Whom To Vote?

Prachal Jitendrakumar Patel
Software Engineering
San Jose State University
San Jose, USA
prachaljitendrakumar.patel@sjsu.edu

Priyam Vaidya
Software Engineering
San Jose State University
San Jose, USA
priyam.vaidya@sjsu.edu

Jay Satishkumar Pathak
Software Engineering
San Jose State University
San Jose, USA
jaysatishkumar.pathak@sjsu.edu

Ashwini Ulhas Talele
Software Engineering
San Jose State University
San Jose, USA
ashwiniulhas.talele@sjsu.edu

***Abstract*—For any country, the government holds an important role for comprehensive development which is expected by the citizens. In order to decide which government to vote for, thorough knowledge is required. But, due to data abuse and digital frauds, people are convinced to vote for a particular candidate which might be misleading. Media and other resources might portray the details which form the biased government. The factors that one should take into consideration to select a government are economic growth, literacy rate, etc. The aim is to gather data from previous elections, analyze economic growth patterns and represent in a visual format that can be easily understood by the voters.**

Keywords—Machine learning, React, Docker Container, Presidential Election Forecasting Artificial neural networks, Support vector regression, Linear regression, Prediction

I. Introduction

Predicting voter turnout has been a persistent obstacle for campaigns and analysts trying to model election outcomes[1]. The stakes for accurately predicting which parts of the electorate will vote are different depending on the interest group: campaigns, for example, have only a limited number of people and dollars at their disposal. Efforts to increase voter turnout can include setting up phone banks, organizing volunteers to pass out fliers, and recording robocalls [2]. Understanding which voters are least likely to vote can guide a campaign's use of money and person-power.

The primary objective of this paper is to model and forecast the United States presidential election via the usage of learning algorithms. Political and economic variables are utilized in the model, and significant variables are identified through further analysis and statistical procedures. The dependent variable is defined as the electoral votes of the incumbent party. The incumbent party is considered as the dependent variable because it presents further related variables such as the incumbent president's approval rate and gross domestic product (GDP)

II. ARCHITECTURE FLOW

The web application for "Whom To Vote" has the following architecture.

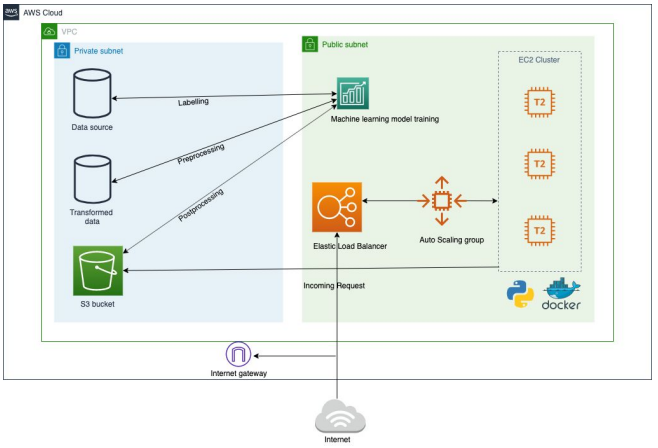

Fig. 1. Architecture Flow

III. Personas and Hill Statement

Hills defines the direction of the project and the team. The hill statement and persona expectations are as below:

1. Voter: Voter can view the results based on economic growth in a particular period and vote for upcoming elections.

Who? Voter

What? Can view voting data

How? in the form of graphs

2. Political Party: A political party can view prediction and plan their campaign accordingly.

Who? Political Party
What? Can view voting data to look at their progress
How? by looking at the predictions

IV. Technology Components

A. Frontend

i. React: Client-side interface.
ii. Chart.js: To display data in different charts.

B. Backend

i. Python: To implement the Machine Learning model.

C. Infrastructure

i. Docker: To containerize the machine learning model and react application.
ii. Classic Load Balancer: To load balance incoming requests.
iii. Auto Scaled EC2: Scaling according to incoming traffic.
iv. S3 bucket: To store intermediate results.

V. Model Components

The following block diagram shows the major components of our prediction model.

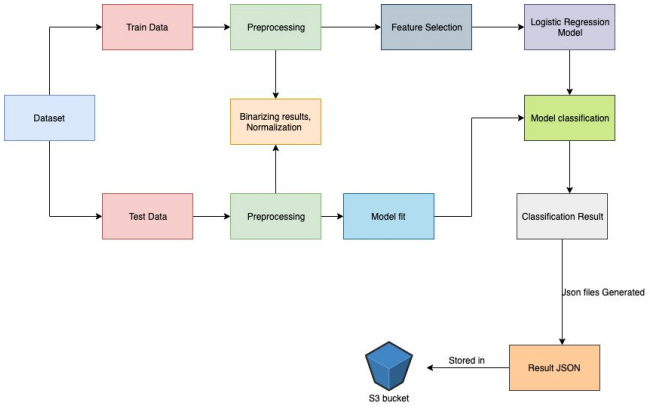

Fig. 2. Machine Learning Model

A. Dataset :

The first component of the model is the dataset. The US election data from 2012 and 2016 having multiple features data is used to train the model. The dataset includes total population, median age, unemployment rate, per capita income, total households, population growth, and the winning party.

B. Pre-processing

The winning party has two possible outcomes. So, the outcomes are binarized in 0 and 1. All the features in the dataset are normalized using Euclidean normalization. Unnecessary features are removed in order to achieve optimum accuracy.

C. Model training

We are using the logistic regression model for training our data. The model further is used to predict the results of the test data. The training accuracy achieved is 84%.

D. Model Prediction

The predictions for our test data were stored in the JSON file and directly uploaded to the S3 bucket[3]. The data is fetched from the S3 bucket by the frontend code and displayed the data in the interactive chart format.

E. Classification Result

The classification results for the democrats were 41.61% and the republicans were 58.39%.

VI. Prediction Insights And Web Interface

1. Main Page

The main page of the application is the welcome page and has some links to contact, about, support.

Fig. 3. Main Page

2. Statistics Page

This page shows the overall chart having all the features view at once. Individual feature charts can be viewed clicking respective buttons.

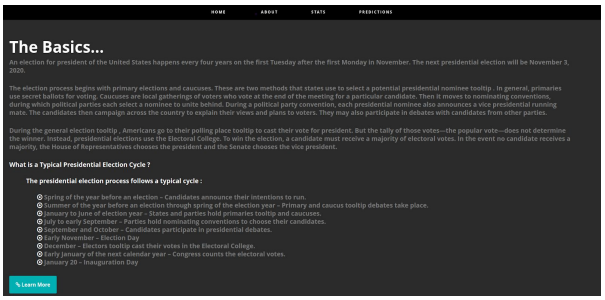

Fig. 4. About Page

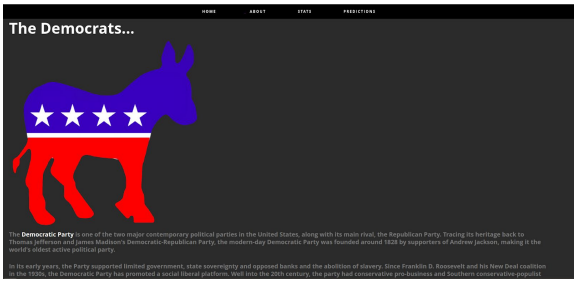

Fig. 5. About Page Democrat Data

3. Prediction Page

This page shows the winning prediction in the 2020 elections on a button click.

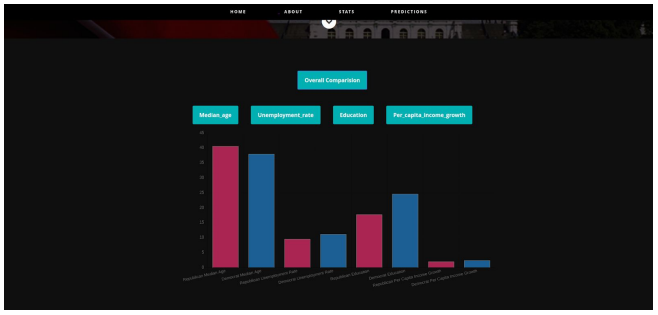

Fig. 6. Overall Features Chart

Fig. 7. Individual Feature Chart

4. About Page

This page shows the information of both the political parties. External links are provided which can be clicked to get more information.

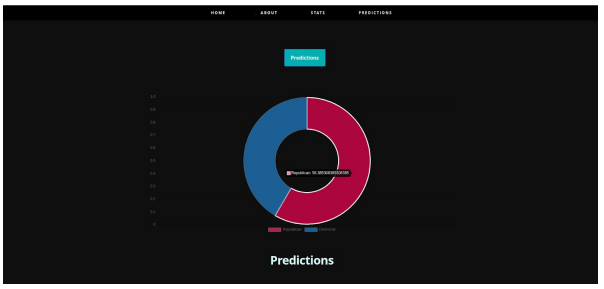

Fig. 8. Prediction Chart

VII. Conclusions

The project has helped us understand the machine learning concepts, integration with full-fledged web components, and dockerizing an application.

Acknowledgment

We would like to take this opportunity to extend our gratitude to Professor Rakesh Ranjan who encouraged us to into developing such an innovative application and helped us in understanding the design phase of the application.

References

[1] R. Igielnik, and S. Keeter 20036 U.-419-4300 | M.-419-4349 | F.-419-4372 | M. Inquiries, Can Likely Voter Models Be Improved?, Pew Research Center, 07-Jan-2016.

[2] D. P. Green and A. S. Gerber, Get Out the Vote: How to Increase Voter Turnout. Brookings Institution Press, 2015.

[3] Storing the data from python to S3 bucket https://dzone.com/articles/boto3-amazon-s3-as-python-object-store

[4] Using the Logistic Regression model for prediction. https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

[5] Using chart.js in the frontend to display the content in the form of charts: https://www.chartjs.org/

[6] Dockerizing the node js application https://nodejs.org/de/docs/guides/nodejs-docker-webapp/

[7] Dockerizing python machine learning model https://hackernoon.com/building-python-data-science-container-using-docker-c8e346295669