Data Science Semester Project



Submitted By:



DataSet Of Year

Election Prediction System Pakistan-2024

Project name: Date: Version:

UMTIAN’s

6-02-2024

1.0

By:



Submitted To:

|  |  |  |  |
| --- | --- | --- | --- |
| Professor | Subject | Project Name | Section |
| Sir Mazhar Javaid | Data Scientist | Election Prediction | W1 |

|  |  |  |  |
| --- | --- | --- | --- |
| Approving party | Version approved | Signature | Date |
|  | Roqia Zahra | F2020065265 |  |
|  | Myra Khan | F2020065256 |  |
|  | Umer Farooq | F2020065103 |  |
|  | Hashir Ahmad | F2020065186 |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Reviewer | Version reviewed | Signature | Date |
|  | Election\_Result\_2002 |  |  |
|  | Election\_Result\_2008 |  |  |
|  | Election\_Result\_2013 |  |  |
|  | Election\_Result\_2018 |  |  |
|  | NA\_WiseList\_2024 |  |  |

# Table of contents



1. Pakistan General Election Predicting Model using Machine Learning
   1. NumPy
   2. Pandas
   3. Matplotlib
   4. Seaborn
   5. Reading Data

2 Data Cleaning Operation on Data Frames

2.1ndardizing Column Names and Resetting Index

2.2 Adding a 'Year' Column to Each Data Frame

2.3 Concatenating Data Frames and Setting Index

2.4 Handling Null and Duplicates in the Combined DataFrame

2.5 Computing and Displaying Correlation Matrix for Numeric Data

3. Univariant Analysis

3.1 Exploring Unique Values and Counts in DataFrame Columns

3.2 Columns and Standardizing Party Names

3.3 Cleaning and Analyzing 'Turnout' Column

3.4 Identifying Outliers in the 'Votes' Column

3.5 Visualizing and Handling Zero Total Votes

3.6 Visualizing and Handling Zero Total Valid Votes

3.7 Visualizing and Handling Zero Total Rejected Votes

6. Dashboard & Habib Akram Survey Comparision

5.1 RandomForest Regressor Model Training and Evaluation

5.2 Making Predictions on New Data Using Trained Model

5.3 Calculating Sum of Predicted Votes for Specific Parties

5.4 Organizing Predictions by Constituency and Party

4.1 Removing Columns from the Election DataFrame

4.2 Setting up a Column Transformer and Pipeline for Preprocessing

3.8 Visualizing Total Registered Voters

3.9 Constructing and Visualizing Vote Percentage Feature

3.10 Identifying Outliers in the 'Vote Percent' Feature

3.11 Creating and Visualizing a Correlation Heatmap

3.12 Creating Scatter Plot and Box Plot Side by Side

3.13 Creating Multiple Scatter Plots and Box Plots Side by Side

3.14 Categorical Bar Plot for Votes by Party

5. Model Training and Testing:

4. One hot Encoding:



# Pakistan General Election Predicting Model using Machine Learning



1

**We’ve used several python libraries for data cleansing, data engineering, pre-processing EDA, data visualization and training, testing models. Details is given below:**

## NumPy (import Numpy as np)

NumPy is a powerful numerical computing library in Python. It provides support for large, multi-dimensional arrays and matrices, along with mathematical functions to operate on these arrays.

## **1.2 Pandas (import pandas as pd)**

Pandas is a data manipulation and analysis library. It provides data structures like Series and Data Frame that are designed for working with structured data (e.g., tables or CSV files) Pandas is commonly used for data cleaning, exploration, and analysis

## 1.3 MatPlotLib

Matplotlib is a popular 2D plotting library for creating static, interactive, and animated visualizations in Python. pyplot is a module within Matplotlib that provides a MATLAB-like interface for creating plots.

## 1.4 Seaborn

Seaborn is a statistical data visualization library built on top of Matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics. Seaborn simplifies the process of creating complex visualizations, and it often works well with Pandas Data Frames.

## 1.5 Reading CSV Files to our DataFrame

Reading data from four CSV files ("NA\_RESULT\_2002.csv", "NA\_RESULT\_2008.csv", "NA\_RESULT\_2013.csv", and "NA\_RESULT\_2018.csv") into separate Pandas DataFrames named NA\_2002, NA\_2008, NA\_2013, and NA\_2018. The data is encoded using the Latin-1-character encoding.

Data Cleaning Operations on Data Frames



2

**Performs various data cleaning operations on different DataFrames, including checking for missing values and removing specific columns to prepare the data for further analysis.**

## **2.1 Standardizing Column Names and Resetting Index:**

Ensures that all four Data Frames have the same column names by assigning them to be equal. Additionally, it resets the index of each DataFrame, providing a consistent index structure for further analysis or merging.

## **2.2 Adding a 'Year' Column to Each Data Frame:**

Adds a new column named 'Year' to each of the four DataFrames (NA\_2002, NA\_2008, NA\_2013, NA\_2018). The 'Year' column is populated with the corresponding year values (2002, 2008, 2013, and 2018) to facilitate tracking the vintage of the data in each DataFrame.

## **2.3 Concatenating Data Frames and Setting Index:**

Concatenates the individual DataFrames for different years into a single DataFrame named election. Additionally, it sets the 'Year' column as the index for the combined DataFrame to facilitate time-based analysis or retrieval of specific year data.

## **2.4 Handling Null Values and Duplicates in the Combined DataFrame:**

performs data cleaning operations on the election DataFrame. It checks for missing values, removes rows with missing values, and identifies and counts duplicate rows. The operations are intended to ensure the data's cleanliness and uniqueness for further analysis.

## **2.5 Computing and Displaying Correlation Matrix for Numeric Data:**

Calculates the correlation matrix for the numeric columns in the election DataFrame and prints the matrix. It helps identify relationships between different numerical variables in the dataset.

# Univariate analysis:



3

## **3.1 Exploring Unique Values and Counts in DataFrame Columns**

Provide information about the unique values, their counts, and sorted unique values in specific columns ('Seat', 'Constituency\_Title', and 'Candidate\_Name') of the election DataFrame. It helps in understanding the diversity and distribution of data within these columns.

## **3.2 Columns and Standardizing Party Names:**

Renaming a column ('Part' to 'Party') and standardizing party names in the 'Party' column. It helps ensure consistency and correctness in the representation of political parties in the dataset.

## **3.3 Cleaning and Analyzing 'Turnout' Column:**

Cleaning and analyzing the 'Turnout' column. It converts the column to numeric format, visualizes its distribution, and identifies potential outliers using a boxplot and statistical measures (IQR). The results are printed to provide insights into the data distribution and the presence of outliers.

## **3.4 Identifying Outliers in the 'Votes' Column**

Identifying potential outliers in the 'Votes' column. It calculates the IQR and sets upper and lower limits to flag data points that fall outside this range. The results, including the limits and the number of identified outliers, are printed to provide insights into the distribution of 'Votes' in the dataset.

## **3.5 Visualizing and Handling Zero Total Votes:**

Visualizing the distribution of 'Total\_Votes' using a distribution plot and a boxplot. It also identifies and handles rows with zero total votes, ensuring data quality for further analysis.

## **3.6 Visualizing and Handling Zero Total Valid Votes:**

Visualizing the distribution of 'Total\_Valid\_Votes' using a distribution plot and a boxplot. It also identifies and stores rows with zero total valid votes in the variable zero\_total\_Valid\_votes. The visualizations help in understanding the data distribution, and handling zero values ensures data quality for further analysis.

## **3.7 Visualizing and Handling Zero Total Rejected Votes:**

Visualizing the distribution of 'Total\_Rejected\_Votes' using a distribution plot and a boxplot. It also identifies and handles rows with zero total rejected votes, ensuring data quality for further analysis.

## **3.8 Visualizing Total Registered Voters:**

Visualizing the distribution of 'Total\_Registered\_Voters' using a distribution plot and a boxplot. These visualizations help in understanding the distribution of registered voters in the dataset.

## **3.9 Constructing and Visualizing Vote Percentage Feature:**

Creating a new feature 'vote percent' in the election DataFrame to represent the percentage of votes each candidate received. It then visualizes the distribution of vote percentages using both a distribution plot and a boxplot. These visualizations help in understanding the spread and central tendency of vote percentages in the dataset.

## **3.10 Identifying Outliers in the 'Vote Percent' Feature:**

Identifying potential outliers in the 'vote percent' column. It calculates the IQR and sets upper and lower limits to flag data points that fall outside this range. The results, including the limits and the number of identified outliers, are printed to provide insights into the distribution of 'vote percent' in the dataset.

**3.11 Creating and Visualizing a Correlation Heatmap:**

Creates a correlation matrix for numeric columns in the election DataFrame and visualizes it as a heatmap. The heatmap uses color intensity to represent the strength and direction of correlations between different pairs of numeric variables. This provides insights into potential relationships within the dataset.

## **3.12 Creating Scatter Plot and Box Plot Side by Side:**

Creates two subplots side by side. The first subplot is a scatter plot showing the relationship between 'Total\_Valid\_Votes' and 'Total\_Registered\_Voters'. The second subplot is a box plot for these two columns. This visualization allows for a comparison of distribution and relationships between the variables.

**3.13 Creating Multiple Scatter Plots and Box Plots Side by Side:**

Creates three sets of subplots, each consisting of a scatter plot and two box plots for different pairs of variables. The visualizations provide insights into the relationships and distributions between the selected variables. The color-coded box plots enhance the distinction between different variables.

## **3.14 Categorical Bar Plot for Votes by Party:**

Generate a categorical bar plot showing the distribution of votes across different political parties. Each bar represents a party, and the height of the bar corresponds to the total number of votes received by that party. The x-axis labels (Party names) are rotated for better visibility. The visualization provides insights into the relative vote counts for each political party.

# One Hot Encoding



4

## **4.1 Removing Columns from the Election DataFrame**

Removes the 'Seat' and 'Candidate\_Name' columns from the election DataFrame, likely to focus on a subset of relevant features for further analysis.

## **Setting up a Column Transformer and Pipeline for Preprocessing**

Sets up a preprocessing pipeline using ColumnTransformer and Pipeline from scikit-learn. It defines transformers for one-hot encoding categorical features and standard scaling numerical features. The resulting preprocessor pipeline can be used for consistent preprocessing of the dataset.

# Model Training and Testing



5

## **RandomForestRegressor Model Training and Evaluation**:

Training and evaluation of a RandomForestRegressor model on the provided dataset. The model is assessed on both a cross-validation set and a separate test set, and various regression metrics are calculated for multiple target variables. The optional cross-validation on the entire dataset provides additional insights into the model's performance.

## **Making Predictions on New Data Using Trained Model:**

Demonstrates how to use the trained RandomForestRegressor model to make predictions on new data (NA\_2024). The predictions for each target variable are added to a DataFrame, allowing for a comparison between the actual and predicted values. This process is useful for assessing how well the model generalizes to unseen data.

## **Calculating Sum of Predicted Votes for Specific Parties:**

Calculates and prints the sum of predicted votes for specific political parties based on the output data from the RandomForestRegressor model. This information provides an overview of the model's predictions for each party in the given dataset.

* 1. **Organizing Predictions by Constituency and Party:**

Organizes the predicted values by constituency and party, creating a structured DataFrame for each constituency. The final DataFrame (na\_wise\_predictions\_df) provides a detailed view of the predicted outcomes for each political party in each constituency.

# Dashboard



6

## 6.1 Dashboard using PowerBi:

We have developed a Dashboard for our Election Prediction System using PowerBI, showcasing various components:

* **Clickable Flags**:

Flags are interactive and clickable.

Each flag presents the predicted Vote Share percentage based on our model.

* **Pie Chart:**

The Pie Chart visually represents the percentage share of votes for each political party.

* **Interactive Map:**

The map highlights different areas.

Colors on the map represent the respective political party names and their majority in each area.

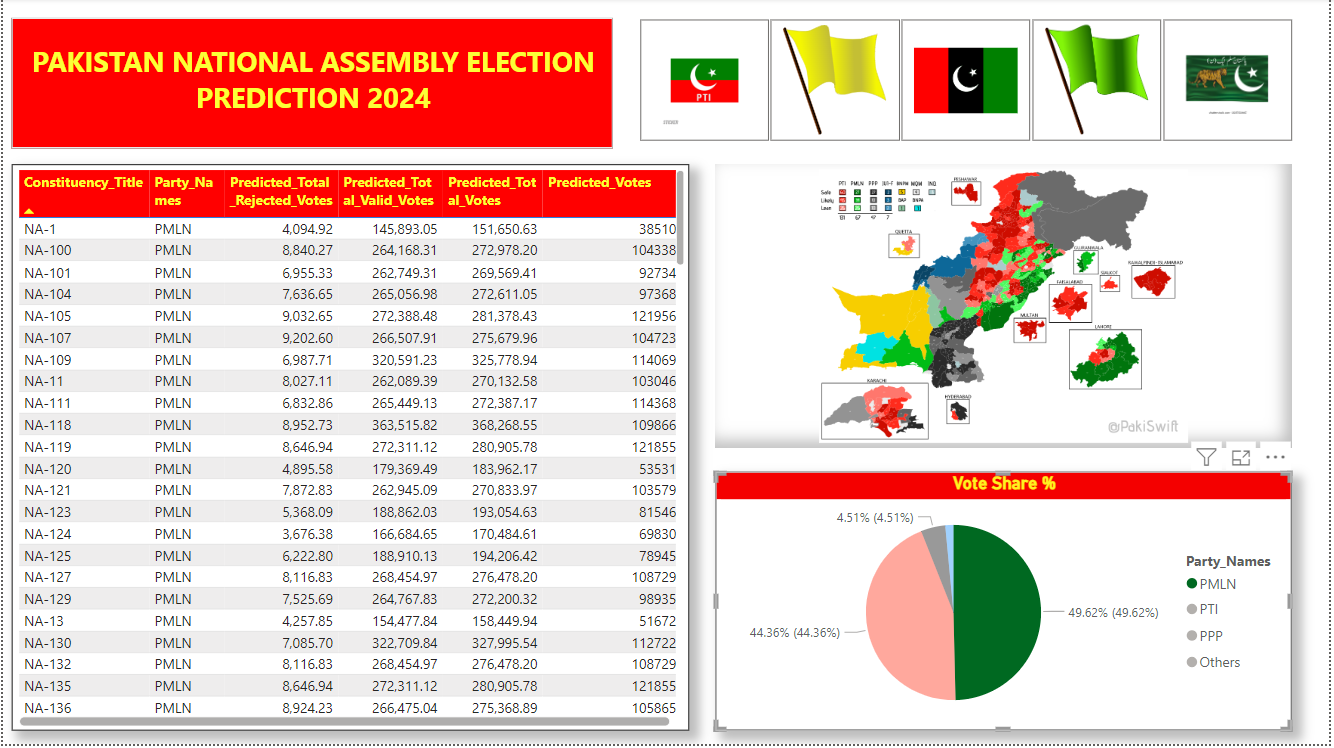
* **Outcome Table:**

A table below the charts provides a detailed breakdown of the predicted outcomes from our model.

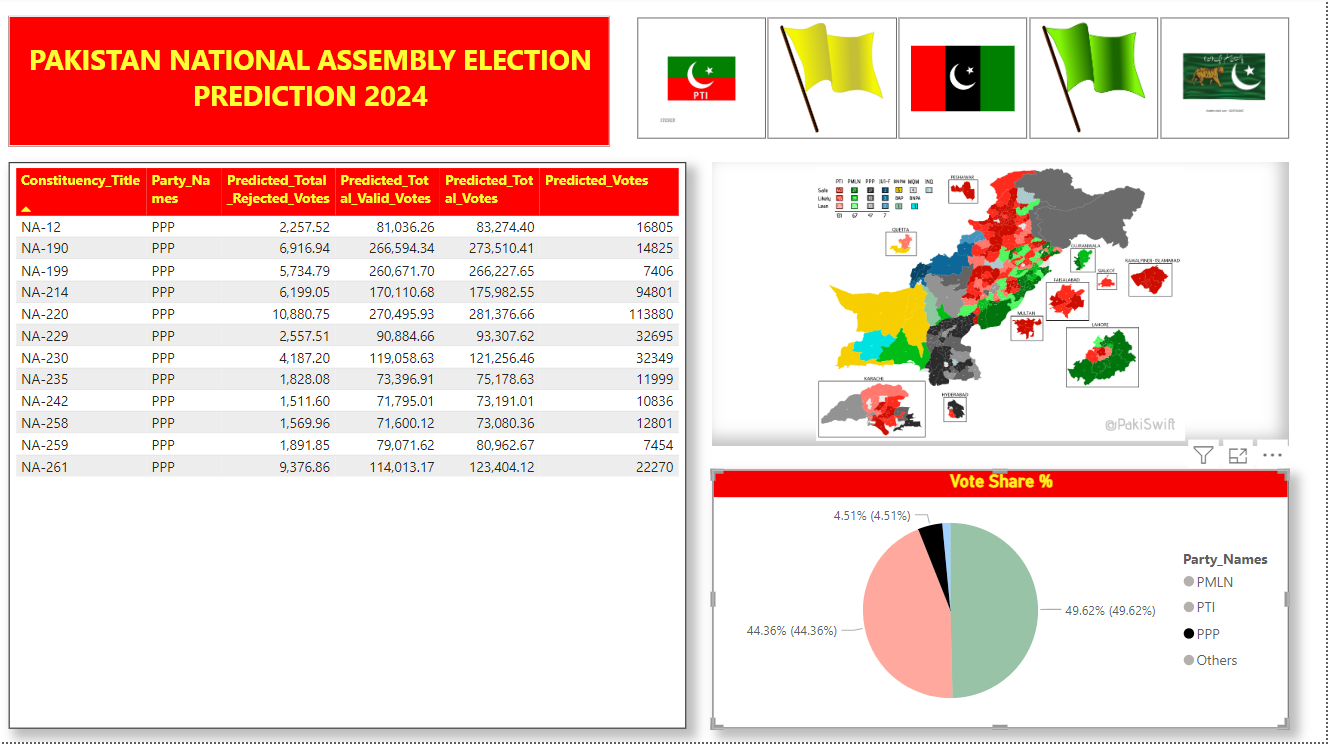
A screenshot of a computer screen

Description automatically generated

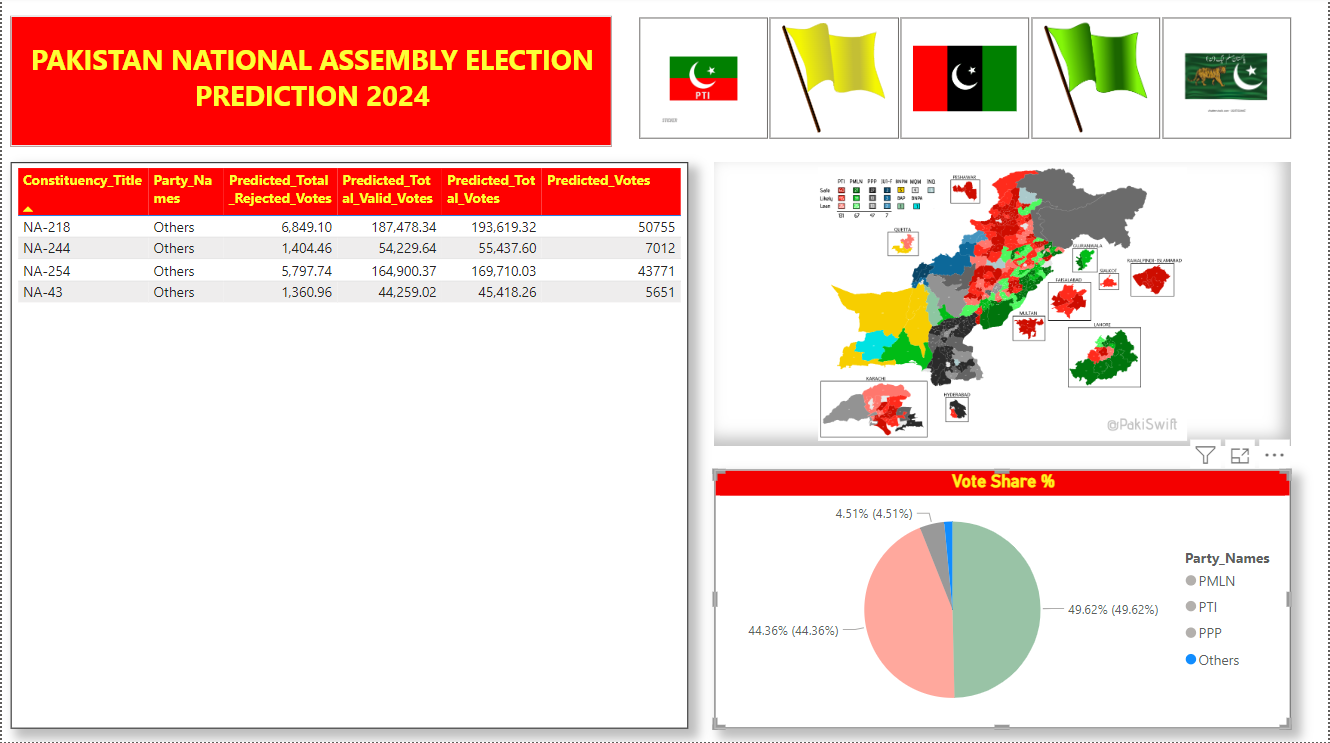
In this picture, by clicking on any flag or Pie Chart % distribution, the dashboard will automatically sort NA’s based on the predicted winning party. This enables you to check for all as mentioned below



As you can see, the green color represents **PMLN**. After clicking on this board, it automatically sorts all predicted winning seats for **PMLN**.



As you can see, the Black color represents **PPP**. After clicking on this board, it automatically sorts all predicted winning seats for **PPP**.



As you can see, the Blue color represents **Others**. After clicking on this board, it automatically sorts all predicted winning seats for **Others**.

According to Above Dashboard predictions results, here is Table in which all predictions counts of Seats for Each Party is represented.

|  |  |  |
| --- | --- | --- |
| **Party\_Name** | **Predicted Votes** | **Seats** |
| PTI | 22616118 | 118 |
| PMLN | 18942454 | 132 |
| PPP | 5179680 | 12 |
| IND | 9748611 | 0 |
| OTHERS | 10,059,540 | 4 |
| **Total** | **66546403** | **266** |

The following table presents the seat-winning predictions made by our model. It's important to note that, according to historical data, independent candidates typically align with a political party after winning a seat. Therefore, our model predicts 0 seats for independent candidates, as they are expected to join another party and lose their independent status upon winning.

# Comparison with Habib Akram’s Survey:

Below shown is the survey conducted by Habib Akram on different constituencies:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **NA** | **PTI** | **PML-N** | **PPP** | **Others** | **Not Voting** | **Total** |
| **NA-1** | 40 | 18 |  | 25 | 17 | 100 |
| **NA-2** | 62 | 8 |  | 15 | 15 | 100 |
| **NA-3** | 60 | 8 |  | 16 | 16 | 100 |
| **NA-4** | 62 | 6 |  | 17 | 15 | 100 |
| **NA-5** | 48 |  | 10 | 25 | 17 | 100 |
| **NA-11** | 58 | 16 | 4 | 11 | 11 | 100 |
| **NA-12** |  |  |  |  |  | 0 |
| **NA-13** |  |  |  |  |  | 0 |
| **NA-14** | 57 | 20 |  | 10 | 13 | 100 |
| **NA-15** | 56 | 22 |  | 8 | 14 | 100 |
| **NA-19** | 62 | 9 |  | 17 | 12 | 100 |
| **NA-22** | 52 |  |  | 35 | 13 | 100 |
| **NA-23** | 58 |  |  | 23 | 19 | 100 |
| **NA-28** | 60 |  |  | 21 | 19 | 100 |
| **NA-32** | 60 |  |  | 23 | 17 | 100 |
| **NA-33** | 48 |  |  | 29 | 23 | 100 |
| **NA-34** |  |  |  |  |  | 0 |
| **NA-44** | 56 | 4 | 4 | 21 | 15 | 100 |
| **NA-45** |  |  |  |  |  | 0 |
| **NA-46** | 51 | 20 |  | 10 | 19 | 100 |
| **NA-47** | 53 | 18 | 5 | 10 | 14 | 100 |
| **NA-48** | 50 | 19 | 4 | 10 | 17 | 100 |
| **NA-49** | 45 | 28 |  | 8 | 19 | 100 |
| **NA-56** | 46 | 25 | 3 | 8 | 18 | 100 |
| **NA-57** | 47 | 26 | 4 | 6 | 17 | 100 |
| **NA-58** | 37 | 38 |  | 8 | 17 | 100 |
| **NA-60** | 45 | 28 | 2 | 9 | 16 | 100 |
| **NA-64** | 60 | 14 |  | 8 | 18 | 100 |
| **NA-65** | 44 | 22 | 12 | 9 | 13 | 100 |
| **NA-66** | 35 | 32 |  | 11 | 22 | 100 |
| **NA-67** | 43 | 31 |  | 9 | 17 | 100 |
| **NA-68** | 50 | 26 | 3 | 7 | 14 | 100 |
| **NA-71** | 46 | 26 |  | 6 | 22 | 100 |
| **NA-73** | 43 | 30 |  | 10 | 17 | 100 |
| **NA-75** | 39 | 36 |  | 8 | 17 | 100 |
| **NA-76** | 30 | 36 |  | 14 | 20 | 100 |
| **NA-77** | 39 | 37 |  | 7 | 17 | 100 |
| **NA-78** | 38 | 35 |  | 8 | 19 | 100 |
| **NA-79** | 37 | 30 |  | 15 | 18 | 100 |
| **NA-82** | 38 | 39 | 2 | 8 | 13 | 100 |
| **NA-83** | 40 | 39 |  | 5 | 16 | 100 |
| **NA-84** | 44 | 31 | 3 | 6 | 16 | 100 |
| **NA-89** | 74 | 9 |  | 6 | 11 | 100 |
| **NA-90** | 60 | 18 | 3 | 5 | 14 | 100 |
| **NA-94** | 38 | 34 | 6 | 6 | 16 | 100 |
| **NA-96** | 47 | 35 |  | 9 | 9 | 100 |
| **NA-100** | 44 | 30 |  | 12 | 14 | 100 |
| **NA-102** | 49 | 29 | 3 | 7 | 12 | 100 |
| **NA-104** | 44 | 32 |  | 9 | 15 | 100 |
| **NA-105** |  |  |  |  |  | 0 |
| **NA-106** | 45 | 34 |  | 8 | 13 | 100 |
| **NA-107** | 48 | 29 |  | 8 | 15 | 100 |
| **NA-109** | 48 | 26 |  | 7 | 19 | 100 |
| **NA-114** | 33 | 36 |  | 17 | 14 | 100 |
| **NA-131** | 28 | 38 |  | 14 | 20 | 100 |
| **NA-132** | 33 | 45 |  | 10 | 12 | 100 |
| **NA-145** | 41 | 26 |  | 17 | 16 | 100 |
| **NA-148** | 38 | 22 | 12 | 7 | 21 | 100 |
| **NA-149** | 48 | 22 | 5 | 6 | 19 | 100 |
| **NA-150** | 41 | 28 | 4 | 5 | 22 | 100 |
| **NA-152** | 40 | 26 | 14 | 6 | 14 | 100 |
| **NA-153** | 39 | 25 | 5 | 12 | 19 | 100 |
| **NA-155** | 43 | 28 |  | 12 | 17 | 100 |
| **NA-158** | 43 | 29 |  | 10 | 18 | 100 |
| **NA-160** | 22 | 34 |  | 24 | 20 | 100 |
| **NA-161** | 34 | 33 |  | 15 | 18 | 100 |
| **NA-162** | 39 | 33 |  | 10 | 18 | 100 |
| **NA-163** | 36 | 27 |  | 24 | 13 | 100 |
| **NA-165** | 38 | 25 |  | 21 | 16 | 100 |
| **NA-168** | 52 | 20 |  | 10 | 18 | 100 |
| **NA-172** | 51 | 21 | 7 | 5 | 16 | 100 |
| **NA-174** | 41 | 16 | 18 | 9 | 16 | 100 |
| **NA-175** | 52 | 17 | 5 | 9 | 17 | 100 |
| **NA-185** | 59 | 16 | 3 | 6 | 16 | 100 |
| **NA-194** | 21 |  | 50 | 7 | 22 | 100 |
| **NA-200** | 33 | 4 | 33 | 7 | 23 | 100 |
| **NA-201** | 22 |  | 37 | 18 | 23 | 100 |
| **NA-202** | 22 |  | 46 | 7 | 25 | 100 |
| **NA-207** | 28 | 3 | 34 | 11 | 24 | 100 |
| **NA-220** | 38 |  | 18 | 21 | 23 | 100 |
| **Total Covered** | | | | | **75** | |

The survey data, encompassing only 75 constituencies, is significantly limited. The data suggests a predominant trend where PTI is projected to secure the highest number of votes across the majority of surveyed constituencies. However, it's crucial to note that the political landscape in Pakistan has undergone substantial changes over the past two years, introducing dynamic elements that are challenging to capture through predictive modeling.

Our model, like any machine learning model, operates based on historical data and patterns. It may struggle to accurately predict outcomes influenced by recent and unforeseen political developments. Additionally, the limited scope of the survey poses constraints on constructing a comprehensive model, especially when extrapolating results to a broader context.

In reality, numerous factors beyond the scope of our model and the survey can impact election outcomes. Pertinent considerations include the presence of PTI candidates outside their home constituencies, evolving political dynamics, and other events that may influence voters' choices.

Given the constraints imposed by the limited survey data and the inherent complexity of the political landscape, it becomes challenging to assert a definitive prediction that a single party, such as PTI, will uniformly dominate all constituencies. The multifaceted nature of Pakistani politics demands a nuanced understanding of the diverse factors at play.

It's essential to approach these predictions with caution and acknowledge the uncertainty associated with extrapolating from a limited dataset. Moreover, considering the dynamic nature of the political landscape, seeking additional expert opinions or incorporating real-time updates may enhance the robustness of any electoral analysis.

In conclusion, while the survey data and machine learning models offer valuable insights, a comprehensive understanding of the upcoming election requires a holistic view that incorporates the broader political context and the unpredictable dynamics that characterize the evolving political scenario in Pakistan.