Summary

The aim of this paper is to predict the 2024 presidential election results in Virginia using the voting, census, and general economic data over the election years of 2008, 2012, and 2016. To accomplish this analysis, our group selected a few key variables from census and general economic data, namely poverty level, per capita income, median house value, GED ratio, unemployment rate, inflation rate, interest rate, and oil prices. All of this data was formatted to line up with the previously mentioned election periods in 2008, 2012, and 2016, which was then initially used in a k-Nearest Neighbors model to predict the general importance of the selected variables. The data was then used as an input to a Lasso-linear model which could be used to predict 2024 data without census data due to it currently being unreleased. With the census and general economic data we aimed to predict the difference in votes between republican and democratic candidates in order to predict the outcome of each county and therefore the state of Virginia for the 2024 presidential election. Running the model produced a vote skew (Democrat - Republican votes) of -32000, predicting an overall win for the Republican presidential candidate in Virginia 2024.

Data

The data used in this analysis consisted of US census data procured by the NHGIS from 2004-2020, Virginia presidential voting data from 2000 to 2020, and FRED economic data from 2008, 2012, and 2016. The census data used consisted of estimates in four year ranges, such as the census estimates from 2006-2010. In order to use census data to predict presidential elections, the census data for these periods was equated to the averages of the years, such that 2006-2010 was used as the census data for the 2008 presidential election. This meant that data was available for presidential elections in 2008, 2012, and 2016, using the census data for 2006-2010, 2010-2014, and 2014-2018 respectively. To predict the presidential election results, a number of variables that we thought may be correlated with voting tendencies were selected. Among these variables were income to poverty level ratio, per capita income vs

population, median house value, and GED ratio. Poverty level ratio, per capita income vs population and GED ratio were calculated by dividing the number of people in various poverty levels, incomes, and educational achievements (GED's) by the total population of the county, while median house value was taken directly as the value given in the census data. The FRED data used to predict the presidential election was the unemployment rate, inflation rate, interest rate, and oil prices in Virginia for the years 2008, 2012, and 2016.

In terms of data cleaning, there wasn't much missing data, though much of it had to be reorganized in order to get it formatted correctly for model creation. This started by sub selecting the census data to only contain counties that are located in Virginia, then sub selecting the columns that contain the relevant data. This process was fairly painstaking due to the fact that all the census data was organized differently than previous and future years, meaning that variables had to be renamed and reinterpreted for each year. Once this data was collected, it was organized into counties by election year, i.e. a row for each Virginia county for the years 2008, 2012, and 2016, with columns showing the variables of each of those values for each county and year. FRED data was compiled by finding the value of the selected metrics for January the first of each election year, and adding it to the respective row according to the year.

Results

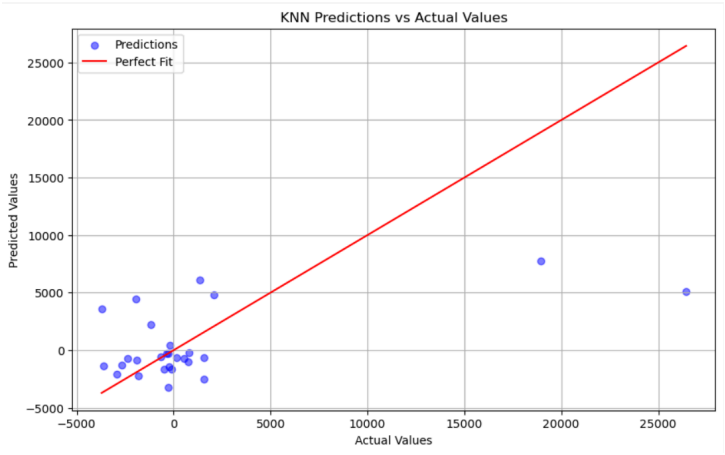
To develop a preliminary understanding of some of the important features in our cleaned dataset we divided up our cleaned dataset into three separate files, representing the 2008, 2012, and 2016 election cycles. A KNN algorithm is used to predict outcomes based on several features, which include Income to Poverty Level Ratio, Median House Value, Per Capita Income vs Pop, and GED Ratio. These features were chosen because they were considered to have some predictive power regarding the target variable, which is "Vote Skew" as our predictor for the 2024 election. The k value for the kNN algorithm was set to 20, meaning that the model considers the 20 nearest neighbors to make a prediction, and the Euclidean

distance metric has been used to calculate the distance between data points. This specific k value was chosen due to its more accurate representation of the strength of the input variables against our target variable through all election cycle years. Each year's dataset is also split into a training set and testing set to evaluate the accuracy of the model.

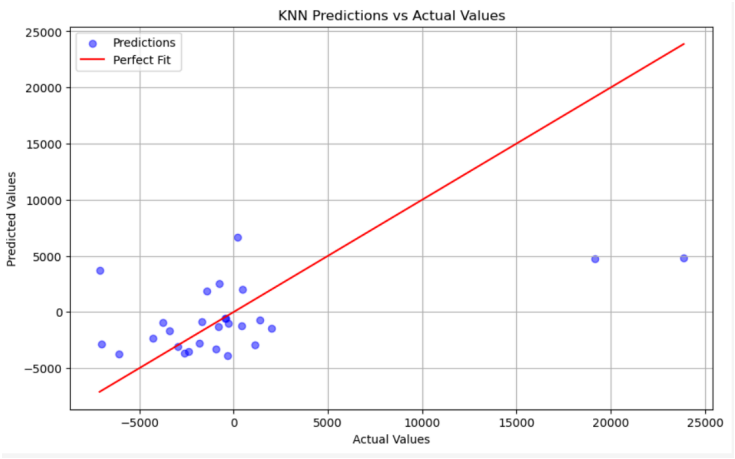
Our K-Nearest Neighbors regression model yielded an R-squared value of ~0.3 for the 2008, 2012, and 2016 election cycles. This suggests that the model explains around 30% of the variability for Vote Skew, which provides meaningful insight into the relationships between the selected socioeconomic indicators and voting patterns given the complexity of the factors influencing voting behavior.

The images below represent visualizations of the performance of the KNN regression model for all three election cycles. The x-axis represents the actual values from the dataset, while the y-axis shows the predicted values generated by the KNN model. The blue dots on the graphs represent individual predictions, with their position indicating the relationships between the predicted and actual values. The red line represents the line of perfect fit, where the predicted values would exactly match the actual values.

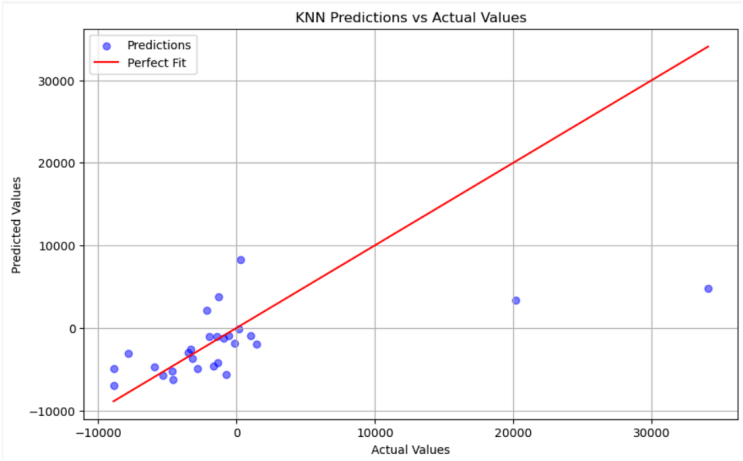
2008 Election:



2012 Election:



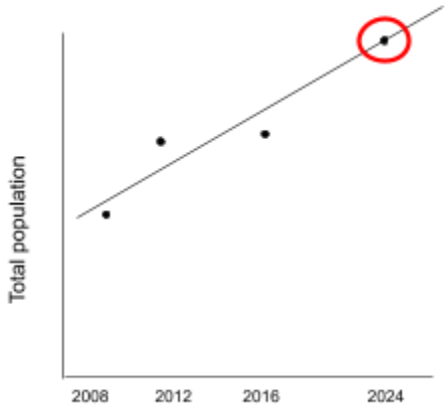
2016 Election:



From the scatter plots, we can observe that while there is a general trend that while the predictions follow the line of perfect fit, there are also points that deviate from this line, indicating errors in the predictions. The spread of the blue dots around the line of perfect fit suggests that the model has varying degrees of accuracy across different data points.

While our KNN regression model demonstrates a certain level of predictive power with an R-squared value of around 0.3, it is important to note that this model cannot be employed to make predictions for the 2024 election cycle without the input data. Therefore, it is important to highlight that the use of KNN regression is mainly to do a preliminary analysis highlighting some important features that should be examined in different models.

Now, to predict the results of the 2024 election, our initial challenge was to create the "test" set using our available data. It's not exactly a test set because we don't have the outcome to evaluate the results against. But once we train our model on the demographic and economic metrics that are related to the 2008-2016 election, we need to pass the model metrics relating to the 2024 election to make predictions. And we don't have the 2024 census data. So, to predict census metrics for 2024, we used linear models. The graphic below shows how it would predict 2024 total population based on the values for the other years.



Predicting 2024 values to create a test set

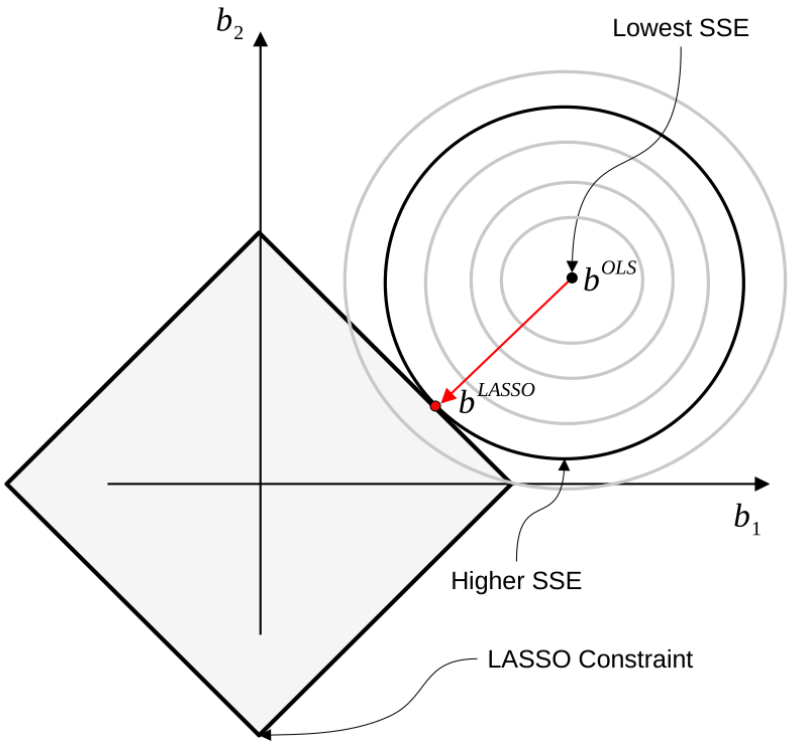
To predict these 2024 values across each of the counties and for each of the census metrics, we used a nested loop. The outer loop ran through the five columns (total population, GED ratio, etc) and the inner loop ran through each of the rows grouping them by County. All of the values were saved to a dataframe which will be used as a test set to create predictions.

In addition to the census columns, we used economic indicators in the analysis. These indicators, such as Fed interest rates, are nationwide so there's no variation between the counties. So, for 2024, the current interest rate and other metrics were used.

Finally, because the counties have a lot of variation in size and vote counts, we wanted to apply a fixed effects model. To do this, we created dummy variables for each of the counties. This means that if a

specific county varies significantly from the mean size and therefore has a larger/smaller vote skew (Dem votes - Rep votes), this effect will be captured by the county-specific dummy variable's coefficient and the coefficients on the census/FRED metrics will not try to predict county size. To further address this issue of variation in the outcome, we took the inverse hyperbolic sine of Vote Skew.

Now we had both the training and test set prepared to run. The next step was to train the model. We decided to use a Lasso-Linear model. Lasso is a powerful canned model that allows us to give a linear model many parameters without facing the issue of overfitting and multicollinearity. Lasso applies a budget constraint to linear models, where the sum of the absolute value of the OLS coefficients cannot exceed a certain value. It uses that value as a hyperparameter to cross-validate and find the constraint that achieves the optimal model complexity. This often leads to many variables used in the prediction to be zeroed out, or not mathematically picked to most effectively reduce SSE.



Visualizing the Lasso constraint

To ensure the Lasso constraint does not bias towards parameters that usually take large values and therefore have small OLS coefficients, the regressors need to be scaled. Sklearn standard scaler is applied to X_train and X_test.

The model, as expected, zeros out many variables. This includes several of the census metrics and all of the economic indicators. It chooses to include Total Population, Income to Poverty Level Ratio, GED Ratio, and some of the fixed effects dummies. It would be interesting to play around with this model more. The alpha value can be adjusted to adjust the aggressiveness of the budget constraint. It could be that economic factors are not great predictors of Democrat vs Republican votes, or that the budget constraint was very aggressive causing them all to be zeroed out.

Then we applied the model to the 2024 predicted data. After it generated vote skew predictions, we needed to undo the inverse hyperbolic sine transformation. After doing so and summing the resulting array, the model predicted that the Vote Skew would be -32000. This suggests that in 2024, the Virginia Republican votes would exceed Democrat votes by 32000 overall.

Conclusion

While our model predicted that the Republican candidate would win Virginia in 2024, there is a very high degree of uncertainty to this prediction. The way the model was built, there were multiple layers of predictive uncertainty. First, we had to predict what the census metrics for Virginia would look like in 2024 to have something to plug into the model. Each of these five metrics, which we predicted with linear models based on the previous three years of data, have different variances and are each calculated with limited data. Then, we used those results to plug into the election prediction model and get a number for the net votes. So, to quantify the variance in the prediction as a whole, there would need to be either a bootstrap framework or stats calculation that combines the uncertainty of both of these rounds of predictions.

Even then, by achieving a number for variance or testing multiple iterations of a modeled election to give a guess at the probability of a certain winner, it still does not consider the complexity of the event being modeled. Elections are particularly vulnerable to “shock” type events, like geopolitical factors or personal scandals of candidates. Those both seem particularly relevant for this upcoming election. How do you model those events in advance? Or, if you decide you can't, how does that factor into the degree of uncertainty you assign to model predictions? This is why Nate Silver gets humiliated every four years.

In terms of changes we could implement to our model in the future, we have several ideas. Firstly, it would be helpful to have a greater number of elections to work with. Including data from the 2020 election would be particularly relevant given the likelihood of a Trump-Biden rematch. It also would be interesting to play around with the effects of our regressors on non-presidential elections for which county-wide data exists (Congressional elections, State-level elections, etc.). Second, our LASSO regression zeroed out many variables that we thought would be key election outcomes. It would be interesting to examine why the cross validation it ran selected a low budget constraint for the coefficients. We could play around with the alpha value and see which other variables pick up nonzero coefficients. Finally, it would be interesting to add in a dummy variable for incumbency. Given the model does not predict votes for democrats and republicans separately, and rather predicts the net vote, I wonder if it would make sense to have that dummy variable take the values of 1, 0, -1. This would correspond to incumbent democrat, no incumbent, and incumbent republican.