US Poll Election Prediction

# US Poll Election Prediction

Under Guidance of
Dr. Senjuti Basu Roy

Aishwarya Bose
12/14/2016

## 1. What are the trends of the polls over time (by month)? Present visualization.

- First csv file is converted to dataframe format using pandas.
- The enddate present in csv file is in String format. It is required to convert it to datetime format i.e. '%m/%d/20%y' . Matplotlib understands in datetime format.
- The new formated enddate i.e. date_tmp is appended in the dataframe as dates
- adj_trump and adj_clinton columns are selected from dataframe format as lists with names trump_adj and clinton_adj respectively.
- In order to plot trend of polls by month, scatter() function is used.

Why scatter plot is used ?

The time series data is very much inter dependent to time and polling percentage. Data is very discrete and widely spread. Scatter plot is the best choice to display how much one variable is dependent upon each other.

Barplot can also be used, but matplotlib python is taking long time to execute.

Following are the steps incorporated in Python.

- For finding trend of poll for Clinton, x-axis (Clinton_adj) and y-axis(dates) and in order to distinguish, color blue is given.
- Since it is necessary to plot the trend by month, set_major_locator() and set_major_formatter() functions are used.
- Xlabel is set as 'Poll end date' and Ylabel as 'clinton adjusted'.
- Same steps are followed for Trump. Only the color of the scatter plot is Red. And Xlabel is set as 'Poll end date' and Ylabel as 'trump adjusted'.
- In the main program the csv file 'presidential_polls.csv' is read and the plots are visualised using draw_visualization().

The scatter plot of both Clinton and Trump over the month is shown in Fig. 1.1 and Fig 1.2.

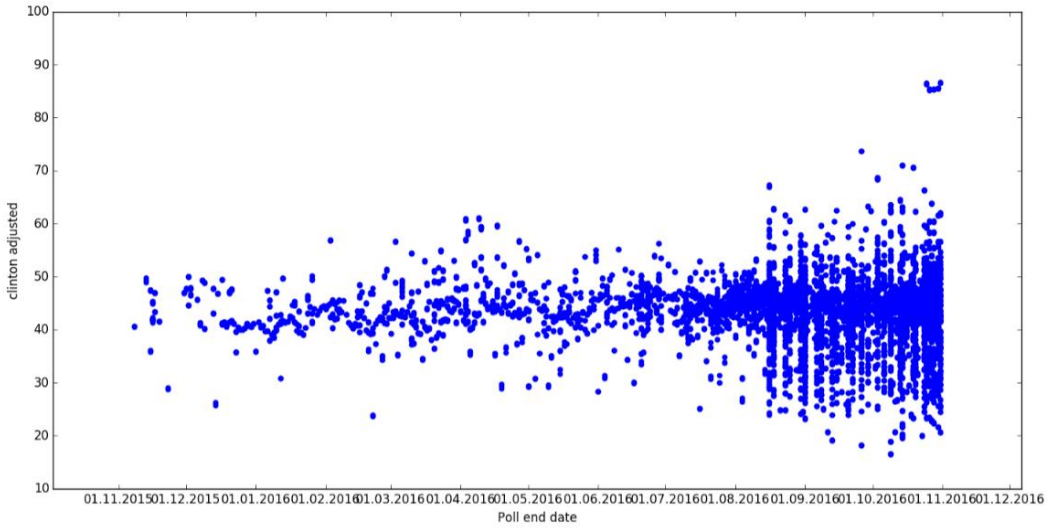


Fig 1.1 Scatter plot of Clinton polls

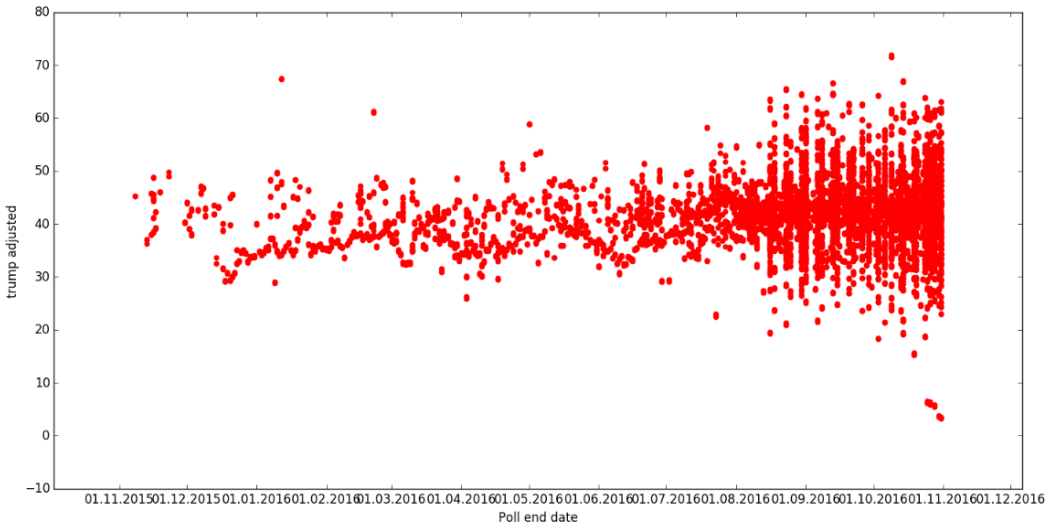


Fig 1.2 Scatter plot of Trump polls

**2. Describe your solutions including any preprocessing steps, predictors, and the classification algorithm.**

Understanding the data**:**

A good sampling strategy needs to consider many factors for presidential election based on which polls prediction can be biased.

The provided csv file consists of different attributes namely cycle, branch, matchup, type, url, which does not provide any insight to the data. The dataset related to poll has following challenges:

- Possible factors affecting predictions are sampling time, state, sample size, the way the sampling is done, any other external effects.
- Attributes for consideration are sample size, polling weight, number of days from start date till today. Weighted average of this data is created using:

```
tmp_weight = polling_weight

tmp_weight *= df.samplesize[count]

tmp_weight /= days_from_poll[count]
```

- Margin of adjustment between raw poll data and adjusted data are taken into consideration.
- States are considered to be a factor because of political preference.
- Binomial data is created trump_win=0 and clinton_win=1.

Attributes like raw_clinton , adjpoll_clinton , raw_trump, adjpoll_trump, samplesize , polling_weight , startdate and enddate are considered. Following table shows how the resultant dataset looks like ( as given in Fig. 2.1).

| | clinton_adjusted | polling_weight | state | trump_adjusted | trump_clinton_win |
|---|---|---|---|---|---|
| 0 | 4.9514 | 149279.0608 | 36 | 5.79509 | 1 |
| 1 | -1.70341 | 2367.272688 | 36 | -1.27016 | 0 |
| 2 | -1.70221 | 1984.950272 | 21 | -1.27396 | 1 |
| 3 | -1.64069 | 532.159137 | 37 | 0.30585 | 1 |
| 4 | -0.67256 | 1028.092789 | 36 | 2.20888 | 1 |
| 5 | 0.6508 | 583.7313193 | 36 | 1.26663 | 1 |
| 6 | 1.61834 | 3334.013116 | 36 | -0.13983 | 1 |
| 7 | -0.10951 | 5175.750875 | 35 | -0.49667 | 1 |
| 8 | -0.47424 | 728.3648616 | 36 | 0.84948 | 1 |
| 9 | 0.21983 | 7915.2476 | 36 | 1.6954 | 1 |
| 10 | 1.25217 | 9298.856095 | 36 | -3.05155 | 1 |
| 11 | -0.18168 | 449.077225 | 37 | 1.92262 | 0 |

Fig 2.1. A part of resultant Dataset after Preprocessing.

Model 1 (Logistic regression model)

From the preprocessed data , the problem is interpreted as probability of trump win = 0 or Clinton win = 1. Logistic regression modeling is used.

Test data and training data:

From the dataset displayed above test and training data is created in a split of 70:30.

```
def test_train_data(new_df):
'''
Source for LogisticRegression code

http://nbviewer.jupyter.org/gist/justmarkham/6d5c061ca5aee67c4316471f8c2ae976
'''
y, X = dmatrices('trump_clinton_win ~ state + polling_weight + trump_adjusted + clinton_adjusted' ,new_df,
return_type="dataframe")
y = np.ravel(y)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=0)
return X, y, X_train, X_test, y_train, y_test
```

Following is the code in Python:

```
print 'Modelling with logistic regression \n'
model2 = LogisticRegression()
model2.fit(X_train, y_train)
predicted_regression = model2.predict(X_test)
probs = model2.predict_proba(X_test)

print ' accuracy on Logistic model training set ', model2.score(X_train, y_train)
print ' \n Done Modelling with logistic regression \n'
```

```
print predicted_regression
fig=plt.figure()
plt.scatter(probs[:,1],predicted_regression,color='r')
plt.xlabel('Clinton adjusted margin probablity')
plt.ylabel('clinton_trump_win_loss')
plt.show()
```

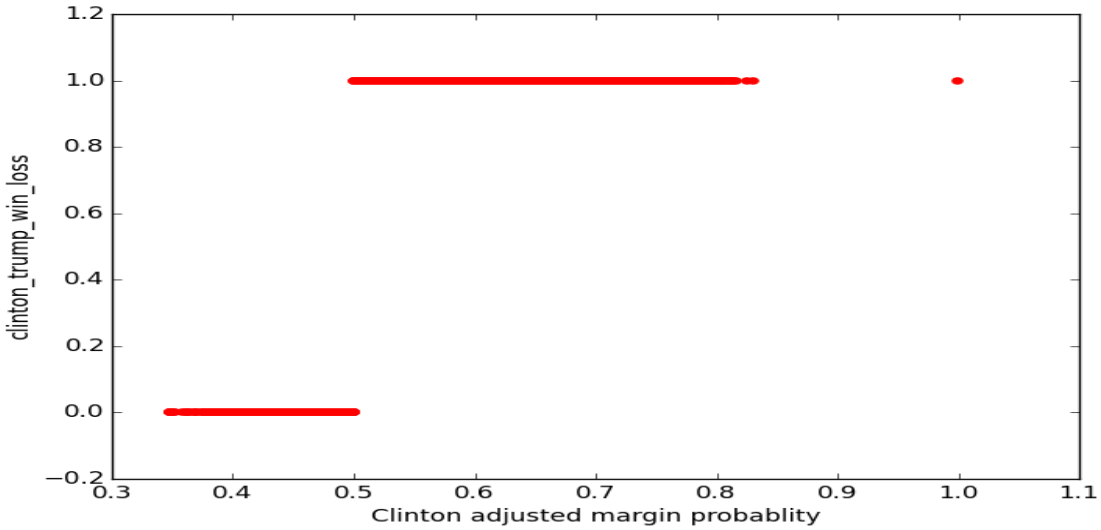


Fig 2.2. Output graph of Logistic Regression Model

The above graph shows the probability of Clinton winning the election i.e. 1 is from 0.5 to 0.85. While the probability of Trump winning the election i.e. 0 is from 0.35 to 0.5. Hence, according to Logistic Regression model Clinton is the winner of presidential poll election.

## Model 2 (Decision Tree model)

Decision Tree is performed on same dataset as given in Fig.2.3.

The decision tree is generated using the following R code.

```
library(party)
library(rpart)
library(rpart.plot)
library(readr)
library(dplyr)

df <- read.csv('president_out.csv')
iris_ctree <- ctree(trump_clinton_win ~ clinton_adjusted + polling_weight + state + trump_adjusted, data =  df)
plot(iris_ctree)
```

The decision tree graph looks like :

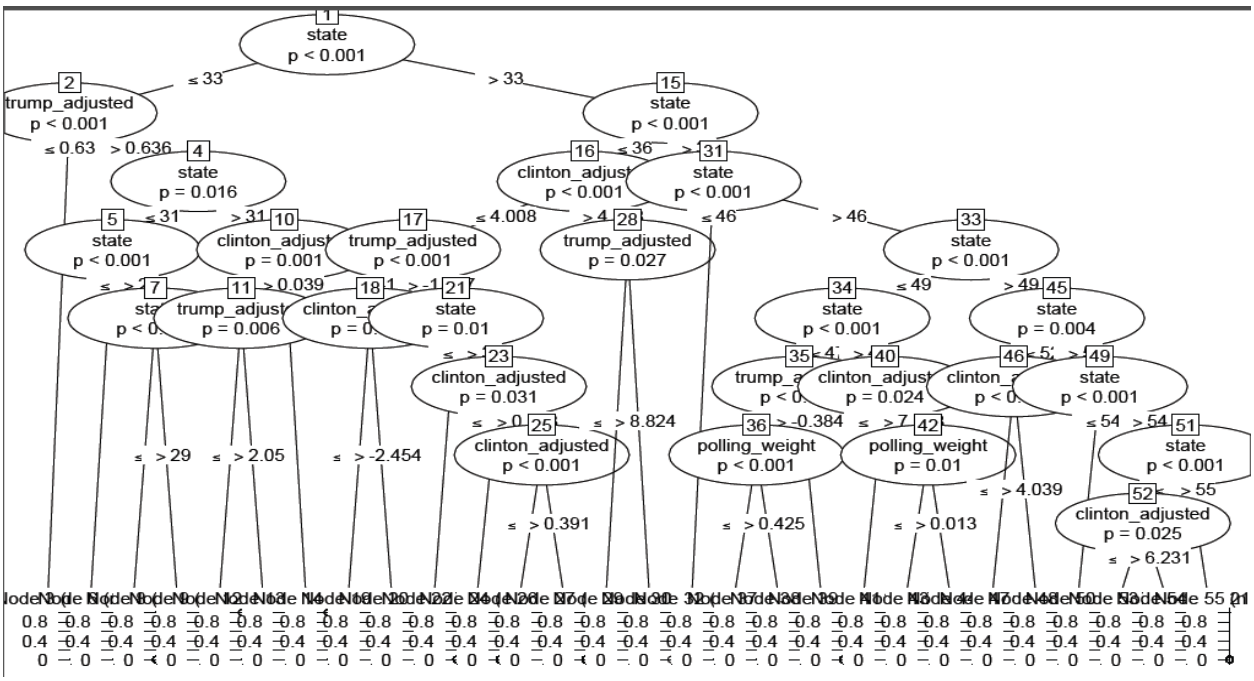


Fig 2.3. Decision Tree Model

Following are the steps incorporated in Python.

- Defining a function clean_prepare_data().
- adjpoll_clinton, rawpoll_clinton, adjpoll_trump, rawpoll_trump are converted to list format.
- Going through both columns of adj_clinton, raw_clinton error in actual is estimated.
- startdate in csv file is converted in list format in order to calculate the number of days wrt Today's date.
- Comparing adj_clinton and adj_trump, if it is greater than 0 then Clinton wins and if adj_clinton and adj_trump is less than 0 then Trump wins.
- All duplicate states are removed and are mapped for eg: {'Wisconsin': 43, 'Mississippi': 0, 'Washington': 49 }
- Creating weighted_data list which contains sample size, days_from_poll.
- Final table is created using the attributes 'trump_clinton_win' : binomial_data,'state' : state_data, 'polling_weight' : weighted_data, 'trump_adjusted' : trump_error_data, 'clinton_adjusted':clinton_error_data}

-Test and Training set

-Test and Training set are created from the dataframe with a split of Testing data as 30% and Training data as 70% where Xtrain- Attribute of Training set , Ytrain- Classlabel of Training set, Xtest- Attribute of Test set, Ytest- Classlabel of Test set.

**3. Of course the election results are not out yet therefore there is no ground truth as such. How could you possibly use the given dataset to augment it with ground truth and perform 5-fold cross validation.**

5 fold cross validation is performed by following code

```
scores = cross_val_score(logistic_model, X_test, y_test, scoring='accuracy', cv= 5
        print '\n for logictic regression after 5-fold cross-validation', scores, scores.mean()
        kfold = KFold(n=len(X_test), n_folds=5)
        results = cross_val_score(logistic_model, X_test, y_test, cv=kfold)
        print("Accuracy for LogisticRegression: %.3f%% (%.3f%%)") % (results.mean()*100.0,
results.std()*100.0)
```

Cross validation is performed using both the models.
For logistic regression the scores are as follows
For logistic regression after 5-fold cross-validation [ 0.62042013 0.64973131 0.61797753 0.63636364 0.64125122]
Mean = 0.633148765497 standard deviation .0357

For decision tree the scores are after 5-fold cross-validation [ 0.94870542 0.92281387 0.90425012 0.91495601 0.9173998 ]
Mean = 0.921625046978 standard deviation = .0187

**4. Based on 4, perform 5-fold cross validation and compute precision of M1 and M2. Based on precision, does one of your model better than the other with 5% significance level? Describe your approach in length.**

Following are the statistical data for both the models i.e.

Logistical Model Classification

logistic_model classification_report

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0.0 | 0.54 | 0.21 | 0.31 | 1154 |
| 1.0 | 0.65 | 0.89 | 0.75 | 1916 |

| | | | | |
|---|---|---|---|---|
| avg / total | 0.61 | 0.64 | 0.59 | 3070 |

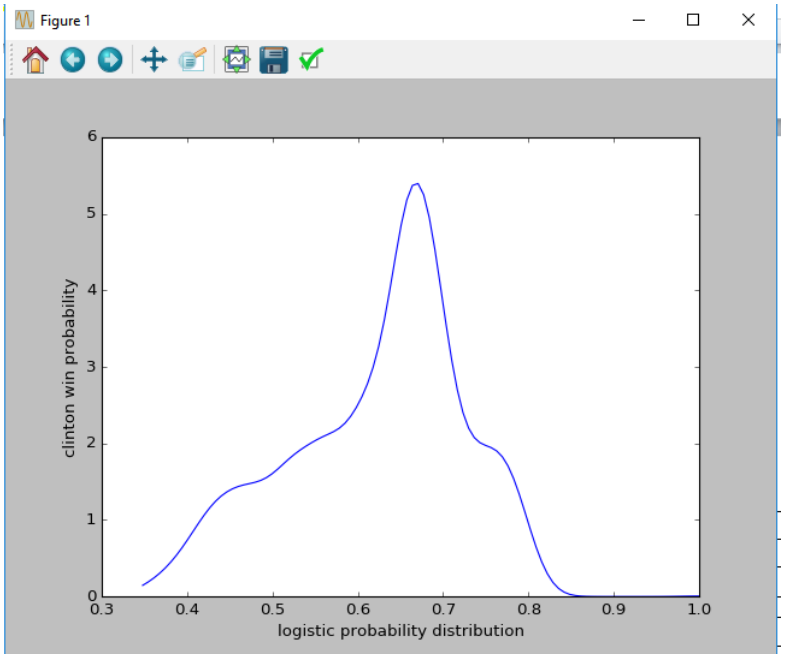


Fig. 4.1. Logistic probability distribution

## Decision Tree Classification

decision_model classification_report

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0.0 | 0.89 | 0.89 | 0.89 | 1154 |
| 1.0 | 0.93 | 0.94 | 0.94 | 1916 |

| | | | | |
|---|---|---|---|---|
| avg / total | 0.92 | 0.92 | 0.92 | 3070 |

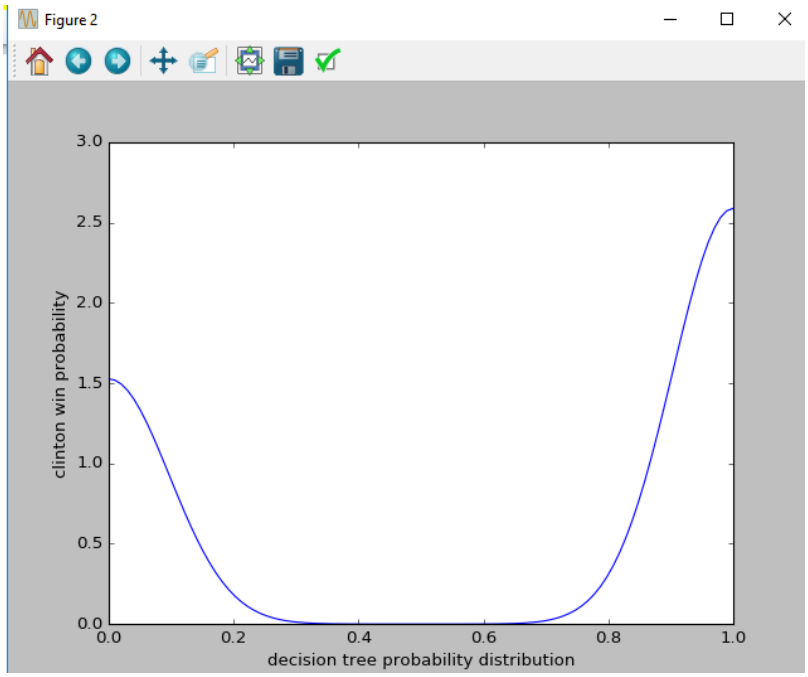


Fig. 4.2. Decision Tree probability distribution

Accuracy is the proximity of measurement results to the true value; precision, the repeatability, or reproducibility of the measurement.

Based on the data the Decision tree has high precision and high accuracy but Logistic Regression has less accuracy and less precision.