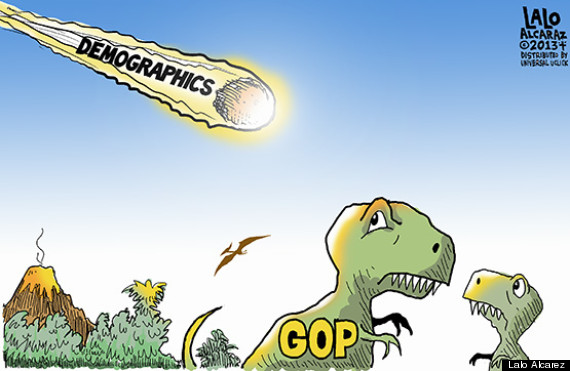
Capstone I

Data Science Career Track



Ryne Schultz

**PART I: DATA WRANGLING**

**Problem Statement**

Over the past 50 years demographic and socioeconomic trends have drastically changed the United States – from a largely homogenous and middle-class nation, to a more diverse and more economically stratified society.

This has driven commensurate political change, as American politics has become more polarized. These changes have upended the traditional political coalitions and has made political predictions more difficult.

The purpose of this paper is to utilize demographic and socioeconomic data to make congressional district party affiliation predictions using a logistic regression model. The goal is to create a model that can incorporate district level demographic data in order to make accurate predictions about election outcomes.

This question is of particular importance to political campaigns. Each political party has a fixed amount of capital (i.e. monetary resources, volunteers, local party infrastructure, etc.) and therefore must make important strategic decisions about how to allocate resources to optimize election outcomes.

Our model strives to formulate a bottoms-up approach to this prediction problem by analyzing underlying demographic changes in order to make political predictions. Our model will helo determine, before a single vote is cast, the likely outcome of a given district race, entirely predicated on that district’s demographic fundamentals.

Using our model, parties can make predictions about how likely a given district will vote Democrat or Republican and make strategic decisions about party resource allocation. This will help political parties avoid wasting resources on districts that are already highly likely to vote for one party or another, instead allocating resources toward swing districts.

**Data Description**

The Census Bureau’s “My Congressional District” tool lets you browse (and download) demographic, socioeconomic, and business data corresponding to each of the country’s 435 congressional districts (plus D.C.).

Political scientist Ella Foster-Molina has compiled a historical dataset containing similar information for the period from 1972 to 2014; it also contains details about each district’s representatives – such as their personal characteristics, the committees they served on, and the number of bills they sponsored.

**Removing Irrelevant Data**

The first step in our analysis is to remove unnecessary features from the dataset. This will help to prevent overcomplication, potential inaccuracies, and will improve the overall processing speed of the model.

Since the goal of our model is to predict district party affiliation from demographic and socioeconomic data, we removed all non-demographic and non-socioeconomic features. After removing these irrelevant features, we are left with the following list of variables that will help us train and develop our model:

* **recentArrivalPrcnt** – percent of the district that recently moved into the district from another county or state. This is a mild approximation, as a county can be in the same district as another, but the census does not track whether people recently moved into a district.
* **totalPopBirthPlace** – total population used for measuring mobility and/or nativity.
* **prcntForiegnBorn** – percent of the district that was born in a foreign country.
* **under10k** – percent of the district’s households that made less than $10,000/year in 2009 dollars. For all of our socioeconomic features (**over10k**, **over15k**, **over25k**, etc.) the exact income changes before and after 2009 due to inflation and the census using different income buckets for reporting income prior to 1990. It is always as close as possible to the 2009 value listed.
* **over10k** – percent of the district’s households that made over $10,000/year.
* **over15k** – percent of the district’s households that made over $15,000/year.
* **over25k** – percent of the district’s households that made over $25,000/year.
* **over35k** – percent of the district’s households that made over $35,000/year.
* **over50k** – percent of the district’s households that made over $50,000/year.
* **over75k** – percent of the district’s households that made over $75,000/year.
* **over100k** – percent of the district’s households that made over $100,000/year.
* **over150k** – percent of the district’s households that made over $150,000/year.
* **over200k** – percent of the district’s households that made over $200,000/year.
* **prcntUnemp** – percent of the district’s population that is unemployed but still in the labor force.
* **prcntBA** – percent of the district with a bachelor’s degree or higher.
* **prcntHS** – percent of the district with a high school degree or higher.
* **prcntAsian** – percent of the district that is Asian.
* **prcntBlack** – percent of the district that is black, including black Hispanics.
* **prcntHisp** – percent of the district that is Hispanic, both white and black.
* **prcntWhiteAll** – percent of the district that is white, including whit Hispanics.
* **gini** – inequality index estimated from the percent of the population in each income bracket.

**Dealing with Missing Values**

The next step in our analysis is to deal with any missing values contained in those features we chose to keep in our model.

Fortunately, once we removed irrelevant features from our dataset, there were very few rows that had any missing values. Those rows that did have missing data were simply removed from the data. As a result, the number of rows only declined from 9,312 to 9,298.

**Processing Our Dependent Variable**

To reiterate, the goal of our analysis is to predict – given the fundamental features of the district in question – whether that district will vote Democrat or Republican. We intend to do this using a logistic regression model, a subtype of classification model.

In logistic regression, the probabilities describing the possible outcomes of a single trial are modeled using the log-odds (i.e. the logarithm of the odds) for values labeled "1", using a linear combination of one or more independent variables (i.e. our independent variables). Logistic regression uses a common "S" (or “sigmoid”) curve to model a binary dependent variable. The resulting probability of the value labeled "1" can vary between 0 and 1, with the model rounding these values to determine categorization.

As such, our dependent variable must be converted to a numerical binary – represented by either a 0 (for Republican) or 1 (for Democrat).

We can accomplish this by applying one-hot encoding, a process by which we convert text-based features into numerical, binary values – represented as “1” or “0” – “True” or “False”.

When we apply one-hot encoding to the **party** feature, we end up splitting our original feature into four new ones – **party\_Democrat, party\_Republican, party\_Independent**, and **party\_Republican-Conservative**.

The variable **party\_Indpendent** represent such a small proportion of the overall observations that we can simply remove rows containing this value.

Rows containing **party\_Republican** and **party\_Republican-Conservative** can easily be combined, leaving only **party\_Democrat** and **party\_Republican**.

Finally, since **party\_Democrat** contains binary party affiliation information, we can remove **party\_Republican**, leaving us with one final, dependent variable: **party\_Democrat**.

Our new **party\_Democrat** variable takes one of two values – 0 or 1, whereby 0 signifies Republican and 1 signifies Democrat.

**PART II: Exploratory Data Analysis**

**Graphical Analysis**

The data is organized by Congress (specifically, the 93rd through 113th Congresses) so it makes sense to visualize the data as a time-series plot, with Congressional Number on the x-axis and the various independent variables on the y-axis.

A time-series plot allows us to see demographic and socioeconomic differences between Democratic and Republican districts, plotted over time.

Those variables that have a persistent difference between Democratic and Republican figures, over time, are likely to be the most significant variables in explaining the variation in party outcomes.

**Initial Findings**

When we look at the timer-series visualizations of our features it appears that there are large, national changes that affect Democratic and Republican districts *similarly*.

For example, according to the graph plotting **recentArrivalPrcnt**, there has been a distinct reduction in the mobility of Americans. Again, the **recentArrivalPrcnt** variable tracks the percent of the district that recently moved into the district from another county or state.

It appears that between the 100th and 110th Congresses, there was a distinct and sudden reduction in the percentage of the district populations that had recently moved, suggesting that fewer people are moving and that people are more geographically stable than they have been historically.

In addition, it appears that the percent of district populations born in a foreign country has increased in both Democratic and Republican districts – though this trend is significantly more pronounced in Democratic districts.

Unemployment rates, High School graduation rates, and higher education completion rates have also increased in both Democratic and Republican districts.

Perhaps most troublingly, it appears that inequality has increased in both Democratic and Republican districts (as measured by the gini coefficient, whereby a larger gini value signifies a more unequal district).

Of course, we are more interested in those features that *differentiate* Democratic and Republican districts, as these differences allow our model to separate Democratic and Republican districts more effectively.

For example, though many of these national trends are represented in both Democratic and Republican districts, we also see many of these trends are *more* pronounced in one type of district versus another.

For instance, though mobility has declined in both Democratic and Republican districts, it has declined *less* in Republican districts; and though the percent of the district populations that were born in a foreign country has also increased in both Democratic and Republican districts, this trend is dramatically *more* pronounced in Democratic districts, and has actually *declined* in recent years amongst Republican districts.

There also appears to be a distinct gap between Republican and Democratic districts when it comes to the percent of households earning more than $35,000. Indeed, there appears to be a large and persistent inequality – with a greater percentage of Republican district households earning more than $35,000.

Though graduation rates have increased for both High School and all forms of higher education in both Democratic and Republican districts, Republican districts haves consistently higher rates of High School graduation, while Democratic districts have only recently outpaced Republican districts in terms of high education graduation rates – a reversal of a long-established trend.

The greatest divergences appear, however, when we compare Democratic and Republican districts by racial composition. Democratic districts have consistently higher concentrations of Asian, Black, and Hispanic Americans, and both Democratic and Republican districts are becoming less white overall.

In sum, the trends point toward a more static, highly educated, financially unequal, and diverse nation. However, these trends appear to be far more pronounced in Democratic districts than in Republican ones.

Therefore, it is our initial contention that as we look to predict a district's party affiliation, we will see that more static, highly educated, and diverse districts will vote Democratic, while more mobile, less educated, and more homogenous districts will vote Republican.

**Variable Significance**

The next step in our analysis is to test our selected features for significance. In determining significance, the logit function is the appropriate test as we will be using a logistic function to classify our districts.

An initial test for significance suggested that **over75k**, **over150k**, **over200k**, and **prcntUnemp** are all statistically insignificant in explaining the variation in our dependent variable (i.e**. party\_Democrat**). Fortunately, after removing these features from the data, the rest of the variables remain significant.

Thus, of the original 21 features tested, only 16 were determined to be statistically significant (determined by a p-value less than 0.05):

* **recentArrivalPrcnt**
* **totalPopBirthPlace**
* **prcntForeignBorn**
* **under10k**
* **over10k**
* **over15k**
* **over35k**
* **over50k**
* **over100k**
* **prcntBA**
* **prcntHS**
* **prcntAsian**
* **prcntBlack**
* **prcntHisp**
* **prcntWhiteAll**
* **gini**

**Feature Correlation**

When thinking about correlations between the independent and dependent variables we have to keep in mind that correlation is simply normalized covariation, and covariation measures how two random variables co-variate – i.e. how changes in one variable are related to changes in another.

Features that are highly correlated will hinder the accuracy of our model as our model will find it difficult to discern the origination of an effect on the dependent variable. In other words, our model will not be able to decide which of two highly correlated independent variables is contributing to the variation in our dependent variable.

When we look at cross-correlation figures in our dataset we do see some strong correlations. For example, there is a high correlation between **prcntForeignBorn** and **prcntHisp**, which is not surprising, given that much of the U.S. Hispanic population are relatively recent arrivals.

Additionally, there is a great deal of correlation amongst our various socioeconomic variables (i.e. **under10k**, **over10k**, **over15k**, etc.). In fact, given these high correlations it makes the most sense to – at the very least – remove either **under10k** or **over10k** since they are simply the inverse of one another.

**PART III: Running the MODEL**

**Prepping the Data**

The first step in running or model is to prep our data by establishing our X and y values – our independent and dependent variables in the model. The dependent variable (y) is our column **party\_Democrat**. Our independent variables are the 15 relevant and significant variables we determined during our exploratory data analysis. After establishing our X and y variables, we then proceed to split the data into a training set and test set (using a 70/30 split), shuffling the data.

**Tuning the Hyperparameters**

When we train a model such as a logistic regression, we must choose hyperparameters that give us the best fit to the training data. This means minimizing the error between our model’s predicted values and the actual values contained in our test data.

Problems arise when we have a lot of independent variables but not a lot of data. In such cases, the model will often tailor the hyperparameter values to idiosyncrasies in the training data. However, because those idiosyncrasies are, by definition, particular to the training data, they do not appear in testing data, and the model ends up being a poor predictor.

To control for this, we minimize a function that penalizes large values of the hyperparameters, using a coefficient called lambda. The larger the lambda value the less likely it is that the hyperparameters will be increased in magnitude simply to adjust for small variations in the data.

In the case of the logistic regression, rather than specifying lambda, we specify *C*, which is equal to 1/lambda, or the inverse of the regularization strength.

In scikit-learn *C* is defaulted to 1, signifying that it has no minimizing effect on the underlying function. What we want to do is select a value of *C* that improves the accuracy of our model – either by increasing *C* (i.e. setting *C* to some number greater than 1) or by minimizing *C* (i.e. by setting *C* to some number less than 1).

Toward that end, we will use a variation of the logistic regression that incorporates cross-validation capabilities to automatically select the best *C* value. Operationally, the model tests a grid of *C* values, set by default to be ten values in a logarithmic scale between 1e-4 and 1e4, with the best *C* value being selected using cross-validation.

*C* isn’t the only hyperparameter to tune in our model – though it is, arguably, the most important. Another important hyperparameter is determining the number of folds to use when cross-validating our model. In the current version of scikit-learn, the default number of folds is 3, but the industry standard – and the default in future versions of scikit-learn – is 5, so bowing to industry standard we manually set our *cv* value to 5.

The next hyperparameter to tune is *class\_weight*. *class\_weight* contains the weights associated with the classes of our dependent variable. If not specified, all classes are given the equal weight of 1. The “balanced” mode uses the values of y to automatically adjust weights inversely proportional to class frequencies in the input data.

When we look at the incidence of Democratic and Republican districts in our data, we see that Democratic districts outnumber Republican districts, with Democratic districts making up ~55% of the total observations. By setting *class\_weight* to “balanced” we correct for this imbalance when running our model and so remove a potential source of bias.

Finally, we must determine which *solver* to implement, i.e. the optimization algorithm to use in order to minimize the difference between our model’s predicted and actual dependent variable values.

Scikit-learn provides us 5 solvers to choose from: “liblinear”, “newton-cg”, “lbfgs”, “sag” and “saga”. Each algorithm has its respective strengths and weaknesses.

For small datasets, “liblinear” is a good choice, whereas “sag” and “saga” are faster for larger data sets. For multiclass problems, only “newton-cg”, “sag”, “saga” and “lbfgs” are able to handle multinomial loss; “liblinear” is limited to one-versus-rest schemes. “newton-cg”, “lbfgs” and “sag” are only able to handle L2 penalty, whereas “liblinear” and “saga” can handle both L2 and L1.

Perhaps the most important aspect to consider when choosing a solver is determining the available error penalties – i.e. L1 or L2. These penalties are two different methods of accounting for error in our model. A regression model that uses the L1 penalty is using Lasso Regression; a model using an L2 penalty is using Ridge Regression.

The key difference here is the penalty term. Lasso Regression (which stands for Least Absolute Shrinkage and Selection Operator) adds the absolute value magnitude of the coefficient as a penalty term to the loss function, whereas Ridge regression adds the squared magnitude of the coefficient as a penalty term to the loss function.

The main difference between these techniques is that Lasso Regression *shrinks* the less important features’ coefficients to zero – in effect removing those features altogether – and thus acts as an additional quasi-feature-selection step.

Since our dataset is not particularly large (less than 10,000 observations), the benefits of “sag” and “saga” are somewhat diminished, and the main differences between the remaining three solvers are in their available penalty combinations, with “liblinear” having both L1 and L2 penalties available, and “lbfgs” and “newton-cg” allowing only L2.

This suggests that liblinear (using either penalty method) may be our best bet. However, instead of trying to pick winners, we can instead test all of the available combinations of solvers and penalties to determine the best overall solver/penalty combination.

After running all 5 solvers, we arrive at the following cross-validated accuracy scores (with standard deviations provided in parentheses):

* liblinear (L1): 0.60 (+/- 0.11)
* liblinear (L2): 0.63 (+/- 0.07)
* lbfgs: 0.57 (+/- 0.08)
* newton-cg: 0.61 (+/- 0.11)
* sag: 0.55 (+/- 0.00)
* saga (L1): 0.55 (+/- 0.00)
* saga (L2): 0.55 (+/- 0.00)

One clear winner stands out: “liblinear (L2)”. This solver/penalty combination both maximizes its mean accuracy score and minimizes its standard deviation.

The “liblinear” solver implements a logistic regression model using a coordinate descent algorithm. Coordinate descent successively minimizes along coordinate directions to find the minimum of a function. The intuition behind this method is based on the idea that the minimization of a multivariate function can be achieved by minimizing it along one direction at a time, i.e., solving univariate (or at least much simpler) optimization problems in a loop.

**PART IV: Analyzing the results**

**Analyzing District Probabilities Over Time**

Now that we have selected our model’s hyperparameters and successfully ran our model on the test data, we can now analyze our results.

The first step in our analysis to look at the predicted probabilities for specific districts over time. For example, let us consider Alabama's First and Seventh Congressional Districts. Alabama’s First Congressional District incorporates southern Alabama – including the greater Mobile and Pensacola areas.

A close up of a map

Description automatically generated

Alabama’s Seventh Congressional District incorporates parts of Birmingham, Montgomery, and Selma – areas that comprise the heart of Alabama’s “Black Belt”.

A close up of a map

Description automatically generated

Our model capable of picking up on these distinctions and trends and successfully categorized Alabama’s First Congressional District as Republican and Alabama’s Seventh Congressional District as Democratic.

**Plotting the Probability Distribution**

One of the main benefits of using a Logistic Regression for categorization problems is that – unlike other categorization methods – logistic regression provides a level of confidence in its prediction in the form of a probability.

This probability denotes the probability of a given observation being categorized as a 1, given that observation’s independent variable values.

A strong model would likely be bi-modal – where most of the predicted values occupy either an area close to 1 (i.e. that the model is certain that a given observation should be categorized as 1) or an area close to 0 (i.e. with the model certain that a given observation should not be categorized as 1). A poor model’s distribution would be concentrated around the middle – i.e. around 50% – signifying a less accurate model with a lot of residual uncertainty.

A close up of text on a white background

Description automatically generated

Sure enough, when we plot the distribution, we see a large congregation of predicted probabilities around the center, largely around 40%. There is a small, put perceptible peak closer to 1 (mostly around 85%), but this peak is noticeably smaller than the peak at the center of the distribution.

This distribution tracks well with our overall model accuracy score. We already knew that the model was only marginally better than chance. If the cross-validated accuracy score of the model had been closer to 0.7 or 0.8 we would expect a decidedly more bi-modal distribution.

**Using K-Means to Categorize District Probabilities**

Now that we have a distribution of probabilities, we can analyze this distribution at a more granular level by applying yet another categorization model to the probabilities – K-means clustering.

K-means clustering is one of the simplest and most popular unsupervised machine learning algorithms in use today. Unsupervised algorithms – in contrast to supervised algorithms like Logistic Regression – make inferences from datasets using only input vectors without referring to known, or labelled, outcomes.

The objective of K-means is straightforward: group similar data points together and discover underlying patterns. To achieve this objective, K-means looks for a fixed number of clusters in a dataset – allocating each data point to the nearest cluster, while at the same time keeping the centroids (the center of the cluster) as small as possible.

Initially, these centroids are randomly assigned. However, the algorithm performs iterative calculations to optimize the positions of the centroids – seeking to continually reduce the in-cluster sum of squares.

The algorithm repeats these steps until either the centroids have stabilized (i.e.  there is no change in their values because the clustering has been successful) or a predetermined number of iterations have been completed.

We can plot the in-cluster sum of squares against k to determine the optimal k value to apply to our K-means clustering model – a graph known as a Scree Plot.

The optimal K value is determined by finding the “elbow” of the graph – i.e. the point or region in the graph where the distortion (i.e. the in-cluster sum of squares) initially falls precipitously and then begins to flatten as K increases (see the below graph).

The “elbow” of our graph appears to be around three, meaning the optimal K value is 3 clusters. This makes intuitive sense since, when we consider our distribution of predicted probabilities, we have three main categories of probability – solidly Republican (signified by those probabilities nearest to 0), solidly Democratic (signified by those probabilities nearest to 1), and swing (signified by those probabilities nearest to 0.5).

A picture containing text

Description automatically generated

Therefore, after setting K to 3 and fitting the K-means model to our predicted probabilities, we can now assign each observation a cluster value, signifying solidly Republican districts, solidly Democratic districts, and swing districts.

Most campaigns are exclusively interested in swing districts, as districts solidly for or against a particular party aree already baked in, so to speak. Therefore, swing districts are battleground districts, where either party has a chance of winning.

**Analyzing the Clusters**

Now that we have successfully clustered our districts, we can analyze each cluster individually.

Our solidly Democratic cluster has a mean **Democratic\_proba** value of 0.794939 (+/- 0.072610), a minimum predicted probability of 0.681358, a maximum predicted probability of 0.956188, and comprises ~13% of our predicted probabilities.

Our solidly Republican cluster has a mean **Democratic\_proba** value of 0.382546 (+/- 0.049380), a minimum predicted probability of 0.256615, a maximum predicted probability of 0.475021, and comprises ~53% of our predicted probabilities.

Finally, our swing cluster has a mean **Democratic\_proba** value of 0.567847 (+/- 0.058248), a minimum predicted probability of 0.475205, a maximum predicted probability of 0.680796, and comprises ~34% of our predicted probabilities.

When we analyze our clusters by unique district, we find the following breakdown:

* Number of Democratic Districts: 63
* Number of Swing Districts: 130
* Number of Republican Districts: 243

It appears that, given the demographic makeup of all 436 congressional districts in the 113th Congress, ~56% are solidly Republican, ~30% are swing, and only ~14% are solidly Democratic. Contrary to popular belief, it seems that Republicans have a distinct demographic advantage. We will analyze this further in our conclusion.

**Analyzing Swing District Trends**

Given our ability to analyze probability trends we can now analyze swing district specific trends. Though the Republicans appeared to have a clear advantage when we analyzed the clusters, perhaps swing district trends show a different story.

To definitively answer this question, we split the swing district probabilities into one of two buckets – Democratic Trending or Republican Trending – based on the historical trend lines of each swing district’s probabilities, just like we did when we analyzed Alabama’s First and Seventh Congressional Districts.

We split the data by designating those swing districts with a positively sloped trend line (i.e. a slope greater than 0) as Democratic, since these districts are trending more Democratic over time. Inversely, we designated those swing districts with a negatively sloped trend line (i.e. a slope less than or equal to 0) as Republican, since these districts are trending more Republican over time.

After performing this split process, we arrive at the following results:

* Number of Democratic Trending Swing Districts: 40
* Number of Republican Trending Swing Districts: 90

If we then add these two figures to our solidly Democratic and solidly Republican figures we arrive at the following results:

* Total Democratic Districts: 103
* Total Republican Districts: 333

In fact, it does not seem that the swing district trends help the Democrats – in fact, the Democrats’ situation worsens. Republican districts rise from ~56% of the total to ~76%, while Democratic districts comprise only ~24% of the total.

**Analyzing National Trends**

Do we see these same trends on the national level? What if we were to subsume the entire country into a single congressional district and extend out national (i.e. average) trends into the future? Would national trends also point to a distinct Republican advantage?

To answer this question, we determined a lead time of 10 congresses (roughly 20 years into the future). We then looked at national average trends and carried them out to the 123rd Congress using a simple straight-line estimate.

Below are the predictions for the 123rd Congress:

* \***recentArrivalPrcent**: -4.016875172725804
* \***totalPopBirthPlace**: 84568.55738736317
* **prcntForeignBorn**: 21.31231173070823
* **over10k**: 97.6961015564552
* **over15k**: 92.82638209372242
* **over25k**: 76.29581465431964
* **over35k**: 65.92055329314522
* **over50k**: 53.771402189957314
* **over75k**: 32.922604798984565
* **over100k**: 21.263867918093588
* **over150k**: 11.042569854562036
* **over200k**: 6.528763957748673
* **prcntUnemp**: 11.579242459099124
* **prcntBA**: 41.29149007652366
* \***prcntHS**: 111.3718880101539
* **prcntAsian**: 9.090032441052031
* \***prcntBlack**: 19.182377024539065
* **prcntHisp**: 24.314302591216702
* \***prcntWhiteAll**: 93.76057441422269
* \***gini**: 0.7916134009698002

There are some initial problems with this simplistc method, and we have flagged some problem predictions with an asterisk above.

For example, **recentArrivalPrcent** has a nonsensical negative value. It is simply impossible to have a negative percentage of a district’s population be a recent arrival.

Indeed, all of the other problem predictions suffer from the same issue: namely, in using the simple straight-line method we arrive at impossible or nonsensical values that we must correct. Therefore, in those cases where the simple straight-line method provides impossible or nonsensical values, we must apply another method.

In the case of **recentArrivalPrcent** it may be better to simply use a floor value of 5%. When we adjust the predictions for the other problem features, i.e. for **totalPopBirthPlace**, **prcntHS, prcntBlack, prcntWhiteAll,** and **gini** we arrive at the following, corrected predictions:

* **recentArrivalPrcent**: 5.0
* **totalPopBirthPlace**: 750000
* **prcntForeignBorn**: 21.31231173070823
* **over10k**: 97.6961015564552
* **over15k**: 92.82638209372242
* **over25k**: 76.29581465431964
* **over35k**: 65.92055329314522
* **over50k**: 53.771402189957314
* **over75k**: 32.922604798984565
* **over100k**: 21.263867918093588
* **over150k**: 11.042569854562036
* **over200k**: 6.528763957748673
* **prcntUnemp**: 11.579242459099124
* **prcntBA**: 41.29149007652366
* **prcntHS**: 90.0
* **prcntAsian**: 9.090032441052031
* **prcntBlack**: 12.0
* **prcntHisp**: 24.314302591216702
* **prcntWhiteAll**: 52.59566496773127
* **gini**: 0.5

In the case of **recentArrivalPrcent** we set the prediction equal to 5%, assuming the average recent arrival percentage will stabilize.

For **totalPopBirthPlace** we set the prediction equal to 750,000, assuming the average total native-born population of a district stabilizes around 750,000 people.

It was obviously unreasonable to assume that high school graduation rates would exceed 100% so we set **prcntHS** equal to 90%, assuming the national average high school graduation rate will top out around the 90% level.

Both the national average percentages of black and white Americans appear to have topped out in the 105th and 100th Congresses, respectively, and so we assume both **prcntBlack** and **prcntWhiteAll** will either plateau or decline.

For **prcntBlack** we assume that the national average percentage black population will decline slightly and stabilize around 12%. For **prcntWhiteAll**, we simply backed into the predicted figure by subtracting the predicted Asian, Hispanic, and Black populations from 100%, arriving at a predicted national average percentage white population of ~52%.

Finally, for **gini**, we assumed that the near-term average of around 0.5 would hold going forward.

We then took these corrected predictions and plugged them into our original logistic regression model to determine a hypothetical, national predicted probability.

Finally, we plugged this predicted probability into our K-means model to assign the predicted probability to a cluster. The following are the results of this process:

* Probability Republican: 0.38957303290081235
* Probability Democratic: 0.6104269670991876
* Cluster Prediction: 1
* Prediction: Swing

It appears that our models predict that national demographic trends are moving the country toward a swing status.

To determine if the country is Democratic or Republican trending, we simply extended out our analysis to the 133rd Congress by changing our lead time to 20 from 10. The following are the results:

* Probability Republican: 0.31643051433774216
* Probability Democratic: 0.6835694856622578
* Cluster Prediction: 2
* Prediction: Solid Democratic

By extending our predictions further out into the future we can see a larger national trend – namely, a country that is becoming more Democratic.

**PART V: Conclusions**

These results are, initially, quite confusing: though national trends point to a more Democratic nation, the individual district level demographics give Republican a definitive advantage.

Of all 436 congressional districts in the 113th Congress, ~56% were solidly Republican, ~30% were swing, and only ~14% were solidly Democratic.

Additionally, when we looked at swing district trends this split only served to widen the gap between Republican and Democratic districts, with the total number of Democratic trending swing districts comprising only ~31% of the total.

And when we considered the breakdown of districts in total – comprising both solid and swing districts – the disparity only became more pronounced, with Republican districts rising from ~56% of the total to ~76%, while Democratic districts comprised only ~24% of the total.

However, when we look at national trends, we see an increasingly Democratic country. What’s going on here? Aren’t these trends mutually exclusive?

We can have national trends pointing towards a more diverse – and be extension more Democratic – nation, while at the same time having Congressional trends that continue to favor Republicans. How?

**Redistricting**

The U.S. Constitution mentions redistricting in Article 1 Section 4:

“The Times, Places and Manner of holding Elections for Senators and Representatives, shall be prescribed in each State by the Legislature thereof; but the Congress may at any time by Law make or alter such Regulations, except as to the Place of Chusing Senators.”

This section of Article 1 was written to determine the authority in charge of redistricting. This authority was given in part to state legislators but gives ultimate authority to Congress. However, Congress has never mandated a redistricting procedure and ultimately lets the states decide.

Because there are no explicit guidelines, it remains relatively unclear how voting power should be apportioned or specifically how representatives should be elected.

Republicans have exploited this ambiguity in recent years by concentrating their efforts on getting Republicans elected to state legislatures, where these legislators would then draw the district maps to favor Congressional Republican incumbents – a process known as “Gerrymandering”.

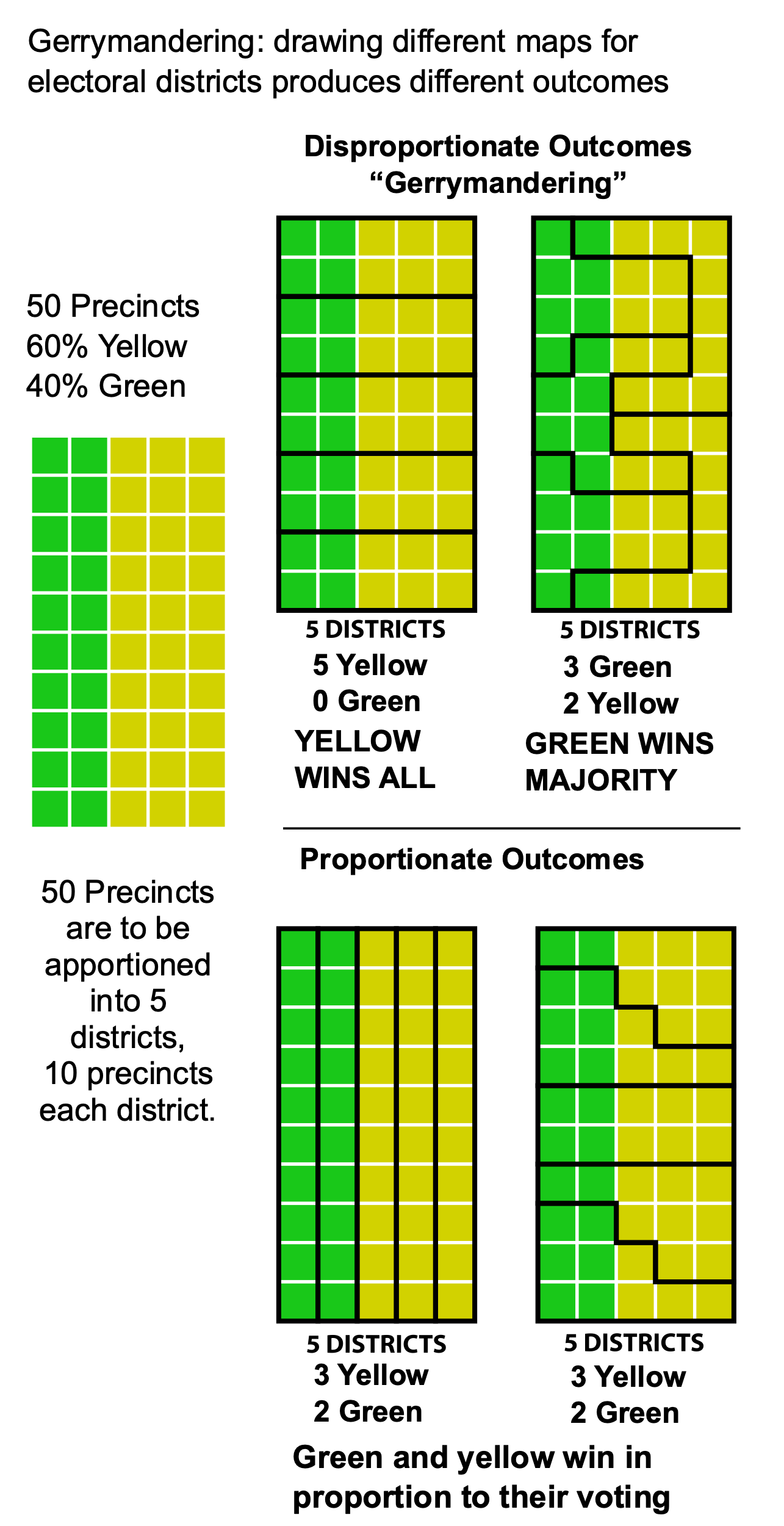
The result are district maps (and by extension district demographics) that do not reflect national realities. To see how, consider the visualization on page 28.

Gerrymandering is effective because of the wasted vote effect – i.e votes that do not contribute to electing a candidate, either because they are in excess of the bare minimum needed for victory or because the candidate lost.

By moving district boundaries, the incumbent party packs opposition voters into a few districts they are likely to win anyway, wasting those extra votes. Other districts are more tightly constructed with the opposition party allowed a bare minority count, thereby wasting all the minority votes for the losing candidate. These districts constitute the majority of districts and are drawn to produce a result favoring the incumbent party.

In 2010 – a redistricting year – the Republican State Leadership Committee formulated the aptly-named REDMAP strategy, short for Redistricting Majority Project. Using this plan, Republicans redrew district maps across the country and managed to hold the U.S. House in 2012, despite earning 1.4 million *fewer* votes than Democratic congressional candidates, and won large GOP majorities in the Ohio,

Wisconsin, Michigan and North Carolina state legislatures even when more voters backed Democrats[[1]](#footnote-1).



These same maps continued to provide Republicans and advantage in 2014 (the last Congressional period in our dataset) and 2018 – despite the blue wave that took place during that year’s midterm elections.

Our model picks up on these redistricting efforts as it simultaneously sees national trends move distinctly Democratic.

1. Daley, David. “How the Republicans Rigged Congress - New Documents Reveal an Untold Story.” Salon, Salon.com, 8 Feb. 2018, www.salon.com/2018/02/06/how-the-republicans-rigged-congress-and-poisoned-our-politics/. [↑](#footnote-ref-1)