Classifying the Political Party of Survey Respondents

David Hutchings

9/25/2020

Goal: I want to see whether a machine learning model can classify whether someone is a Democrat or Republican using demographic data. I intentionally left off many survey response questions that would have made this classification task too easy. For example, there were questions asking about the respondent's opinion of Trump.

Data Source: Pew Research Center January 2020 Political Survey

[https://www.pewresearch.org/politics/dataset/january-2020-political-survey/]

The variables used in this analysis:

1. **cregion:** The region that the respondent lives in

   levels = Northeast, Midwest, South, West

2. **usr:** Whether the respondent is from an urban, suburban or rural region

   levels = Urban, Suburban, Rural

3. **income:** The income level of the respondent

   levels = Less than $10,000, 10 to under $20,000, 20 to under $30,000, 30 to under $40,000, 40 to under $50,000, 50 to under $75,000, 75 to under $100,000, 100 to under $150,000, $150,000 or more, Don't know/Refused

4. **relig:** The respondent's religion

   levels = Protestant, Roman Catholic (Catholic), Mormon, Orthodox such as Greek or Russian Orthodox, Jewish (Judaism), Muslim (Islam), Buddhist, Hindu, Atheist, Agnostic, Something else, Nothing in particular, (VOL)Christian, (VOL) Unitarian (Universalist), (VOL) Don't know/Refused

5. **attend:** How often the respondent attends religious gatherings

   levels = More than once a week, Once a week, Once or twice a month, A few times a year, Seldom, Never, (VOL) Don't know/Refused

6. **racecmb:** The race of the respondent

   levels = White, Black or African-American, Asian or Asian-American, Mixed Race, Other race, Don't know/Refused (VOL.)

7. **sex:** The respondent's sex

   levels = Male, Female

8. **age:** The age of the respondent ranging from 18 to 99 years old

9. **educ:** The education level of the respondent

   levels = Less than high school, High school incomplete, High school graduate, Some college (no degree), Two year associate degree, Four year college or university degree, Some postgraduate or professional school (no postgraduate degree), Postgraduate or professional degree, (VOL) Don't know/refused

10. **party:** the party that the respondent belongs to

    levels: Republican, Democrat

```r
library(caret)
library(tidyverse)
```

```r
colnames(data)
```

```
##  [1] "cregion" "usr"     "income"  "relig"   "attend"  "racecmb" "sex"
##  [8] "age"     "educ"    "party"
```

Breakdown of number of Republican's and Democrats in our data sample and the dimensions of our dataframe

```r
table(data$party)
```

```
##
## Republican   Democrat
##        446        403
```

```r
dim(data)
```

```
## [1] 849  10
```

Chi Sq Tests Now, I run a chi sq test to discover if there are any interactions between different categorical variables and the respondent's party association. Low p-values correspond to a low chance of the observed categorical frequencies matching the expected categorical variable's frequencies. For example, a low p-value indicates that men and women might have a different breakdown between support for Republicans and Democrats.

```r
for (name in colnames(data)){
  if(name != "age" & name != "party"){
    t <- table(data[,c(name)], data$party)
    test <- chisq.test(t)
    p.value <- test[[3]]
    print(paste(name, ": p-value =",p.value))
  }
}
```

```
## [1] "cregion : p-value = 0.000668320651386766"
## [1] "usr : p-value = 1.05643574978038e-09"
## [1] "income : p-value = 0.15625243905461"
## [1] "relig : p-value = 2.20553583270183e-19"
## [1] "attend : p-value = 8.44926108926972e-10"
## [1] "racecmb : p-value = 9.03361353760262e-13"
## [1] "sex : p-value = 3.87314298397813e-13"
## [1] "educ : p-value = 0.000287692417452741"
```

Crosstabs Lets examine some of the crosstabs

```
## [1] "Region:"
```

```
##
##              Republican  Democrat
##   Northeast  0.3809524 0.6190476
##   Midwest    0.5765306 0.4234694
##   South      0.5709677 0.4290323
##   West       0.5102041 0.4897959
```

```
## [1] "Sex"
```

```
##
##           Republican  Democrat
##   Male     0.6359833 0.3640167
##   Female   0.3827493 0.6172507
```

```
## [1] "Attend Religious Gathering:"
```

```
##
##                             Republican  Democrat
##   More than once a week      0.7692308 0.2307692
##   Once a week                0.5779817 0.4220183
##   Once or twice a month      0.5677966 0.4322034
##   A few times a year         0.5000000 0.5000000
##   Seldom                     0.5200000 0.4800000
##   Never                      0.2689076 0.7310924
##   (VOL) Don’t know/Refused   0.5000000 0.5000000
```

Logistic Regression

Now, we are going to try to model the data using a simple logistic regression with the caret package.

Note that we don’t need to split the data between train and validation sets because we are performing cross validation with 15 folds. In other words, we will inspect the model results using 15 different validation sets during training. This will give us a more accurate look at the success of the model.

```r
set.seed(400)
model_log <- train(
  party ~ .,
  data = data,
  method = "glm",
  family = "binomial",
  trControl = trainControl(method="cv", number=15)
)
print(model_log)
```

```
## Generalized Linear Model
##
## 849 samples
##   9 predictor
##   2 classes: 'Republican', 'Democrat'
##
## No pre-processing
## Resampling: Cross-Validated (15 fold)
## Summary of sample sizes: 792, 793, 793, 792, 793, 792, ...
## Resampling results:
##
##   Accuracy   Kappa
##   0.7338972  0.4649652
```

A cool feature of logistic models is that it tells us which variables were the most import in predicting a respondents political party. Here we can see the top 20 most import features.

```
importance <- varImp(model_log, scale = FALSE)

importance
```

```
## glm variable importance
##
##   only 20 most important variables shown (out of 50)
##
##                                                                         Overall
## sexFemale                                                                 7.628
## ‘racecmbBlack or African-American‘                                        7.602
## attendNever                                                               4.564
## religAtheist                                                              4.435
## religAgnostic                                                             4.050
## ‘attendA few times a year‘                                                3.891
## attendSeldom                                                              3.666
## religJewish                                                               3.464
## ‘attendOnce a week‘                                                       3.175
## religCatholic                                                             2.870
## ‘racecmbOr some other race‘                                               2.831
## ‘educTwo year associate degree from a college or university‘              2.759
## ‘relig(VOL) Christian‘                                                    2.509
## cregionSouth                                                              2.419
## ‘educHigh school graduate (Grade 12 with diploma or GED certificate)‘    2.400
## ‘educSome college, no degree (includes some community college)‘          2.171
## ‘racecmbDon’t know/Refused (VOL.)‘                                        2.117
## ‘educHigh school incomplete (Grades 9-11 or Grade 12 with NO diploma)‘   2.013
## ‘attendOnce or twice a month‘                                             1.985
## ‘religNothing in particular‘                                              1.851
```

This confusion matrix shows us the accuracy of our model in more detail

```
confusionMatrix(model_log)
```

```
## Cross-Validated (15 fold) Confusion Matrix
##
## (entries are percentual average cell counts across resamples)
##
```

```
##               Reference
## Prediction   Republican Democrat
##   Republican       40.5     14.6
##   Democrat         12.0     32.9
##
##  Accuracy (average) : 0.7338
```

KNN Model

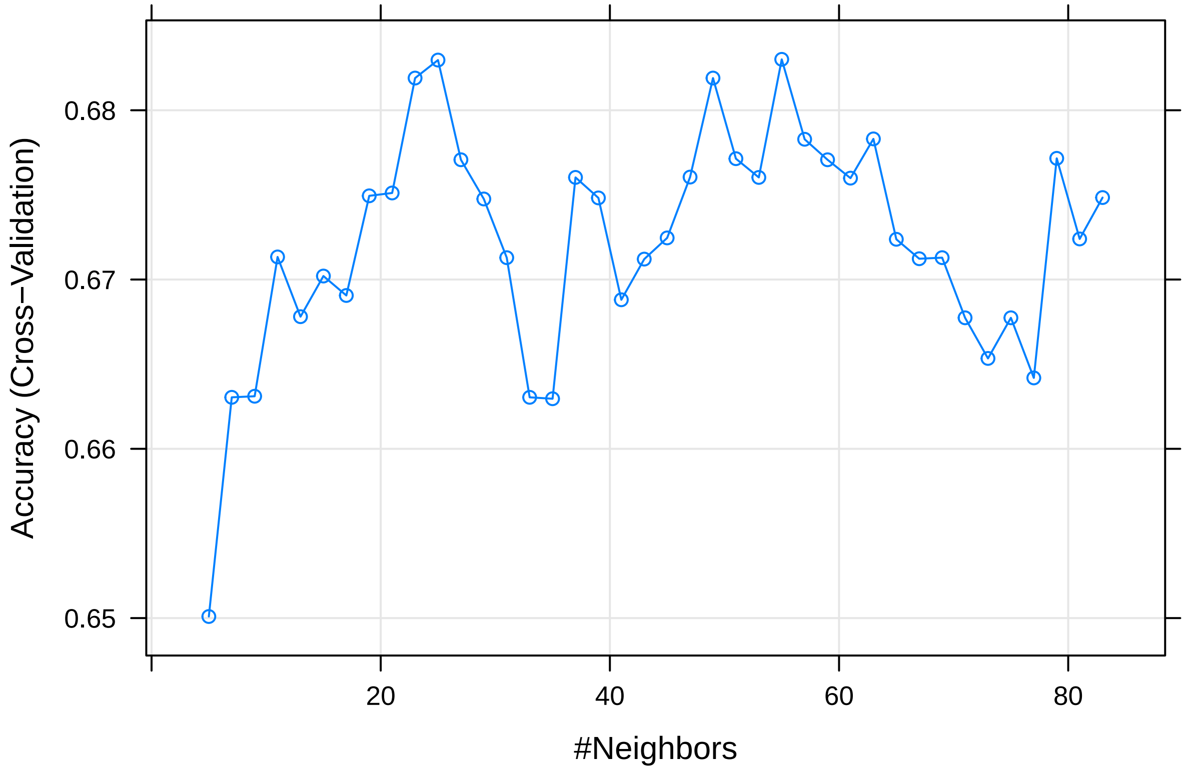
Some notes about the model:

- "nzv" means that we remove Near Zero Variance variables. In other words, we remove variables with little to no predictive power
- "center" & "scale" as worded will center and scale the continuous variables
- tuneLength will try 40 different values of hyperparameter "k" and choose the value k that produces the best model results

```
model_knn <- train(party ~ ., data = data, method = "knn",
                   trControl = trainControl(method="cv", number=15),
                   preProcess = c("nzv","center","scale"), tuneLength = 40)
```

Our model ended up selecting the value of k based on which model produced the best accuracy

```
plot(model_knn)
```



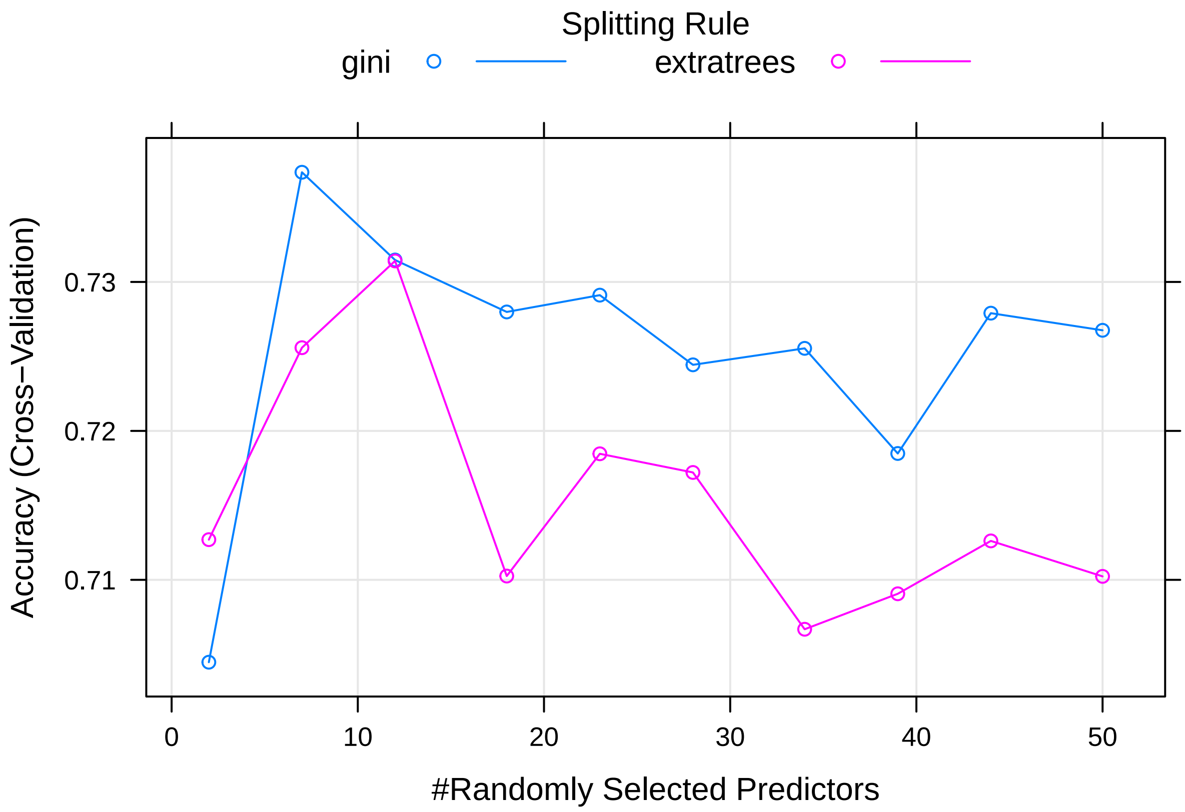
```
confusionMatrix(model_knn)
```

```
## Cross-Validated (15 fold) Confusion Matrix
##
## (entries are percentual average cell counts across resamples)
##
##             Reference
## Prediction   Republican Democrat
##   Republican       45.3     24.5
##   Democrat          7.2     23.0
##
##  Accuracy (average) : 0.6832
```

Random Forest

```
model_rf <- train(
  party ~ .,
  data= data,
  method = "ranger",
  tuneLength = 10,
  trControl = trainControl(method="cv", number=15)
)
```

```
plot(model_rf)
```



```
confusionMatrix(model_rf)
```

```
## Cross-Validated (15 fold) Confusion Matrix
##
## (entries are percentual average cell counts across resamples)
##
##             Reference
## Prediction   Republican Democrat
##   Republican       42.0     15.8
##   Democrat         10.5     31.7
##
##  Accuracy (average) : 0.7373
```

Support Vector Machine

```
model_svm <- train(
  party ~ .,
  data= data,
  method = "svmLinear",
  tuneLength = 10,
  trControl = trainControl(method="cv", number=15),
  preProcess = c("nzv","center","scale")
)
```

```
confusionMatrix(model_svm)
```

```
## Cross-Validated (15 fold) Confusion Matrix
##
## (entries are percentual average cell counts across resamples)
##
##             Reference
## Prediction   Republican Democrat
##   Republican       41.9     16.6
##   Democrat         10.6     30.9
##
##  Accuracy (average) : 0.7279
```

Comparing Models

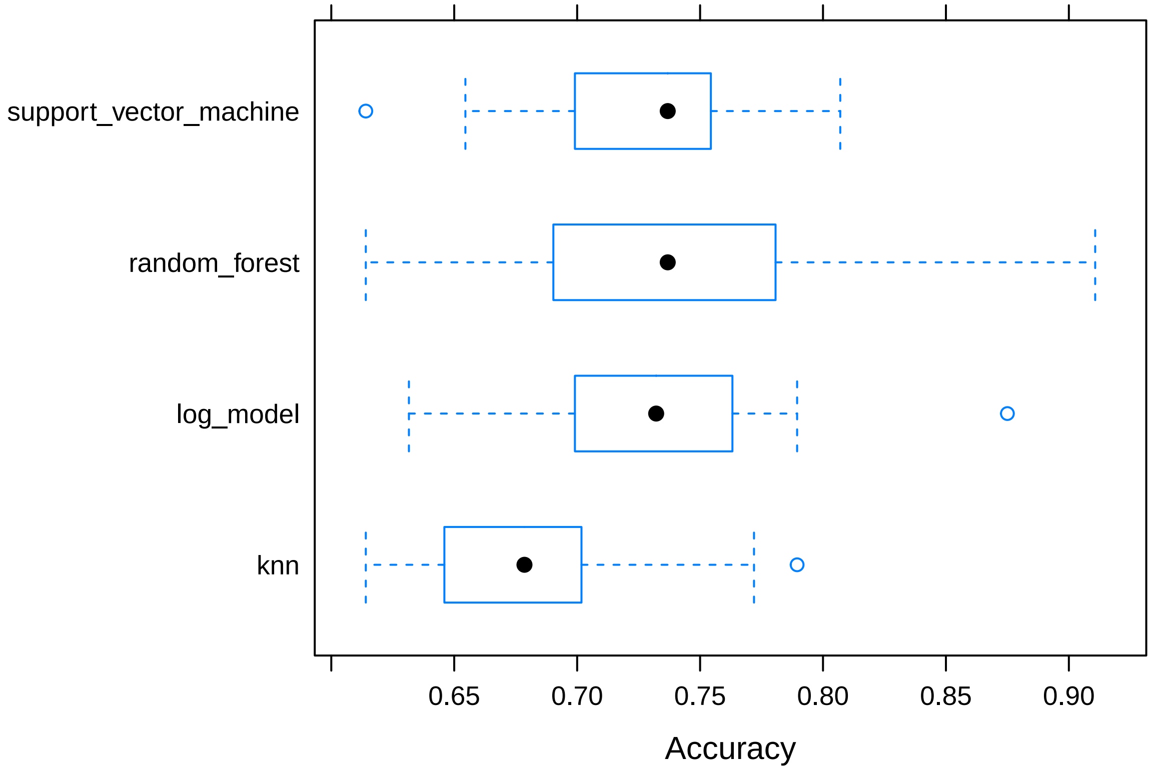
```
model_list <- list(log_model = model_log, random_forest = model_rf, knn = model_knn,
                   support_vector_machine = model_svm)
```

```
resamples <- resamples(model_list)
summary(resamples)
```

```
##
## Call:
```

```
## summary.resamples(object = resamples)
## 
## Models: log_model, random_forest, knn, support_vector_machine 
## Number of resamples: 15 
## 
## Accuracy 
##                             Min.   1st Qu.    Median      Mean   3rd Qu.
## log_model              0.6315789 0.6990915 0.7321429 0.7338972 0.7631579
## random_forest          0.6140351 0.6903195 0.7368421 0.7373642 0.7807018
## knn                    0.6140351 0.6459900 0.6785714 0.6830064 0.7017544
## support_vector_machine 0.6140351 0.6990915 0.7368421 0.7278678 0.7543860
##                             Max. NA's
## log_model              0.8750000    0
## random_forest          0.9107143    0
## knn                    0.7894737    0
## support_vector_machine 0.8070175    0
## 
## Kappa 
##                             Min.   1st Qu.    Median      Mean   3rd Qu.
## log_model              0.2678899 0.3949073 0.4545455 0.4649652 0.5236601
## random_forest          0.2287823 0.3726588 0.4692737 0.4705294 0.5589106
## knn                    0.2113208 0.2824968 0.3463035 0.3529958 0.3917137
## support_vector_machine 0.2201493 0.3947777 0.4692737 0.4511406 0.5037313
##                             Max. NA's
## log_model              0.7493606    0
## random_forest          0.8200514    0
## knn                    0.5681818    0
## support_vector_machine 0.6122449    0
```

```
bwplot(resamples, metric = "Accuracy")
```



Summary

Based on the plot above, there is no clear answer as to which model is the best. Most of the median accuracies were around ~73%. The KNN model was probably the worst of the bunch with the lowest median accuracy. In the future it would interesting to try some other machine learning models on this data and see if better results can be achieved. The neat takeaway from this project is that with only demographic information you can predict whether someone is a Republican or Democrat with pretty decent accuracy.