Local Voters

An analysis of one district's Registered Voter data.
Then building a model to predict likely voters.

By Alison Kline (December 2018)

Introduction

As part of preparing for his wife's run for office, my friend has purchased the Registered Voter data for our city from the Registrar. He has kindly shared District 3's data with me. It consists of two data sets, one for every registered voter and one for every household. The registered voter data includes voting patterns for the last 3 full election cycles; November and June elections for 2012, 2014, and 2016.

I wanted to help him answer the following questions to inform an effective candidate marketing campaign. The key problem statements I'm targeting to help him solve are:

1. Which households should I target for phone calling and mailing?
    a. Who is likely to vote
    b. What is their party registration so I can tailor my message
2. Are there any gross voter trends that will help me with targeting:
    a. Do people in apartments vote as much as others?
    b. What characteristics do very frequent voters share?

To answer these questions I completed a statistical analysis of the data, identifying various characteristics that impact a voters vote rate. I then built a Random Forest supervised machine learning model, training and optimizing it before using it to predict likely voters for the June and November 2018 elections.

All data cleaning, analysis and modeling was completed in various iPython Notebooks available in the my github repository LocalVoters.

Summary of Findings

- Voters are 30% less likely to vote in primary elections than general elections.
- Older Voters are more likely to vote
- Voters with a party affiliation are 12.4% more likely to vote than unaffiliated voters.
- Voters with Democrat and Republican affiliations were more likely to have a higher vote rates than someone belonging to one of the minor parties or holding No Party Preference.
- Holding a permanent Absentee Ballot increases the chance of you always voting by ~20%
- No significant difference in Male and female vote rate
- Voters who live in apartments are less likely to vote than voters whose address does not include an apartment number (obs: ≈4.9%).
- If everyone in your household is affiliated with a political party you are 12% more likely to vote than if you don't live in such a household.
- Living in a household with people who are affiliated with another party meant you were 4.9% (SS at 99%) less likely to vote than if you lived in a household where all the voters in the household were affiliated with the same party.

A Random Forest Supervised learning model fully tuned with a Random Search 5-fold Cross Validation algorithm was able to reach 93% prediction accuracy on the labeled training data we had. The top predictive features were the two features engineered by my friend. When using the model for prediction it identified 4032 voters likely to cast votes in the 2018 primary and 5861 voter likely to cast votes in the 2018 general election. I was able to re-identify the people on these lists and pass them to my friend.

Data Cleaning and Wrangling

The data has come from the official voter registration office and so contained a significant amount of Personally Identifiable Information (PII). It also contains household level information as well as voter level information. A number of steps were taken to clean and process the data ready for further analysis and predictive modeling.

Voter Data: 13307 total rows (each row representing one voter) and 56 columns of data.

Household Data: 6930 total rows (each row representing one household) and 23 columns of data.

Data that was Dropped:

Columns that included no valuable information or was too sparsely populated were dropped. This included:

Status (all values were the same), LastVoted (only 6 entries), Salutation (6388 missing values), HouseNumberSuffix and StreetPrefix (empty), Building number, City & State (all the same), Military (only 9 entries)

Personally Identifiable Information (PII),

All PII was removed from the data so I could share my analysis freely. In come cases (as noted below) I used the full data to fill missing values in other columns or to extract usable summary information prior to dropping. Columns treated this way included:

Affidavit, Last name, First name, used with a name database to fill missing gender data before dropping. Middle name, & Suffix, House number, Street, used to create StreetType and then dropped, ApartmentNumber used to create a boolean 'isApt' field and then dropped. ImageID, Phone fields were converted into True/False fields, Birth date was cleaned into BirthYear and the full date dropped, email cleaned to service provider level only

Household and Voter relationship

The data had an identifier for all households (voters living at the same street address), I created my own unique ID for voters and households populated the data and then dropped the original keys to maintain anonymity of the data. I kept a record of the original key and a lookup table to enable final predictions to be linked back to the original data for use by my friend in his campaigning.

All columns were analyzed for missing values and consistent categories. Where necessary data was converted and 'UNK' or had a '-1' used for missing values.

Final Voter Data

Original Data Column	Description of action	output column(s)
'VoterID'	Rows of table were randomly shuffled, the index reset and the new index used as new UID.	'vid'
'Abbr'	Kept as is, although it's not understood it is a clean number.	'Abbr'
'StreetType'	Populate missing values using 'Street': 'Common' => 'CMN' 'GREEN' => 'GRN' and two cross streets => 'UNK'. Combine small categories together 'PL' and 'TER' => 'PL/TER', 'RD', 'LN', 'PKWY', 'LOOP', 'GRN' AND 'CIR' => 'OTH'	'StreetType'
'ApartmentNumber'	Converted to a True/False field.	'isApt'
'Zip'	Cleaned all to 5 digit numerical zip code entries.	'Zip'
'Precinct'	Converted to number and kept.	'Precinct'
'PrecinctSub'	Converted to number and kept.	'PrecinctSub'
'Party'	Converted to category and kept. Also created a 'PartyMain' field by combining all the small parties into an 'OTH' category	'Party', 'PartyMain'
'RegDate'	Converted to a dateTime and kept.	'RegDate'
'Phone1'	5266 NaN's 8041 values, converted to True/False.	'havePhone'
'Gender'	5223 NaN's 1743 'F' and 1717 'M' were added by comparing FirstName data to a database of name genders (https://github.com/organisciak/names), remaining missing data was set to 'UNK'.	'Gender'
'PAV'	Is voter a Permanent Absentee Voter, converted to category and kept.	'PAV'
'BirthPlace '	This mixed two and three letter code was assumed to be a two USA state code, and only if that failed to match assumed to be a two or three letter country code. Output was 2 clean columns with state and or country, plus 2 clean columns with State and Country Region information. State, Country and Region code data gathered from wikipedia, 'UNK' added for the 1296 NaN's.	'BirthPlaceState', 'BirthPlaceStateRegion', 'BirthPlaceCountry', 'BirthPlaceCountryRegion '
'BirthDate'	Cleaned full birthday into 'BirthYear', rest dropped as PII.	'BirthYear'
	Used BirthDate to calculate Oldest in Household Birth Year and if you are the oldest in your household	'OldestInHouseBirthYear', 'IsOldestInHouse'
Mailing Address columns	Compared with main address to create a True/False, Country kept as a category.	'sameMailAddress', 'MailCountry'
'email'	9009 NaN's, Cleaned to keep the service provider with UNK for NaNs.	'EmailProvider'
'RegDateOriginal'	Converted to a dateTime and kept.	'RegDateOriginal'
'District'	Kept as is in case we need to add in other district data.	'District'
'VoterScore'	Score assigned by my friend based on which election someone has reported data for and voted (A or V)	'VoterScore'
'VoterScorePossible'	Score assigned by my friend assuming all reported data was 'vote' (A or V)	'VoterScorePossible'
'VoterScorePctOfPoss'	'VoterScore'/'VoterScorePctOfPoss'	'VoterScorePctOfPoss'
Household	Unique key linking each voter to a household, looked up and converted to anonymized Hid.	'Hid', cHid

Full details in '05_Cleaning Voter' notebook

Final Household Data

Original Data Column	Description of action	output column(s)
'Household_Id'	Rows of table were randomly shuffled, the index reset and the new index used as new UID.	'hid', 'cHid'
'StreetType'	‘CMN’ ‘GREEN’ => ‘GRN’ and two cross streets => ‘UNK’. Also combined some smaller groups.	'StreetType'
'ApartmentNumber'	Converted to a True/False field.	'isApt'
'Zip'	Cleaned all to 5 digit numerical zip code entries.	'Zip'
'Precinct'	Converted to number and kept.	'Precinct'
	The Precinct was also used to create a 'CityArea' column	'CityArea'
'PrecinctSub'	Converted to number and kept.	'PrecinctSub'
'District'	Kept as is in case we need to add in other district data.	'District'

3 extra households were identified as having duplicate entries due to one or more members of the household entering their house number as an apartment number. the cHid field was created to clean this up - correctly combining these households and the 11 affected voters.

Full details in '05_Cleaning HouseHold' notebook

Voting behavior data

Data on Voting behavior for 6 prior elections was included in the data. A particular voter had an entry if they have been registered to vote for that election and that entry contained the following key:

Entry	Description
A	The voter voted using an Absentee Ballot
V	The voter voted in person
N	The voter didn't vote

y(xxx) : For some of the primary elections the voter status was communicated as y and the type of ballot used was captured by the xxx, A(REP) indicated a vote cast on an Absentee Republican ballot. In these cases the ballot types were extracted into their own fields and then different combinations of election vote data processed to enable vote rate analysis to be completed on the different election combinations. I also created a 'Ground Truth' column for each of the 6 elections where 11 indicated the voter voted (ie had a 'A' or 'V') and 00 indicated they didn't (ie had a 'N'), this was to facilitate training of our predictive model.

Original Data Column	Description of action	output column(s)
'E1_110816'	Code indicated vote, converted to category and kept.	'E6_110816', E6_GndTth
'E2_060716'	Code indicated vote and ballot used, Cleaned into 'Vote' and 'BallotType' and kept.	'E5_060716', 'E5_060716BT' ,E5_GndTth
'E3_110414'	Code indicated vote, converted to category and kept.	'E4_110414', E4_GndTth
'E4_060314'	Code indicated vote, converted to category and kept.	'E3_060314', E3_GndTth
'E5_110612'	Code indicated vote, converted to category and kept.	'E2_110612', E2_GndTth
'E6_060512'	Code indicated vote and ballot used, Cleaned into 'Vote' and 'BallotType' and kept.	'E1_060512','E1_060512BT', E1_GndTth

For each election or group of elections to be analyzed I calculated The total number of votes that could have been cast from voters in my data, the number of votes actually cast and the proportion of successes (ie cast votes).

Description of action	output column(s)
Column indicating number of elections voter has been registered for. Ie how many times they had an entry ('A','V', or 'N') in one of the election columns of interest	'_nVotesPos'
Column indicating number of elections voter actually voted in. Ie how many of their entries were 'A' or 'V'	'_nVotes'
Column indicating % of possible elections actually voted in. In _nVotesPos_nVotes_nVotesPos_nVotes	'_nVotesPct'

The groups of elections analyzed included:

Prefix used	Elections
'E6'	a General Presidential election held on Nov 8th 2016
'E5'	a Primary election held on Jun 7th 2016
'E4'	a General Congressional election held on Nov 4th 2014
'E3'	a Primary election held on Jun 3th 2014
'E2'	a General Presidential election held on Nov 6th 2012
'E1'	a Primary election held on Jun 5th 2012
'E12'	a combination of both elections held in 2012
'E14'	a combination of both elections held in 2014
'E16'	a combination of both elections held in 2016
'E34'	also a combination of both elections held in 2012
'E56'	a combination of both elections held in 2012 & both elections held in 2014
'E78'	a combination of all 6 elections
'Eap'	a combination of the 3 primary elections
'Eag'	a combination of the 3 general elections

Full details in '07_Vote Rates' notebook

Election general information

I also collected some general information about the environment at the time of each of these elections. Mostly from wikipedia:

Output column(s)	Description of data
'election'	Unique ID for each election held between 2012 and 2018, including this years target for predicting voting.
'dates'	Actual date of the election.
'cycle '	Is this a Congressional only election year or a Presidential election year.
'etype'	Is this a 'Primary' or 'General' election.
'president'	What is the party of the president in power at the time of the election.
'us_senate_maj'	How big is the controlling margin in the US Senate. Positive numbers indicate a REP maj, negative a DEM one.
'us_repre_maj'	How big is the controlling margin in the US House of Representatives. Positive for REP maj, negative for DEM.
'ca_governor'	Which party did the Governor of CA belong too at the election.
'ca_lt_govnor'	Which party did the Lieutenant Governor of CA belong too at the election.
'ca_senate_maj'	How big is the controlling margin in the CA Senate. Positive for REP maj, negative for DEM.
'ca_assembly_maj'	How big is the controlling margin in the CA Assembly. Positive for REP maj, negative for DEM.

Full details in '05_Election information' notebook

Finally some additional fields were calculated and added to the data set

Mainly relating to household level derived features from the individual voter data:

- Number of voters in HH,
- Number of PAV,
- Number with party affiliation
- Number of DEM party affiliation in HH
- Number of REP party affiliation in HH
- Number of NPP party affiliations in HH
- Party with most affiliations in HH
- Mixed affiliations True/False (all affiliated with same party)
- All voters affiliated
- Uniform affiliations (all same party or all NPP)

And at the Voter level

- Is Oldest in household

All these fields were made available in the household data and at the Voter level in the voter data

Full details in '09_Clean Data Features' notebook

Analysis of the Data

Our data covers one district, containing information about 13307 Voters and 6930 Households. It contains information about both primary and general elections held in 2012, 2014, 2016

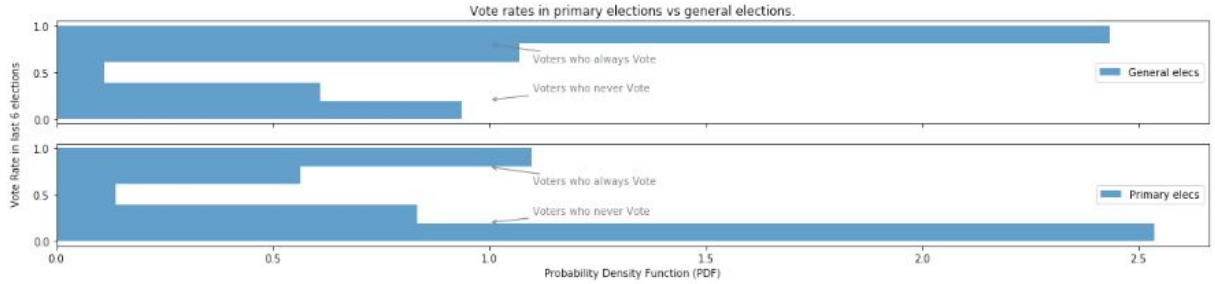
	Number of Votes Possible	Number of Votes Cast	Votes Percent cast/possible
E6_110816	12342	9220	0.747043
E5_060716	11101	4846	0.436537
E4_110414	9987	4314	0.431962
E3_060314	9727	2807	0.288578
E2_110612	9296	6937	0.746235
E1_060512	8577	2577	0.300455

An Initial analysis was completed of some of the high level commonly repeated 'true'isums about voter behavior to see how similar to these 'standards' our voters were.

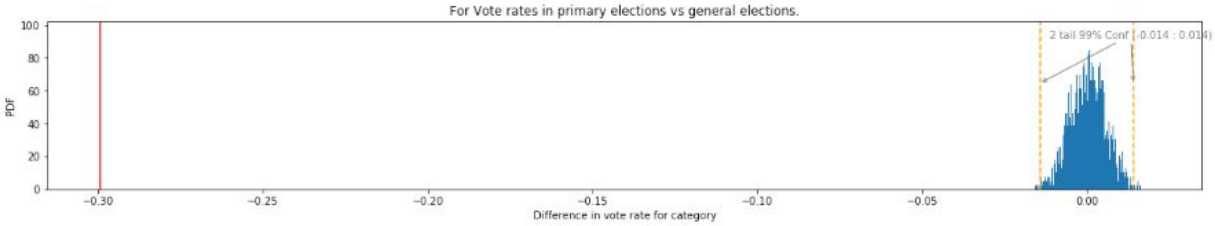
These included:

- People vote less in Primaries
- Older votes are more likely to vote

People vote less in Primaries



	Number of Voters		Voters as a %	
	General elecs	**Primary elecs**	**General elecs_pct**	**Primary elecs_pct**
Always	5725	2334	46.3	20.9
Over Half	2647	1260	21.4	11.3
Half	298	337	2.4	3.0
Under Half	1506	1862	12.2	16.7
Never	2200	5384	17.8	48.2
Totals	12376	11177	100.1	100.1

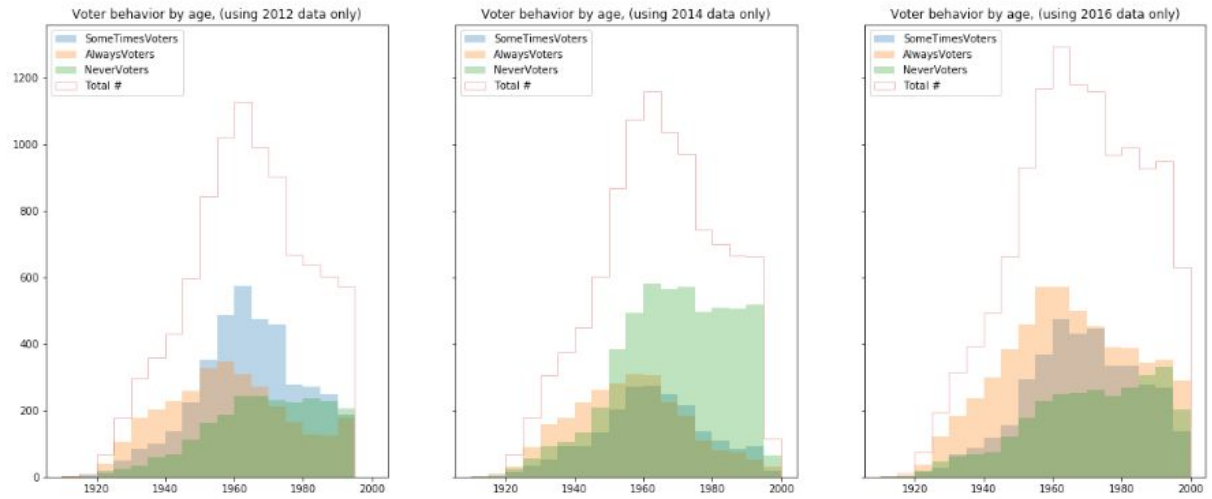


	votes_s0	elec_n0	rate_r0	votes_s1	elec_n1	rate_r1	emp_diff	perm_p
All primary vs general elections	10230	29405	34.79	20471	31625	64.7304	-29.9404	0

Our data confirmed that across the 3 primary elections and 3 general elections people were 30% less likely to vote in primary elections than general elections. Using a two sample permutation test we can conclude that this difference is statistically significant at a confidence level of 99%.

Older Voters are more likely to vote

There are 12 people over 100 (inc 7 people entering 1900 who were removed as likely bad data)



When I completed a 2 sample boot hypothesis test to identify if the mean age of the always vote and never vote groups could be zero and us still see the variation in group mean by chance I was able to confirm the following:

	Always Vote Mean birth year	Never Vote Mean birth year	Difference	estimated p-value
Year				
2012	1958	1968	10.10	0
2014	1956	1969	12.90	0
2016	1964	1971	6.29	0

You can see in these histograms that the Always Voters skew older (having birth years to the left of the graphs) and the Never Voters skew younger with birth years to the right of the graphs. This is a particularly strong effect in the Congressional year of 2014.

You can also see that the young are less likely to have registered to vote in the first place as the total voter histograms all show a peak around birth year 1960. It is also possible that this effect is caused not by younger voters not registering but by our district having fewer younger voters living here. It is also plausible that younger voters are more mobile and so even if they were here for the 2012 or 2014 vote they have since moved out of the district and so dropped out of our data set and you can't forget that there may just be more people born in the 1960's. We would need additional demographic data about our voting district to identify which of these hypothesis held true.

Other Observations

In the full '14_Milestone Report' notebook you can see a few additional data Observations and correlations explored. I found that voters with a party affiliation are 12.4% more likely to vote than unaffiliated voters. Voters with Democrat and Republican affiliations were statically more likely to have a higher vote rate than someone belonging to one of the minor parties or holding No Party Preference.

Having a permanent Absentee Ballot significantly increases the likelihood of your being in the always voter category. Our observations indicate ~20% greater chance of being an always voter.

Following statistical analysis we can conclude that the difference seen in male and female vote rate in the data is likely due to chance.

Voters who live in apartments are less likely to vote than voters whose address does not include an apartment number (obs: ≈4.9%).

If everyone in your household is affiliated with a political party you are 12% more likely to vote than if you don't live in such a household. This result was statistically significant at 99% confidence. This included single voter households.

It is likely that this is an effect caused related to the greater propensity for people affiliated with REP or DEM to vote. To understand the relationships here a little more I also looked at those living in multi voter households were they more or less likely to vote if everyone in the household was affiliated with the same party

Living in a household with people who are affiliated with another party meant you were 4.9% (SS at 99%) less likely to vote than if you lived in a household where all the voters in the household were affiliated with the same party. (NPP voters did not count as affiliated and single voter households were removed from this analysis)

Building a Predictive Model

Having cleaned and prepared my data in earlier related notebooks I now set out to create a model that can be used to identify the likely voting habits of voters and hence be used by my friend to target marketing and outreach efforts.

Needing a binary prediction output of either 'will' vote or 'won't' vote I chose to explore two different models, a random forest and a logistic regression. Following initial results I noted that the random forest seemed to be providing better results and used a grid search and visual optimization techniques to tune the model further.

Data Reshaping

The clean data is organized by voter with one voter and data on up to 6 elections per row. For modeling I wanted each row to relate to a unique election and for predicting future election we needed to extend the data to cover the 2018 elections.

I identified groups of columns, Data columns, and sets associated with each of the elections, including the Ground Truth of each voter. Reshaping to gather the data into one row per election that the user had voted in. I also joined in the election information data collected. At this stage I included information on the voter's and household last election vote rate, as well as their vote rate in the last full election cycle as features.

The final shape of the data for modeling was 106456 rows (one for each of our 13307 user for each of 8 elections) with 56 columns

Preparing Data for Model

I checked for category balance:

	Gnd truth value	**Count of values**
Vote cast in election	1	30701
No Vote was cast	0	30329
Data not available	-1	45426

I also checked for variance and found that 'ca_governor', 'ca_lt_govnor' didn't vary across the elections we are considering so I dropped those columns.

Prior to building a model we have to one-hot-encode all the category features. This is when you switch a single multi value column for multiple columns each one taking a 1 or 0 for of the multiple values in the original column. I used the get_dummies() method for this.

For training the model I removed the 45k rows of voters/election combination where we don't have data (including all E7 and E8 rows) - for predictions I left in just the rows for Election 7 (2018 primary) and Election 8 (2018 general) as those are the elections we are interested in predicting voter behavior for.

This left a training data set with shape 61030 rows and 530 features.

Initial Random Forest Model

As my data is looking for binary classification - will vote/won't vote I decided to try using a random forest supervised machine learning model first. These models are well known for performing well on many different data sets as they are easy to use and often produces a great result, sometimes without hyper-parameter tuning.

Given that my data doesn't have Last Election and Last Election Cycle information for the elections held in 2012, I'm going to build my models using the 4 elections we do have complete data for. The additional vote data could be procured if we were motivated and willing to make the additional investment so doesn't impact of the validity of the model. (I tested using all the election data and as expected our accuracy dropped).

When considering only training data from E3, E4, E5, & E6 the training data set had 43137 rows.

Creating a Random Forest and using a standard hold out test/train split of the data we were able to achieve 90.5% accuracy.

Looking at feature importance allowed us to confirm we were not seeing data leakage and an unexpectedly important single feature.

	importance
VScorePct	0.202490
nVScorePctInHH	0.134275
us_repre_maj	0.042719
lastCycleVoteRatehh	0.040229
lastElecVoteRate	0.039531
lastElecVoteRatehh	0.037942
etype_Primary	0.034404
BirthYear	0.028919
lastCycleVoteRate	0.027851
OldestInHouseBirthYear	0.026430

This model is looking good - there are a number of features each helping predict voter behavior and they are the features we would expect based on the statistical analysis completed earlier eg your birth year.

There are a couple of interesting points to note:

- The top features are the ones my friend engineered into the data, a weighted combination of which elections someone had voted in considering which they were available for.
- The us_repre_maj or size of the majority in the US house of Representatives is a surprisingly important feature in this particular set of data - this may or not hold broadly as the republican's power may be unusually motivating in this Californian left-leaning district.
- The next set of features are Vote Rates in the last election & cycle and the type of election
- As expected BirthYear and whether you are the Oldest in your household make up the rest of the top ten features.

I tried removing my friends engineered features and as you would expect given their importance the models accuracy dropped to 76%.

Tuning the Random Forest

Even though we have a nicely performing model I wanted to see how much better it would get with a bit of tuning. I completed a randomized grid search using 5 fold cross validation and 100 iterations on the key Random Forest parameters:

- n_estimators, the number of trees grown,
- max_features, the number of features considered when looking for the best split
- max_depth, max depth (aka number of splits) in each tree
- min_samples_split, the min number of samples required to split an internal node
- min_samples_leaf, the number of samples required to be at a leaf node

Random Forest Feature	Parameters explored during random grid search	Best Value
n_estimators	[4, 8, 16, 32, 64, 100, 250, 400, 550, 700]	400
max_features'	['sqrt', 'log2']	‘sqrt’
max_depth	[10, 20, 30, 40, 50, 60, None]	30
min_samples_split	[2, 5, 10]	5
min_samples_leaf	[1, 2, 4]	1
bootstrap	[True, False]	False

Creating a model using the best_parameters setting found from the random grid search and assessing its accuracy using the same standard hold out test/train split of the data used above we were achieved 93.5% accuracy giving the model’s performance a relatively modest 3% boost.

Logistic Regression Model

I somewhat briefly explored using a Logistic Regression Model and turning its ‘C’ parameter. With Logistic Regression we were getting an accuracy in the 60%-70% range so it made sense to refocus on the Random Forest and use it for predictions.

Making predictions using the model

Having identified a good model I want to use it on E7 and E8 data to predict who is most likely to vote in these two upcoming elections. For Predictions we use a model that has been fully trained on all the labelled data that we have given that our parameters have already been selected and our accuracy measured. In this case we will not be able to validate our final predictions until/unless we wait for the election outcome and purchase updated voter data.

I created a dataframe with just the 26614 rows of E7 and E8 election data and run it through the fully trained model.

The model predicted 4032 people will vote in E7 the 2018 Primary and 5861 will vote in E8 the 2018 General.

These predictions seem reasonable. Particularly if you consider that we should expect the E8 results to be an underestimation as everyone's 'last election' vote will incorrectly be -1 for E8 at this time as we don't have the vote data for E7's election yet. Running the model again between the E7 and E8 elections (ie in Jul or Aug 2018) would enable more accurate predictions to be made.

I use the Vid and the previously stored look up table to let my friend know who the likely voters are.