Andrew Carr

Logistic Regression Analysis

Purpose:

Predict the probability that a given person will show up to vote at the upcoming elections given previous voter history and personal information.

# Keywords:

**Voter file, Logistic Regression, Probability, Multicollinearity, voter turnout.**

# System Used:

The bulk of this analysis was done using the python library “scikit learn” which has a large number of modeling tools easily available.

# Assumptions made:

1. All voter variables are independent of one another to a degree. Ex: higher income does not affect voter’s party choice. Therefore, all the variables were used with equal weight to predict voter turnout.
2. Let it be noted that the use of mail in ballots will significantly change the percentage of voter turnout, and previous data will be obsolete in predicting future events. We are ignoring this fact in our prediction.
3. Year of vote does not affect the likelihood of turnout. In other words, if someone voted in 2006 and did not vote in 2008 it was not because of the year.
4. The process of transformation of data from non-numeric values to numeric values will not result in multicollinearity.

# Results:

## Model: Logistic Regression

The logistic regression model is used to estimate the probability of an event based on one or more independent features. This regression is also used to measure the relationship between dependent variables and their independent counterparts. It uses a logistic function to estimate these distributions.

This type of regression is optimal for predicting a probability that a specific voter will turn out to vote.

## Features:

In ‘training’ a model for prediction, there are two types of data that are required for an accurate prediction. First, it is necessary to have a list of independent variables. In this prediction, all personal information is used with equal weight as independent variables.

Second, there must be a list of ‘targets’ that will be the outcome of our prediction. In other words, previous information will be used to train features on targets. In this situation, the target is an average of all the previous voter turnout from the years 2000 to 2014. These binary values are summed and averaged to get a percentage. The model is then trained against these target values.

## Table:

This is an excerpt from the turnout.csv file that contains all of the predictions. If the vote\_probability was greater than 0.50 then we assumed that this person would turn up to vote. Otherwise, they were said to not vote.



# Overview of code:

## Data Formatting:

The data was converted from non-numeric values to numeric values while still in the csv file, using the built in find and replace functionality of excel. If this process needed to be repeated for multiple records, a programmatic way to convert the data would need to be implemented. However, for the sake of brevity, that functionality was not included in this system.

Numbers were assigned per column to similar values. A more detailed taxonomy is included in notes.txt. Please refer to this document for more information regarding the change of variables.

After the prediction process, if required, find and replace can be used again to reinstitute the original variable values for easier inference.

## Import Statements:

**import** csv

**from** sklearn **import** linear\_model

**import** numpy **as** np

## Read in the CSV:

Note: The function of this bit of code is to read in the file from a csv that has been converted into numerical form. *(See Overview of Code->Data formatting and notes.txt for information regarding variable changes)*

data **=** **[]**

personal **=** **{}**

**with** open**(**'voterfileNym.csv'**,** newline**=**''**)** **as** csvfile**:**

data\_csv **=** csv**.**reader**(**csvfile**)**

**for** row **in** data\_csv**:**

data**.**append**(**row**)**

## Format Data:

Here the data is organized into an array, and the 0ptimus\_id is removed, so that it is not included with the prediction’s data.

**for** i **in** range **(**0**,** len**(**data**)):**

personal**[**data**[**i**][**0**]]** **=** **[**x **for** x **in** data**[**i**]]**

personal**[**data**[**i**][**0**]].**pop**(**0**)**

numlist **=** **[]**

targets **=** **[]**

**for** key **in** personal**.**keys**():**

numlist**.**append**([**personal**[**key**][**x**]** **for** x **in** range**(**0**,** len**(**personal**[**key**])-**1**)])**

summation **=** 0

**for** i **in** range**(**10**,** 25**):**

summation **+=** int**(**personal**[**key**][**i**])**

average **=** summation**/**14

targets**.**append**(**average**)**

Note: In the above code the data is trained with an average of all of the previous years of voter turnout data. *(See Results->Features for more information)*

## Fit Model:

npNumlist **=** np**.**array**(**numlist**).**astype**(**np**.**float**)**

npTargets **=** np**.**array**(**targets**).**astype**(**np**.**float**)**

model **=** linear\_model**.**LogisticRegression**()**

model**.**fit**(**npNumlist**,** npTargets**)**

expected **=** npTargets

predicted **=** model**.**predict\_proba**(**npNumlist**)**

## Append to Array:

Note: The optimus\_id is added back to the beginning of each array, even though it is included as the dictionary key. This is done to simplify the writing process to the new csv.

counter **=** 0

**for** key **in** personal**.**keys**():**

personal**[**key**].**insert**(**0**,**key**)**

personal**[**key**].**append**(**"%.3f" **%** predicted**[**counter**][**1**])**

**if(**predicted**[**counter**][**1**]** **>=** 0.5**):**

personal**[**key**].**append**(**'1'**)**

**else:**

personal**[**key**].**append**(**'0'**)**

counter **+=** 1

## Write to file:

Note: The row headers will have to be added back on manual to the csv file after it has been written to, however this is trivial. Again, if this process needed to be repeated, that could be handled programmatically.

writer **=** csv**.**writer**(**open**(**'new.csv'**,** 'w'**))**

**for** key**,** value **in** personal**.**items**():**

writer**.**writerow**([**x **for** x **in** value**])**

# Conclusion:

While this prediction could be made more accurate by further analysis of independent variables’ relations with one another, it fulfills the requirements of prediction. The data used to train the model is an average of the sum of the previous years of voter turnout. This data was used to help predict future turnout.

In addition to this document: voterfile.csv, voterfileNym.csv, turnout.csv, and notes.txt are included to provide supplemental information if desired.