County Primaries Predictions

Short Term Project

## Objective

In this short term project you will build a model to predict different stages of the 2016 county primary election outcomes given demographic data, as well as results from previous stages. You will analyze at the accuracy of various regression and classification models to find the best model that fits the data both for the Democratic and Republican primaries.

## Data Description

Elections results data for 2016 is in datasets/geo/elections/2016/primary.csv file. County facts data is in datasets/geo/.

## Methodology and Deliverables

Submit all your work in geo/elections/elections\_forecast/.

### Data Gathering

Since you will be doing a backtest of a predictive model, for this project you will need to know the timeline of each primary contest. The data can be found in the corresponding Wiki pages:

* Democratic Primary timeline: Can be found in the Schedule and Results section in <https://en.wikipedia.org/wiki/Democratic_Party_presidential_primaries,_2016>
* Republican Primary timeline: Can be found in the Schedule section in <https://en.wikipedia.org/wiki/Republican_Party_presidential_primaries,_2016>

Using the Schedule data, add the following columns to data in primary.csv

* primary\_date: the date of the primary
* primary\_type: (string) the primary type. E.g. ‘closed caucus’, ‘open primary’, etc.

and save the result in primary\_processed.csv file.

### Structure of The Backtest

The backtest that you will create will mimic the way information was actually revealed during the primaries. In other words you cannot see into the future, but you are allowed to use any available information up to that point. Primaries which occur on the same day (say as the March 1 Super Tuesday primary) will all be assumed to have their results become available at the same time. Since prior to March 1 there isn’t much information, start your backtest after Super Tuesday as follows:

* First, skip predicting primaries such as Mar 1-8 Democrats abroad for which you have no County Facts data.
* At every stage of the primaries, use all of the available information up to and including that stage later stages.
  + For example, given the results up to Super Tuesday as well as all the county facts, predict the Democratic winner in Kansas counties on March 5, Florida counties in March 15, Arizona Counties in March 22 etc.
  + Note that this way you will be generating multiple predictions for primaries which are further away than the current primary. For example, you would have 3 predictions for Michigan primaries, coming from the March 1, March 5 and March 6 stages.

### Regression vs Classification Approach

* Compare both regression and classification models.
* For regression models, your goal is to predict *contestant\_edge* for each candidate i defined as edge\_i = %votes\_i - 1/N, where N is the number of candidates running in the contest at that stage.
  + Note that for 2 contestants the contestant edge for the winner is %votes\_winner - ½ = (2 \* %votes\_winner - 1) / 2 = (%votes\_winner - %votes\_loser) / 2 = %winner\_margin / 2.
  + Basically the contestant edge measures how far from a purely random outcome is the percentage of votes for the given candidate.
* For classification models, your goal is to predict the winner contestant in a given race.
  + Note that most classification models also produce a classification score, i.e. the probability that each participant can win. So you can technically also use these classification scores and compare them with the actual realized edge of each winner.

### Models to Use

In your model comparison, investigate the following models

* For classification:
  + Use SGDClassifier with both ‘log’ and ‘hinge’ loss corresponding to logistic regression and SVM classification respectively
  + What are the parameters in each model? How do you find the optimal set of parameters?
  + Construct an sklearn pipeline that would let you easily change the classification model and its parameters, as you do your backtests.
* For regression:
  + Use sklearn packages of both Elastic Nets and Gaussian Process Regression. Make sure to read the [sklearn Gaussian Process page](http://scikit-learn.org/stable/modules/gaussian_process.html).
  + What are the parameters in each model? How do you find the optimal set of parameters?
  + Construct an sklearn pipeline that would let you easily change the regression model and its parameters, as you do your backtests.

### Evaluation of Your Models

* At every stage of your backtest, your model will produce a set of predictions for each county outcome in the next stage of your model. For every given stage of the election, use the predictions for the counties in that stage:
  + For regression, scatter the predicted candidate edge vs the realized candidate edge.
  + For classification, compute the Precision-Recall curves.
* Which models perform best? Can you explain why? Are any of the models producing more variance? Are any of them producing higher bias? Explain.

### README file

Create a README\_db.txt file containing:

* The names and emails of all the teammates so you can be contacted by the next user of the dataset
* Description of the file naming convention, fields for each dataset
* Any comments on data features other users should be aware of when they use your data.