The Presidential Analysis

Project Report

22:960:641-**ANALYTICS FOR BUSINESS INTELLIGENCE**

By
Nishant Sharma

Guided by: **Professor Jaideep Vaidya(Rutgers University)**

DATA OVERVIEW

The analysis primarily consists of three Datasets-

- Presidential Polls- This dataset is a collection of state and national polls conducted from November 2015-November 2016 on the 2016 presidential election. Data on the raw and weighted poll results by state, date, pollster, and pollster ratings are included. This contains 27 relevant socio geographical survey variables. The original dataset is from the FiveThirtyEight 2016 Election Forecast. Poll results were aggregated from HuffPost Pollster, RealClearPolitics, polling firms and news reports.
  Description/Metadata
  - cycle
  - branch
  - type
  - matchup
  - forecastdate
  - state:
  - startdate
  - enddate
  - pollster
  - grade
  - samplesize
  - populaion
  - poll_wt
  - rawpoll_clinton
  - rawpoll_trump
  - rawpoll_johnson
  - rawpoll_mcmullin
  - adjpoll_clinton
  - adjpoll_trump
  - adjpoll_johnson
  - adjpoll_mcmullin
  - multiversions
  - url
  - poll_id
  - question_id
  - createddate
  - timestamp

- Primary Results- This contains data relevant for the 2016 US Presidential Election, including up-to-date primary results of 8 variables. Each row contains the votes and fraction of votes that a candidate received in a given county's primary. Sample
  Description/Metadata
    - state: state where the primary or caucus was held
    - state_abbreviation: two letter state abbreviation
    - county: county where the results come from
    - fips: FIPS county code
    - party: Democrat or Republican
    - candidate: name of the candidate
    - votes: number of votes the candidate received in the corresponding state and county (may be missing)
    - fraction_votes: fraction of votes the president received in the corresponding state, county, and primary

- County Facts-
  Description/Metadata
  - PST045214 Population, 2014 estimate
  - PST040210 Population, 2010 (April 1) estimates base
  - PST120214 Population, percent change - April 1, 2010 to July 1, 2014
  - POP010210 Population, 2010
  - AGE135214 Persons under 5 years, percent, 2014
  - AGE295214 Persons under 18 years, percent, 2014
  - AGE775214 Persons 65 years and over, percent, 2014
  - SEX255214 Female persons, percent, 2014
  - RHI125214 White alone, percent, 2014
  - RHI225214 Black or African American alone, percent, 2014
  - RHI325214 American Indian and Alaska Native alone, percent, 2014
  - RHI425214 Asian alone, percent, 2014
  - RHI525214 Native Hawaiian and Other Pacific Islander alone, percent, 2014
  - RHI625214 Two or More Races, percent, 2014
  - RHI725214 Hispanic or Latino, percent, 2014
  - RHI825214 White alone, not Hispanic or Latino, percent, 2014
  - POP715213 Living in same house 1 year & over, percent, 2009-2013
  - POP645213 Foreign born persons, percent, 2009-2013
  - POP815213 Language other than English spoken at home, pct age 5+, 2009-2013
  - EDU635213 High school graduate or higher, percent of persons age 25+, 2009-2013
  - EDU685213 Bachelor's degree or higher, percent of persons age 25+, 2009-2013
  - VET605213 Veterans, 2009-2013
  - LFE305213 Mean travel time to work (minutes), workers age 16+, 2009-2013
  - HSG010214 Housing units, 2014

- HSG445213 Homeownership rate, 2009-2013
- HSG096213 Housing units in multi-unit structures, percent, 2009-2013
- HSG495213 Median value of owner-occupied housing units, 2009-2013
- HSD410213 Households, 2009-2013
- HSD310213 Persons per household, 2009-2013
- INC910213 Per capita money income in past 12 months (2013 dollars), 2009-2013
- INC110213 Median household income, 2009-2013
- PVY020213 Persons below poverty level, percent, 2009-2013
- BZA010213 Private nonfarm establishments, 2013
- BZA110213 Private nonfarm employment, 2013
- BZA115213 Private nonfarm employment, percent change, 2012-2013
- NES010213 Nonemployer establishments, 2013
- SBO001207 Total number of firms, 2007
- SBO315207 Black-owned firms, percent, 2007
- SBO115207 American Indian- and Alaska Native-owned firms, percent, 2007
- SBO215207 Asian-owned firms, percent, 2007
- SBO515207 Native Hawaiian- and Other Pacific Islander-owned firms, percent, 2007
- SBO415207 Hispanic-owned firms, percent, 2007
- SBO015207 Women-owned firms, percent, 2007
- MAN450207 Manufacturers shipments, 2007 ($1,000)
- WTN220207 Merchant wholesaler sales, 2007 ($1,000)
- RTN130207 Retail sales, 2007 ($1,000)
- RTN131207 Retail sales per capita, 2007
- AFN120207 Accommodation and food services sales, 2007 ($1,000)
- BPS030214 Building permits, 2014
- LND110210 Land area in square miles, 2010
- POP060210 Population per square mile, 2010

PROBLEM STATEMENTS AND OBJECTIVES

- Which are the major contributors towards the Adjusted Poll Prediction of the winner of Presidential Election 2016?
- Checking non variance and biasness of the survey
- Which variables are major contributors towards a county voting for a particular party?
- Predicting and selecting best model for who will win a county based on demographics
    - Random Forest
    - Naïve Baye's
- Which model classifies best the winner class for Presidential Polls Survey?
    - Neural Networks
    - Naïve Baye's
- A correct prediction from the big-name surveys from simple mathematics (Surprise Analysis?)

STATISTICAL INTERPRETATION

5 Number summary for Primary Results dataset-

- For both votes and fraction votes the mean is greater than the median and hence both are likely to be positively skewed.

```
summary(primary)
##      state            state_abbreviation    county
##  Length:24611       Length:24611       Length:24611
##  Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##       fips              party            candidate              votes
##  Min.   :     1001   Length:24611       Length:24611       Min.   :     0
##  1st Qu.:    21091   Class :character   Class :character   1st Qu.:    68
##  Median :    42081   Mode  :character   Mode  :character   Median :   358
##  Mean   :26671525                                          Mean   :  2306
##  3rd Qu.:90900125                                          3rd Qu.:  1375
##  Max.   :95600036                                          Max.   :590502
##  NA's   :100
##  fraction_votes
##  Min.   :0.0000
##  1st Qu.:0.0940
##  Median :0.2730
##  Mean   :0.3045
##  3rd Qu.:0.4790
##  Max.   :1.0000
##
```

Best Fit Regression model for Surveyor's opinion on Adjusted Trump Poll

- Adjusted Poll from trump regressed against Sample size, population type, Grade of voter, poll weightage and raw polls figured.

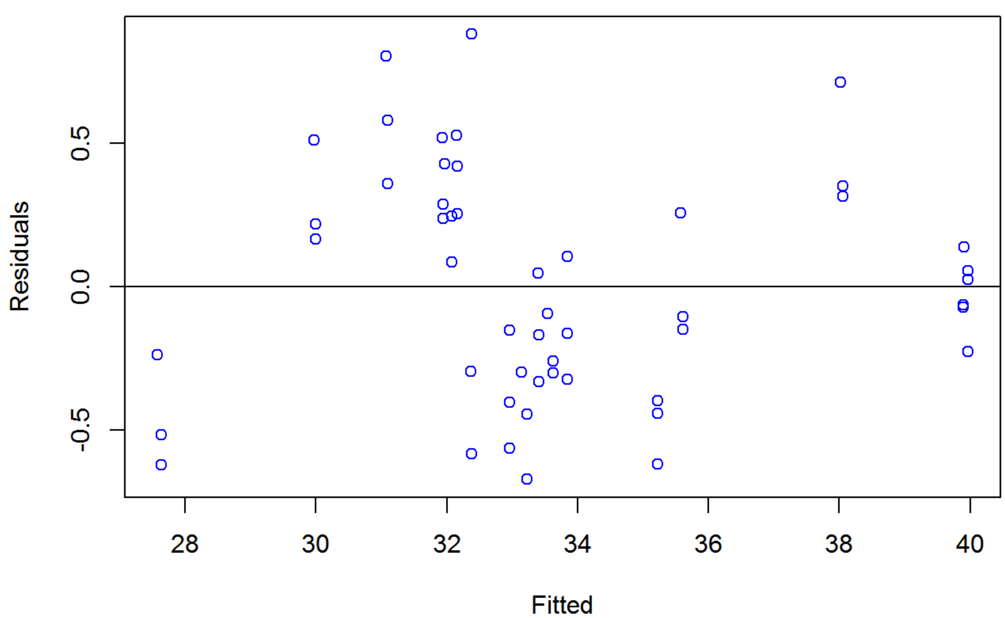
```
##Selected best fit model towards predicted adjusted polls for the actual win
ner

##Adjusted R square=98%

g_trump<- lm(adjpoll_trump~grade+samplesize+population+poll_wt+rawpoll_trump+
rawpoll_clinton

             +rawpoll_johnson+rawpoll_mcmullin, data = pres)

summary(g_trump)
```

```
##
## Call:
## lm(formula = adjpoll_trump ~ grade + samplesize + population +
##     poll_wt + rawpoll_trump + rawpoll_clinton + rawpoll_johnson +
##     rawpoll_mcmullin, data = pres)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.67340 -0.30102 -0.06513  0.27136  0.88111
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      -46.738056   7.405659  -6.311 1.92e-07 ***
## gradeB             0.914385   0.653717   1.399 0.169791
## gradeB+           -0.834665   0.700329  -1.192 0.240535
## gradeC-           -7.679775   1.026985  -7.478 4.77e-09 ***
## gradeC+           -0.370639   0.448819  -0.826 0.413935
## samplesize         0.004613   0.001168   3.948 0.000319 ***
## populationlv     -19.060519   1.389947 -13.713  < 2e-16 ***
## poll_wt           -0.298549   0.125180  -2.385 0.022038 *
## rawpoll_trump      1.732397   0.101335  17.096  < 2e-16 ***
## rawpoll_clinton    0.392129   0.086683   4.524 5.56e-05 ***
## rawpoll_johnson    1.441052   0.151402   9.518 1.02e-11 ***
```

```
## rawpoll_mcmullin   0.849171   0.091825   9.248 2.23e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4503 on 39 degrees of freedom
##   (10185 observations deleted due to missingness)
## Multiple R-squared:  0.9848, Adjusted R-squared:  0.9805
## F-statistic: 230.1 on 11 and 39 DF,  p-value: < 2.2e-16
```

Residue Fitted Plot for our model

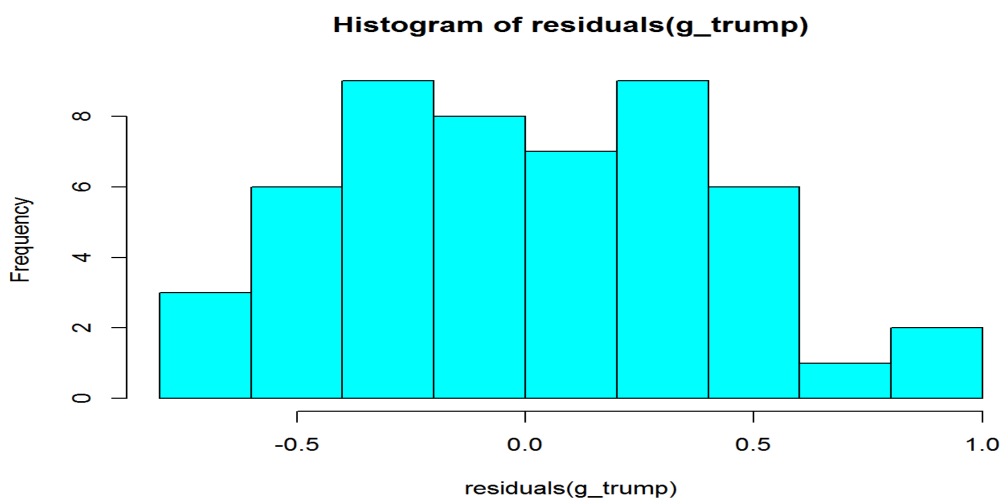
```
##Residue vs Fitted Plot pattern
##Checking non-constant variance, non-normality
par(mfrow=c(1,1))
plot(fitted(g_trump), residuals(g_trump), xlab="Fitted", ylab="Residuals", co
l="blue")
abline(h=0)
```



- The residual plot does not exhibit a prompt pattern and puts up a constant variance.

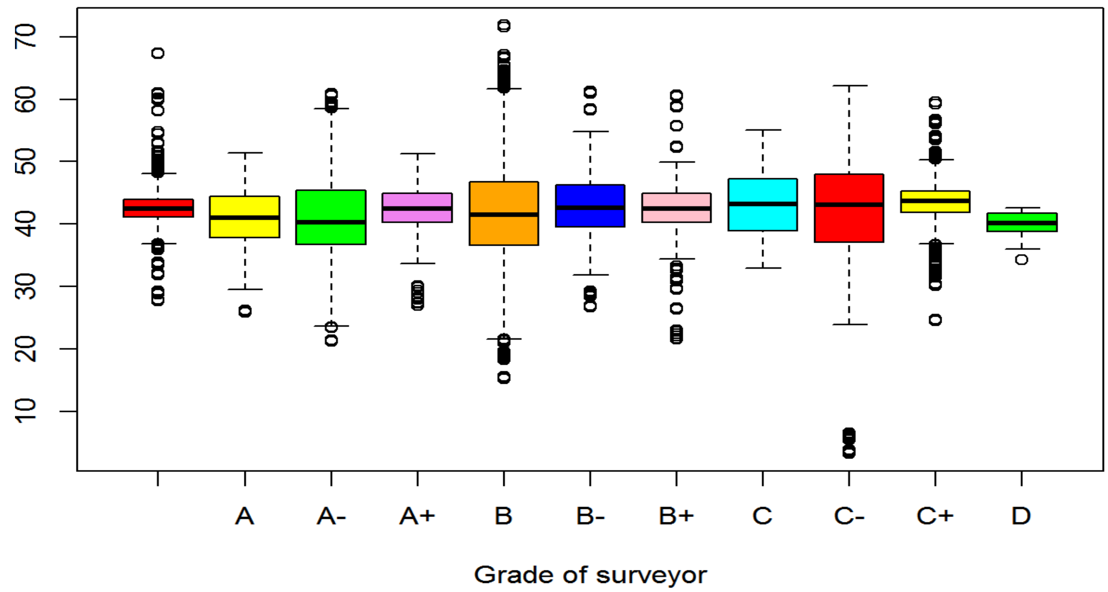
- The histogram of the residuals show a near bell curve indicating the normality assumption likely to be true.

```
##Distribution of residuals
hist(residuals(g_trump), col="cyan")
```

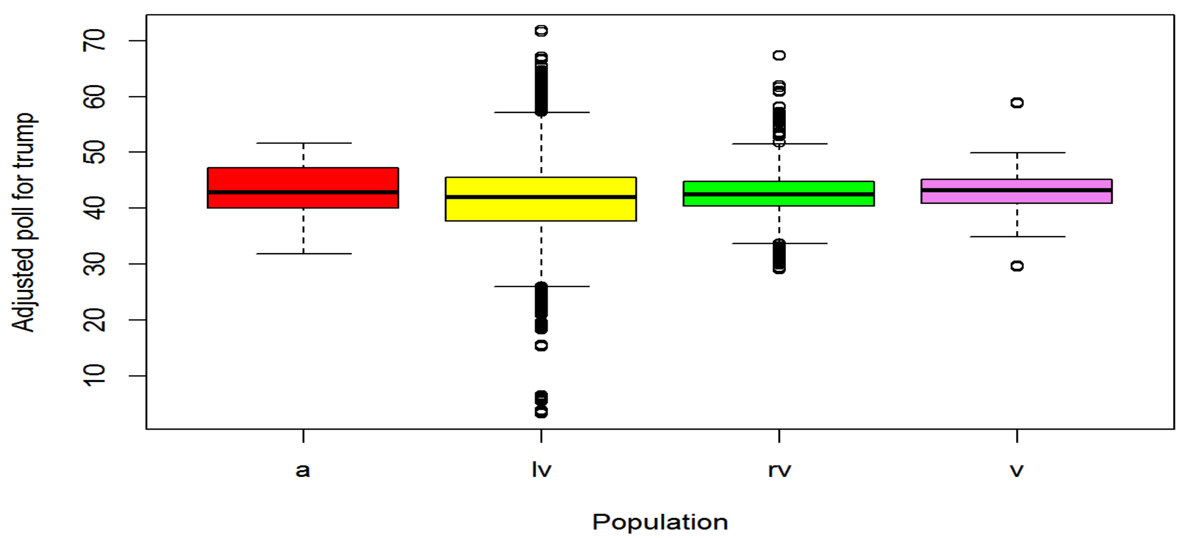


Detecting Outliers in the dataset

```
##To detect outliers in regards to prior grades of surveyors
colors = c("red", "yellow", "green", "violet", "orange",
            "blue", "pink", "cyan")
boxplot(adjpoll_trump~grade, data=pres, xlab = "Grade of surveyor", ylab = "A
djusted poll for trump", col=colors)
```

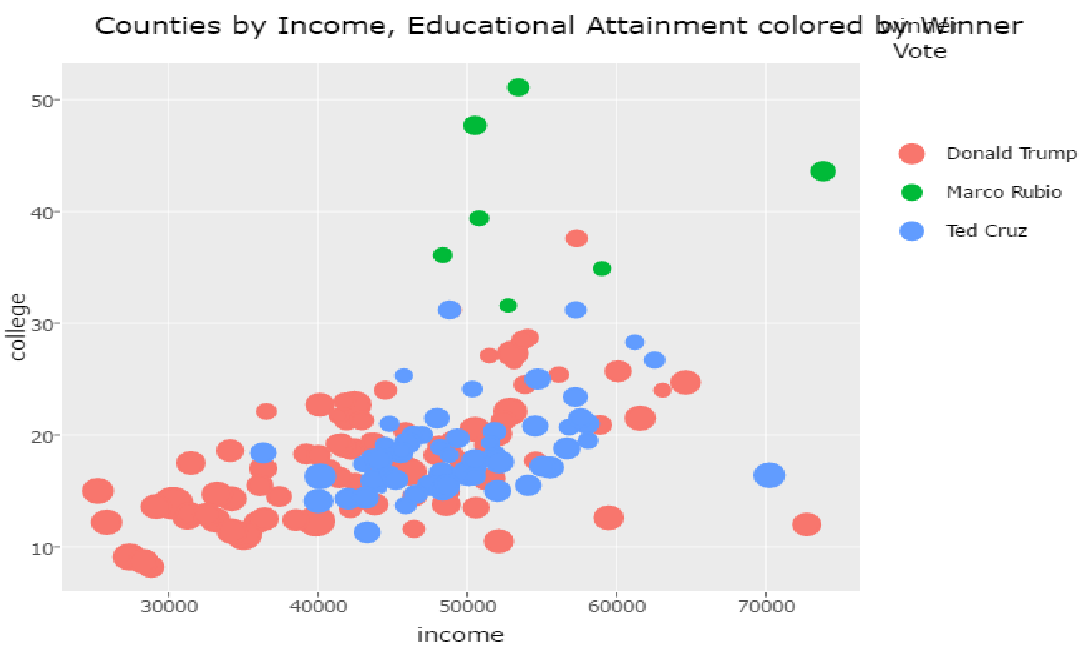



```
##To detect outliers in regards to population

boxplot(adjpoll_trump~population, data=pres, xlab = "Population", ylab = "Adj
usted poll for trump", col=colors)
```

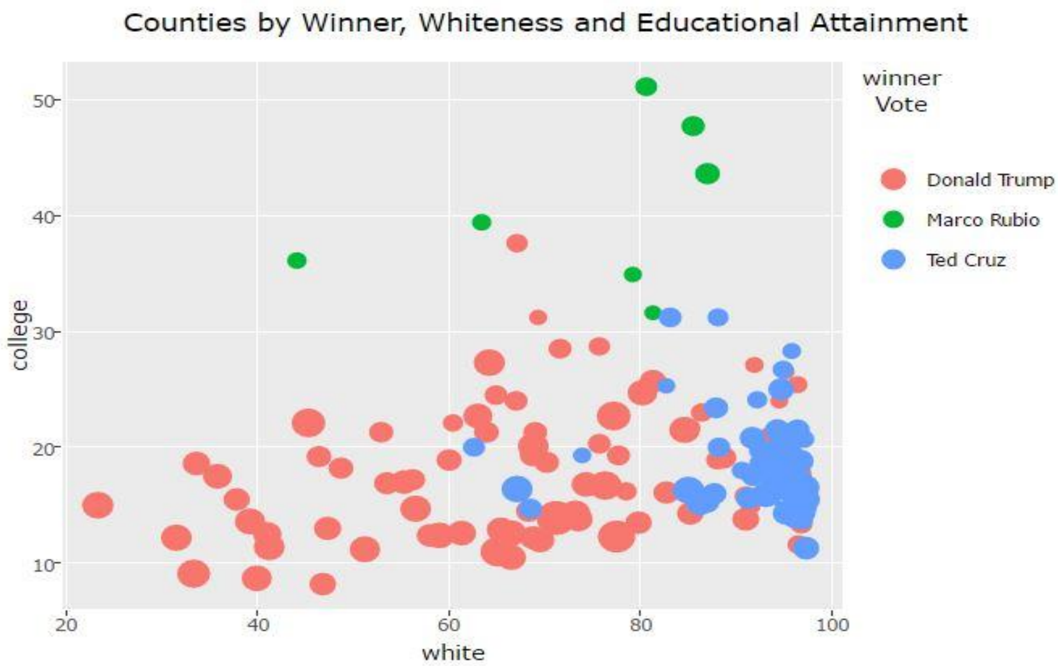


CLASSIFICATION ANALYSIS OF PRIMARIES

```
ggplotly(qplot(x =  white, y = college, data = votes,
                color = winner, size = Vote) +
            ggtitle("Counties by Winner, Whiteness and Educational Attainment"
))
```




```
ggplotly(qplot(x =  income, y = college, data = votes,
                color = winner, size = Vote) +
            ggtitle("Counties by Income, Educational Attainment colored by Win
ner"))
```



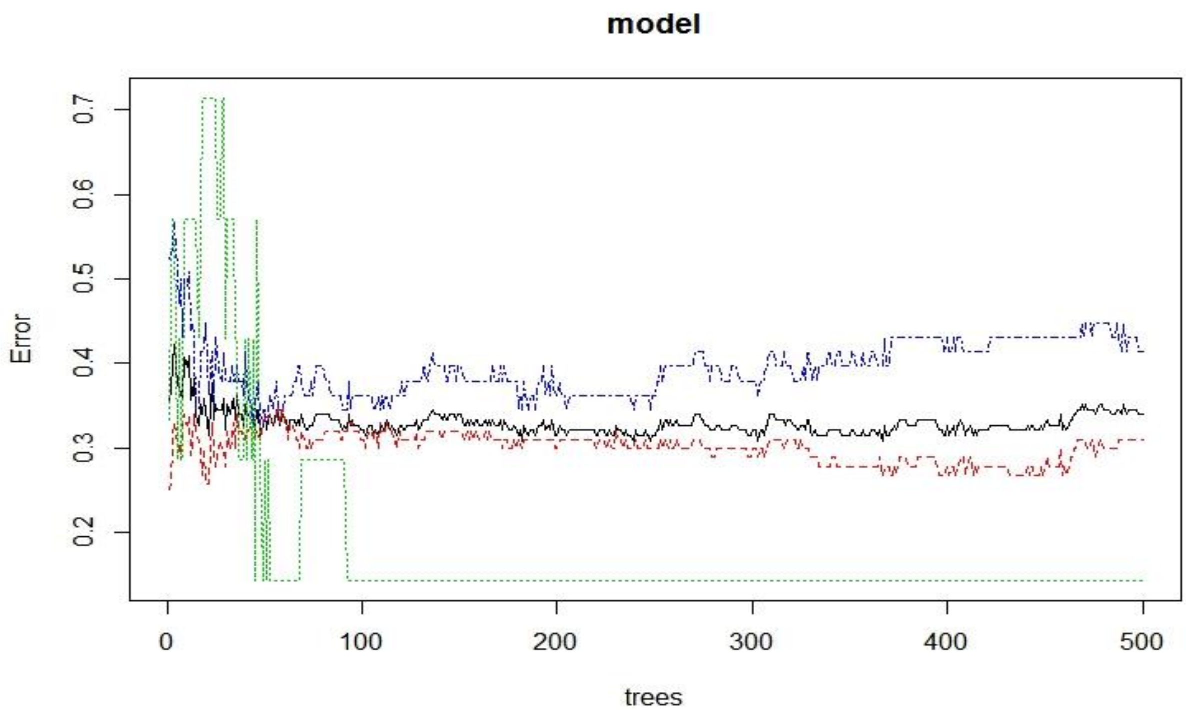
Applying Random forest:

→Using: winner ~ income + hispanic + white + college + density.

→it has as roughly 70% accuracy.

→The blue line represents the error for Cruz, red is Trump and green is Rubio. Rubio's errors are more erratic since we only have a few data points for him.

```
votes$winner <- as.factor(votes$winner)





model <- randomForest(winner ~ income + hispanic + white + college + density,
data = votes)

plot(model, ylim = c(0, 0.7))
```




```
votes
## Source: local data frame [162 x 10]
## Groups: state_abbreviation [?]
##
##    state_abbreviation     county        winner  Vote votes income hispanic
##                 <chr>      <chr>        <fctr> <dbl> <int>  <int>    <dbl>
## 1                  IA      Adair Donald Trump 0.256   104  47892      1.7
## 2                  IA      Adams      Ted Cruz 0.297    81  45871      1.1
## 3                  IA  Allamakee Donald Trump 0.281   193  48831      5.7
## 4                  IA  Appanoose Donald Trump 0.348   292  39208      1.5
## 5                  IA    Audubon      Ted Cruz 0.361   135  48313      1.1
## 6                  IA     Benton      Ted Cruz 0.365   596  56669      1.3
## 7                  IA Black Hawk      Ted Cruz 0.268  1585  45747      4.2
## 8                  IA      Boone      Ted Cruz 0.322   566  51826      2.4
## 9                  IA     Bremer      Ted Cruz 0.274   408  61216      1.4
## 10                 IA   Buchanan      Ted Cruz 0.368   308  55553      1.4
## # ... with 152 more rows, and 3 more variables: white <dbl>,
## #   college <dbl>, density <dbl>
```

```
Call:
 randomForest(formula = winner ~ income + hispanic + white + college +      density, data = votes)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 2

        OOB estimate of  error rate: 33.95%
Confusion matrix:
             Donald Trump Marco Rubio Ted Cruz class.error
Donald Trump           67           2       28   0.3092784
Marco Rubio             1           6        0   0.1428571
Ted Cruz               24           0       34   0.4137931
```

2.)Applying Naïve Bayes Classifier:

```
library(e1071)
classifier <- naiveBayes(winner ~ income + hispanic + white + college + densi
ty, data = votes)
classifier
```

```
##
## Naive Bayes Classifier for Discrete Predictors
##
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace)
##
## A-priori probabilities:
## Y
## Donald Trump  Marco Rubio     Ted Cruz
##   0.59876543   0.04320988   0.35802469
##
## Conditional probabilities:
##               income
## Y                  [,1]     [,2]
##   Donald Trump 44333.96 9181.732
##   Marco Rubio  55527.29 8745.836
##   Ted Cruz     49607.84 6204.443
##
##               hispanic
```

```
## Y                  [,1]     [,2]
##   Donald Trump 6.556701 6.494788
##   Marco Rubio  5.628571 1.499683
##   Ted Cruz     4.722414 5.877971
##
##               white
## Y                  [,1]      [,2]
##   Donald Trump 73.12680 19.708102
##   Marco Rubio  74.44286 15.438789
##   Ted Cruz     91.61552  7.484576
##
##               college
## Y                  [,1]     [,2]
##   Donald Trump 17.84639 5.192544
##   Marco Rubio  40.62857 7.127813
##   Ted Cruz     18.77069 4.062365
##
##               density
## Y                   [,1]      [,2]
##   Donald Trump  81.70000 100.72958
##   Marco Rubio  354.74286 223.25174
##   Ted Cruz      37.46207  46.34941
```

```r
summary(classifier)
```

```
##         Length Class  Mode
## apriori 3      table  numeric
## tables  5      -none- list
## levels  3      -none- character
## call    4      -none- call
```

```r
nb_test_predict <- predict(classifier,votes[,4:8])
```

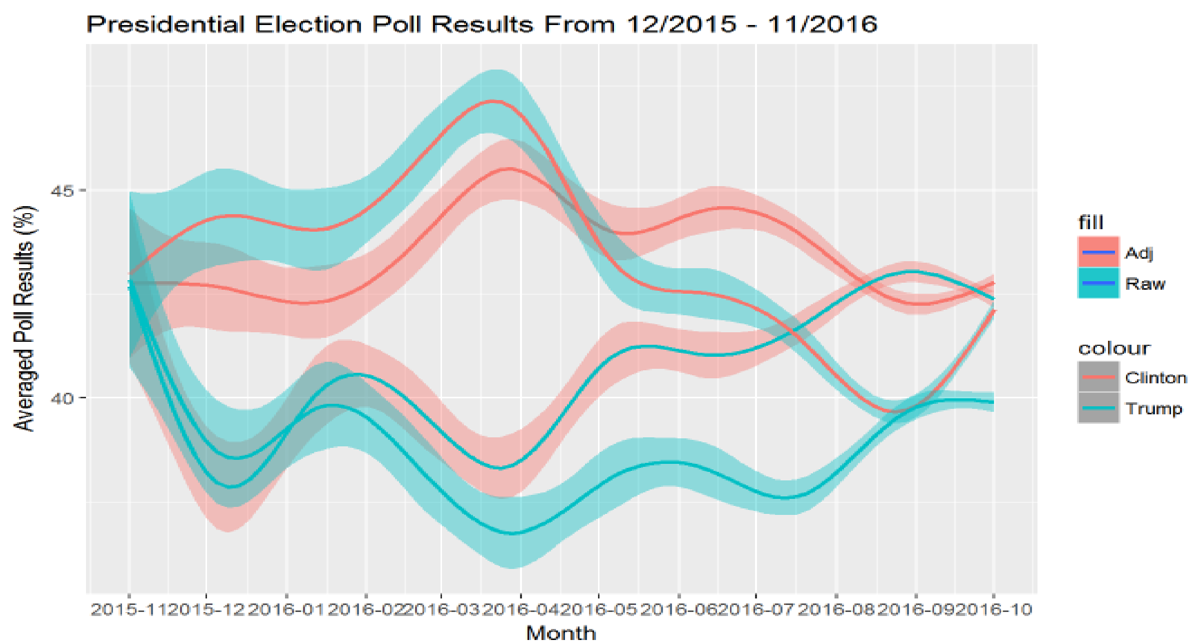
```
#nb_test_predict nb_test_predict)
## nb_test_predict
##                  Donald Trump Marco Rubio Ted Cruz
##   Donald Trump           62           0       35
##   Marco Rubio             3           1        3
##   Ted Cruz                6           0       52
Overall Statistics
##
##                  Accuracy : 0.8
##                    95% CI : (0.6143, 0.9229)
##       No Information Rate : 0.6333
##       P-Value [Acc > NIR] : 0.03992
##
##                     Kappa : 0.6334
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                       Class: Donald Trump Class: Marco Rubio
## Sensitivity                        0.7895            1.00000
## Specificity                        0.9091            0.96429
## Pos Pred Value                     0.9375            0.66667
## Neg Pred Value                     0.7143            1.00000
## Prevalence                         0.6333            0.06667
## Detection Rate                     0.5000            0.06667
## Detection Prevalence               0.5333            0.10000
## Balanced Accuracy                  0.8493            0.98214
##                       Class: Ted Cruz
## Sensitivity                    0.7778
## Specificity                    0.8095
## Pos Pred Value                 0.6364
## Neg Pred Value                 0.8947
```

ANALYSIS INSIGHTS AND GRAPHS

```r
library(choroplethrMaps)
p<-read.csv("presidential_polls.csv")
colum <-names(p)
poll <- fread("presidential_polls.csv",stringsAsFactors = T,select = colum)
poll$enddate <- as.Date(as.factor(poll$enddate), "%m/%d/%Y") # Format date
poll$month <- as.Date(cut(poll$enddate,breaks = "month"))
poll <- poll[order(enddate)] # Order by Date
#Step2
ggplot(data = poll, aes(month)) +
  geom_smooth(aes(y = adjpoll_clinton, colour = "Clinton",fill="Adj")) +
  geom_smooth(aes(y = adjpoll_trump, colour = "Trump",fill="Adj")) +
  geom_smooth(aes(y = rawpoll_clinton, colour = "Clinton",fill="Raw")) +
  geom_smooth(aes(y = rawpoll_trump, colour = "Trump",fill="Raw")) +
  scale_x_date(labels = date_format("%Y-%m"),
               date_breaks = "1 month")  +
  labs(x = "Month", y = "Averaged Poll Results (%)",
       title = "Presidential Election Poll Results From 12/2015 - 11/2016")
```

1.)Raw VS Adjusted Polls over time:

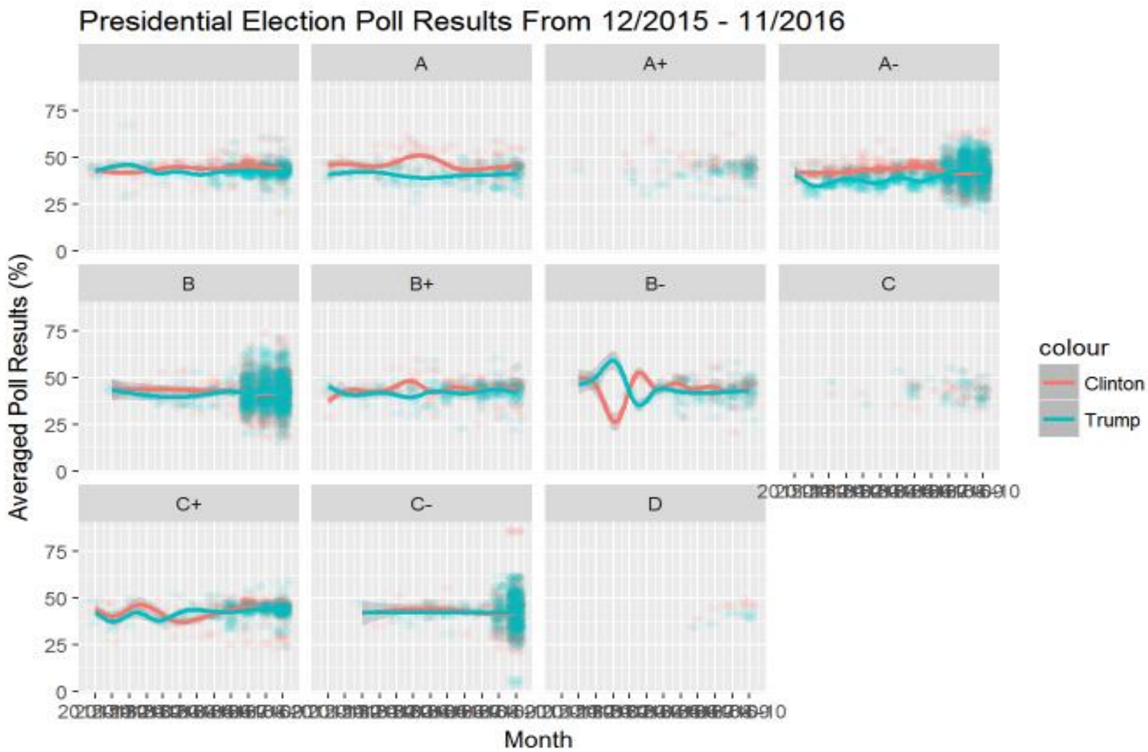
→Clintons adjustments were much higher.




```
# Now Poll Results by Grade -points and lines
ggplot(data = poll, aes(month)) +
geom_jitter(aes(y=adjpoll_clinton,colour="Clinton"),alpha=.04)+
  geom_jitter(aes(y=adjpoll_trump,colour="Trump"),alpha=.04)+
  geom_smooth(aes(y = adjpoll_clinton, colour = "Clinton")) +
  geom_smooth(aes(y = adjpoll_trump, colour = "Trump")) +
  scale_x_date(labels = date_format("%Y-%m"),
               date_breaks = "1 month")  +
  facet_wrap(~grade)+
  labs(x = "Month", y = "Averaged Poll Results (%)",
       title = "Presidential Election Poll Results From 12/2015 - 11/2016")
```

2.) Polling with respect to surveyors(grade wise):

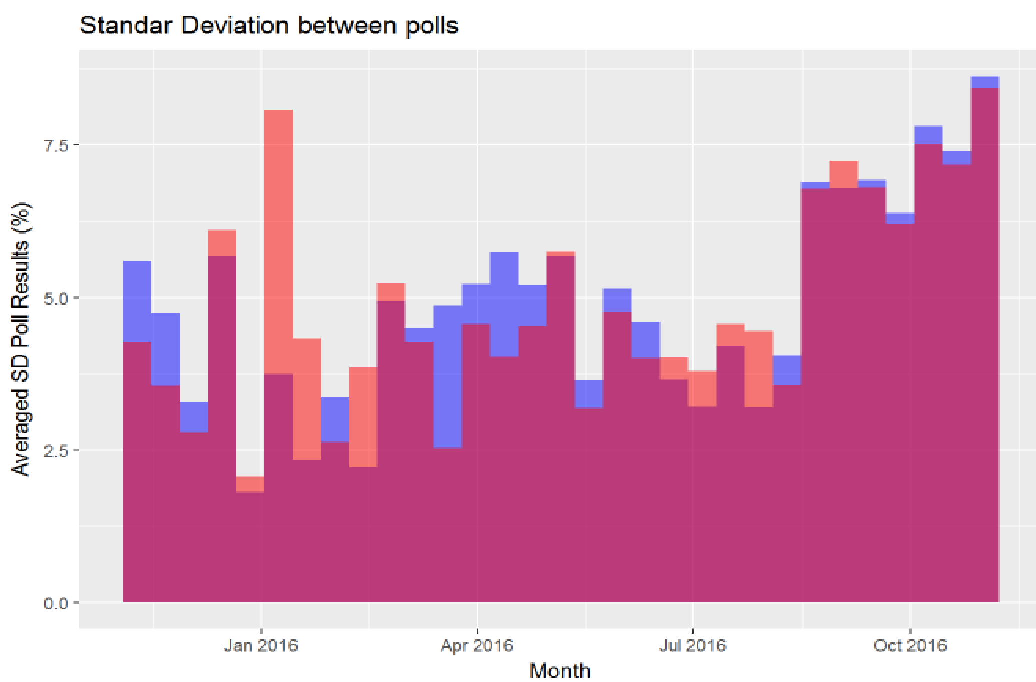
→Lower grade pollsters were better at predicting, perhaps due to their lower adjustments.



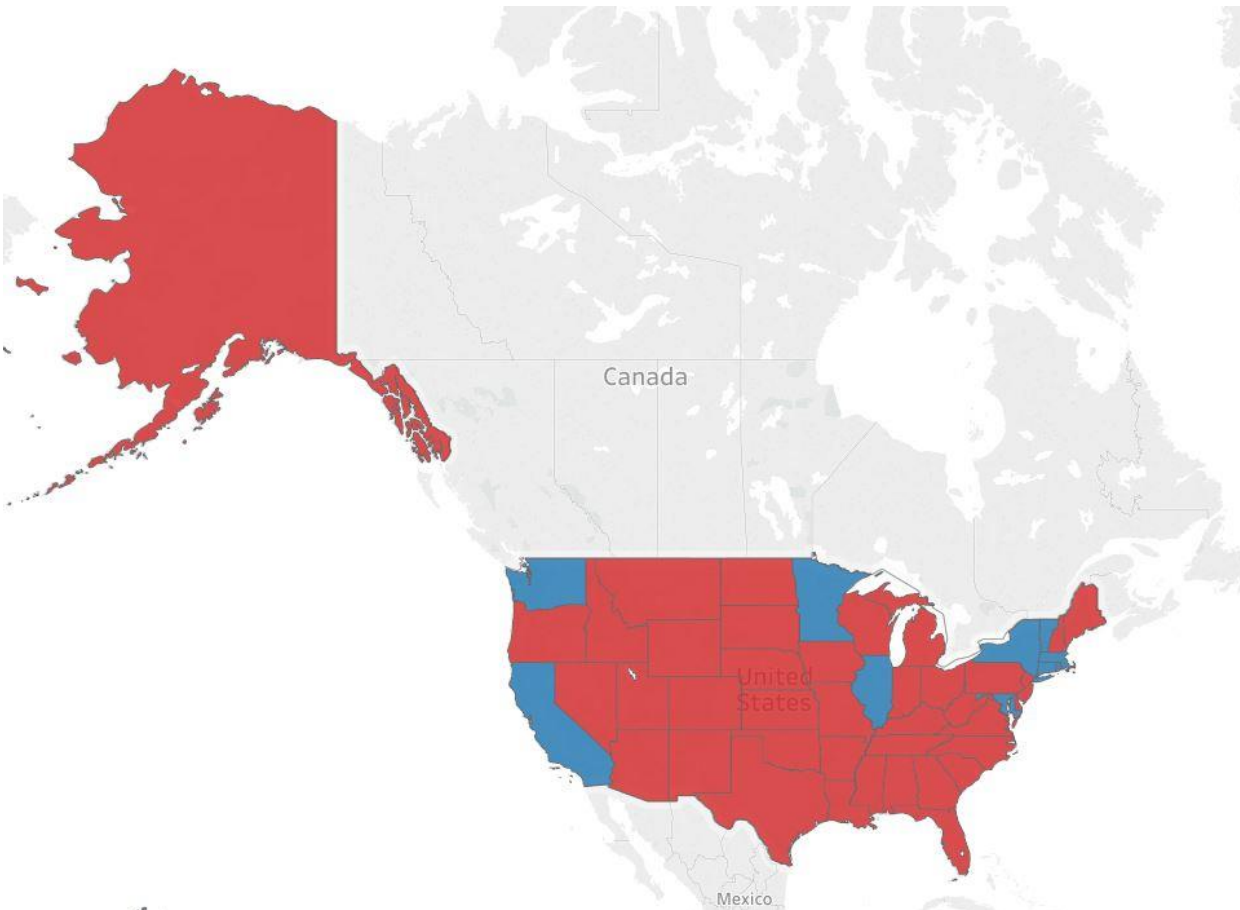
```
The polls tend to converge or not?

 ggplot(poll,aes(x = enddate))+

  stat_summary_bin(aes(y = adjpoll_clinton),fun.y = "sd",geom="bar",fill="blu
e",alpha=.5)+

  stat_summary_bin(aes(y = adjpoll_trump),fun.y = "sd",geom="bar",fill="red",
alpha=.5)+

  ggtitle("Standard Deviation between polls")+labs(x = "Month", y = "Averaged
SD Poll Results (%)")
```

3.)Standard Deviation between polls:

→Here we can see that the difference between polls are increasing, thus they don’t converge.



PREDICTIVE ANALYSIS



Here in order to predict whose winning from each pollster our dataset we used a simple formula:

If(adjClinton-adjTrump)>0 then make the “winner "column as Clinton, Else Trump

Correctness of the formula is confirmand by this map.

Finding the best model for classification?

1.) Applying Neural Networks to Presidential Data for our Formula:

→Two input nodes,

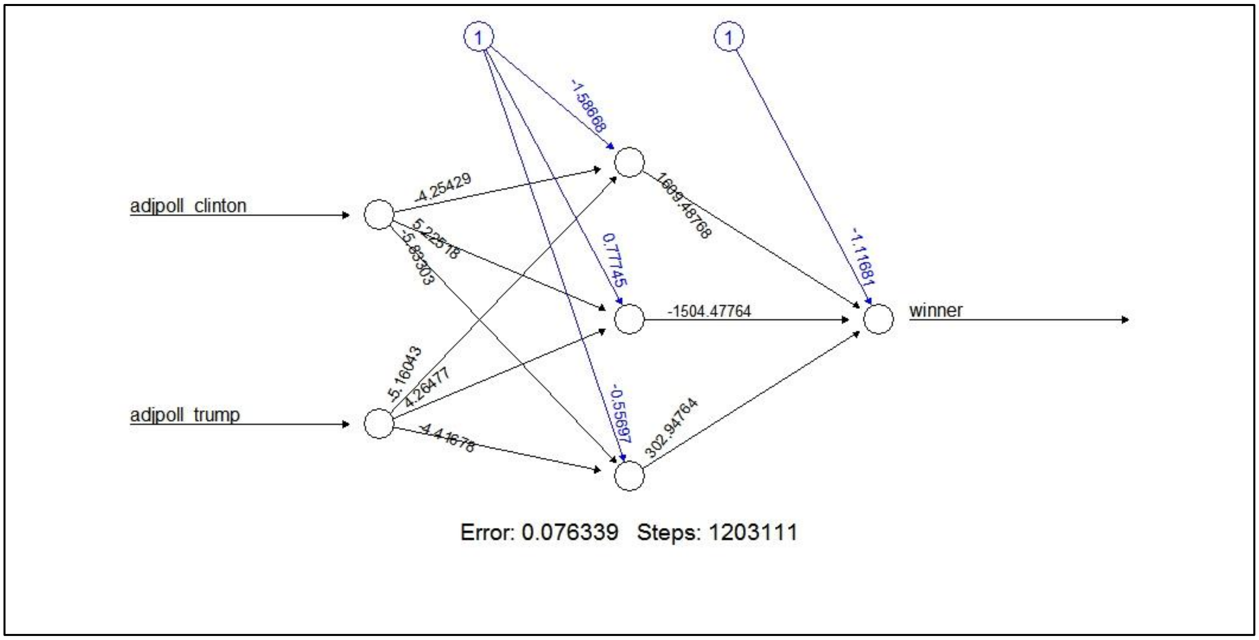
→stepmax to 1e9

→Single hidden layer(3 nodes).

→Trainset(70% of data).

→Testset(30% of data)

```
library(neuralnet)
library(data.table)

#creat vector of column max and min values
maxs <- apply(polls[,1:2], 2, max)
mins <- apply(polls[,1:2], 2, min)

# Use scale() and convert the resulting matrix to a data frame
scaled.data <- as.data.frame(scale(polls[,1:2],center = mins, scale = maxs - mins))

# Check out results
print(head(scaled.data,2))

# Convert Private column from Yes/No to 1/0
winner = as.numeric(polls$winning)-1
data_neos = cbind(winner,scaled.data)

data_reduced<-head(data_neos,2000)
data_reduced1

library(caTools)
set.seed(101)


# Create Split (any column is fine)
split2<-sample.split(data_reduced1$winner,SplitRatio = 0.70)

# Split based off of split Boolean Vector
train_set1<-subset(data_reduced1,split=TRUE)
test_set1<-subset(data_reduced1,split=FALSE)
train_set1

feats<-names(scaled.data)

# Concatenate strings and Convert to formula
func_neurons<-paste(feats1,collapse = '-')
func_neurons<-paste('winner ~',func_neurons)
func_neurons<-as.formula(func_neurons)
func_neurons

#train neural net
neuronsZ1<-neuralnet(func_neurons,train_set1,hidden = c(3),linear.output = FALSE,stepmax = 1e9)
plot(neuronsZ1)
```




```r
#predict results
predict2<-compute(neuronsZ,test_set1[,1:2])
#print results
print(head(predict2$net.result))
#confusion matrix
table(test_set1$winner,predict2$net.result)

```

```
> neuronsZ1$result.matrix
                                        1
error                        7.633866e-02
reached.threshold            9.983591e-03
steps                        1.203111e+06
Intercept.to.1layhid1       -1.586678e+00
adjpoll_clinton.to.1layhid1 -4.254294e+00
adjpoll_trump.to.1layhid1   -5.160427e+00
Intercept.to.1layhid2        7.774471e-01
adjpoll_clinton.to.1layhid2  5.225184e+00
adjpoll_trump.to.1layhid2    4.264773e+00
Intercept.to.1layhid3       -5.569675e-01
adjpoll_clinton.to.1layhid3 -5.833030e+00
adjpoll_trump.to.1layhid3   -4.416782e+00
Intercept.to.winner         -1.116810e+00
1layhid.1.to.winner          1.609488e+03
1layhid.2.to.winner         -1.504478e+03
1layhid.3.to.winner          3.029476e+02
```

Confusion Matrix for Neural Networks

```
     0
0 1166
1  834
```

Result: Ann is not efficient for classification as it only predicts partially.

2.) Applying Naïve Baye's to Presidential Data for our Formula:

→Folds=10,Trainset(90%),test(10%).

→82% accuracy

3.) Applying Support Vector Machine's:

→Folds=10,Trainset(90%),test(10%).

→84.5% accuracy

```
polls1 <- fread('presidential_polls.csv', stringsAsFactors = TRUE, select = c
("type", "state", "enddate", "pollster", "grade", "samplesize", "population",
"poll_wt", "rawpoll_clinton", "rawpoll_trump", "adjpoll_clinton", "adjpoll_tr
ump", "multiversions", "poll_id"), showProgress = TRUE)

polls1$grade <- sub("^$","F",polls1$grade)

polls1$grade <- factor(polls1$grade, ordered = TRUE, levels = c("F","D","C-",
"C","C+","B-","B","B+","A-","A","A+"))

polls1$enddate <- as.Date(polls1$enddate, format = "%m/%d/%Y")

polls1 <- na.omit(polls1)

for (i in 1:nrow(polls1)) {

  if(polls1$adjpoll_clinton[i]<polls1$adjpoll_trump[i]) {

    polls1[i,'labels'] <- as.factor('trump')

  }

  else {

    polls1[i,'labels'] <- as.factor('clinton')

  }

}


folds <- cvFolds(nrow(polls1), K=10, type = "random")

index <- 1:round(nrow(folds$subsets)*0.9)
```

```r
control<-trainControl(method = "repeatedcv",number = 10,repeats = 3)

# CART

set.seed(7)

fit.cart <- train(labels~., data=polls1[-index,-1], method="rpart", trControl
=control)
```

```
## Loading required package: rpart
```

```r
# SVM

set.seed(7)

fit.svm <- train(labels~adjpoll_clinton - adjpoll_trump, data=polls1[-index,-
1], method="svmRadial", trControl=control)

# naive_bayes

set.seed(7)

fit.nb <- train(labels~adjpoll_clinton - adjpoll_trump, data=polls1[-index,-1
], method="nb", trControl=control)

#confusion matrix of svm and naive bayes

predicted_svm <- predict(fit.svm, polls1[index,-1])

cm_svm <- confusionMatrix(predicted_svm, polls1$labels[index])

cm_svm
```

```
## Confusion Matrix and Statistics
##
##             Reference
## Prediction clinton trump
##    clinton    5428  1243
##    trump       182  2357
##
##                Accuracy : 0.8453
##                  95% CI : (0.8377, 0.8526)
##     No Information Rate : 0.6091
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.657
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9676
```

```
##             Specificity : 0.6547
##          Pos Pred Value : 0.8137
##          Neg Pred Value : 0.9283
##              Prevalence : 0.6091
##          Detection Rate : 0.5894
##    Detection Prevalence : 0.7243
##       Balanced Accuracy : 0.8111
##
##        'Positive' Class : clinton
```

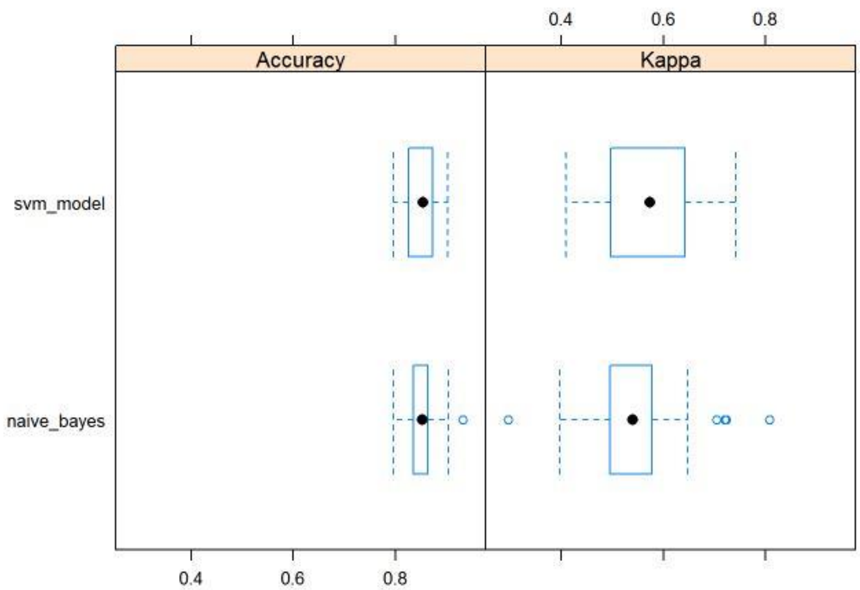
```
predicted_nb <- predict(fit.nb, polls1[index,-1])
cm_nb<- confusionMatrix(predicted_nb,polls1$labels[index])
cm_nb
```

```
## Confusion Matrix and Statistics
##
##             Reference
## Prediction clinton trump
##    clinton    5512  1569
##    trump        98  2031
##
##                Accuracy : 0.819
##                  95% CI : (0.811, 0.8268)
##     No Information Rate : 0.6091
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.5899
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9825
##             Specificity : 0.5642
##          Pos Pred Value : 0.7784
##          Neg Pred Value : 0.9540
##              Prevalence : 0.6091
##          Detection Rate : 0.5985
##    Detection Prevalence : 0.7688
```

```
##       Balanced Accuracy : 0.7733
##
#results
results <- resamples(list(svm_model=fit.svm, naive_bayes=fit.nb))


#summary
summary(results)
##
## Call:
## summary.resamples(object = results)
##
## Models: svm_model, naive_bayes
## Number of resamples: 30
##
## Accuracy
##                Min. 1st Qu. Median   Mean 3rd Qu.   Max. NA's
## svm_model   0.7961  0.8252 0.8537 0.8492  0.8725 0.9020    0
## naive_bayes 0.7961  0.8370 0.8522 0.8541  0.8627 0.9314    0
##
## Kappa
##                Min. 1st Qu. Median   Mean 3rd Qu.   Max. NA's
## svm_model   0.4090  0.4992 0.5728 0.5666  0.6397 0.7420    0
## naive_bayes 0.2966  0.4950 0.5403 0.5456  0.5764 0.8078    0
```

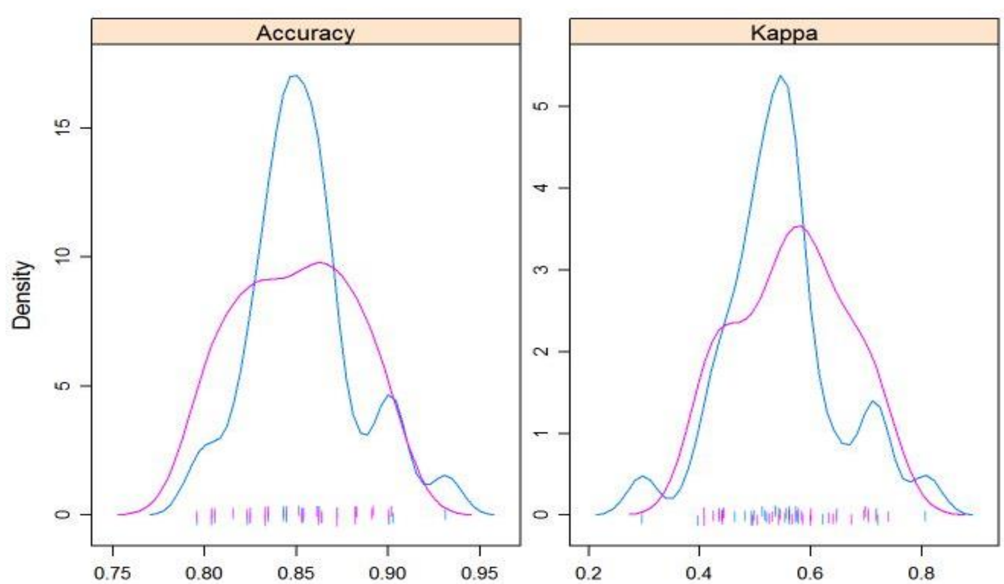
PREDCTIVE ANAYLSIS(EXTENDED)

```
#boxplot
bwplot(results)
```



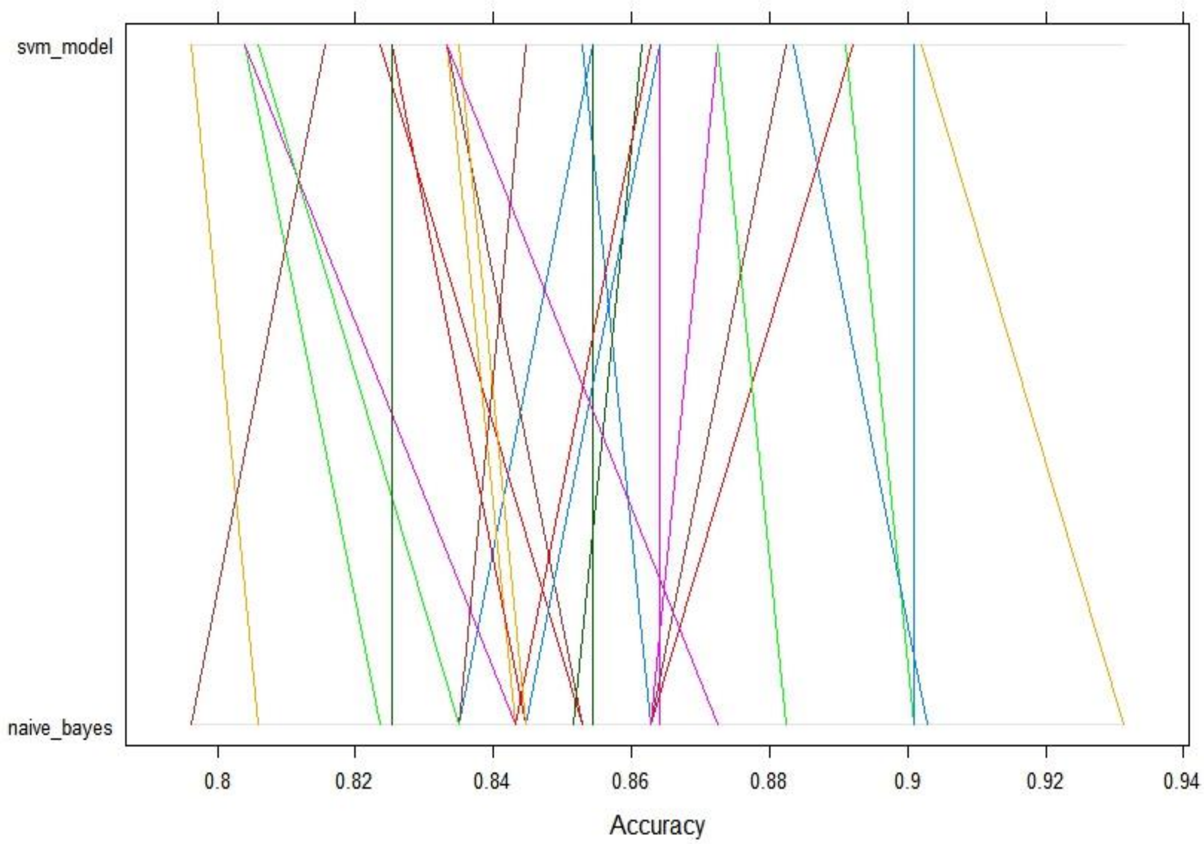
Svm's have the highest mean accuracy

```
# density plots of accuracy
scales <- list(x=list(relation="free"), y=list(relation="free"))
densityplot(results, scales=scales, pch = "|")
```



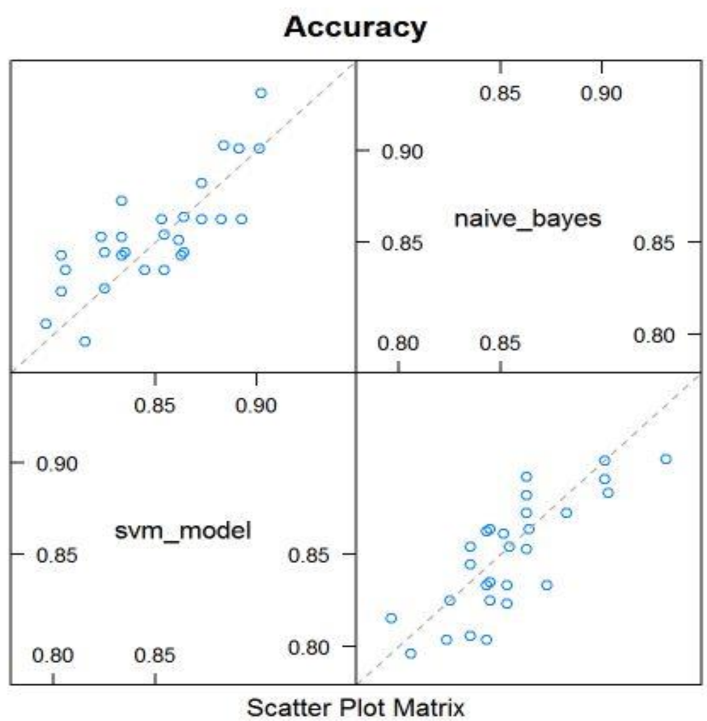
Parallel Plots and Scatter Plots:

1.) Each trial of each cross validation fold behaved for svm and naïve bayes in a random manner.



2.)Svm and naïve bayes are strongly correlated to a certain extent.

```
#scatter plots
splom(results)
```



CONCLUSION

1. The Adjusted Poll was mostly influenced by the grade of the surveyor, population type, sample size and poll weightage apart from the raw poll counts.
2. Surprisingly lower grade surveyors and registered voter type populations consistently predicted the actual winner in most cases.
3. We ran two classification models to see which variables are major contributors in primaries and the main takeaway however is, Donald Trump seems to have a much broader appeal than his two main rivals, at least among Republican primary voters and he is most successful in counties that have:
   - low median income
   - low college attainment
   - large(er) hispanic population.
4. Higher Adjustment in votes for Clinton than trump, perhaps due to media bias.
5. Formulation of class labels which match the final outcome of the 2016 election results, using adjusted votes.
6. Using the above prediction formula we used three classifiers to find the best result, and concluded that naïve Bayes and SVM's both works quite efficiently in prediction on test set.