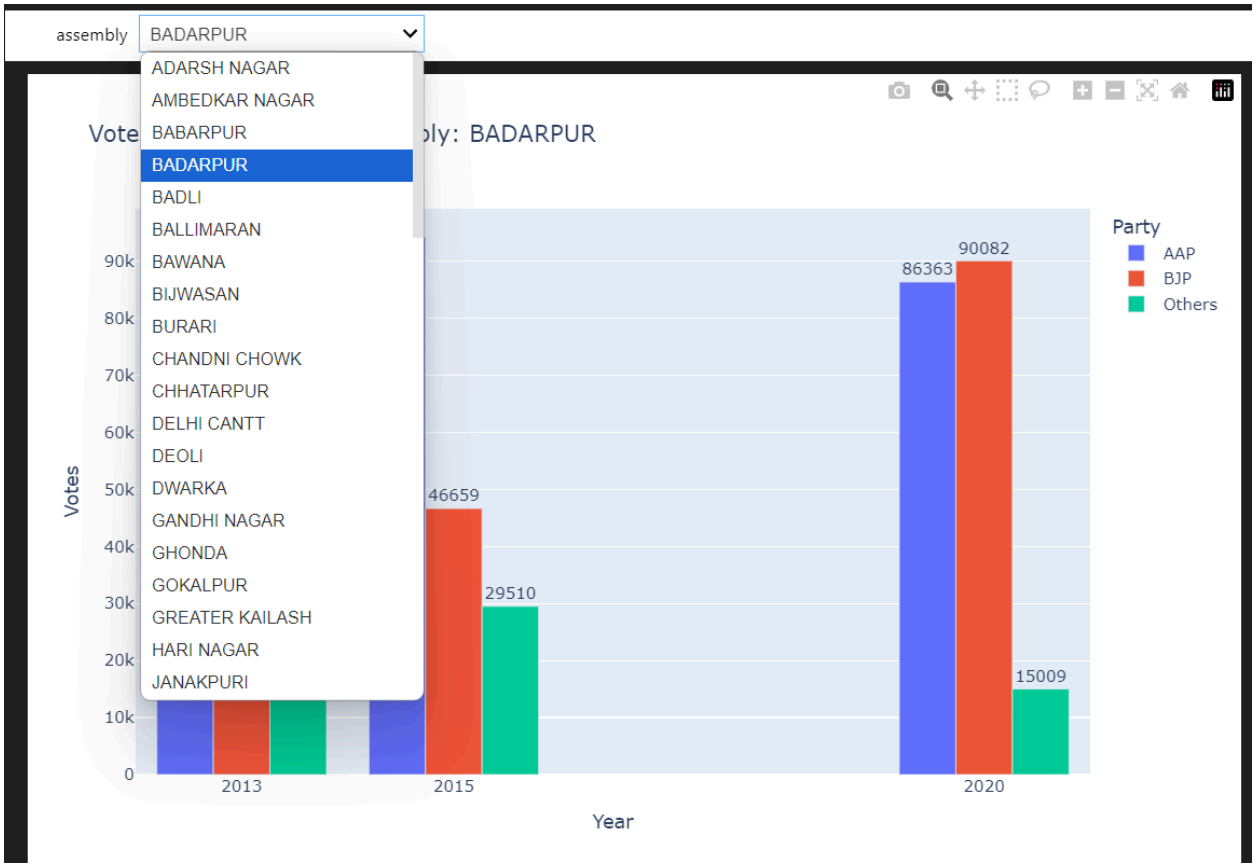
Model
The best model selected was the LGBMClassifier with the following hyperparameters:
- `learning_rate`: 0.01
- `n_estimators`: 200

The model pipeline included:
- Imputation of missing values using `SimpleImputer`.
- Classification using `LGBMClassifier`.

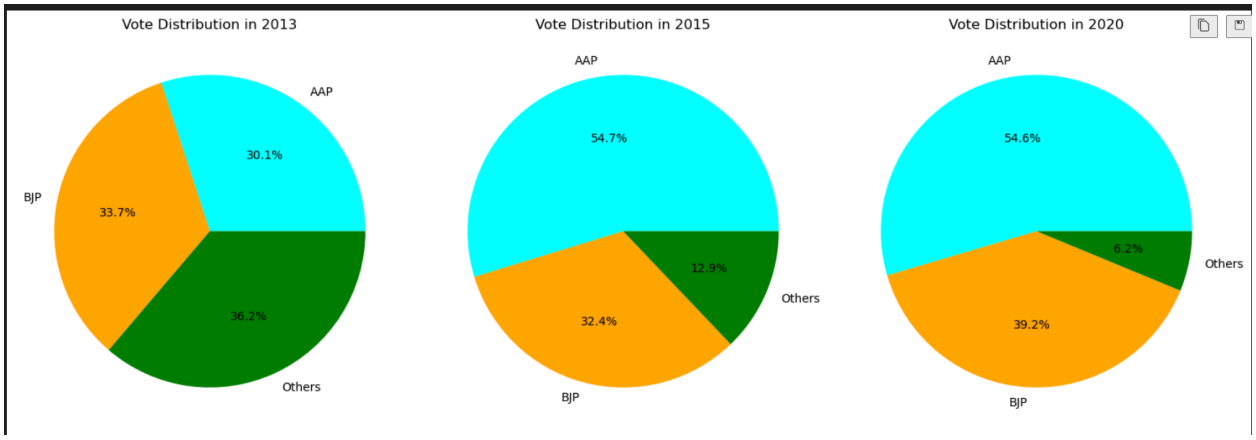
Outcome
- The best model achieved a validation accuracy of 83.33% and a test accuracy of 73.81%.
- The model was saved for future predictions.
- The analysis provided insights into the voting patterns and the factors influencing election outcomes in Delhi.

The final model and encoders were saved using `joblib` for future use, and the results were visualized using various plots to understand the trends and patterns in the election data.

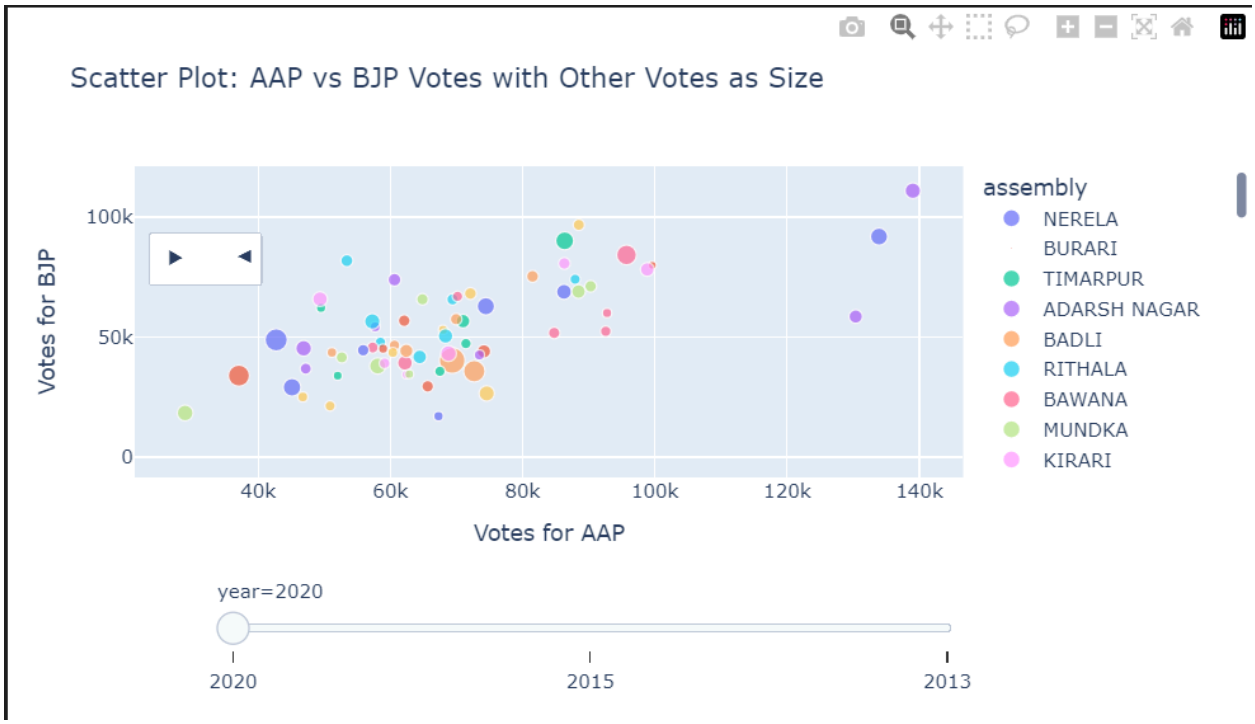
Some of the useful plottings,



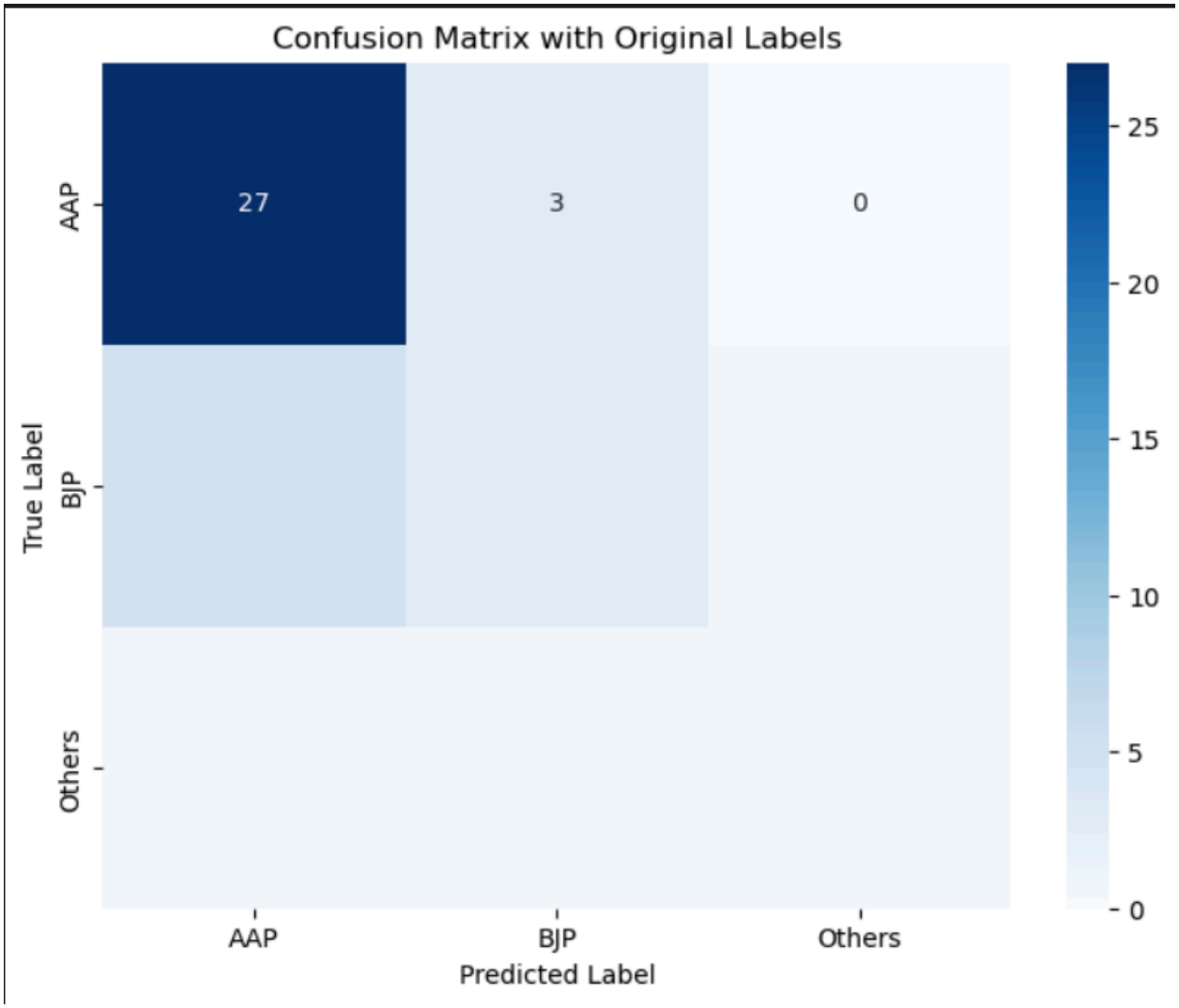
Interactive plot to observe the vote trends YoY for every Assembly. Gives a brief idea about polarization across Assembly.



Vote Distribution Across Parties



AAP and BJP Votes Share



Prediction of Future Winner

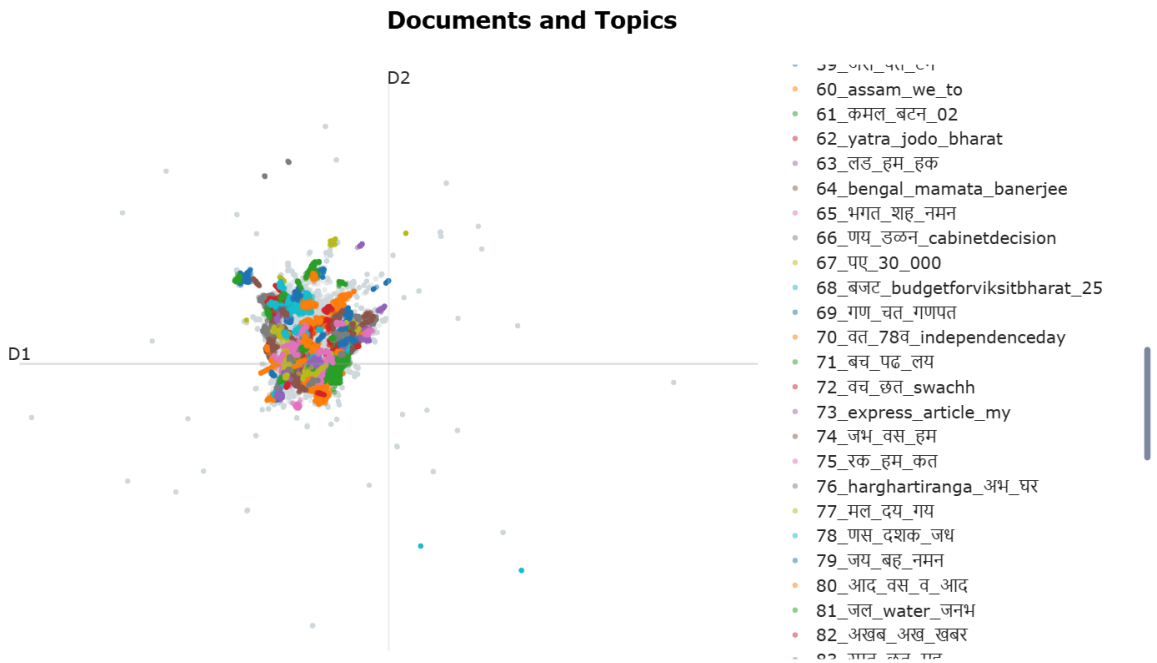
Title 2 : Predicting Political Leaning Using Machine Learning: Analysis of Tweets from BJP and INC

Objective: Build an ML model to classify political leaning (BJP or INC) based on tweets.

Dataset: Utilized a dataset of 40,000 tweets from BJP and INC politicians, sourced from Kaggle.

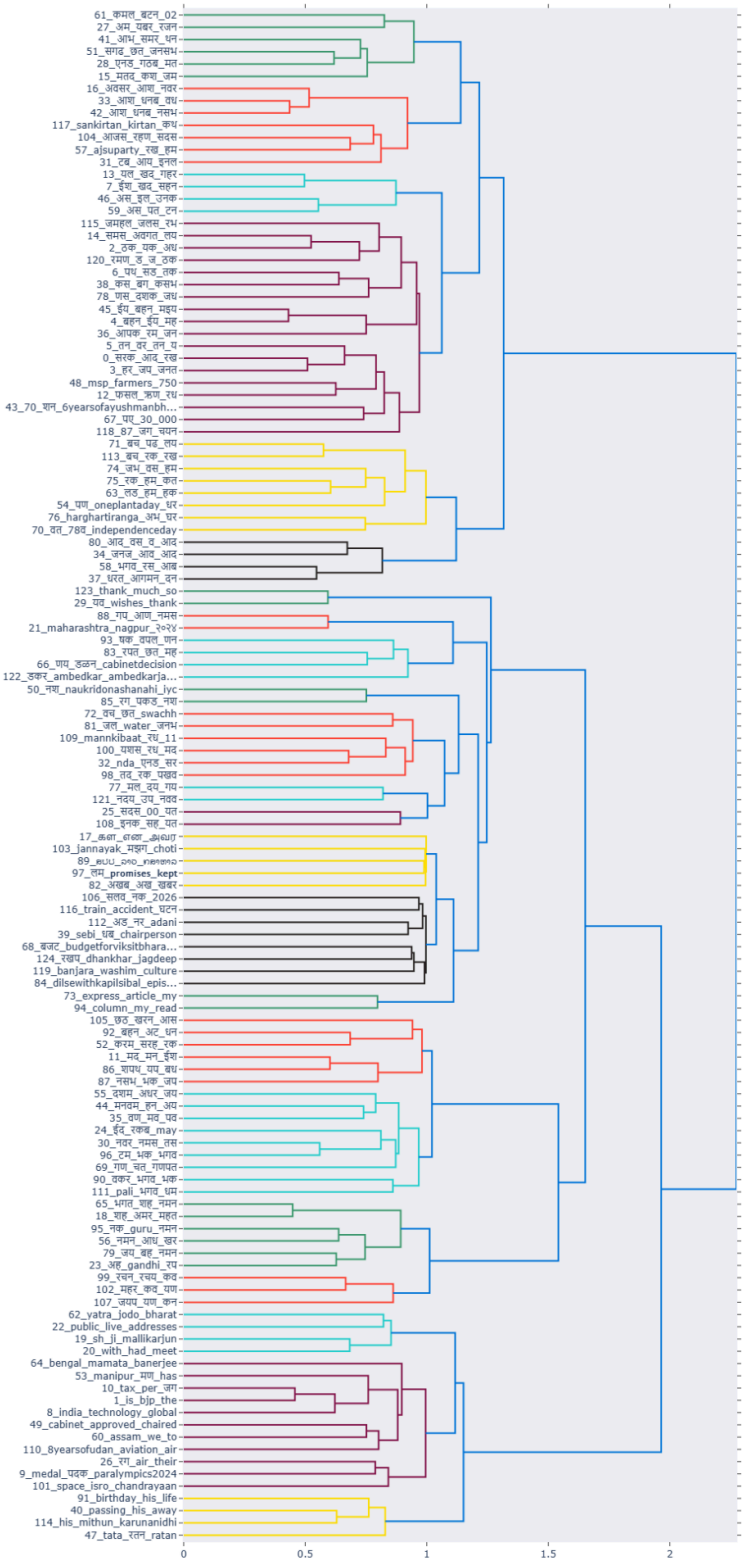
Data Preprocessing:
- Imported necessary libraries: `kagglehub`, *sentence transformers*, *pandas*, *UMAP*, *HDBSCAN*, *BERTopic*, etc.
- Downloaded the dataset using `kagglehub.dataset_download()`.
- Loaded the data from a text file into a pandas DataFrame.
- **Handled missing values** (NaN) in the 'Tweet' column by removing rows with null tweets, reducing the dataset size.

Modeling and Topic Modeling:
- Employed a pre-trained Hindi sentence similarity *SentenceTransformer* model (`*l3cube-pune/hindi-sentence-similarity-sbert*`) to generate embeddings for the tweets.
- Performed dimensionality reduction on the embeddings using ***UMAP***, reducing from 384 to 5 dimensions.
- Clustered the reduced embeddings using ***HDBSCAN***.
- Built a ***BERTopic*** model using the generated embeddings, ***UMAP*** model, and ***HDBSCAN*** model to perform topic modeling on the tweets.
- Visualized topics and documents using BERTopic's visualization methods (*visualize_documents*, *visualize_barchart*, *visualize_heatmap*, *visualize_hierarchy*).
- Updated the topic representations using KeyBERTInspired and then the ***Flan-T5*** model for more descriptive topic labels, comparing the differences before and after updates.



Topic Modelling

Hierarchical Clustering



5. Political Leaning Classification:

Leveraged a **Flan-T5 text generation pipeline** to classify the political leaning of tweets.

- Defined a prompt template instructing the model to classify tweets into one of four categories: 'Left-leaning,' 'Right-leaning,' 'Centrist,' or 'Unclear.'
- Implemented a classification function and applied it to the dataset, storing predictions in the `political_leaning` column.
- Optimized the model's output by fine-tuning parameters like `max_length` and `num_return_sequences` to ensure concise and accurate classifications.

Employed the **Phi-3-mini-4k-instruct encoder-decoder model** for an alternative classification of political leaning.

- Utilized a similarly structured prompt to maintain consistency in the classification approach across models.
- Applied the model's predictions to the dataset, storing results in the `predicted_political_leaning` column.
- Verified outputs by ensuring minimal token truncation and robust decoding methods for accurate text interpretation.

Conducted comparative evaluation between the two models:

- Assessed classification accuracy using metrics such as confusion matrix, precision, recall, and F1-score.
- Identified alignment and discrepancies in predictions, analyzing cases where models disagreed to understand the nuances in their outputs.

6. Outcome:

- Produced a DataFrame enriched with a `political_leaning` column, capturing the predicted political leaning ('Left-leaning,' 'Right-leaning,' 'Centrist,' or 'Unclear') for each tweet.
- Trained a topic model on the tweets to uncover prevalent themes and discussions within the dataset.
- Enhanced topic model insights by updating keywords using **KeyBERT** and **Flan-T5 models**, ensuring accurate and contextually relevant topic representations.

- Visualized the topic model results to provide a qualitative understanding of the key topics, aiding in the exploration of political discourse trends within the dataset.

7. Recommendations: The provided code does not include the evaluation metrics (Accuracy, Precision, Recall, F1-score) as the prompt requests. A more thorough analysis would require evaluation of the classification task using standard metrics. Further, the dataset bias and improvements for multilingual data analysis would be the next steps.

GitHub Link - *https://github.com/SaibalPatraDS/Delhi-Election-Trend-Analysis*