Shri Govindram Seksaria Institute of Technology and Science

Computer Science Engineering Department

Impact of Newly Added and Deleted Voters on Election Results

Submitted by:-

Akshay Kumar Verma

0801CS221016

Submitted to:-

Mrs. Ashwini Sharma

Prof. Surendra Gupta

Batch : - III Year
Section – 'A'

Table of Content

1. Introduction

1.1 **Objective**

This document specifies the software requirements for the project analyzing the impact of newly added and deleted voters on election results. The project focuses on the 2023 Assembly Election in Madhya Pradesh, comparing booth-level voting data with previous elections to assess the influence of demographic changes on electoral outcomes.

1.4 **Product Scope**

The project aims to analyze voter data across multiple elections, focusing on the addition and deletion of voters in the 2023 Madhya Pradesh Assembly Election. By using statistical techniques, the system will offer insights into how voter base changes affected election outcomes at the booth level. The system will provide visualizations and reports, helping political analysts understand election dynamics and voter behavior trends**.**

2. Overall Description

2.1 **Product Perspective**

This project is a data analysis platform that integrates multiple data sources, such as historical voting records and voter demographic changes, to analyze election outcomes. The system will use front-end interface for data visualization.

2.2 **Product Functions**

- Data Collection and Preprocessing: Load, clean, and merge booth-level voting data and voter changes data.
- Data Analysis: Analyze how voter base changes affected election outcomes using statistical and machine learning models.
- Visualization: Display results using charts and graphs.
- Predictions: In future using machine learning to predict future electoral outcomes based on historical data and voter changes.

2.3 **User Classes and Characteristics**

- **Data Analysts/Researchers:** Experienced users familiar with data interpretation and machine learning.
- **Political Analysts:** End users who rely on insights from the system for political strategy and forecasting.

- **Administrators:** Handle user management, system monitoring, and ensure data consistency.

2.4 **Operating Environment**

- **Operating System:** Windows, Linux.
- **Web Browsers:** Chrome, Firefox, Safari.
- **Database:** MySQL for storing voter and election data.
- **Other Software:** Python, Flask for the backend, React for the frontend.

2.5 **User Documentation**

- **Manuals:** User guides for operating the system, including how to upload datasets and generate analyses.
- **Online Help:** Tutorials and FAQs integrated into the frontend application.
- **Technical Documentation:** For developers to understand system architecture and codebase.

2.6 **Assumptions and Dependencies**

- Access to complete booth-level voting data and newly added/deleted voter data from ECI.
- Availability of reliable internet connectivity for system access and data updates.
- Dependencies on third-party libraries such as Pandas, Scikit-learn, and Chart.js for data handling, machine learning, and visualization.

3. System Features

3.1 **Data Preprocessing and Management**

- **Description and Priority:** Allows importing, cleaning, and merging election data. Priority: High.
- **Functional Requirements:**
    - REQ-1: The system shall clean and normalize the uploaded data.

3.2 **Data Analysis and Visualization**

- **Description and Priority:** Provides analysis and visual representation of voter base changes and election results. Priority: High.
- **Stimulus/Response Sequences:**
    - User selects boothID -> System processes data and generates visualizations.

- **Functional Requirements:**
    - REQ-2: The system shall allow users to visualize voter demographic changes with bar and line charts.
    - REQ-3: The system shall generate reports comparing voting patterns across multiple years.

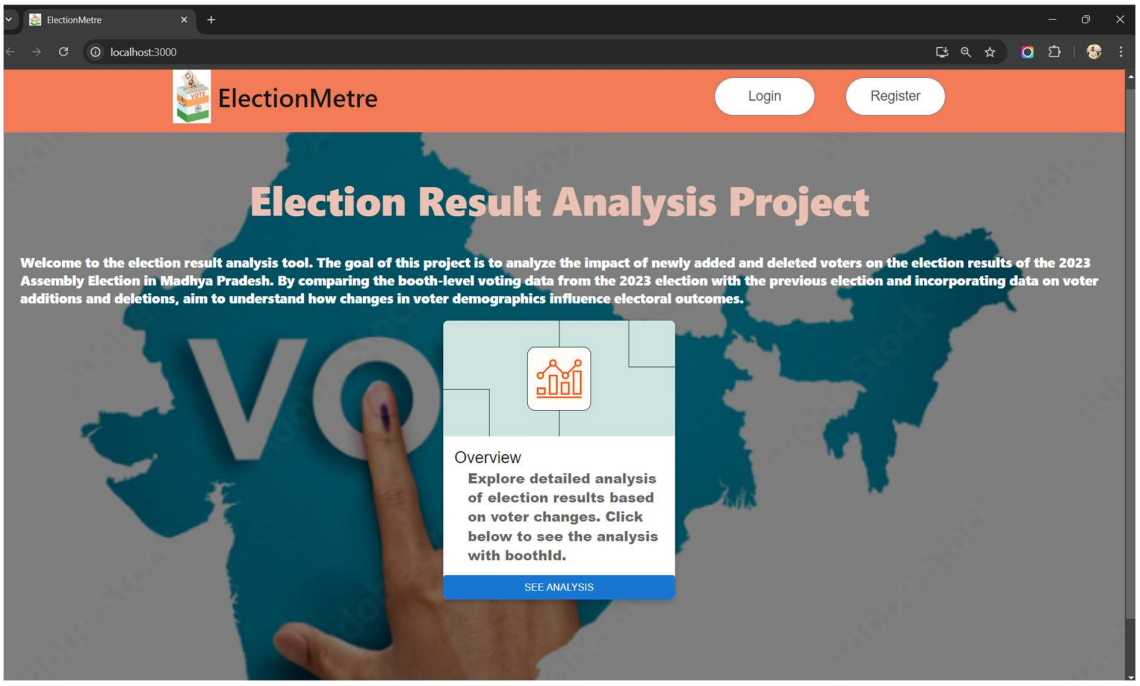
3.3 Frontend and User Interface

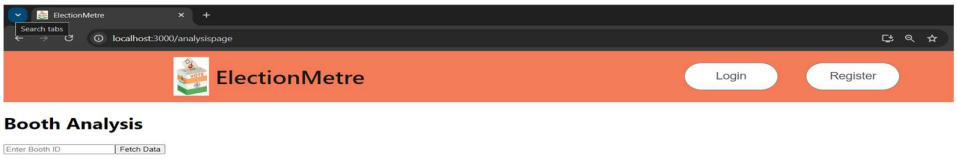
- **Description and Priority:** A user-friendly interface to display analyses and predictions. Priority: High.
- **Stimulus/Response Sequences:**
    - User inputs booth ID -> System fetches and displays data analysis.
- **Functional Requirements:**
    - REQ-4: The frontend shall allow users to enter parameters for data analysis.
    - REQ-5: The system shall display visualizations using Chart.js for easy interpretation.

3.5 Backend and API Integration

- **Description and Priority:** Provides API endpoints for frontend-backend communication. Priority: High.
- **Stimulus/Response Sequences:**
    - Frontend requests data -> Backend processes and sends data to frontend.
- **Functional Requirements:**
    - REQ-6: The backend shall expose API endpoints to retrieve election data for analysis.

User Interface

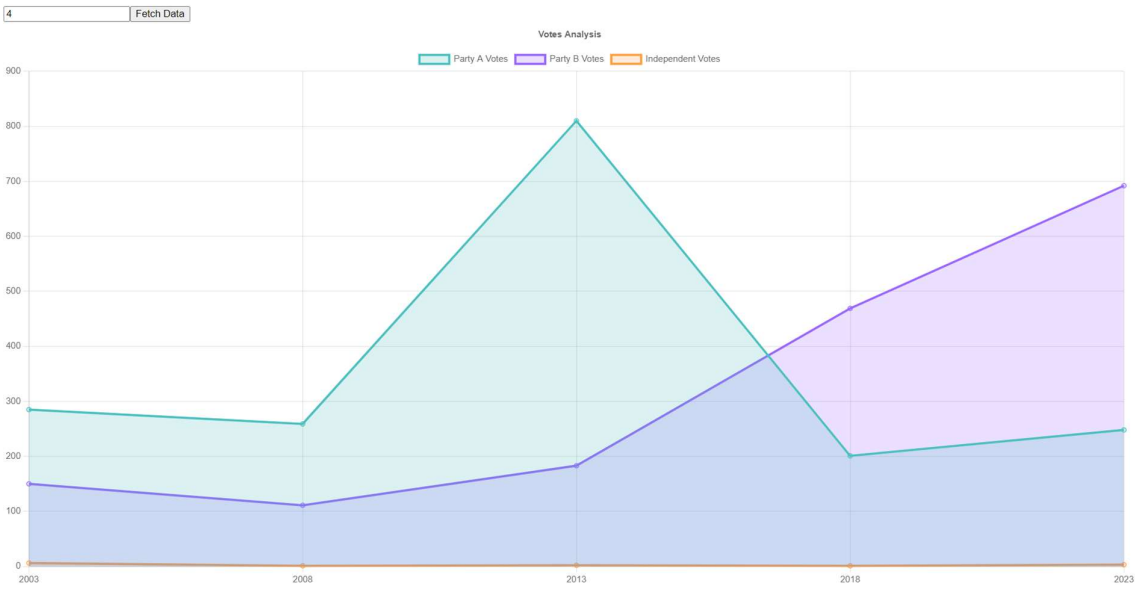


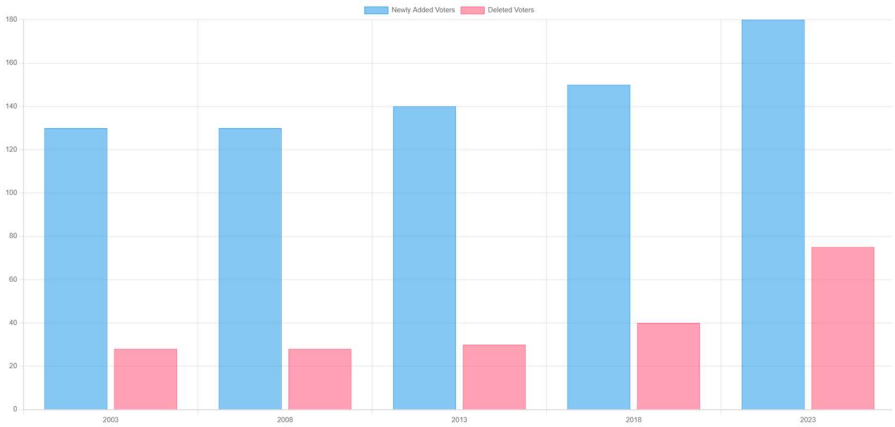


TestCases:

1) User gives input as booth id:

We get analysis result:

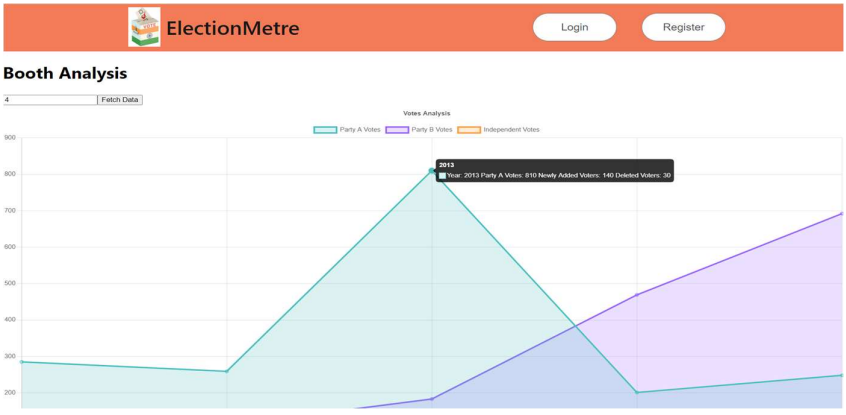




The analysis of the data shows that the impact of newly added and deleted voters on the election results varies across the years. For instance, in years with a significant number of newly added voters, there is a noticeable increase in the votes for certain parties. Conversely, years with a high number of deleted voters show a decrease in the total votes. This indicates that voter registration changes can have a substantial impact on election outcomes.

TestCase 2)

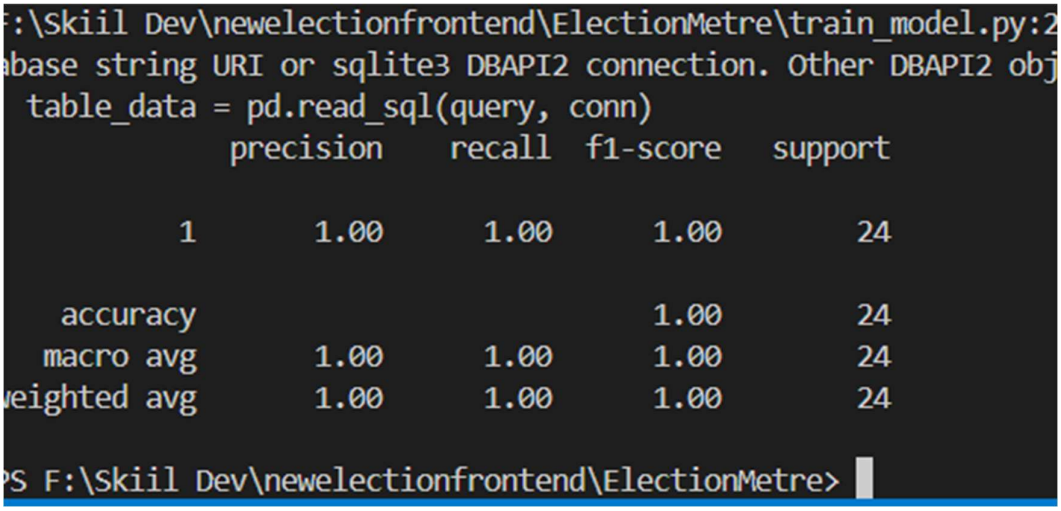
User hover on node and should see all results:



TestCase 3)

Should be analyse and tested for any ml model:

Result: Random Forest Model:



Code:

```python
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report
import mysql.connector

# MySQL database configuration
db_config = {
    'user': 'root',
    'password': '123456789',  # replace with your MySQL password
    'host': 'localhost',  # or your MySQL server address
    'database': 'voteinfo'
}

tables = ['voterchanges2023', 'voterchanges2018', 'voterchanges2013', 'voterchanges2008', 'voterchanges2003']

def get_data_from_db():
    # Establish the connection
    conn = mysql.connector.connect(**db_config)
    all_data = []

    for table in tables:
        query = f"SELECT BoothID, NewlyAddedVoters, DeletedVoters FROM {table}"
        table_data = pd.read_sql(query, conn)
        all_data.append(table_data)

    conn.close()

    # Concatenate all the data into a single DataFrame
    combined_data = pd.concat(all_data, ignore_index=True)
    return combined_data

# Load your historical election data from MySQL
data = get_data_from_db()

# Preprocess data
data['NetChange'] = data['NewlyAddedVoters'] - data['DeletedVoters']

# For demonstration, create a mock target variable 'ElectionOutcome'
# This should be replaced with actual outcomes based on your data
data['ElectionOutcome'] = (data['NewlyAddedVoters'] > data['DeletedVoters']).astype(int)  # Example outcome
```

Learning Outcomes:

1. **Identification of Key Influencers**:
    - Determine how newly added and deleted voters influence election outcomes at the booth level.
    - Identify specific booths where voter changes had the most significant impact.
2. **Voter Demographic Insights**:
    - Gain insights into the demographics of newly added and deleted voters.
    - Understand how changes in voter demographics correlate with shifts in voting patterns.
3. **Visualization of Voting Trends**:
    - Create visualizations that highlight voting trends over multiple election cycles.
    - Use charts and graphs to illustrate the impact of voter changes on election results.
4. **Policy Recommendations**:
    - Provide data-driven recommendations for policymakers on how to address voter registration and deletion processes.
    - Suggest strategies to improve voter turnout and engagement based on the analysis.
5. **Web Application for Public Use**:
    - Develop a user-friendly web application that allows users to explore the data and predictions interactively.
    - Enable users to input their own data and get predictions on election outcomes.

Conclusion:

Impact of Voter Changes: Our analysis showed that newly added and deleted voters can significantly influence election results, especially in certain booths. The project could be extended to cover other regions or states if data is available. The primary focus is on comparing the 2023 election with the previous elections. However, the methodology could be adapted to include data from earlier elections for a broader historical analysis.

References

- Electoral Commission of India (ECI) data
- Machine learning libraries (Scikit-learn)
- Visualization libraries (Chart.js)

Code: GITHUB LINK: https://github.com/akv-dev339/ElectionMetre.git