Predicting Voter Turnout in Virginia Elections

Jordan Denish (jfd6twz), Malcolm Mashig (mjm6jy), Christian Rogers (cdr5zk)

Project Category: Politics and Economics

Abstract

The following is a report discussing our group's project for the CS 4774 Machine Learning for Virginia Objective. We were interested in predicting voter turnout as a percentage for the 2020 Presidential Election and wanted to develop a model that can predict voter turnout in any county in Virginia using data from several sources over the past decade. We found data corresponding to voter turnout in Virginia and two other states, along with demographic and economic data at the county level for these states. We fit multiple regression models to this data, ultimately deciding on a Random Forest Regression model after tuning. Fortunately, we were able to run our model for the just-completed 2020 Presidential election, and evaluated our predictions against actual data. We ended up with a mean absolute error of 3.8% in our predictions. We were pleased with the results from our live 2020 predictions, and our project resulted in fascinating conclusions surrounding the impact of demographic, economic, and electoral factors on voter turnout. For instance, we discovered that economic data such as county income and unemployment was far more important for predicting turnout than the demographic makeup of the county. Ultimately, we believe that our model can generate useful turnout predictions for the counties in Virginia (and other states), which is very useful to the state as well as individual campaigns.

Introduction

Many people around the country were focused on the 2020 Presidential Race, so we wanted to work on a project associated with the election. As we were discussing the first Presidential Debate, we thought of the idea to tackle the problem of predicting voter turnout in such an uncertain time. We wanted to project voter turnout in Virginia and possibly other states for any year, for several different types of elections. We found a data set that contains voter turnout for elections in the past 13 years, divided by county and precinct in Virginia. We planned on predicting turnout using voter registration, population, and demographic data provided in the data set.

Predicting voter turnout is an important step in projecting elections and their outcomes, especially at the granular county level. This type of project would be extremely helpful in politics to improve the projection systems for elections. For politicians and their campaigns, improved voting turnout projections can help them better allocate their resources at the state or local level. Civic participation is an important part of the United States democratic process, and it is in the best interest of Virginia that as many voices as possible are heard on election day. Therefore, the state can use this project to determine which factors are important for increasing voter turnout, and allocate resources to increase voter participation in areas that are projected to have low turnout.

We found voter turnout data at the county and precinct level, spanning from 2007 to 2019 in Virginia, Colorado, and Wisconsin (not all years are represented for each state). We received permission from the TAs to use supplemental data from additional states in order to increase our sample size and hopefully improve our model. We are trying to predict voter turnout at the county level, so we believe that the counties in Colorado and Wisconsin will be representative of counties in Virginia. We tried searching for additional voter turnouts from states that were related to Virginia in terms of political or demographic composition, and we think it will also be very interesting to see if the turnout differs by state.

We are most interested in the November general elections as well as party primaries. We also found several different sources of demographic and economic data that we hope are predictive of voter turnout each year for varying election types. For example, we found county demographic information for every state in the United States that contains information about the age, gender, and racial composition for each county. Additionally, we found data sets that measured the unemployment levels, reported the average income by county, and classified the counties as rural and urban. We are extremely interested in how data from these unrelated sources that we have combined together, which have become features in our model, will relate to our outcome variable, voter turnout.

Method

Our method was to follow the traditional end-to-end machine learning steps. That said, after deciding the problem we want to solve as described above, we began to explore and visualize the data from the various data sources, prior to the cleaning process. This was an attempt to understand the nature of our outcome (voter turnout) and predictors, and to hypothesize the relationship they might have with each other. To do this, we analyzed a correlation matrix and visualized the scatter plots for each pair of variables. We also explored the distribution of each variable, and took note of which variables were numerical/categorical, and which contained outliers or missing values that we would later have to deal with. This initial discovery allowed us to gain preliminary insight before conducting the steps of cleaning and model-fitting, where this insight and view of the big picture will be helpful to have.

The next step was to preprocess our data using machine learning techniques. In this step, we did some minor feature engineering to make sure our variables were consistent across our data sources. For example, we had to ensure our outcome variable was a percentage and so we had to actually calculate this for a portion of our data. Additionally, we used the imputer, scaler, and one-hot-encoder functions from sklearn to ensure missing, skewed, and non-numeric values would not cause a problem in our model-fitting efforts later on. We added all preprocessing steps to a pipeline so that we could apply it to both our training and test data sets, and any new data that we come across. To split our data, we conducted an 80-20 split stratified by state and year, to guarantee our test set was representative of our training set. Due to the low number of observations we are working with, we decided not to set aside a validation set because we do not want to further decrease our training set, which we will use to craft our model. Instead, we will take advantage of cross-validation to tune our models, before the final evaluation on the test set, which comprised of 2020 voter turnout. Our preliminary modeling efforts are discussed in the following section.

Experiments

Once we cleaned and combined all of our disparate sources of data into one final dataframe, and created a pipeline to easily clean and process all of this data, we fit some preliminary models to get a sense of whether we have useful data for predicting voter turnout. Since we have three different states in our data, each with a varying number of observations, we used a stratified split to get similar percentages of each state in our train and test sets. We then ran our train data through the pipeline and were ready to fit some models. Thus, we fit a few regression models on our initial data, deploying ridge regression, random forests, and boosting. To evaluate our models, we chose to use RMSE as this is a common error metric for these methods.

We started off with a simple linear regression model through sklearn, and received a testing error of 14.6% versus a training error of 11.9% (using percent because we are predicting voter turnout percentage, so percent is our unit of measurement). We were happy with these initial results, as even though 15% error is probably not too useful for prediction, we think it serves as a solid baseline, and something we can certainly improve on with feature engineering and more powerful models. We also fit some other regression models, in order to narrow down what else we will pursue in the future. These included decision tree regression, random forest regression, and ridge regression. We found that these models had similar error rates to the least squares model, but random forest performed slightly better with a 13.8% test error. All in all, we learned from these initial experiments that our methodology and assumptions are good to move forward with and try to improve upon our baseline error of roughly 15%.

Next, before we tried any additional modeling, we want to continue adding to our feature engineering. Previously, we had many features and a lot of them seemed likely to be related to each other, but we believed we could combine and transform features to create stronger features that will be more predictive of voter turnout. Also, by narrowing our feature space, we hoped to avoid some of the overfitting that might have been occurring in these initial models. Thus, we performed some feature engineering to create our own variables for the type of election (presidential, congressional, or local) and also for separating the counties' demographic compositions into our own created divisions by age and race. This narrowed our feature space and made our models simpler to interpret. The most important steps in our feature engineering process were imputing missing values using available data, which resolved data issues (missing data entries for voting turnout), and creating this election type feature that became our most important feature in the model.

After our feature engineering, our next task was determining what the final model looks like. We tuned each of the models we tried previously (simple linear regression, random forests, and boosting), using cross-validation to find the best hyperparameters. After finalizing the tuning framework and running the tuning with each of our models, we ultimately determined that our tuned random forests model performed best and should be the final model we choose for prediction for the 2020 election.

While building a model to predict voter turnout is our project goal, we are also interested in finding out which factors lead to higher voter turnout. We believe that it is important for the State of Virginia to know what these factors are, so that they can be remedied (if possible) in areas of low turnout. To find this, we looked at the importance of our features in our final model and their positive or negative effect on voter turnout. We think that this information will also be very useful, beyond just reporting the predictions that the model gives us.

Results

Fortunately, we had an excellent opportunity to evaluate our model's predictions with the 2020 election. As such, we did not feel the need to split our data into a formal testing set, meaning we could use more of our data when training. We were able to get 2020 turnout data from the same source that we got our Virginia turnout data for past elections. However, there were some issues with this data, as many precincts clearly had incomplete numbers (for example, Virginia Beach had 0 votes reported, which is obviously incorrect). Therefore, we did not want to run our model on this raw data. To fix this, we decided to only consider counties that had already reported 60% turnout within this data. Since this was a Presidential election (and a very hyped one at that), we knew that turnout would be high, and thus only wanted to include counties that we think have fully reported results or are very close to doing so. When subsetting to these counties, our model achieved a mean absolute error of 3.8% (indicating that, on average, our predicted turnout was 3.8% off the actual turnout for that county). We thought that this was a solid error number, and indicated that our model is useful for prediction on real elections. When more of the data comes in, we will be able to get a clearer picture of our actual error, but we don't believe it will change meaningfully from this number.

In addition to building a model that could predict voter turnout, we also learned a lot about voter turnout in America. Since we ended up using a random forest model for our predictions, we were able to extract feature importances. By far the most important feature was the type of election, which makes sense as we know historically that presidential elections have far higher turnout than other types. After the type of election, the other important features in the model were unemployment percentage and per capita income, which go hand in hand. Higher income and lower unemployment indicate a higher turnout for that county, according to our model. It's fairly well documented that more affluent areas tend to have higher turnout in elections, and our model confirms this notion. Interestingly, some of the other features that we incorporated into the model, especially demographic data, didn't seem to be particularly important in the random forest. Perhaps this information is already captured by the income and unemployment percentage of the county, or it is just not too important in general. Ultimately, we were satisfied with the performance of our model and were also able to gain some knowledge about which factors really drive turnout in American elections.

Here is a link to our colab notebook which contains all of the code written for this project: link

Conclusion

A model like ours, with some improvements, could be very helpful for the state of Virginia and its residents. With enough county-based demographic and economic data, our model can be utilized to speculate voter turnout throughout Virginia in upcoming elections. Those who use the model can have confidence that the predictions will be fairly accurate, based on the performance of our model in the training and testing phases. Accurate voter turnout predictions would be extremely valuable to political marketing campaigns who need to know where their advertising dollars will be best spent. Nonpartisan organizations dedicated to promoting civic participation can target areas where voter turnout is projected to be weak, and to increase the involvement of citizens is in the best interest of the country's and Virginia's democracy. It is undesirable if a large portion of the Virginia population is unheard in an election of government officials who will be tasked to represent them. In conjunction with data on the political party membership of residents in each county, a partisan organization can use our model to identify the ideal counties to advertise in – counties where turnout is low and where most residents are bound to vote in their favor. Alternatively, in this way, our model can be used to predict election outcomes, which would be valuable information to many. With accurate voter turnout projections, media organizations can predict election winners before all votes are counted, and other businesses/organizations can foresee and adapt for future policy changes that may affect them.

To improve our model, it would likely be most important to obtain more data, either looking farther back into Virginia's documented election history, and/or looking at elections in other similar states. In this way, we can more effectively train our model so that it is more robust when applied to counties in other states and more robust to different election situations / conditions. For example, if we incorporate the 2020 presidential election into our training data, then if the 2024 presidential election also involves a large number of mail-in ballots, our model will be prepared to handle this nuance, and still make accurate projections. We could also look to add more predictors to our model, such as more economic indicators, demographic factors, or even predictors that are totally different than what our model uses now, such as county-based internet search interest (into topics related to voting / upcoming elections).

Team Member Contributions

Much of our time on this project has been spent processing this data by accounting for NA values and joining our data from many different sources into a single dataframe that we can preprocess to fit models. This procedure of loading and gathering the data, preprocessing the data, and concatenating and merging the data was a significant step in our project that we are proud to have completed. This process really made us understand how data scientists in the real world spend the majority of their time cleaning data.

Malcolm worked predominantly on gathering the data by loading the voting turnout data from each state for each year, combining the data into a large CSV, and uploading that data to his GitHub, where we can easily load that data into our Colab Notebook.

Jordan and Christian both worked on searching for data online, finding several sources of historical demographic, economic, and geographic data, combing the data into a CSV, and sending the data to Malcolm, to be uploaded on his GitHub.

All three of us worked to preprocess the dataframes from all of our sources to make sure the column naming convention was consistent and the individual values were identical, so we could subsequently concatenate and merge the dataframes into a single dataframe that we will put into a Pipeline.

Jordan built a Pipeline that preprocesses this final dataframe, imputing NaN values, experimenting with some feature engineering, and scaling the numerical and categorical data to generate our design matrices.

Christian fit several initial models, such as linear regression, ridge regression, and random forests, looking at the coefficient values and importance of our features.

After the checkpoint, all three of us worked together to tune the models, select our final model, and use this final model to predict processed 2020 turnout data. All three of use worked to organize and shoot our video and write this report.

References

https://apps.elections.virginia.gov/SBE$_CSV/ELECTIONS/ELECTIONTURNOUT/$

https://www.sos.state.co.us/pubs/elections/Results/archive2000.html

https://www.census.gov/data/datasets/time-series/demo/popest/2010s-counties-detail.html

https://www2.census.gov/programs-surveys/popest/technical-documentation/file-layouts/2010-2019/cc-est2019-alldata.pdf

https://elections.wi.gov/sites/elections.wi.gov/files/page/presidential$_general_election_turnout_2012_2016_co_13582.xlsx$

https://www.bls.gov/lau/cntyaa

https://apps.bea.gov/iTable/iTable.cfm?reqid=70step=1isuri=1acrdn=7reqid=70step=1isuri=1acrdn=7

https://github.com/MalcolmMashig/mach-learn