Feature Engineering for Election Results Prediction Using Machine Learning

SONALI MAHENDRAKAR
Sreenidhi Institute of Science and Technology

Abstract— In this present generation where political landscape as in political parties based on election commissions for the respective countries has always been complex in nature as it involves not only the elector's opinion but also the general public opinion from which the ruling party will be decided. Several studies have shown how public opinion matters in analyzing various decisions such as in the case of political elections. This has been done by analyzing and working on numerous social media networking platforms and websites. The consistent rise of social media has given people the opportunities to openly discuss and debate various issues and consequences happening around the world there by considering issues such as political elections predictions, weather analysis and its prediction, credit card fraud prediction and many more. All these prediction cases can be generally be calculated by implementing them by using various technologies such as machine learning, data science etc. In this following project which deals with the technology machine learning it predicts the chances of winning political party in the upcoming elections. This has been predicted by using machine learning algorithms which also derive the accuracy for the given dataset used. The dataset has been collected from twitter social media platform where general public that is citizens of that particular country can openly participate, which is based on citizens of that specific country which is dependent on elections and political campaigns. Based on this scenario we have generated a machine learning model based on data preprocessing.

Indexed Terms – Machine Learning, Feature Engineering, Data preprocessing, Supervised Learning

I. INTRODUCTION

Electoral systems are the detailed constitutional arrangements and voting systems that convert the vote into a political decision which has to be made by evaluating the decisions made by public as well the members of the politics like who will be the ruling party.. An election is a way people can choose their candidate or their preferences in a representative democracy or other form of government or their democractic party based on their preferences . The use of the Twitter social media networking platform as a tool to predict the outcomes of social phenomena that is winning political party in the elections is a task and it is useful where numerous public people can come and declare their decisions and reasons regarding hundreds of issues happening around the world. Eventually, the rise of social media has given people the opportunities to openly discuss and debate various issues and consequences happening around the world there by considering issues such as political elections predictions, weather analysis and its prediction, credit card fraud prediction,pandemics detection and also the recent covid-19 pandemic like the %of vaccinated people % alive cases etc can be known through the social networking sites such as twitter, facebook.

II. LITERATURE SURVEY

Election prediction and analysis are very popular as well as a complex problem amongst the machine learning community as it deals with huge data corresponding to the public decisions made. In the past various approaches and methods have been taken in to solve those raising problems based on the prediction of such issues. This case has been previously solved by using deep learning but in this project, we will be adding some new features to get better performance and accuracy for the machine learning data model being implemented.

III. PROPOSED SYSTEM

The proposed system consists of a feature engineering tool that takes the raw data and transforms that existing data into features and also creates features or variable which are not present in the training data which further can be used in the creation of predictive models using machine learning or deep learning as such. This feature engineering comes into picture to increase the performance and enhancing accuracy of the model .

Advantages:

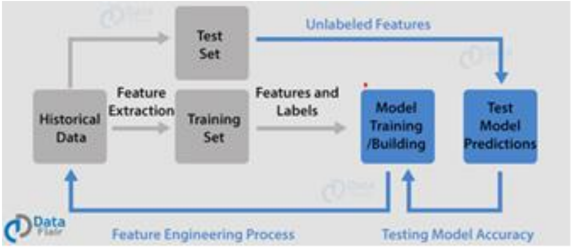
1. Its goal is for speeding up and simplifying data transformation for better flexibility of the features created in the model.
2. It is also used in enhancing model accuracy.

IV. TECHNOLOGY

The technology that deals with the working of this project is python and machine learning. The libraries which I have imported for this project are Numpy which refers to as numerical python and is used for working with arrays in python language, Pandas which deals with better performance by using multi-dimensional arrays, Matplotlib and seaborn are used for graphical visualizations, Scikitlearn which is basically a toolkit which imports packages related to machine learning.

The algorithms which I have used in this project are supervised machine learning algorithms that are Logistic Regression, K-NN classifier, and Random Forest classifier.

V. SYSTEM ARCHITECTURE



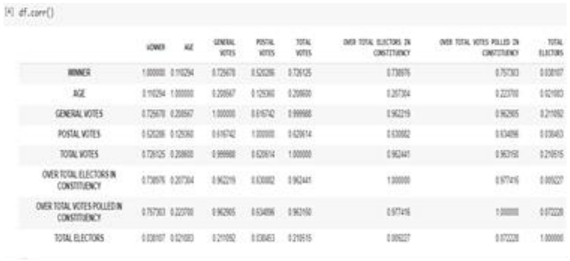
VI. DATASET

The data is collected from twitter social media platform based on the real time data of the upcoming political elections from the year 2015 country India.

The features utilized in the dataset which has been derived from twitter social media platform which is based on numerous citizens opinion on the electoral campagains are: state, constituency, name, winner, party, symbol, gender, age, criminal cases, category, education, assests, liabilities, general votes, postal votes and they consists of over 1600 records.

VII. IMPLEMENTATION STEPS

After importing the libraries and the dataset, the determining correlation matrix which deals with the changes of two variables to show which variable is having how much correlation with the other, and this is visualized by using Heatmap which is imported from Seaborn Library which deals with graphical and mathematical visualizations.



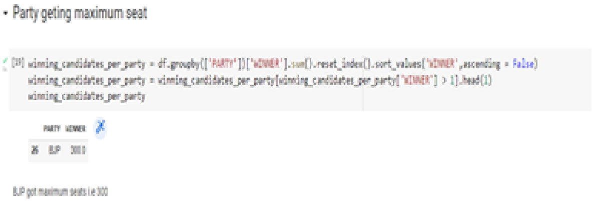
```
[4] df.corr()
```

	WINNER	AGE	GENERAL VOTES	POSTAL VOTES	TOTAL VOTES	OVER TOTAL ELECTORS IN CONSTITUENCY	OVER TOTAL VOTES POLLED IN CONSTITUENCY	TOTAL ELECTORS
WINNER	1.000000	0.110294	0.725678	0.520286	0.726125	0.738976	0.757303	0.030107
AGE	0.110294	1.000000	0.208567	0.129360	0.208600	0.207304	0.223700	0.021083
GENERAL VOTES	0.725678	0.208567	1.000000	0.616742	0.999988	0.962219	0.962905	0.211092
POSTAL VOTES	0.520286	0.129360	0.616742	1.000000	0.620614	0.630082	0.634096	0.030453
TOTAL VOTES	0.726125	0.208600	0.999988	0.620614	1.000000	0.962441	0.963150	0.210515
OVER TOTAL ELECTORS IN CONSTITUENCY	0.738976	0.207304	0.962219	0.630082	0.962441	1.000000	0.977416	0.009227
OVER TOTAL VOTES POLLED IN CONSTITUENCY	0.757303	0.223700	0.962905	0.634096	0.963150	0.977416	1.000000	0.072228
TOTAL ELECTORS	0.030107	0.021083	0.211092	0.030453	0.210515	0.009227	0.072228	1.000000

Now we have clean the dataset as it consists of some missing values as it takes around 10%, so we can just delete those parts as provides us with better efficiency. After doing the necessary changes, we find the total constituencies per state and thereby plot a graph by using features based on party count.

As this project deals with feature engineering used to improve the performance of the model, we will create some new features based on the available raw features.

1. Party getting maximum seats



- Party geting maximum seat

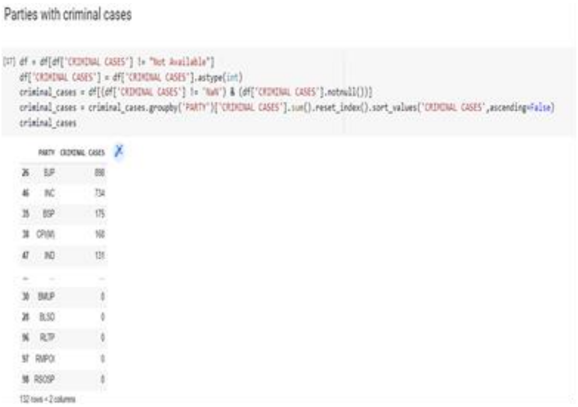
```
[19] winning_candidates_per_party = df.groupby(['PARTY'])['WINNER'].sum().reset_index().sort_values('WINNER',ascending = False)
     winning_candidates_per_party = winning_candidates_per_party[winning_candidates_per_party['WINNER'] > 1].head(1)
     winning_candidates_per_party
```

```
    PARTY  WINNER
26    BJP   300.0
```

BJP got maximum seats i.e 300

2. Parties with criminal cases

Generally, citizens will often consider the number of criminal records against a party before voting. If in case a party has more number of criminal records means less votes thereby less chances of winning the election. Therefore, we can use this by making a new feature which represents the number of criminal records against a party.



Parties with criminal cases

```
[17] df = df[df['CRIMINAL CASES'] != "Not Available"]
     df['CRIMINAL CASES'] = df['CRIMINAL CASES'].astype(int)
     criminal_cases = df[(df['CRIMINAL CASES'] != 'NaN') & (df['CRIMINAL CASES'].notnull())]
     criminal_cases = criminal_cases.groupby('PARTY')['CRIMINAL CASES'].sum().reset_index().sort_values('CRIMINAL CASES',ascending=False)
     criminal_cases
```

```
     PARTY  CRIMINAL CASES
26   BJP    898
46   INC    734
35   BSP    175
38   CPI(M) 160
47   IND    131
...  ...    ...
30   BMUP   0
28   BLSD   0
96   RLTP   0
97   RMPOI  0
98   RSOSP  0
132 rows × 2 columns
```

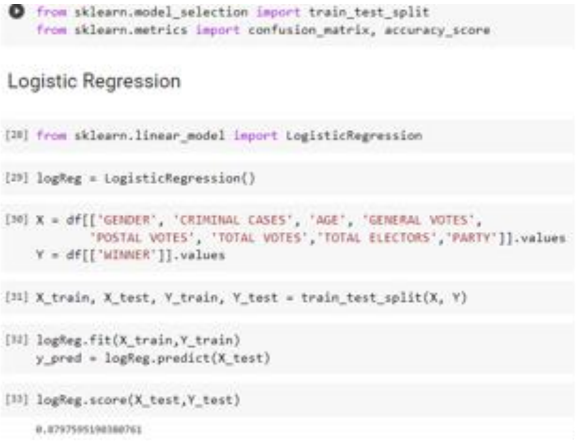
3. We can also consider education level, age which shows the candidate's knowledge and experience and create a new feature respectively.

Using all these newly created features we train the data on three machine learning models that are Logistic regression, K-NN and Random Forest which are all imported from sklearn library.

VIII. OUTPUT

1. Logistic regression

Logistic regression is the process of modelling probability of discrete outcomes for a given input data variable. For examples it predicts outcomes such as yes/no, true/false etc. For this project we have imported logistic regression package from sklearn linear model library using which we have divided our twitter dataset into test data and train data respectively and thereby predicting the performance of the model. For logistic regression we have obtained 87% accuracy.



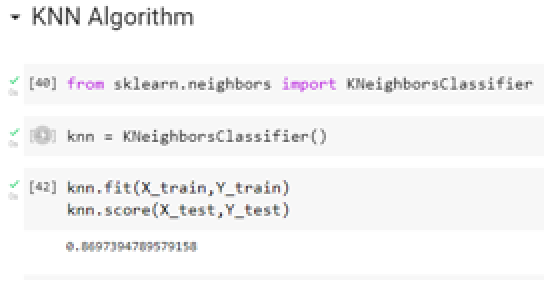
```
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix, accuracy_score
```

Logistic Regression

```
[28] from sklearn.linear_model import LogisticRegression

[29] logReg = LogisticRegression()

[30] X = df[['GENDER', 'CRIMINAL CASES', 'AGE', 'GENERAL VOTES',
            'POSTAL VOTES', 'TOTAL VOTES','TOTAL ELECTORS','PARTY']].values
     Y = df[['WINNER']].values

[31] X_train, X_test, Y_train, Y_test = train_test_split(X, Y)

[32] logReg.fit(X_train,Y_train)
     y_pred = logReg.predict(X_test)

[33] logReg.score(X_test,Y_test)

     0.8797595190380761
```

2. K – Nearest Neighbor Classification

K-NN is a supervised learning algorithm which deals with the minimum eucliden distance between the new dat point created and the all available categorical datapoints. Hence it is also referred to as lazy learner algorithm.

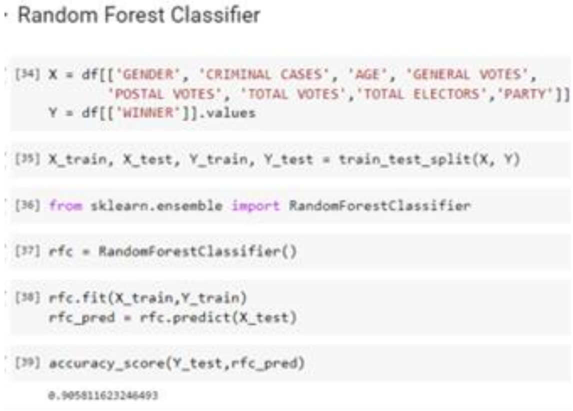


KNN Algorithm

```
[40] from sklearn.neighbors import KNeighborsClassifier

[41] knn = KNeighborsClassifier()

[42] knn.fit(X_train,Y_train)
     knn.score(X_test,Y_test)

     0.8697394789579158
```

3. Random Forest Classification

Random Forest is derived from ensemble learning method which operates by constructing multiple decision trees for the training data. By using random forest classifier for this project I have overcome the problem of overfitting, thereby improving the efficieny of the model.

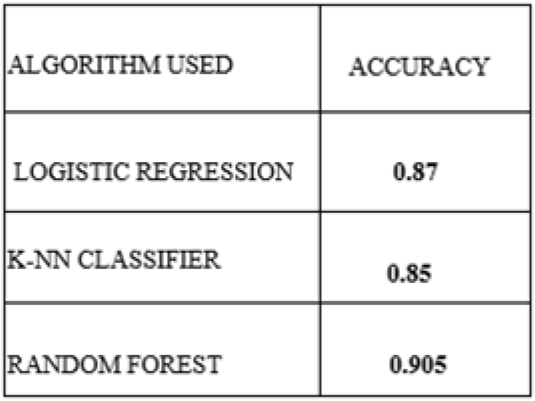


Random Forest Classifier

```
[34] X = df[['GENDER', 'CRIMINAL CASES', 'AGE', 'GENERAL VOTES',
            'POSTAL VOTES', 'TOTAL VOTES','TOTAL ELECTORS','PARTY']]
     Y = df[['WINNER']].values

[35] X_train, X_test, Y_train, Y_test = train_test_split(X, Y)

[36] from sklearn.ensemble import RandomForestClassifier

[37] rfc = RandomForestClassifier()

[38] rfc.fit(X_train,Y_train)
     rfc_pred = rfc.predict(X_test)

[39] accuracy_score(Y_test,rfc_pred)

     0.905811623246493
```

IX. CONCLUSION

In this paper, we engage on the task to predict the Indian political elections by performing feature engineering technique on the dataset. The problems encountered during the course of implementing the project was collecting large dataset as it involves thousands of general public opinion on the elections, and thereby removing missing values, so as to obtain higher accuracy and enhance the performance of the data model. The model uses Feature Engineering in order to better focus on important features and develop a model with better efficiency and performance such that it can handle real world data. The primary reason for using the Feature Engineering technique is to overcome the problem of directly implementing the model on raw data and thereby deriving poor performance. Hence, by the execution of this technique in this project we have obtained a better accuracy.



ALGORITHM USED	ACCURACY
LOGISTIC REGRESSION	0.87
K-NN CLASSIFIER	0.85
RANDOM FOREST	0.905

X. ACKNOWLEDGEMENT

I want to thank everyone who worked behind the scenes to help me turn an idea into a real time working application model. Iam incredibly thankful to my parents for giving me this opportunity to pursue and accomplish my goals. Without them, I would not have been able to do so. The institution is headed by CEO Mr.K. Abhijit Rao, Director Prof. C.V Tomy and Principal Dr. T. Ch. Shiva Reddy, who helped me complete my research work, all played an important role. I would like to express my sincere gratitude to Dr. Aruna Varanasi, Head of Department of Computer Science Engineering, who has served as a wonderful mentor and inspiration for my research work.

REFERENCES

[1] M. Coletto , C. Lucchese , S. Orlando, and R. Perego, “Electoral Predictions with Twitter: a Machine-Learning approach”.

[2] S. JayaKumar, Priyank Patel, Rajat Kumar Singh, Shivraj, Akkshansh Paul, “Election Result Prediction Using Deep Learning Techniques”.

[3] Harsh patil, “What is Feature Engineering — Importance, Tools and Techniques for Machine Learning towardsdatascience.com”.

[4] T.D.Jawawickarma “Election Result Prediction (in Python) towardsdatascience”.

[5] P.Kalyan, “A Comprehensive Guide on Feature Engineering analyticsvidya”

[6] Haider Ali ,Haleem Farman ,Hikmat Yar, “Deep Learning-Based Election Results Prediction Using Twitter Activity”.

[7] José Antonio León Borges,Roger-Ismael-Noh-Balam, “The machine learning in the prediction of elections”.