# Examining Machine Learning Models to predict constituents' voting from the 2016 election

Ellen Burrell: PSTAT 131

## Contents

```
# Libraries imported
library(tidyverse)
library(tidymodels)
library(ISLR)
library(rpart.plot)
library(vip)
library(janitor)
library(randomForest)
library(xgboost)
library(here)
library(tune)
library(parsnip)
```

## Introduction

American Politics are becoming increasingly partisan at an alarming rate. While the United States has two main political parties, the division among primary voters for each party is increasing. According to a study done by the Pew Research Center, "Across 30 political values – encompassing attitudes about guns, race,

immigration, foreign policy and other realms – the average partisan gap is 39 percentage points" between the two parties. The average difference for other grouping factors such as Education, Age, or Gender were around 6-10 percentage points (Nadeem). From this data, we can conclude that the political values that each party represents must be antithetical to have created such a sharp division.

If given such data, could we predict how a person voted in the 2016 presidential election? Would someone's party affiliation or political values be more important in determining who they voted for?

In this project we look at the data from 'The Views of the Electorate Research (VOTER) Survey' conducted by YouGov. Although we will be looking at data from the 2016 election, the respondents were originally interviewed in 2011, 2012, and then in 2016 for a third time. The entire dataset contains responses by 8,000 adults (age 18+) with internet access on 668 questions.

The questions in this survey are VERY comprehensive.

To begin, the survey asks who one voted for in the 2016 election. Then, some of the questions ask about opinions on a specific person.

**"Do you have a favorable or an unfavorable opinion of the following people? BARACK OBAMA"**

Or about honesty in US Elections. . .

**"How confident are you that the votes in the 2016 election across the country were accurately counted?"**

Or about general opinions.

**"Which do you think is more important for a child to have? Curiosity OR Good manners?"**

And some biographical questions such as

**"Have you smoked at least 100 cigarettes in your entire life?"**

Some of the most valuable data comes from a section labeled "Feeling Thermometer", where the survey asks respondents to rate various groups on a scale from 0-100. The question is framed as

**"We'd like to get your feelings toward some groups who are in the news these days. Ratings between 50 degrees and 100 degrees mean that you feel favorable and warm toward the group. Ratings between 0 degrees and 50 degrees mean that you don't feel favorable toward the group and that you don't care too much for that group. You would rate the group at the 50 degree mark if you don't feel particularly warm or cold toward the group. If we come to a group who you don't recognize, you don't need to rate that group. Click on the thermometer to give a rating."**

The survey then frames this question about each upcoming group individually - 'Blacks','Whites','Hispanics','Asians','Muslims' 'Immigrants','Black Lives Matter', 'Wall Street Bankers', 'Gays and Lesbians', 'Labor Unions', 'Police Officers', and the 'The alt-right movement'.

For more information on the Survey, look at the citations at the bottom of this report.

## Variable Selection and Data Cleaning

```r
setwd("~/pstat/131/pstat131/final_project_edited")
VOTER_Survey_December16_Release1 <- read_csv("archive/VOTER_Survey_December16_Release1.csv")
# only taking data from 2016 survey respondent
cleaned_data_2016 <- VOTER_Survey_December16_Release1 %>%
  dplyr::select(ends_with("2016"))
#dim(cleaned_data_2016)
```

By only looking at the questions asked in 2016, we cut down our variables from 668 to 272.

Although it would be interesting to analyze, I didn't think it would be realistic or possible to use all 272 variables. Therefore I choose 25 questions that I found the most interesting. These included all of the feeling thermometer questions, a question on smoking, drinking, religion, opinions about the accuracy of ballot counting, one's political affiliation, and ideology. To see the exact breakdown of questions, I invite you to look at the survey_codebook.txt within the data file.

```
more_cleaned_data_2016 <- VOTER_Survey_December16_Release1 %>%
  dplyr::select(ends_with("2016")) %>%
  dplyr::select(starts_with(c("izip_2016","presvote16post_","ft","accurately_counted2","alcohol","smoke
  dplyr::select(-c("presvote16post_t_2016","presvote16post_rnd_2016"))
#dim(more_cleaned_data_2016)
```

From here I had some importing cleaning to do of the variables. First of all, all of the answers to the surveys were in strings, so I had to convert all of the integers within strings to numeric values. This was not easy, because some of feeling thermometer questions had strings before the numbers as shown below.

```
table(more_cleaned_data_2016$ft_white_2016)
```

```
##
## 0 - Unfavorable feeling                        1                10
##                       8                        5                11
## 100 - Favorable feeling                       11                12
##                     879                       16                 6
##                      14                       15                16
##                       4                        4                 3
##                      17                       18                19
##                       3                        2                 5
##                       2                       20                21
##                       6                       10                 8
##                      22                       23                24
##                       5                        4                 6
## 25 -Unfavorable feeling                       26                27
##                      25                       11                 7
##                      28                       29                 3
##                      14                        4                 6
##                      30                       31                32
##                      20                       29                 6
##                      33                       34                35
##                       6                        9                13
##                      36                       37                38
##                      13                        9                 7
##                      39                        4                40
##                      15                        5                37
##                      41                       42                43
##                      40                       13                13
##                      44                       45                46
##                      21                       25                15
##                      47                       48                49
##                      25                       30                69
##                       5  50 - No feeling at all                51
##                       6                      750               244
##                      52                       53                54
```

```
##                     101                        87                      32
##                      55                        56                      57
##                      46                        49                      30
##                      58                        59                       6
##                      24                        54                       8
##                      60                        61                      62
##                     115                        61                      33
##                      63                        64                      65
##                      48                        26                      52
##                      66                        67                      68
##                      61                        30                      40
##                      69                         7                      70
##                      97                         2                     152
##                      71                        72                      73
##                      86                        80                      43
##                      74  75 - Favorable feeling                       76
##                      70                       285                     127
##                      77                        78                      79
##                      89                       137                     188
##                       8                        80                      81
##                       1                       370                     220
##                      82                        83                      84
##                      57                        64                      97
##                      85                        86                      87
##                     123                        81                      59
##                      88                        89                       9
##                      94                       137                       5
##                      90                        91                      92
##                     372                       208                      82
##                      93                        94                      95
##                      92                       126                     175
##                      96                        97                      98
##                     120                       138                     152
##                      99                  Don't know
##                     282                        97
```

Therefore I had to re-code all of those values as strings, and then convert them to integers.

```
more_cleaned_data_2016$ft_white_2016 <- recode(more_cleaned_data_2016$ft_white_2016, "100 - Favorable fe
more_cleaned_data_2016$ft_white_2016 <- recode(more_cleaned_data_2016$ft_white_2016, "75 - Favorable fee
more_cleaned_data_2016$ft_white_2016 <- recode(more_cleaned_data_2016$ft_white_2016, "25 -Unfavorable fe
more_cleaned_data_2016$ft_white_2016 <- recode(more_cleaned_data_2016$ft_white_2016, "50 - No feeling at
more_cleaned_data_2016$ft_white_2016 <- recode(more_cleaned_data_2016$ft_white_2016, "0 - Unfavorable fe
more_cleaned_data_2016$ft_white_2016 <- recode(more_cleaned_data_2016$ft_white_2016, "Don't know" = "400

# changing the strings to integers
more_cleaned_data_2016 <- more_cleaned_data_2016 %>%
  mutate(ft_white_2016 = as.numeric(ft_white_2016))
#is.numeric(more_cleaned_data_2016$ft_white_2016) #TRUE
```

I cleaned all of the feeling thermometer questions in this matter, and had to decide what to do about answers such as "Don't Know" and values unanswered (NA). While I experimented with different approaches to this problem, I ultimately decided that the best solution was to code any answers of "Don't Know" as NA, and then remove all un-answered observations. This was a tough call to make, because I didn't want to lose any

data or statistic power. However, the goal of this project for me was to look at opinions of respondents and who they subsequently voted for, so I interpreted someone as not answering or answering "Don't Know" as not very clear opinions. Additionally, I didn't want to make any assumptions for the respondent if they didn't respond, so I decide to not include the rest of their data.

To see more specifics of my data cleaning, I invite people to look at the cleaning folder.

## Explanatory Data Analysis (EDA)

Here I am reading in the cleaned data file, and factoring all of the variables that are non-numeric.

```r
# reading in our data
survey <- read_csv("data/no_na.csv")
# factoring data
survey <- survey %>%
  mutate(accurately_counted2_2016 = factor(accurately_counted2_2016, levels = c("Not at all confident",
                                                        "Somewhat confident","Ve
  mutate(alcohol_2016 = factor(alcohol_2016, levels = c("Yes","No"),exclude=NULL)) %>%
  mutate(smoke100_2016 = factor(smoke100_2016, levels = c("Yes","No"),exclude=NULL)) %>%
  mutate(pid7_2016 = factor(pid7_2016)) %>%
  mutate(ideo5_2016 = factor(ideo5_2016, levels = c("Very liberal","Liberal","Not sure","Moderate","Cons
  mutate(pew_religimp_2016 = factor(pew_religimp_2016, levels = c("Not at all important","Not too import
  mutate(presvote16post_2016 = factor(presvote16post_2016))
```

Let's take a look at our outcome variable, presvote16post_2016.

```r
ggplot(data = survey, mapping = aes(fill=presvote16post_2016,x = reorder(factor(presvote16post_2016), pr
```
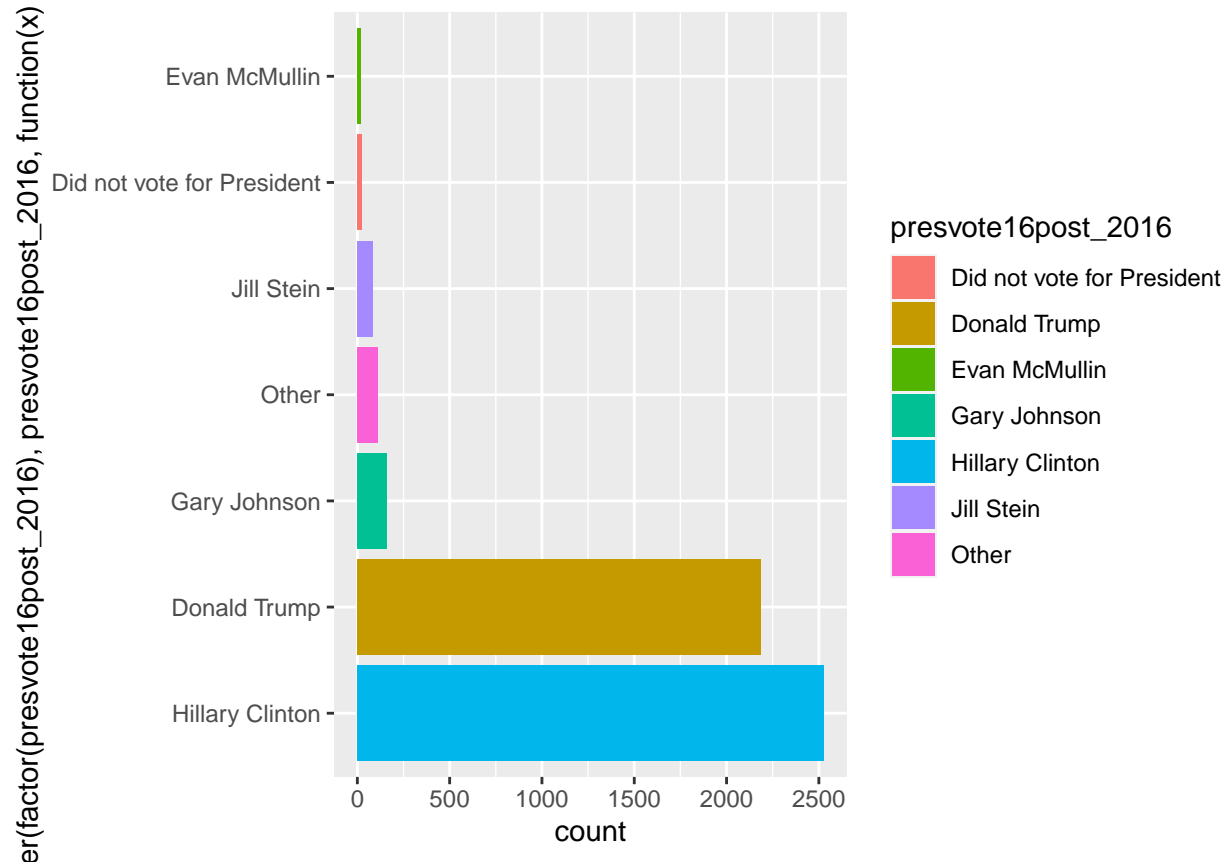
The survey has the most responses for Hilary Clinton and Donald Trump, with only a small fraction representing responses for Evan McMullin, Did not vote for President, Jill Stein, Other, and Gary Johnson.

Let's look at some plots with two different feeling thermometer scales on each axis, with presidental vote as the color highlighting the differences.

**Feeling thermometers for Police and Labor Unions**

```
ggplot(survey, aes(ft_police_2016, ft_unions_2016, colour = presvote16post_2016)) +
  geom_point()
```
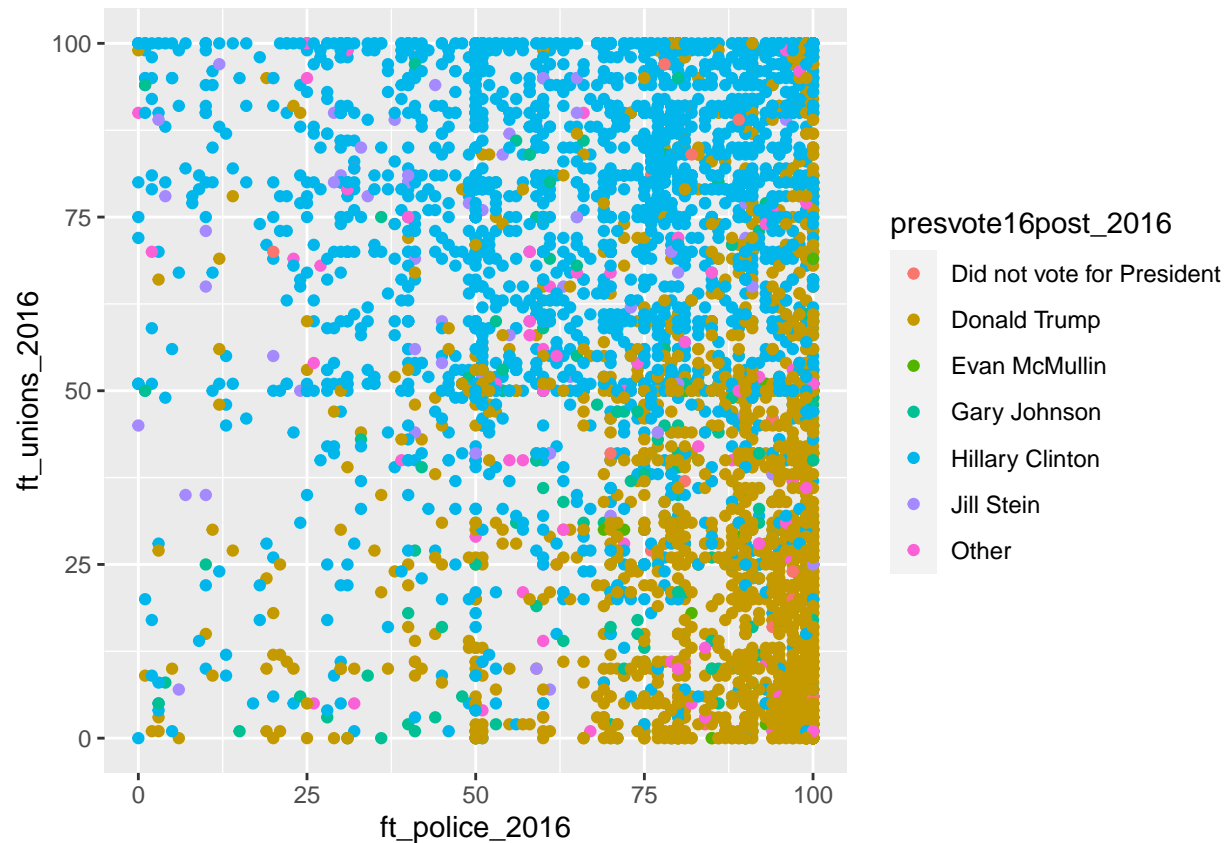
Here we see a large clustering of yellow dots in the lower right hand corner, and a cluster of blue dots in the right upper most quadrant. It seems as if the voters for Donald Trump tend to rank police officers with a high rating and labor unions with a lower than 50 rating. Clinton supporters tend to be more variable, but on average tend to rank police officers pretty highly, as well as labor unions above 50 points.

**Feeling thermometers for BLM and Feminists**

```
ggplot(survey, aes(ft_blm_2016, ft_fem_2016, colour = presvote16post_2016)) +
  geom_point()
```
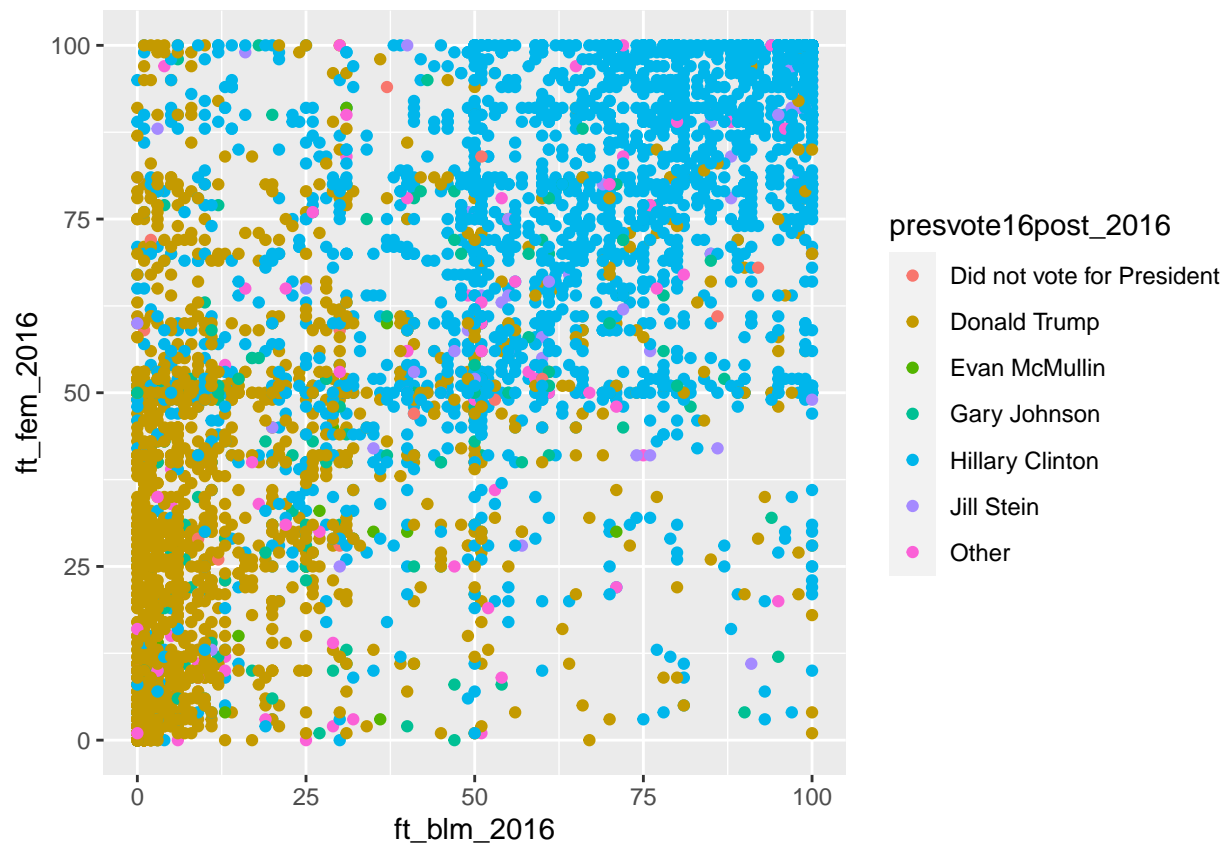
This seems to be our most polarized graph, with one large yellow cluster in the left bottom quadrant, and a strong blue cluster in the upper right quadrant. People who voted for Hillary Clinton tend to rank Feminists and BLM higher than those who voted for Trump. I also think it's worthwhile to mention that the other nominees (Evan McMullin, Gary Johnson, Hilary Clinton, Jill Stein, Other, Did not vote for President) seem to be scattered throughout, with no real strong cluster noticeable. We might be able to notice the purple dots (Jill Stein) following the trend of blue towards the middle right, but it's not very clear.
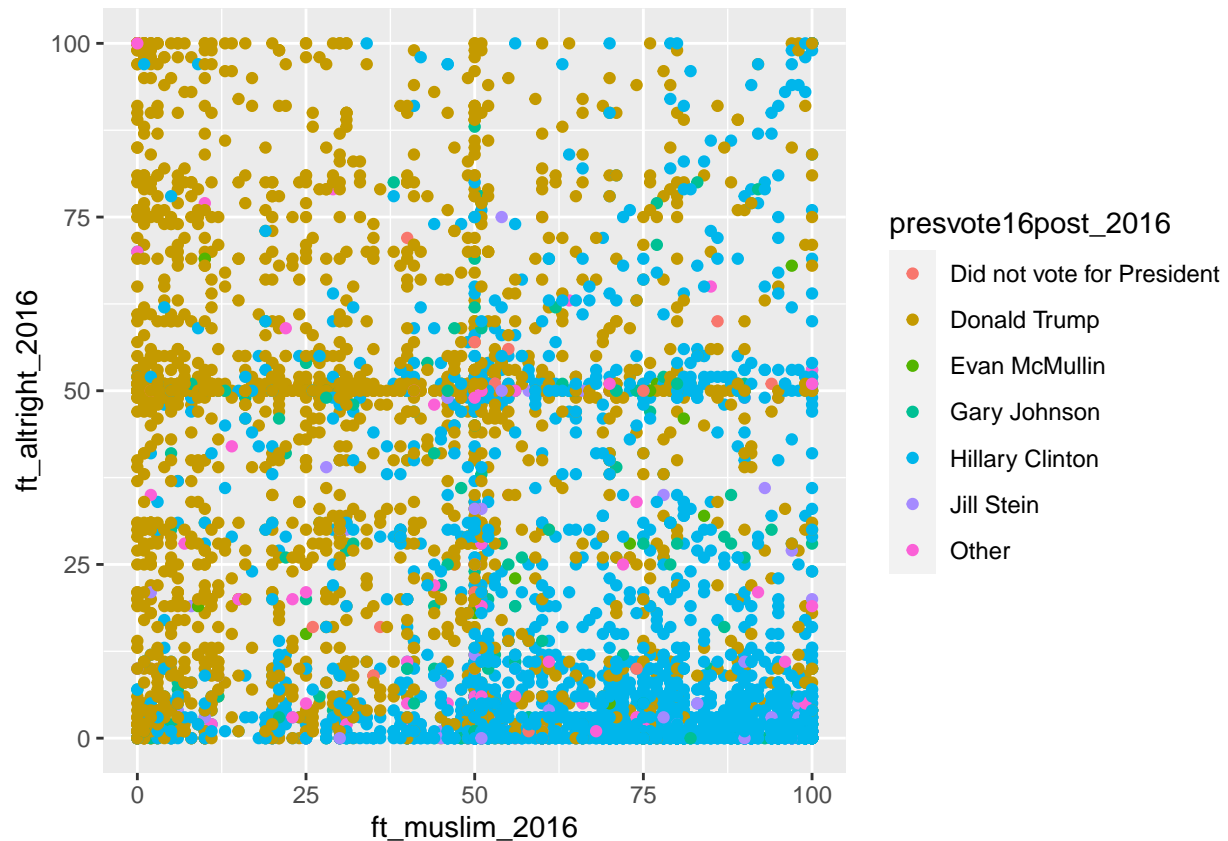
**Feeling thermometers for Jews and Christians**

```
ggplot(survey, aes(ft_jew_2016, ft_christ_2016, colour = presvote16post_2016)) +
  geom_point()
```
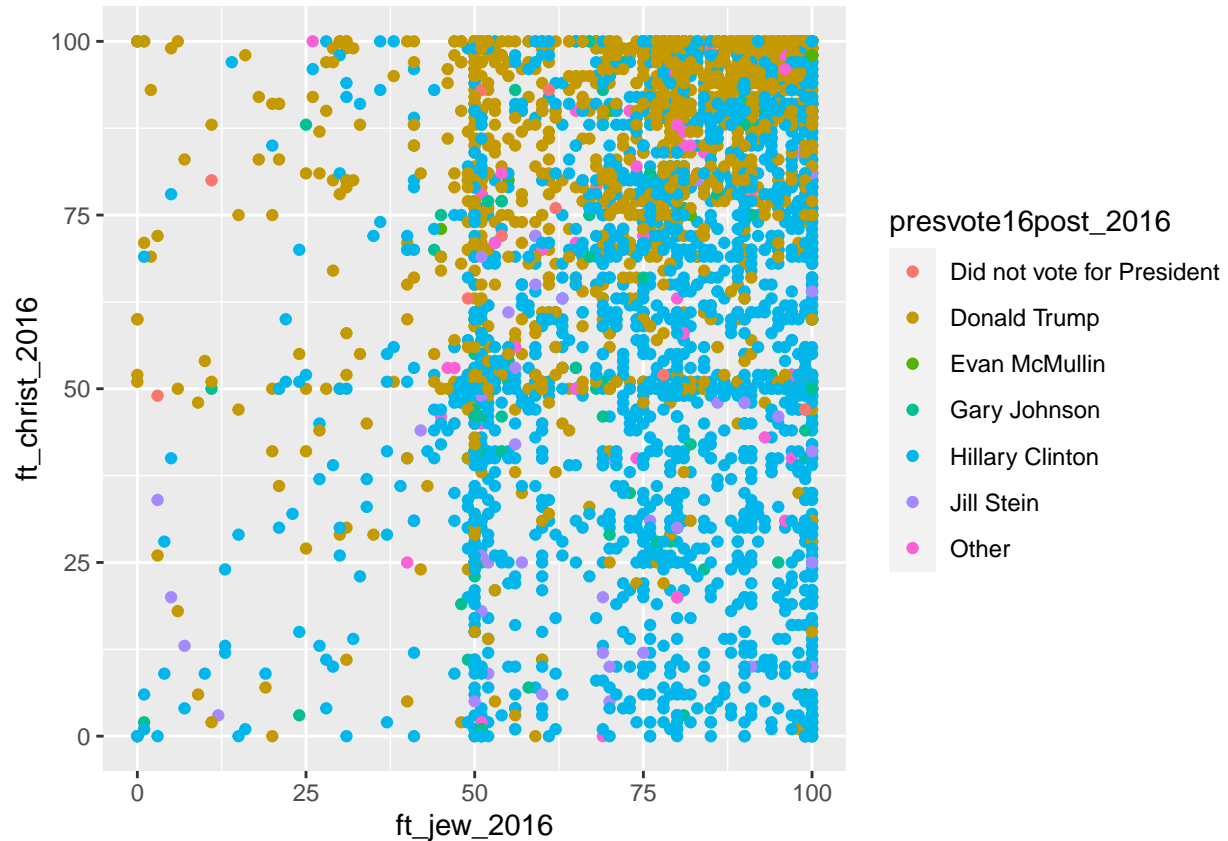
From this graph, we see a very interesting pattern. People who voted for Hilary Clinton tended to cluster near the 50 mark and on wards for ranking Jewish people, yet are very variable from 0-100 ranking Christians. People who voted for Donald Trump cluster near the top right hand quarter, ranking both Christians and Jews favorably.

Now, let's divide on political ideology.

**Ideology dividing feeling thermometer of feminist and BLM**

```
ggplot(survey, aes(ft_blm_2016, ft_fem_2016, colour = ideo5_2016)) +
  geom_point()
```
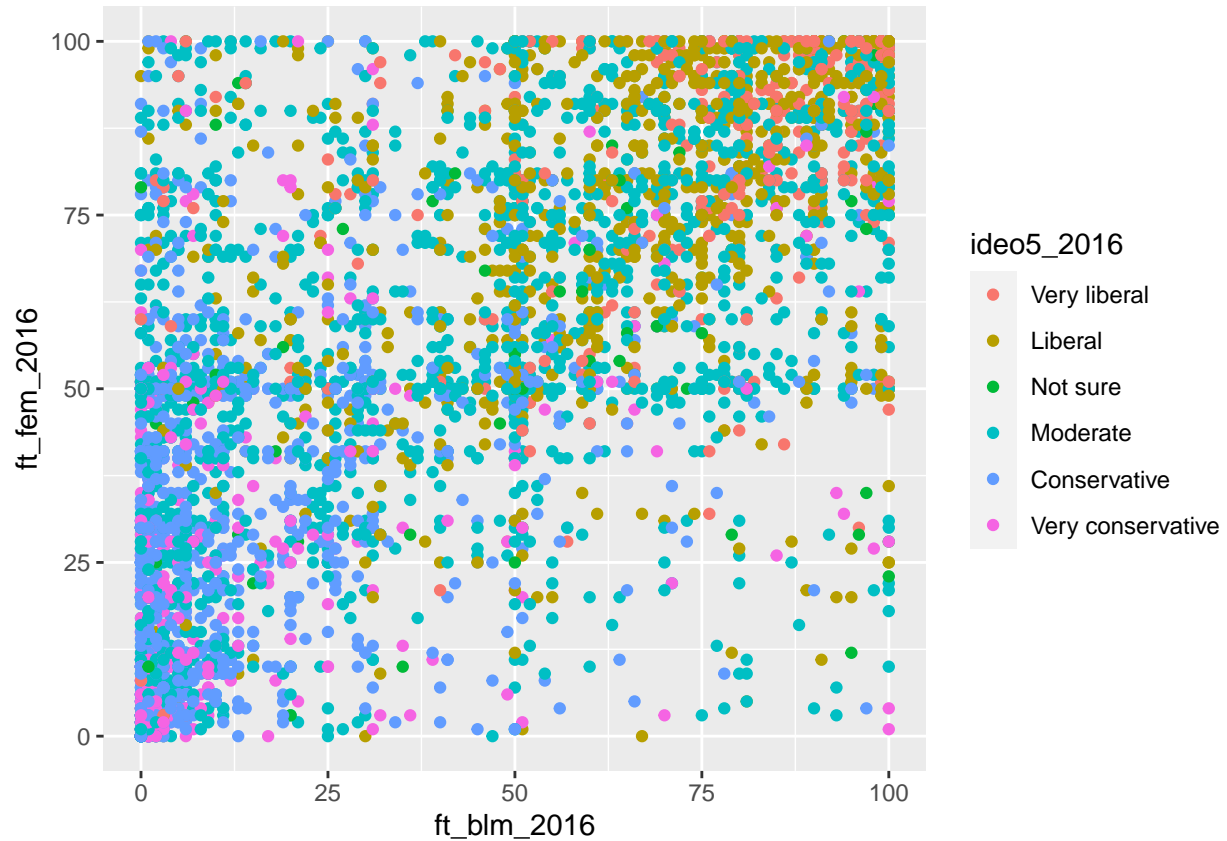
When we divide on ideology, the results are not as clear. There seems to be a small cluster of pink, purple, and blue in the bottom left corner, which would depict moderate, conservative and very conservative as ranking feminists and BLM low. There also seems to be a red and yellow cluster in the top right corner, showing liberal and very liberal as ranking feminists and BLM high. Also, there seems to be a long green cluster around the 50 mark, depicting 'Not sure' people as staying around the halfway mark of ranking. However these results are not as polarized, and for the most part the colors are scattered throughout.

**Ideology dividing feeling thermometer of Asians and Hispanics**

```
ggplot(survey, aes(ft_asian_2016, ft_hisp_2016, colour = ideo5_2016)) +
  geom_point()
```
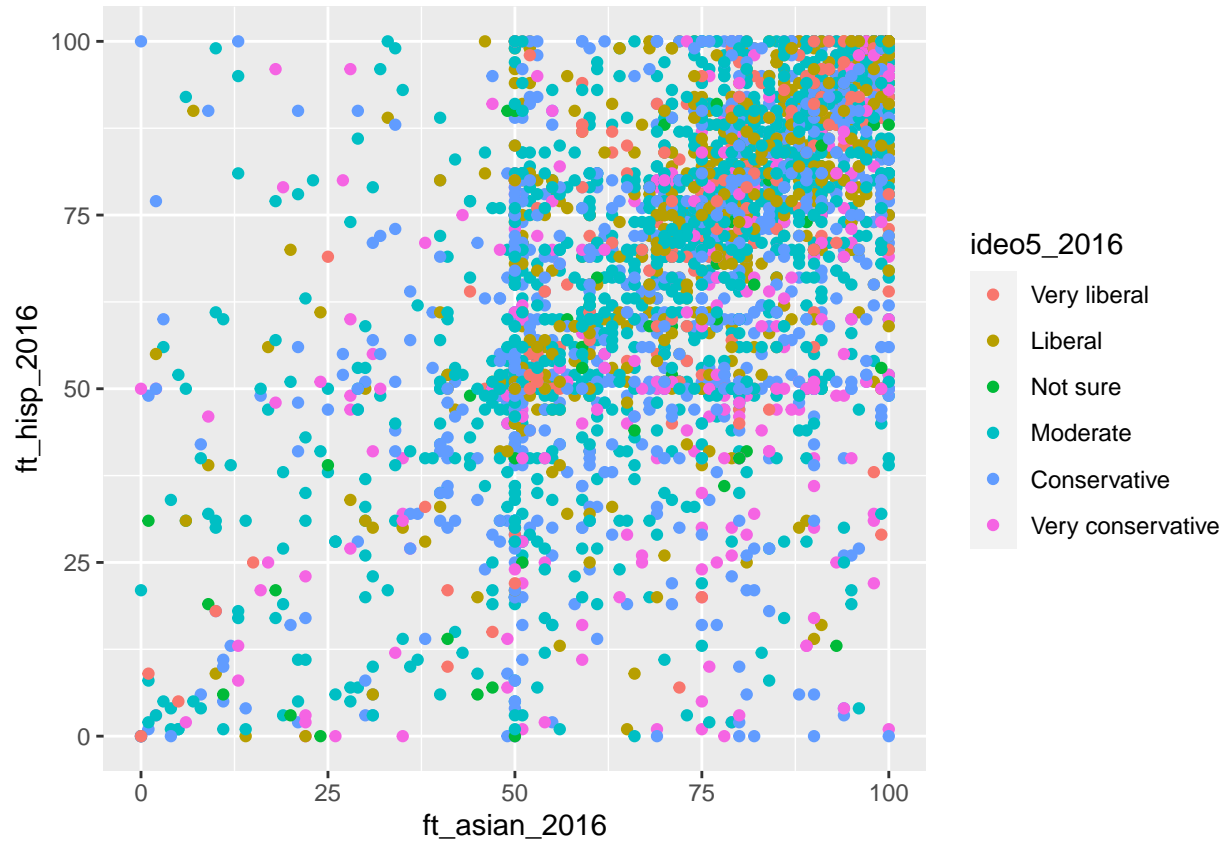
From this plot, we can see a huge cluster of all ideologies in the upper right corner, depicting a general positive feeling thermometers for 'Hispanics' and 'Asians'. There doesn't seem to be any huge polarization within ideology for these feeling thermometers.

For the next two plots, let's keep the feeling thermometer for 'the alt-right movement' on the y-axis, and look at the differences in plots for feeling thermometer of 'Christians' and 'Muslims'.

**Feeling thermometers for Alt-right and Christians**

```
ggplot(survey, aes(ft_christ_2016, ft_altright_2016, colour = presvote16post_2016)) +
  geom_point()
```

Here we see a clustering at the bottom of the graph of people who voted for Hilary Clinton, depicting a general trend of low rankings for the Alt-right movement, and various feelings towards Christians. As we look at the yellow clustering towards the very right of the graph, we can see people who voted for Trump tended to rank their feelings towards Christians highly, and their opinions towards the Alt-right movement varied substantially.

```
ggplot(survey, aes(ft_muslim_2016, ft_altright_2016, colour = presvote16post_2016)) +
  geom_point()
```

However when we switch Christians for Muslims, we see a complete reversal in the opinions of people who voted for Trump. While there is some cluster near the left of the graph, the feeling thermometers for Muslims are more varied than Christians for the Trump supporters. For the people who voted for Clinton, there is the same general trend of low support for the Alt-right movement, but a general higher ranking for Muslims than for Christians.

Lastly, let's look at a correlation plot of all of the feeling thermometers.

**Correlation Plot**

```
library(corrplot)
edited_survey <- survey %>% select(starts_with("ft"))
M <- cor(edited_survey)
corrplot(M)
```
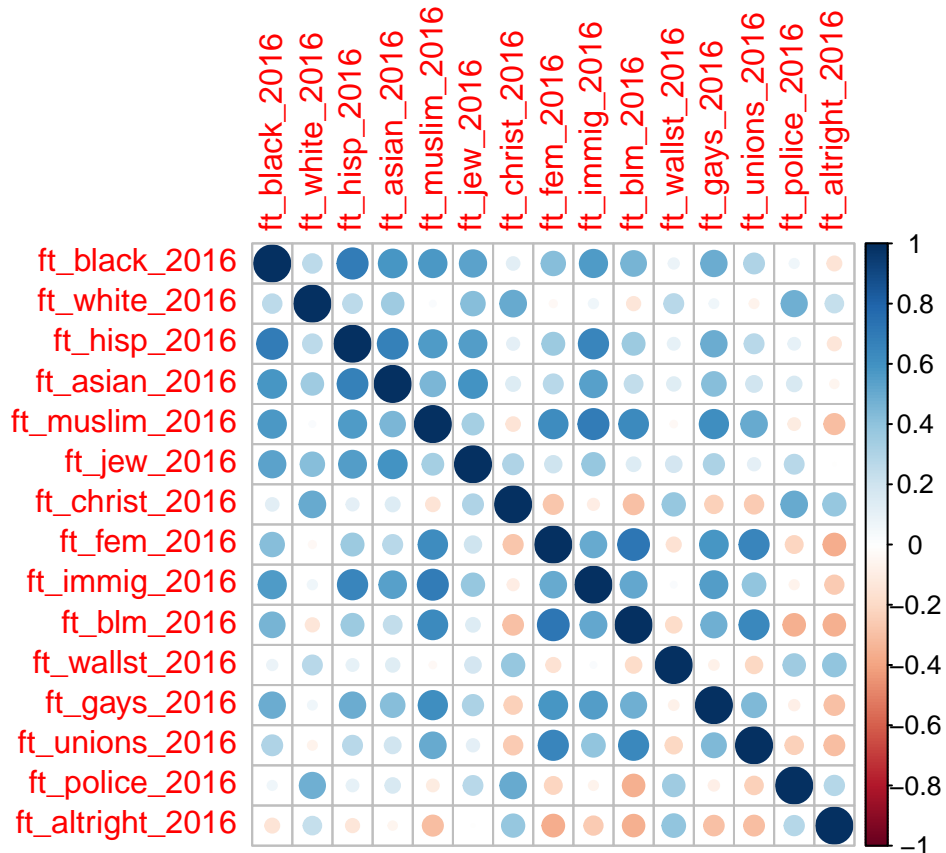
There are many relationships present as we explore the correlations of feeling thermometers. Most of the correlations are blue, depicting more positive correlations than negative. The feeling thermometer with the most orange/red relationships would be "ft_altright_2016" or feelings towards the Alt Right Movement. This makes sense, because the Alt-right movement traditionally wouldn't pair as well as other groups might pair together, such as BLM and Feminists. Some of the strongest associations would be between the feeling thermometers of immigrants with the feeling thermometers of Hispanics and Muslims. This relationship would hold, because the United States holds many Hispanic immigrants and Muslim immigrants, so a positive feeling between the two groups makes sense.

## Data Splitting & Model Fitting

To begin fitting our data to appropriate models, I first set a seed and created a survey training data set and a survey testing data set. I decided to stratify on 'presvote16post_2016' because this is our outcome variable, and stratified sampling helps to replicate the same proportions in each dataset.

```
set.seed(10) # setting seed
survey_split <- initial_split(survey, prop = 0.70,
                              strata = presvote16post_2016)
survey_train <- training(survey_split)
survey_test <- testing(survey_split)
```

Next I created a variable called 'survey_folds' to create divisions of data (called folds) to test our models on using Cross-Validation. Cross-validation is a re-sampling method that uses different portions of the data to test and train a model on different iterations (folds). The goal of cross-validation is to test the model's ability to predict new data that was not used in estimating it, in order to flag problems like over-fitting or

14

selection bias. If we were to use the entire training set, we would be using the validation set approach where the estimate of the test MSE is highly variable and is considered a 'waste' of data, because only training data is used to fit the model.

```
survey_folds <- vfold_cv(survey_train, v = 5, strata = presvote16post_2016)
survey_recipe <- recipe(presvote16post_2016 ~ ft_black_2016 + ft_white_2016 + ft_hisp_2016 +
                          ft_asian_2016 + ft_muslim_2016 + ft_jew_2016 + ft_christ_2016 + ft_fem_2016 +
                         ft_immig_2016 + ft_blm_2016 + ft_wallst_2016 + ft_gays_2016 + ft_unions_2016 +
                         ft_police_2016 + ft_altright_2016 + accurately_counted2_2016 + alcohol_2016 +
                         smoke100_2016 + pid7_2016 + ideo5_2016 + pew_religimp_2016, survey_train) %>%
  step_dummy(accurately_counted2_2016,alcohol_2016,smoke100_2016,pid7_2016,ideo5_2016,pew_religimp_2016)
  step_normalize(all_predictors())
#survey_recipe %>% prep() %>% juice()
```

Then I create a singular recipe to use for each of the 5 models I'm testing. I added in all of the variables listed in the 'survey_codebook.txt' besides zipcode, time started, and time ended. With more time in the future I would like to explore those variables, however I found the other variables easier to work with. I dummy coded all of the categorical variables and normalized all of the feeling thermometer numeric data.

As I built each model, I used the same general process:

1. Set up the type of model, engine, and mode to Classification.
2. Set up a tuning grid with the specific parameters we want tuned and specific levels of tuning.
3. Run the Model and select the one with the best roc_auc, and finalize the workflow with tuning parameters.
4. Save the model to RDA file to avoid re-runing and save time.

I used roc_auc as my metric of performance because it calculates the the area under the curve for the receiver operating characteristic (ROC) curve, and is a great metric for efficency in a multi-classification model.

Time for the most important part of the project: building our models. As previously stated in the introduction, we will be trying out seven different machine learning techniques all using the same recipe. This took quite a while because some of the models took multiple hours to run while they were tuning. The actual model building is a fairly straightforward process, but it takes up a lot of space, so the code can be seen in a seperate rmarkdown file if you want to see it. I decided to set my metric of performance as **roc_auc**, because that is what shows the most significant level of efficiency in a binary classification model where the data is not perfectly balanced. This essentially calculates the area under the curve for the receiver operating characteristic (ROC) curve, which highlights the trade-off between sensibility and sensitivity. In the end, I think it was a great success! Nearly every model built had the same process, which I will detail right now.

1. Set up the type of model, engine, and mode to Classification.
2. Set up a tuning grid with the specific parameters we want tuned and specific levels of tuning.
3. Run the Model and select the one with the best roc_auc, and finalize the workflow with tuning parameters.
4. Save the model to RDA file to avoid re-runing and save time.

To see the specific code for each Model, click the code button to the right!

## Model 1 (Multinomial logistic regression)

```r
# logistic regression model using 'glm'
reg <- multinom_reg(mixture=NULL,penalty=0) %>%
  set_engine('glmnet') %>%
  set_mode('classification')

# creating a workflow
reg_wkflow <- workflow() %>%
  add_model(reg) %>%
  add_recipe(survey_recipe)

# fitting workflow w/ training data
reg_fit <- fit(reg_wkflow, survey_train)

log_reg_acc <- augment(reg_fit, new_data = survey_train) %>%
  roc_auc(truth = 'presvote16post_2016', estimate = `.pred_Did not vote for President`:.pred_Other)

# Cross validation
reg_fold <- reg_wkflow %>%
  fit_resamples(survey_folds)
collect_reg <- collect_metrics(reg_fold)
log_reg_acc_est <- collect_reg$mean[2]
#log_reg_acc_est
#reg_fold <- fit_resamples(reg_wkflow,survey_folds)
#collect_reg <- collect_metrics(reg_fold)
```

## Model 2 (Regularized Regression w/ Elastic Net)

```r
library(parsnip)
elastic_net <- multinom_reg(penalty = tune(),
                            mixture = tune()) %>%
  set_mode("classification") %>%
  set_engine("glmnet")

en_workflow <- workflow() %>%
  add_recipe(survey_recipe) %>%
  add_model(elastic_net)

en_grid <- grid_regular(penalty(range = c(-5,5)),
                        mixture(range = c(0,1)), levels = 10)

tune_res <- tune_grid(en_workflow,resamples = survey_folds,grid=en_grid)


best <- select_best(tune_res,metric='roc_auc')
en_final <- finalize_workflow(en_workflow,best)
en_final_fit <- fit(en_final, data = survey_train)
predicted <- augment(en_final_fit, new_data = survey_test) %>%
  select(presvote16post_2016, starts_with(".pred"))  %>%
  roc_auc(presvote16post_2016, `.pred_Did not vote for President`:.pred_Other)
reg_reg <- predicted$.estimate
#predicted #roc_auc
```

## Model 3 (Decision Tree)

```r
# setting up decision tree model
tree_spec <- decision_tree() %>%
  set_engine("rpart") %>%
  set_mode("classification") %>%
  set_args(cost_complexity=tune())

#setting up the workflow
wrkflow <- workflow() %>%
  add_model(tree_spec) %>%
  add_recipe(survey_recipe)

param_grid <- grid_regular(cost_complexity(range = c(-3, -1)), levels = 10)
```

```r
tune_res <- tune_grid(
  wrkflow,
  resamples = survey_folds,
  grid = param_grid,
  metrics = metric_set(roc_auc)
)

write_rds(tune_res, file = "data/decison-tree-res.rds")
```

```r
decision_tree <- read_rds("data/decison-tree-res.rds")
decision <- decision_tree %>%
  collect_metrics() %>%
  arrange(desc(mean)) %>%
  filter(row_number()==1)
decison_tree_roc <- decision$mean
```

## Model 4 (Random Forest)

```r
rf <- rand_forest() %>%
  set_engine('ranger',importance='impurity') %>%
  set_mode('classification') %>%
  set_args(mtry=tune(),trees=tune(),min_n=tune())

rf_wrkflow <- workflow() %>%
  add_recipe(survey_recipe) %>%
  add_model(rf)

rf_grid <- grid_regular(mtry(range=c(1,21)),
                        trees(range = c(0,50)),
                        min_n(range = c(1,20)),
                        levels = 5)
```

```r
library(ranger)
tune_forest <- tune_grid(rf_wrkflow, resamples = survey_folds, grid = rf_grid, metrics = metric_set(roc_
write_rds(tune_forest, file = "data/rand-forest-res.rsd")
```

17

```
tune_forest <- read_rds("data/rand-forest-res.rsd")
```

```
rand_forest_tuned <- tune_forest %>%
  collect_metrics() %>%
  arrange(desc(mean)) %>%
  filter(row_number()==1)
rf_mean <- rand_forest_tuned$mean
```

## Model 5 (Boosted Tree)

```
library(xgboost)
boasted_tree <- boost_tree() %>%
  set_engine('xgboost') %>%
  set_mode('classification') %>%
  set_args(trees=tune())

boasted_wrkflow <- workflow() %>%
  add_recipe(survey_recipe) %>%
  add_model(boasted_tree)

boasted_grid <- grid_regular(trees(c(10,2000)), levels = 10)

boasted_tune_res <- tune_grid(boasted_wrkflow, resamples=survey_folds, grid = boasted_grid, metrics = m

write_rds(boasted_tune_res, file = "data/boasted_tune_res.rsd")
```

```
boasted_tune_res <- read_rds(file = "data/boasted_tune_res.rsd")
boasted <- boasted_tune_res %>%
  collect_metrics() %>%
  arrange(desc(mean)) %>%
  filter(row_number()==1)
boasted_mean <- boasted$mean
```

## Summary of Models

I decided to evaluate my models based on the roc_auc score (as explained above), and here we see that Regularized Regression w/ an Elastic Net won! However the margins were very close, and all of the models did pretty well.

```
roc_auc <- c(log_reg_acc_est,reg_reg,decison_tree_roc,rf_mean,boasted_mean)
models <- c("Multinomial Regression", "Regularized Regression (Elastic Net)","Decision Tree", "Random F
results <- tibble(models=models,roc_auc= roc_auc)
results <- results %>%
  arrange(-roc_auc)
results
```

```
## # A tibble: 5 x 2
##   models                           roc_auc
##   <chr>                              <dbl>
```

```
## 1 Regularized Regression (Elastic Net)    0.766
## 2 Multinomial Regression                   0.758
## 3 Random Forest                            0.749
## 4 Boasted Tree                             0.727
## 5 Decision Tree                            0.688
```

```
modesl_bar_plot <- ggplot(results,
        aes(x = models, y = roc_auc)) +
  geom_bar(stat = "identity", width=0.2, fill = "pink", color = "black")

models_lollipop_plot <- ggplot(results, aes(x = models, y = roc_auc)) +
    geom_segment( aes(x =models, xend = 0, y = roc_auc, yend = 0)) +
  geom_point( size=7, color= "black", fill=alpha("#FB4F14", 0.3), alpha=0.7, shape=21, stroke=3)
modesl_bar_plot
```
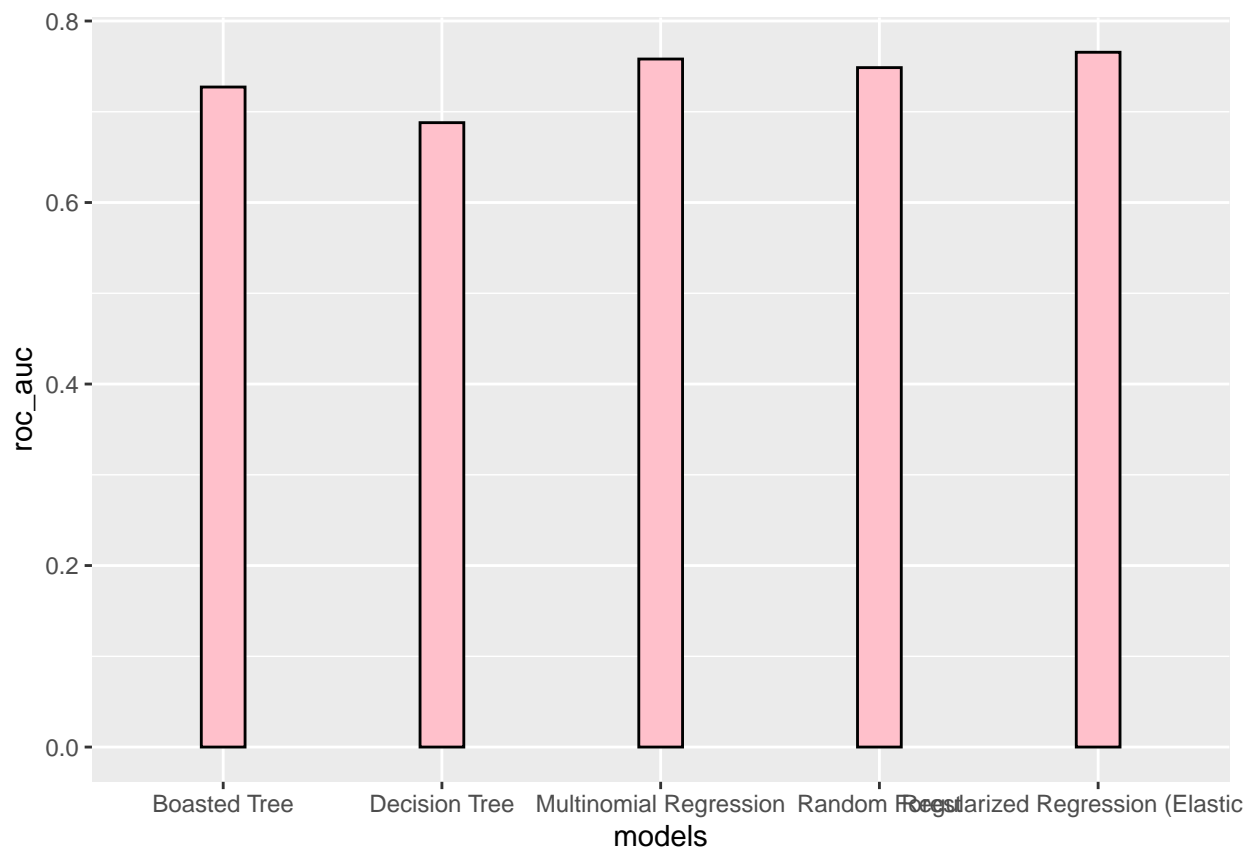


```
models_lollipop_plot
```
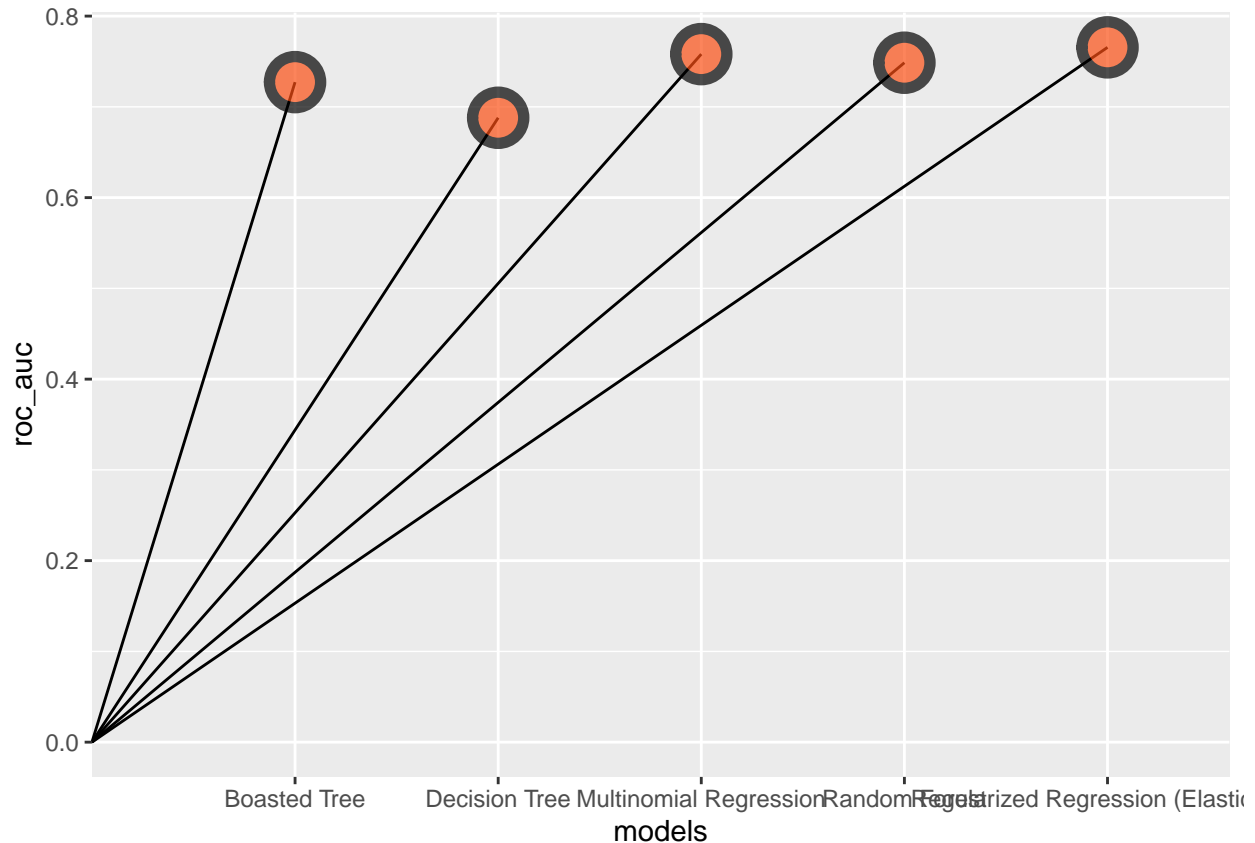
**Let's fit the best model to our training/testing data**

```
prediction_accuracy <- bind_cols(predict(en_final_fit, survey_train, type='prob'))
prediction_accuracy_test_acc <- augment(en_final_fit, new_data = survey_train) %>%
  accuracy(truth = 'presvote16post_2016', estimate = .pred_class)
prediction_accuracy_test_acc # testing accuracy
```

```
## # A tibble: 1 x 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy multiclass     0.885
```

```
prediction <- bind_cols(predict(en_final_fit, survey_test, type='prob'))
test_acc <- augment(en_final_fit, new_data = survey_test) %>%
  accuracy(truth = 'presvote16post_2016', estimate = .pred_class)
test_acc # testing accuracy
```

```
## # A tibble: 1 x 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy multiclass     0.877
```

Here we can see the our Regularized Regression w/ an Elastic Net has an accuracy rating of 0.8845938 on the survey training data, and an 0.8766319 accuracy rating on the survey testing data. This is fairly high, and a really great result. A potential reason for the higher rating on the training data could be over fitting, which is common in training data sets.

## Misc. Visualizations

Here are some interesting Visualizations from our models!

```
augment(en_final_fit, new_data = survey_test) %>%
  conf_mat(truth = presvote16post_2016, estimate = .pred_class)  %>%
  autoplot(type='heatmap')
```
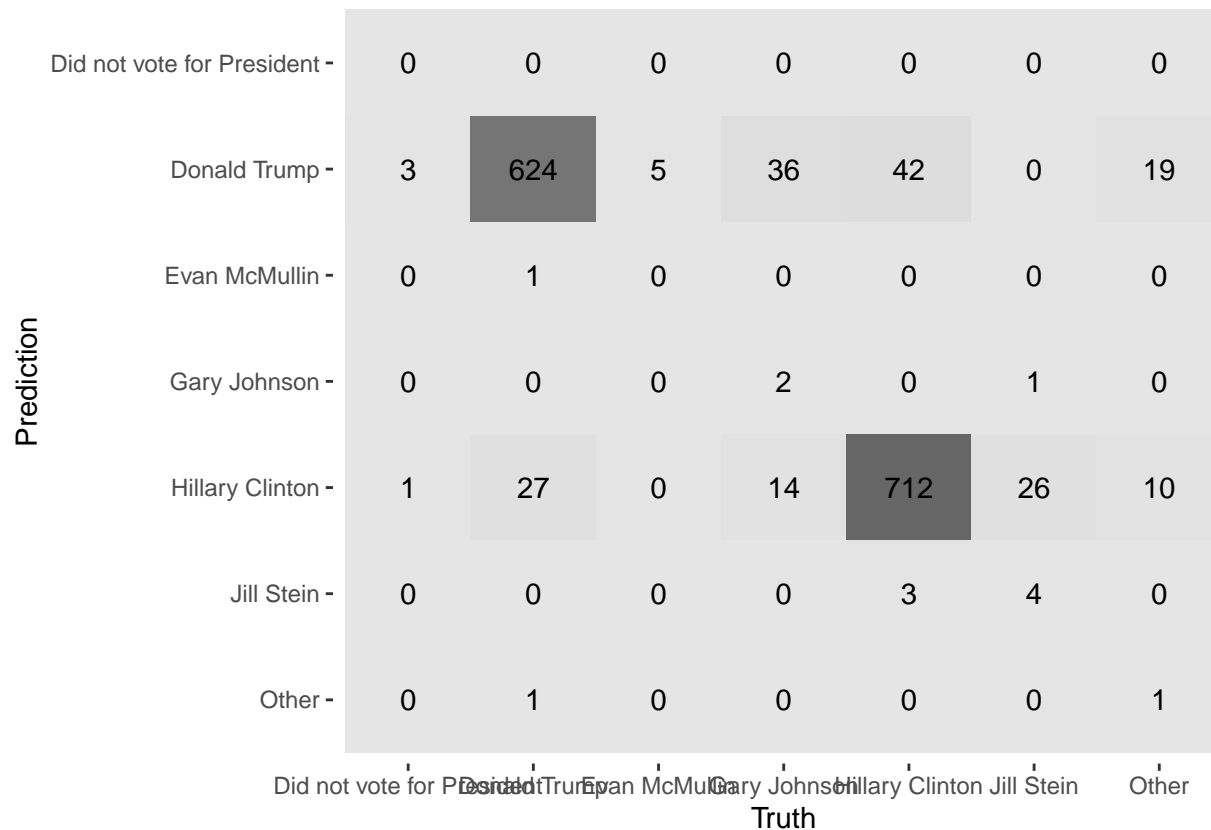


This is a heat map for the Regularized Regression w/ an Elastic Net. A heat map shows the amount of correct/incorrect guesses by the model, and can also be described as a false color image. Th results here tell me that the model incorrectly guessed on the darker boxes, however these are small numbers compared to the size of the dataset.

```
rand_forest <- read_rds(file = "data/rand-forest-res.rsd")
rand_forest %>%
  autoplot()
```
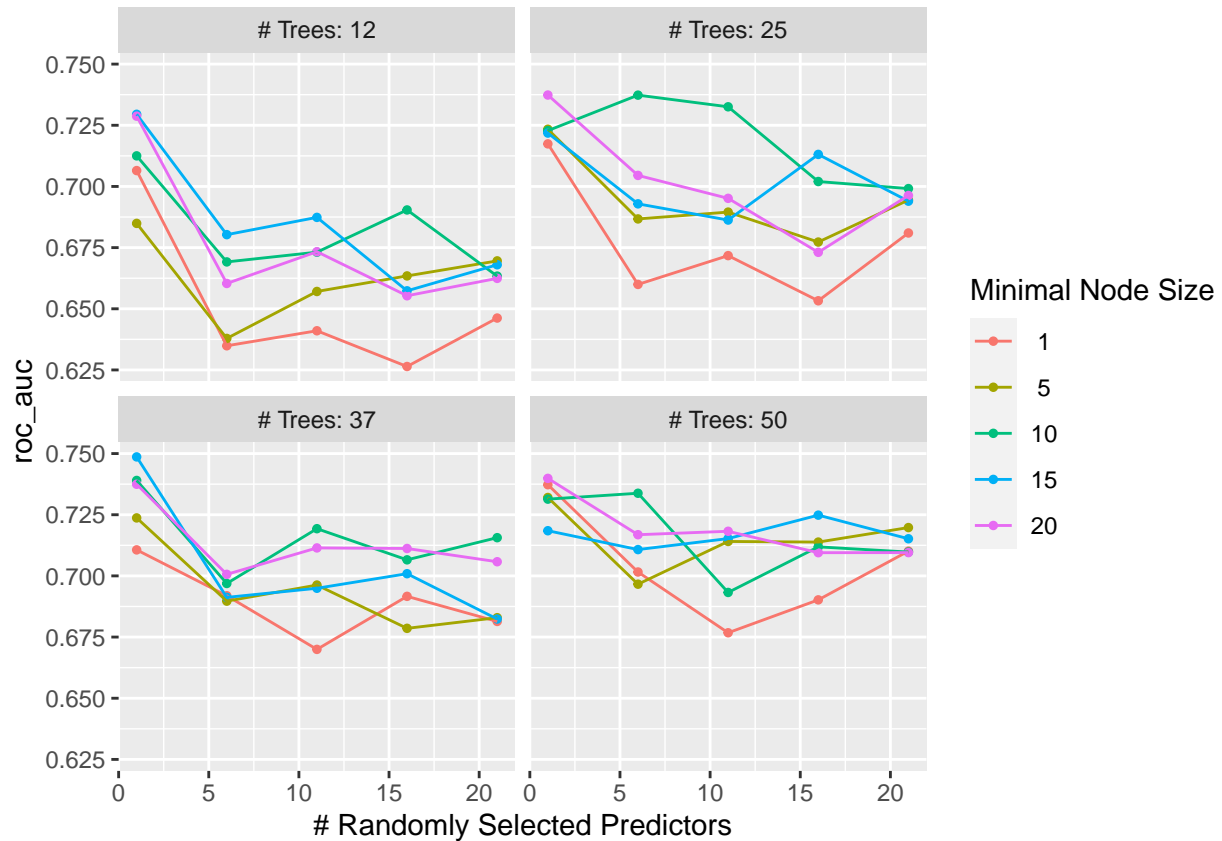
For our random forest model we tuned three different parameters: mtry - the number of predictors that would be randomly sampled, trees - the number of trees to grow in the forest, and min_n - the minimum number of data values needed to create another split. However it doesn't seem that as the number of predictors had a strong positive effect on the accuracy. It does look like as the number of trees increased, the ROC AUC also increased. The most optimal model was around 37 trees at a node size of 15 with a very low number of predictors.

```r
# Random Forest
best_rf <- select_best(rand_forest, metric = "roc_auc")
rf_final <- finalize_workflow(rf_wrkflow, best_rf)
set.seed(10)
class_tree_final_fit <- fit(rf_final, data = survey_train)
class_tree_final_fit %>%
  extract_fit_engine() %>%
  vip()
```
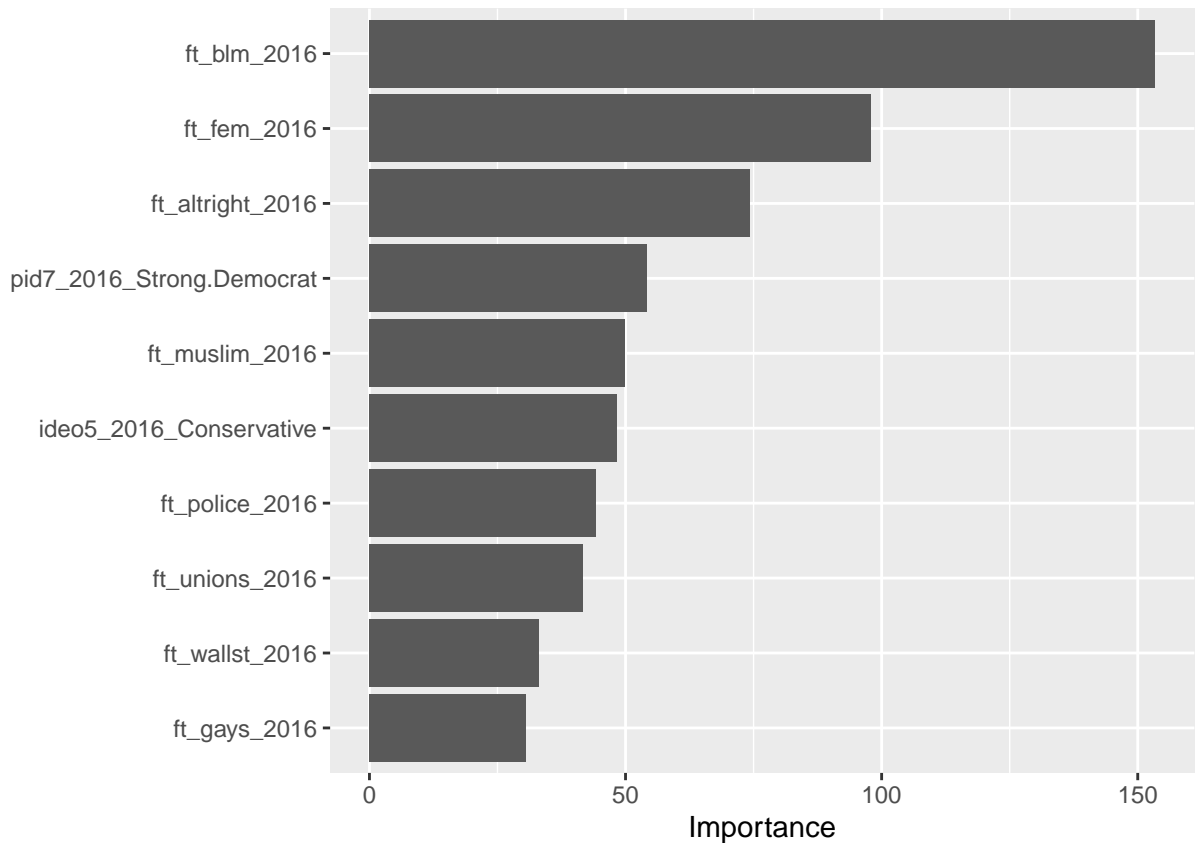
This is a very descriptive and unique plot, especially with the data we are working with. Here we can see that for our Random Forest Model the most important variable when determining who someone would vote for was "ft_blm_2016" or how they ranked Black Lives Matter. The Second Most important variable was how one ranked "ft_fem_2016"or how they ranked Feminists. I think these two determinations make sense, because both Democrats and Republicans feel very different about the two groups. It's interesting to me though because both of these factors were more important than how someone ranked themselves ideologically (liberal to conservative), and I would have thought that would be more telling to the model.

```
# BOASTING
boasted_tune_res <- read_rds(file = "data/boasted_tune_res.rsd")
boasted_tune_res %>%
  autoplot()
```
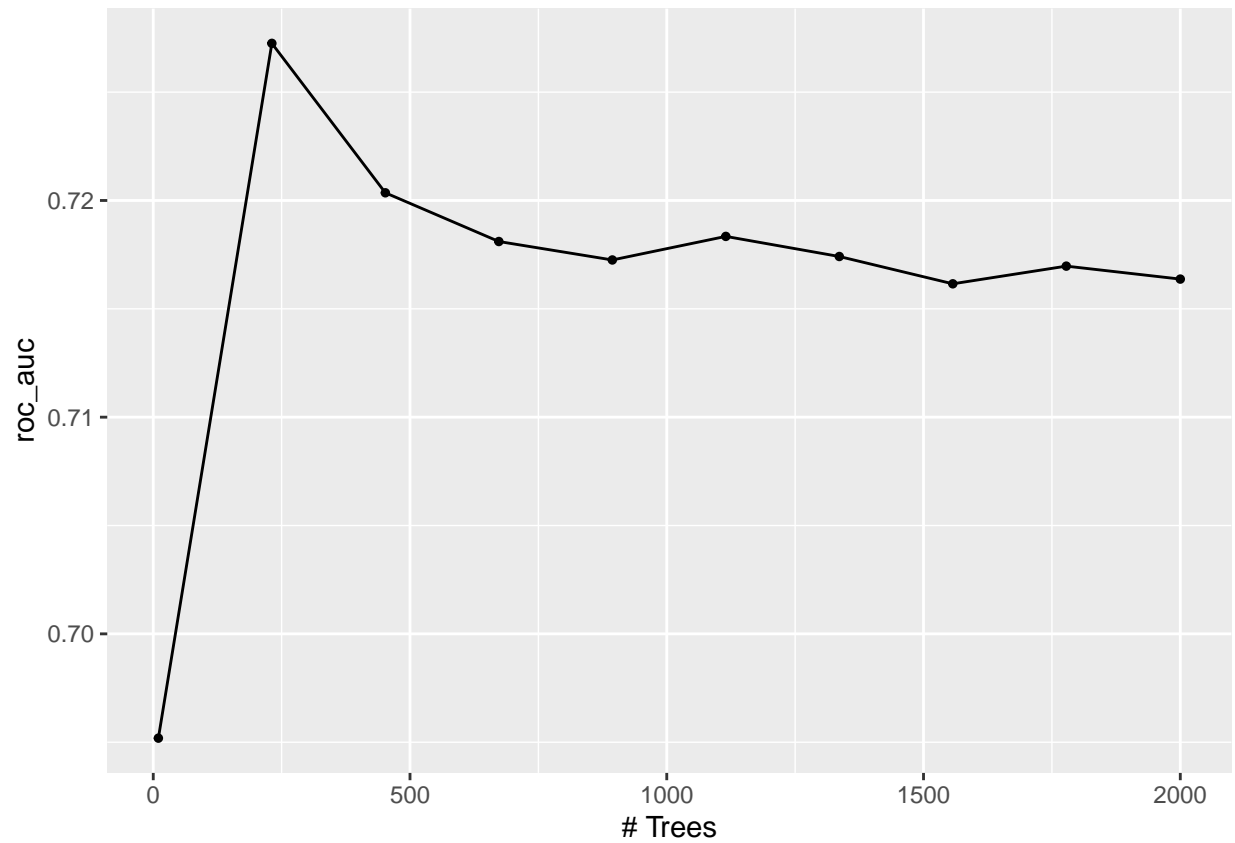
Here we can see with our Boasting Tree Model that the best number of trees was just below 250.

```
decision_tree <- read_rds("data/decison-tree-res.rds")
autoplot(decision_tree)
```
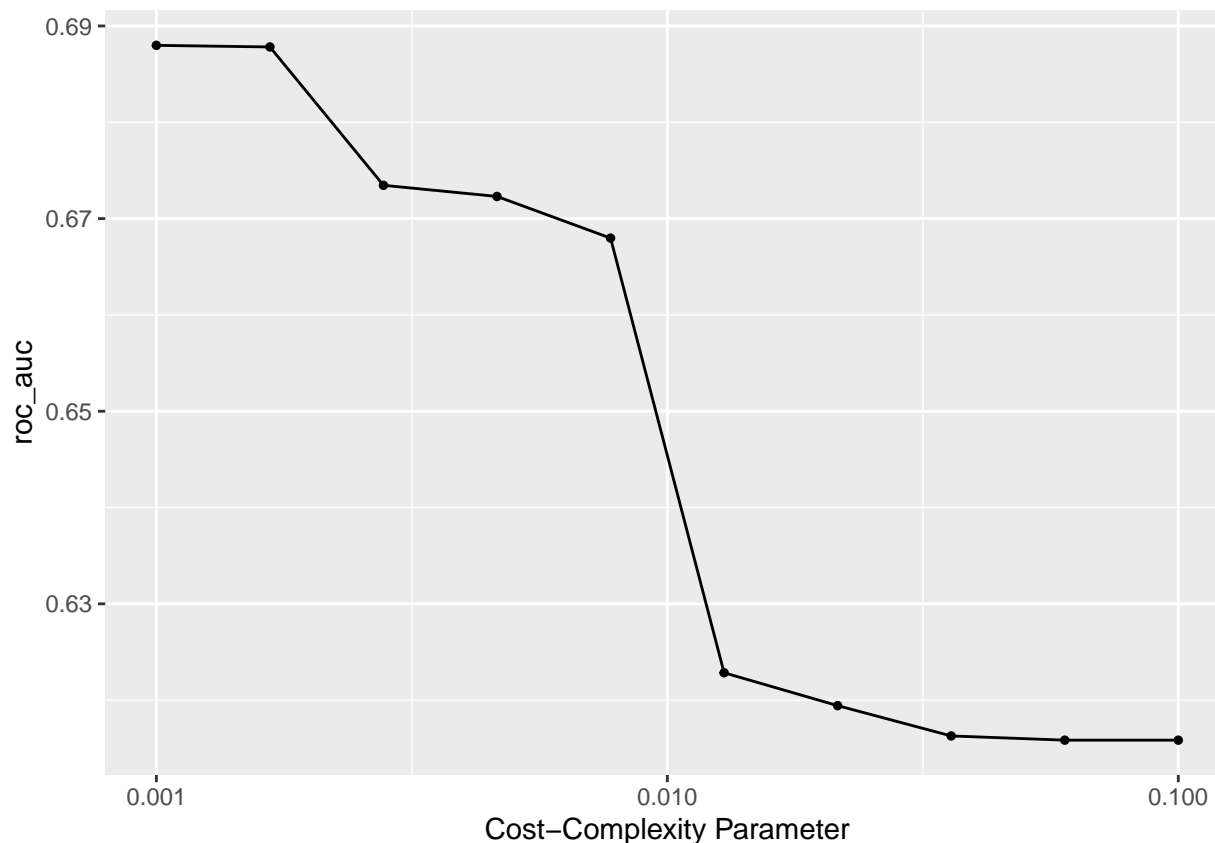
The cost complexity parameter is used to control the size of the decision tree and to select the optimal tree size. Here we can see a sharp decline in roc_auc as our cost-complexity parameter increases, depicting a high cost of adding variables to the decision tree from the current node.

## Conclusion

We set out to determine if we could predict how a person voted in the 2016 presidential election if given someone's party affiliation or political values. We also wanted to predict which would be more important in determining who they voted for.

From the data from our Random Forest, we found that the two most important variables in determining the way one would vote in the 2016 election were how they ranked BLM and how they ranked Feminists. I think this is particularly interesting because respondents also stated their party affiliation/leanings, yet this was not as important. If it was already not apparent, the data here suggests that those two groups are the most polarizing within all of the feeling thermometer variables. Additionally, although our EDA graphs show clear contrasts between voting groups, none of the ethnic groups show up as particularity polarizing besides Muslim.

While all of our Models had similar and high roc_auc ratings, Regularized Regression w/ an Elastic Net performed the best. When tested on the survey training and testing data set, the model recorded an accuracy rating of 88% and 87%, respectively. I think that this is out-standing, given that we are working off of survey data. This model might be the most best because of it's complexity in combining the penalties of ridge regression and lasso "to get the best of both worlds" (Oleszak).

Overall this was a really interesting project, and I'm really grateful for the exposure to such high level concepts and the opportunity to explore themes really important to me.

Thank you for reading through!

# Citations

Nadeem, Reem. "In a Politically Polarized Era, Sharp Divides in Both Partisan Coalitions." Pew Research Center - U.S. Politics & Policy, Pew Research Center, 30 May 2020, https://www.pewresearch.org/politics/ 2019/12/17/in-a-politically-polarized-era-sharp-divides-in-both-partisan-coalitions/.

Oleszak, Michał "Regularization Tutorial: Ridge, Lasso & Elastic Net Regression." *DataCamp*, DataCamp, 12 Nov. 2019, https://www.datacamp.com/tutorial/tutorial-ridge-lasso-elastic-net.

Democracy Fund Voter Study Group. VIEWS OF THE ELECTORATE RESEARCH SURVEY, December 2016. [Computer File] Release 1: August 28, 2017. Washington DC: Democracy Fund Voter Study Group [producer] https://www.voterstudygroup.org/.