***Data Treatment***

- Explanation and reasoning behind any encodings, normalizations, composites or

other transformations on the original features

- How you leveraged the graph data

Many of the data fields were categorical in nature. To process these fields, I used either scikit-learn or pandas. For non-binary categories, I created dummy variables via One-Hot Encoding with panda’s “get\_dummies” functionality. For binary fields (sex, demplus, repplus, ismarried, home, renters) I used scikit-learn’s “LabelBinarizer” to avoid redundancy.

Following intuition, I wanted to leverage the graph data to understand how the voters in the dataset might be grouped. To do this, I used the “connected\_component\_subgraphs” function within networkX. In brief, what this does is

***Attempted Machine Learning Algorithms***

To make use of the graph data applicable with a classification method, I used the networkX package to (see the data treatment section for discussion).

***Model Evaluation***

Train test set…

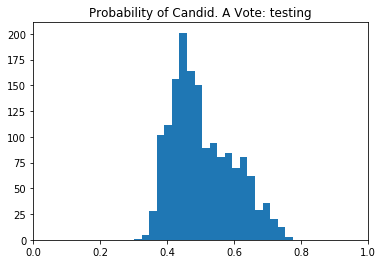
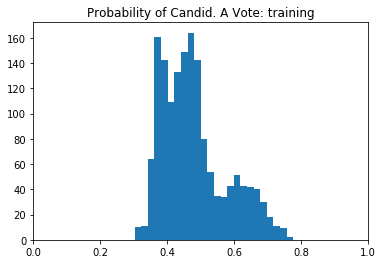
Accuracy score (in vs out of sample)

|  |  |  |
| --- | --- | --- |
| Accuracy Score | Training Set | Test Set |
| w/o Sub-Group Field | 61.6% | 56.4% |
| w/ Sub-Group Field | 65.4% | 57.4% |

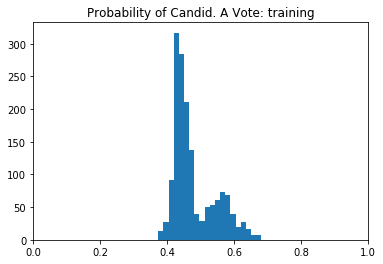
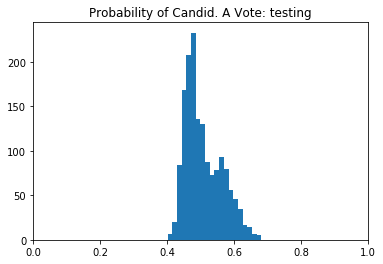
***Analysis of Results***

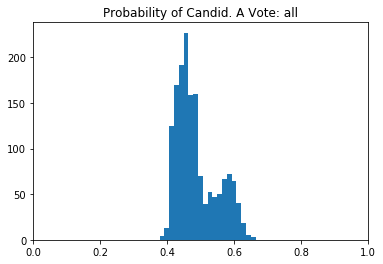
First question: do we do better with this sub group data? Yes maybe because the distributions are narrower?

w/o group



w/ group data



Tried with and without the graph data included…

***Conclusions Future Directions***

As used in this exercuise, the graph data id not improve accuracy of prediction but had some real effects on the distributions of probability. Means there may be promise?

- What you would do with this data if you worked with it daily (as opposed to this

brief time).