Predicting German Election Outcomes Using Machine Leaning Techniques

Dang Ngoc Huy
h.dang@mpp.hertie-school.org

Maximillian Rekuts
m.rekuts@mia.hertie-school.org

Bailey Sutton
b.sutton@exchange.hertie-school.org

Abstract

The ability to predict election outcomes has long been an aim of social science research due to the significant political, social, and economic implications of such a task. However, a voter's behavior is not always rational and there are multiple confounding factors that can effect the vote share of a candidate or political party. Stoetzer, Neunhoeffer, Gschwend, Munzert, & Sternberg (2019) attempted to use classical statistical models to predict the 2017 German federal election and were able to do so within the margin of error for 6 of the 7 parties. The purpose of this paper is to use the same data and scope as Stoetzer et al., but with the aim of simplifying their model through the utilization of machine learning techniques. The main challenge of the data is the inconsistent frequencies in the time-series resulting from different polling process by individual institutions. Therefore, this paper evaluates five different machine learning algorithms to identify the best model for predicting the changes in public support for a particular party based on previous polls. Thus far, the Gradient Boosting model outperforms all others in terms of the chosen evaluation metrics.

1. Methodology [1]

The purpose of this project is to take the data from Stoetzer, Neunhoeffer, Gschwend, Munzert, & Sternberg (2019) [7] and determine whether or not there is a comparable or better machine learning model in terms of accurately predicting the results of the 2017 German federal election. The primary assessment metric for accuracy is RMSE, which mirrors what Stoetzer et al. used for their findings. In terms of predictive ability, machine learning models have the potential to perform as well as classical statistical models, such as the one used by Stoetzer et al., but with significantly greater interpretability.

[1]Project GitHub: `https://github.com/huydang90/Predicting-German-Election`

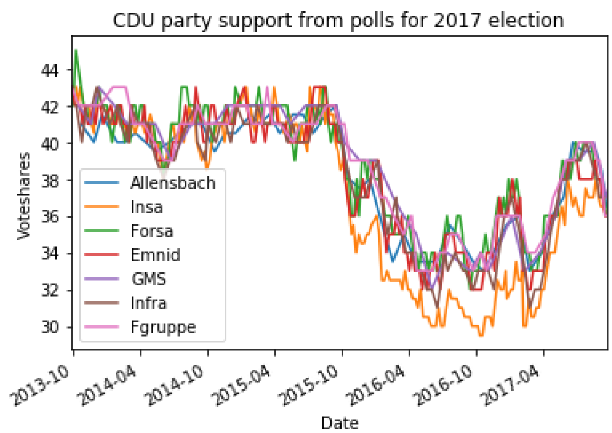

Figure 1. CDU Party Support From Polls for 2017 Election, October 2013 - September 2017

There are two main concerns with using time-series polling data in relation to this project. First, there needs to be greater weight given to polling results released closer to an election than earlier results. Stoetzer et al. accomplished this in their model using a random walk backwards approach [7]. In order to ensure accuracy over time, a machine learning model must accomplish a similar task. Second, by compiling data from several official polling organizations, the resulting data set has multiple frequencies, which can also reduces a model's accuracy. The multiple frequencies are a result of the various sampling methods used by each polling organization. Some samples are taken more frequently than others, some have a greater sample size, and some have a more limited demographic scope. This creates additional noise in the model that is difficult to control or limit.

When examining data from each polling company for each political party, all but one party has trends of support that are more or less the same across all of polls from different institutions. Figure 1 illustrates this for the CDU (see appendix for all other figures). Therefore, even though the sampling methods vary across different polling organizations, the polling numbers seem to be able to reflect true public opinion at any given point in time. The final polling numbers from all of the institutions are lower but extremely close to the actual results. This is a consistent observation for all other parties, except for Die Linke, which saw a much greater variations in public support that is not well captured through the polls.

There are a few possible solutions to managing this multi-frequency time-series data. Casals, Jerez, & Sotoca (2009) note the limits of fully aggregating data as it may result in some observations being over or under weighted [2]. Therefore, they recommend only partially aggregating the sample data in order to reduce the amount of frequencies while minimizing the negative effects of aggregation [2].

Cachucho, Meeng, Vespier, Nijssen, & Knobbe (2014) expand on this and test several possible aggregation techniques in the pre-processing phase of a machine learning model. The primary method in their analysis is termed "the accordion method" and includes several feature constructions [1]. The first is a set of aggregate functions based on the mean, median, maximum value, minimum value, standard deviation, inter-quartile range, and root mean squared error [1]. The second involves a grid search of these aggregate functions to find the optimal value for the entire set [1]. Finally, the authors tuned their accordion model to continually search for the best aggregate feature at every step of the modeling process [1]. Ultimately, their model proved to be efficient in the pre-processing stages when working with this kind of data [1].

Beyond the concerns about how to approach this kind of data in the pre-processing phase, many researchers have also struggled to find an accurate forecasting model for multi-frequency time series data. Kourentzes, Petropoulos, & Trapero used a three step process for their analysis: aggregation, forecasting, and then combining the forecasts into one results [5]. Their most novel application was during the aggregation stage where they used time series components as the basis for their aggregation, which considerably simplified the process [5].

Finally, Huber (2017) used polling data to inform predictions about German state elections and had similar roadblocks in the aggregation and pre-processing of the data [3]. The author found that as a result of the differences in poll quality and frequency across German states, some models were more accurate than others [3]. This will likely pose a similar problem for this project.

Based on the methodologies implemented in previous literature, this project will seek to solve these three main data challenges as follows:

1. **Multiple Time Series Sampled at Inconsistent Frequencies:** There are 7 polling institutions included in the data from Stoetzer et al. and each of them conducted their polls at different intervals. There is a gross imbalance in regards to the availability of data points, which makes modeling impossible. Therefore, the data needs to be upsampled by interpolating additional data with moving averages so that the number of data points are consistent across different institutions;

2. **Transforming Time Series Data into a Form Suitable for Machine Learning Algorithms:** An unique challenge of time series data is that it cannot be directly split into training and testing sets for machine learning modeling as each data point does not simply belong to a separate variable that can be selected as feature, but instead it's a part of a sequence of information. Therefore, most of the features in the models need to be engineered. For the main features, changes in public support for the current day and for 3 days in the future have been created together with the percentage point changes in public support, as well as moving averages of support for time periods of 3, 5, 10, 20, 30, 50, 100 and 200 days. This approach resembles the random-walk method utilized in Stoetzer et al (2019) as it takes into consideration the influence that polling numbers on different dates might have on the final prediction [7];

3. **Modeling Multiple Time Series with Machine Learning Algorithms:** The final obstacle to overcome is how to input these time series data into each model. There are two possible approaches: creating individual models for data from each institutions or aggregating the information into one combined model. While more data usually means better predictions, as exploratory analysis has shown in Figure 1 and then Figures 5-10, the poll numbers from all institutions show surprisingly similar trends and patterns in public support for nearly every party. Therefore, a simple model with data from a single institution might still be able to accurately predict an outcome because it contains the main signal while reducing additional noise. However, by concatenating all the series into a single data set and then training a single model, this aggregated model may be able to generalize better as it has more data to learn and train from. However, it might do so at the expense of simplicity and accuracy. In the current scope, this project will first explore using a single institution to determine whether or not it can be advanced towards reasonable accuracy in comparison to Stoetzer et al. Following the analysis of this model, the next step will be to implement the second approach by combining the time series for each party into a single, aggregate data set and model.

2. Experiments

Data: Data from the 2017 German federal election will be used to create a time series data set that can be processed by the machine learning models. The main data set contains information about the percentage of party support from 7 different polling organizations in Germany. Relevant information includes the date and year the polls were conducted, name of the organization, sample size of polling surveys, party support, and the number of days until the election from the date the survey data was released.

Evaluation method: The primary assumption is that the polling numbers closely reflect the underlying trends in party support from the public at any given point in time, as demonstrated in the methodology. The goal of this current phase in the project, therefore, is to build a machine learning model that can predict the changes in party support from public polls with reasonable accuracy. At this stage, predicting the exact vote share in the final election is not the target, but rather the goal is to assess the changes in public support and how accurately the model can predict shifts in sentiment over time. If the results are reasonably accurate, the next stage is to use this model to forecast the final outcome of the election. For this phase, the project will utilize root-mean-square error (RMSE) as the main evaluation metric.

Experimental details: For this first phase, the focus is on time-series data from one institution. For modeling, Allensbach's polls numbers for AfD were chosen as the primary data for analysis.

After pre-processing the time-series data with the procedures outlined in Methodology, we experimented with five separate models, including:

- **Linear Regression Model:** This model was chosen as the baseline upon which to improve predictive performance. It was able to deliver a surprisingly small RMSE result despite being a relatively simple and straightforward model;
- **Decision Tree Model:** After experimenting with different tree heights, the best maximum height for the decision trees was 3. However, it was still not a strong model and did not perform better than the baseline;
- **Random Forest Model:** The model was fitted with default parameters values at first, which did not yield satisfactory result (RMSE = 2.11). A dictionary of hyperparameters was created to search for the optimized options, including n_estimators, max_depth, and max_features. The model was then looped with a parameter grid with different settings of the hyperparameters and the scores were then saved. The grid search was able to determine the optimal tuning option which yielded a significantly improved result (RMSE = 1.49);
- **Gradient Boosting Model:** After grid searching, the optimal hyperparameters for this model are a learning_rate of 0.01, n_estimators of 200, and max_features of 4. It seems that for this particular problem, by building the decision trees successively, each subsequent tree fits the residual errors from the previous tree. This improves the overall fit through each iteration and this model outperformed all other models;
- **XGBoost Model:** Using GRIDSEARCHCV from the SCIKIT-LEARN library, the optimal hyperparameters were identified to be: colsample_bytree = 0.7, learning_rate = 0.01, max_depth = 7, min_child_weight = 4, n_estimators = 500, nthread = 4, objective = 'reg:linear', silent = 1, and subsample = 0.7. However, even with these hyperparameters, the model was not able to perform better than the Gradient Boosting model.

Method	RMSE
Linear Regression	1.95
Decision Tree	1.97
Random Forest	1.49
Gradient Boosting	1.3
XGBoost	1.84

Table 1. Comparison of Model RMSEs

Results: The results of all five models are illustrated in Table 1. As previously discussed, Gradient Boosting outperforms all other models with regards to minimizing RMSE.

The results of these models are as expected since data from only one institution was used and, thus, it was anticipated that the machine learning algorithms would have good predictive power with relative ease of computation. However, as the project moves to the next phase with the aggregated data model, this may no longer be the case as adding more data and engineering new features will likely increase the dimensionality of the data set.

An interesting conclusion after fitting the different models is the significance of different features to each model. By analyzing feature importance, it is possible to identify the most relevant features that drive the model's final prediction, which helps to better obtain an understanding of the model's logic and further work on improving it by focusing on the most important variables. A surprising commonality amongst these different models is the importance of the 200-day moving average. It seems that this technical indicator was able to distinguish long-term trends in the models

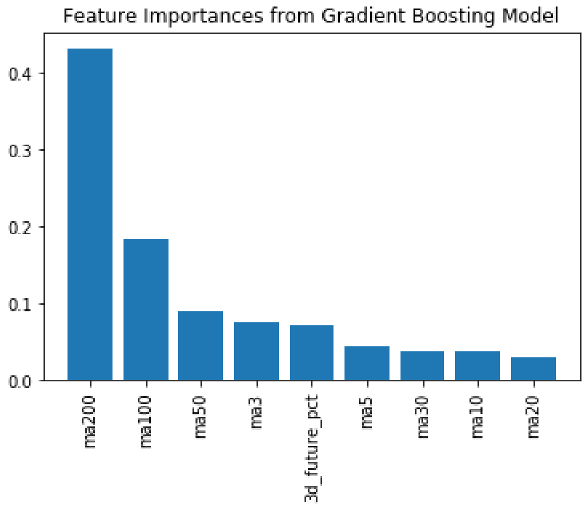

Figure 2. Feature Importance from Gradient Boosting Model

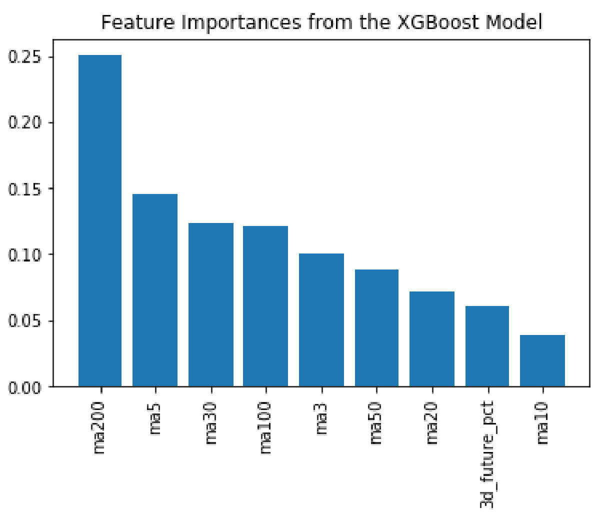

Figure 3. Feature Importance from XGBoost Model

that could potentially provide a more accurate prediction of future public support for a party. Figure 2 and Figure 3 present the feature in terms of importance for the Gradient Boosting and XGBoost models respectively. The figures compare how each variable affects the model in terms of its predictive ability.

This is further illustrated by the heat map in Figure 4, which shows the correlations between the feature and target variables. The 200-day moving average, once again, dominated other predictors in terms of its abiltiy to predict public support for AfD through polling data from Allensbach.

3. Future work

For the next stage of the research, we will attempt the following tasks:

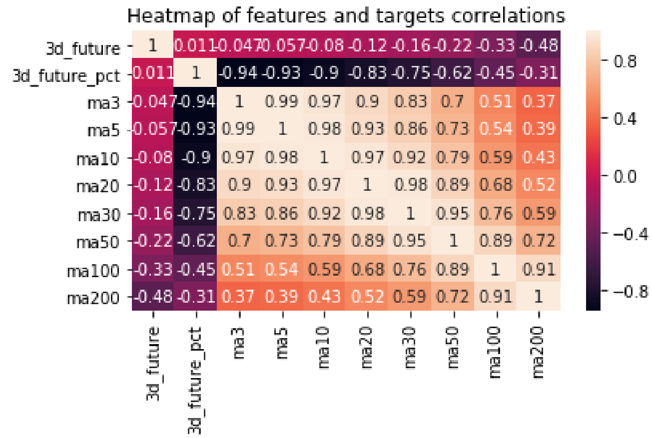

Figure 4. Correlations between Features and Target

- Forecast results using the combined data of all institutions for each party;
- Engineer more features (lags, differences of support, etc.) that can help to improve predictions;
- Implement a Neural Network model with Deep Neural Networks and Recurrent Neural Network with LSTM;

References

[1] R. Cachucho, M. Meeng, U. Vespier, S. Nijssen, and A. Knobbe. Mining multivariate time series with mixed sampling rates. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '14, pages 413–423, New York, NY, USA, 2014. ACM.

[2] J. Casals, M. Jerez, and S. Sotoca. Modelling and forecasting time series sampled at different frequencies. *Journal of Forecasting*, 28:316 – 342, 07 2009.

[3] S. Huber. *The role of polls for election forecasting in German state elections*. PhD thesis, Universität Passau, 2018.

[4] R. Kennedy, S. Wojcik, and D. Lazer. Improving election prediction internationally. *Science*, 355(6324):515–520, 2017.

[5] N. Kourentzes, F. Petropoulos, and J. Trapero. Improving forecasting by estimating time series structural components across multiple frequencies. *International Journal of Forecasting*, 30:291–302, 04 2014.

[6] M. S. Lewis-Beck and R. Dassonneville. Forecasting elections in europe: Synthetic models. *Research & Politics*, 2(1):2053168014565128, 2015.

[7] L. F. Stoetzer, M. Neunhoeffer, T. Gschwend, S. Munzert, and S. Sternberg. Forecasting elections in multiparty systems: A bayesian approach combining polls and fundamentals. *Political Analysis*, 27(2):255–262, 2019.

[8] M. Zolghadr, S. A. A. Niaki, and S. Niaki. Modeling and forecasting us presidential election using learning algorithms. *Journal of Industrial Engineering International*, 14(3):491–500, 2018.

[2] [1] [5] [3] [7] [8] [6] [4]

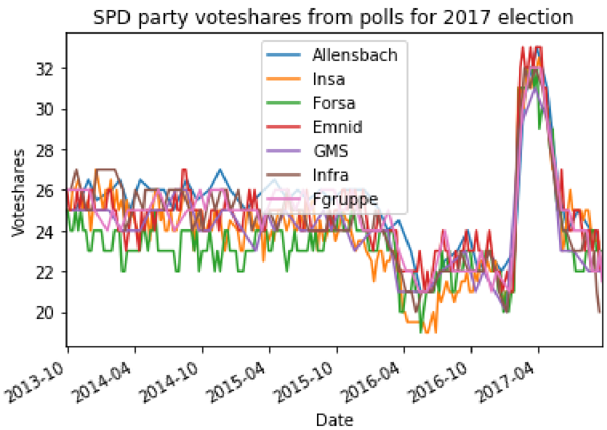

Figure 5. SPD Party Support From Polls for 2017 Election, October 2013 - September 2017

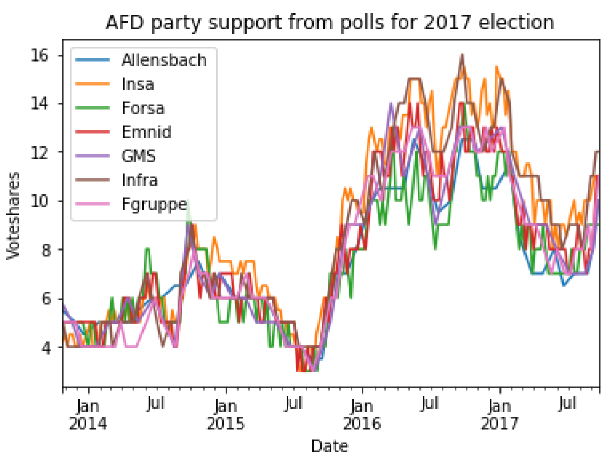

Figure 6. AfD Party Support From Polls for 2017 Election, October 2013 - September 2017

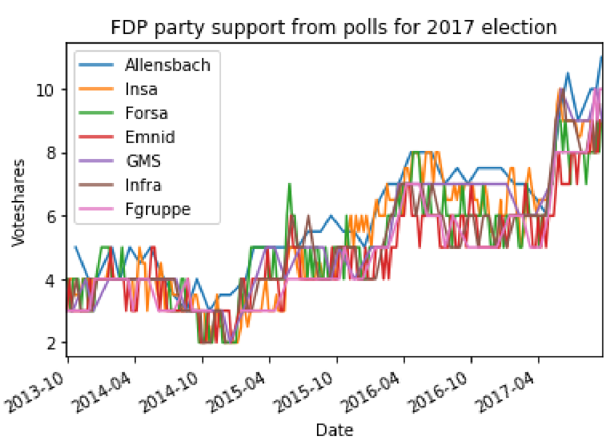

Figure 7. FDP Party Support From Polls for 2017 Election, October 2013 - September 2017

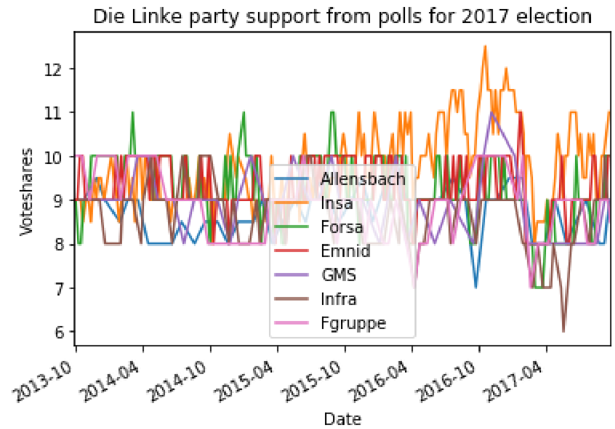

Figure 8. Die Linke Party Support From Polls for 2017 Election, October 2013 - September 2017

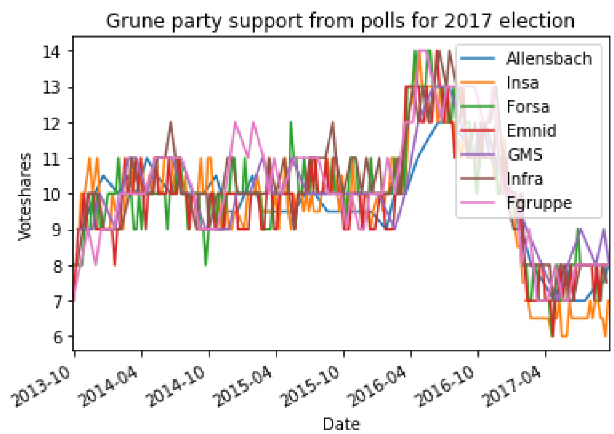

Figure 9. Green Party Support From Polls for 2017 Election, October 2013 - September 2017

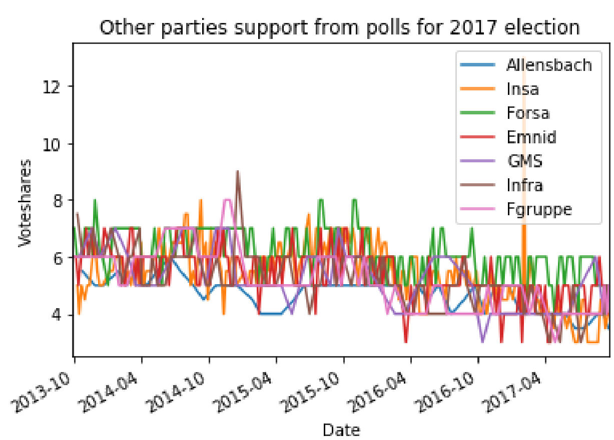

Figure 10. Other Parties Support From Polls for 2017 Election, October 2013 - September 2017