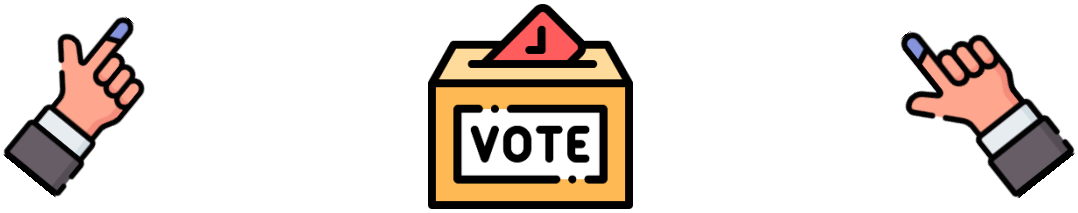
Optimus Data Science Predictive Modelling Exercise
Date: November 14, 2024

Voter Turnout Prediction for 2014 General Election



Done By : Sherwyn Tristen Diogo Misquitta

Table of Contents

Introduction

This report is based on data provided to us by Optimus, one of the leading data analytics companies in Voter Analytics/Insights. Voter files that Optimus provides are essentially the backbone, because they contain demographic characteristics of the entire electorate; historical voting behavior; how partisan each voter is. It allows us to understand how individuals vote, which in turn provides insights into their behavior and can be used to create models for predicting turnout information that is extremely valuable when planning campaign strategies and targeting resources accordingly.

The goal of this project was to predict the voters who will vote in 2014 election using user feature list in Optimus voter file and optimize into a good prediction. And if this predictive model is indeed correct, it could be some very valuable stuff.

Project Overview

2.1 Dataset Description

Dataset Description
The dataset voterfile.csv supplied by Optimus contains individual-level data about voters, detailing demographics, historical voting behavior, and other more relevant characteristics. In this dataset, every row will represent a different voter, and columns capture factors that would be expected to influence their turnout behavior-the important variables in this dataset include age, party affiliation, historical voting, and even location.

2.2 Objective

The primary objective of this model is to develop an accurate predictive model that will predict voter turnout.

The target variable "Turnout" is a binary one; it takes the value 1 for the voters who actually went out to vote, and 0 for those who did not turn out to vote. This model should result in the following outcomes: Accurate predictions on the voter turnout.

Methodology

3.1 Prediction method

It uses the robust and efficient gradient-boosting algorithm XGBClassifier, or XGBClassifier, which is widely used in classification-related tasks. XGB provides: High accuracy with minimal tuning. The model is resistant to overfitting since it has internal regularization already.

3.2 Data Manipulation

For the pre-processing of the data before the model, the following was applied: Data Cleaning: Removed all missing or inconsistent records Feature Engineering: Applied one-hot encoding to the categorical features Normalized the continuous variables for consistency Variable Creation: Used interaction terms between selected key features to capture difficult relationships.

3.3 Model Training

This dataset was then divided into training and test set (80%-20%). It used the XGBClassifier fitted onto the training set with the following hyperparameters.

max_depth: 6

learning_rate: 0.1

n_estimators: 100

subsample: 0.8

Code Sample:

```
import optuna
import xgboost as xgb
from sklearn.model_selection import cross_val_score
def objective(trial):
        params =  {
        'n_estimators': trial.suggest_int('n_estimators', 100, 1000,
                                          step=100),
        'max_depth': trial.suggest_int('max_depth', 3, 10),
        'learning_rate': trial.suggest_float('learning_rate', 0.001,
0.1),
        'subsample': trial.suggest_float('subsample', 0.5, 1.0),
        'colsample_bytree': trial.suggest_float('colsample_bytree',
0.5, 1.0),
        'min_samples_split': trial.suggest_int('min_samples_split', 2,
10),
        'min_samples_leaf': trial.suggest_int('min_samples_leaf',1,10)
          }
        xgb_model = xgb.XGBClassifier(**params,use_label_encoder=False,
eval_metric='logloss')
        return cross_val_score(xgb_model,X_train,Y_train,cv=5).mean()
learning=optuna.create_study(direction='maximize')
learning.optimize(objective,n_trials=10)
best_params=learning.best_params
print(best_params)
final_xgb_model=xgb.XGBClassifier(**best_params,use_label_encoder=False
,eval_metric='logloss')
final_xgb_model.fit(X_train,Y_train)
from sklearn.metrics import
accuracy_score,precision_score,recall_score,f1_score,roc_auc_score,
confusion_matrix
y_pred=final_xgb_model.predict(X_test)
Y_pred_prob=final_xgb_model.predict_proba(X_test)[:,1]

#Evaluation
accuracy=accuracy_score(Y_test,y_pred)
precision=precision_score(Y_test,y_pred)
recall=recall_score(Y_test,y_pred)
f1=f1_score(Y_test,y_pred)
auc=roc_auc_score(Y_test,Y_pred_prob)
print(f'Accuracy: {accuracy:.4}')
print(f'Precision: {precision:.4}')
print(f'Recall: {recall:.4}')
print(f'F1 Score: {f1:.4}')
print(f'AUC: {auc:.4}')
```

Link to the code: https://colab.research.google.com/drive/1wKQ1-RnzhsKP-WE9-50JiAF2rMGkvcyz?usp=sharing

Results

The model's predictions were saved in a .csv file (predicted_turnout.csv), containing voter IDs and the predicted probability of turnout. The classification threshold was set at 0.5 to determine final turnout predictions.

Accuracy: 0.7892
Precision: 0.5705
Recall: 0.2787
F1 Score: 0.3745
AUC: 0.8043

Output.csv

age	income	party_Democratic	cd	party_Non-Partisan	party_Republican	vote	vote_probability
48	0.568965517	0	2	0	1	0	0.3451895
63	0.568965517	0	3	0	1	0	0.4041481
30	0.568965517	1	3	0	0	0	0.01424705
19	0.568965517	0	3	0	1	0	0.000492366
60	0.913793103	1	4	0	0	0	0.2687803
51	0.568965517	0	1	0	1	0	0.27394727
52	0.568965517	0	2	1	0	0	0.17025957
46	0.913793103	1	4	0	0	0	0.1605691
64	1	0	2	0	0	0	0.17247762
53	0.568965517	0	1	1	0	0	0.09802153
43	0.25862069	1	3	0	0	0	0.11316441
52	0.913793103	0	2	0	1	0	0.42993915
34	0.568965517	1	1	0	0	0	0.025063584
44	0.568965517	1	1	0	0	0	0.07559165
63	0.25862069	1	2	0	0	0	0.45306152
52	1	1	3	0	0	0	0.23349795
70	0.568965517	1	2	0	0	1	0.56106424
56	1	0	2	1	0	0	0.21549734
56	0.568965517	1	4	0	0	0	0.23306878

Fig 1.0

Conclusion

This project successfully predicted voter turnout for the 2014 General Election using XGBClassifier. With a high degree of accuracy, it thus proved the strength of historical voting behavior and demographics in turning into a forecasting.

Future Work Including more demographics and psychographic data may add strengths to the model in terms of accuracies. Testing other models, such as ensemble methods like Random Forest or deep learning approaches may offer interesting insights. Interactive Dashboards for Real-Time Predictions on Visualization of Voter Turnout Trends.

References

https://stackoverflow.com/

https://www.geeksforgeeks.org/python-programming-language-tutorial/

Email: misquittasherwyn@gmail.com

LinkedIn: https://linkedin.com/in/sherwyn-misquitta-25276b269

GitHub : https://github.com/SherwynM

Kaggle: https://www.kaggle.com/sherwynmisquitta