Predicting Trademark Application Outcomes Using United States Patent and Trademark Office Datasets

Capstone Report for SlideRule: Introduction to Data Science

Michael Berlet

Project Description

Every year since 2013, the total number of Fiscal Year Trademark Applications has exceeded over 400,000, with over 250,000 successful trademark registrations processed per fiscal year.[1] While this implies a successful short-term registration rate of over sixty percent, this still leaves a substantial portion of applications to be abandoned outright or delayed for not meeting the minimum acceptable criteria for successful registration. Were federal registration a straightforward, economical, and non-time intensive task, registrants might have little cause to consider the likelihood of successful trademark registration according to any predictive criteria, but as the filing process typically requires several months of correspondence with the United States Patent and Trademark Office (USPTO)--as well as thousands of dollars in application and/or attorney fees--one can make a compelling case to afford some attention to the likelihood of success prior to application.

This project analyzes annual trademark applications processed by the USPTO in an attempt to isolate any compelling criteria corresponding, correllating, or conducive to successful federal registration of trademarks using predictive modeling in the form of Classification and Regression Trees (CART), RandomForest, and Logistic Regression.

[1] http://www.uspto.gov/dashboards/trademarks/main.dashxml

Available Data

As of April 2016, the USPTO has provided yearly updates[2] on the ongoing statuses of over seven million trademark applications filed from January 1870 to December 2015:

```
> str(case2012)
Classes 'data.table' and 'data.frame':      7047786 obs. of  79 variables:
 $ serial_no          : int  60000001 60000002 60000003 60000004 60000005 60000006 60000007
60000008 60000009 60000010 ...
 $ abandon_dt         : chr  "" "" "" "" ...
 $ amend_reg_dt       : chr  "" "" "" "" ...
 $ amend_lb_for_app_in: int  0 0 0 0 0 0 0 0 0 0 ...
 $ amend_lb_for_reg_in: int  0 0 0 0 0 0 0 0 0 0 ...
 $ amend_lb_itu_in    : int  0 0 0 0 0 0 0 0 0 0 ...
 $ amend_lb_use_in    : int  0 0 0 0 0 0 0 0 0 0 ...
 $ reg_cancel_cd      : chr  "" "" "" "" ...
 $ reg_cancel_dt      : chr  "" "" "" "" ...
 $ cancel_pend_in     : int  0 0 0 0 0 0 0 0 0 0 ...
 $ cert_mark_in       : int  0 0 0 0 0 0 0 0 0 0 ...
 $ chg_reg_in         : int  0 0 0 0 0 0 0 0 0 0 ...
 $ coll_memb_mark_in  : int  0 0 0 0 0 0 0 0 0 0 ...
 $ coll_serv_mark_in  : int  0 0 0 0 0 0 0 0 0 0 ...
 $ coll_trade_mark_in : int  0 0 0 0 0 0 0 0 0 0 ...
 $ draw_color_cur_in  : int  0 0 0 0 0 0 0 0 0 0 ...
 $ draw_color_file_in : int  0 0 0 0 0 0 0 0 0 0 ...
 $ concur_use_in      : int  0 0 0 0 0 0 0 0 0 0 ...
 $ concur_use_pend_in : int  0 0 0 0 0 0 0 0 0 0 ...
 $ file_location      : chr  "" "" "" "" ...
 $ draw_3d_cur_in     : int  0 0 0 0 0 0 0 0 0 0 ...
 $ draw_3d_file_in    : int  0 0 0 0 0 0 0 0 0 0 ...
 $ exm_attorney_name  : chr  "" "" "" "" ...
 $ lb_use_file_in     : int  0 0 0 0 0 0 0 0 0 0 ...
 $ lb_for_app_cur_in  : int  0 0 0 0 0 0 0 0 0 0 ...
 $ lb_for_reg_cur_in  : int  0 0 0 0 0 0 0 0 0 0 ...
 $ lb_intl_reg_cur_in : int  0 0 0 0 0 0 0 0 0 0 ...
 $ lb_for_app_file_in : int  0 0 0 0 0 0 0 0 0 0 ...
 $ lb_for_reg_file_in : int  0 0 0 0 0 0 0 0 0 0 ...
 $ lb_intl_reg_file_in: int  0 0 0 0 0 0 0 0 0 0 ...
 $ lb_none_cur_in     : int  0 0 0 0 0 0 0 0 0 0 ...
 $ filing_dt          : chr  "" "" "" "" ...
 $ for_priority_in    : int  0 0 0 0 0 0 0 0 0 0 ...
 $ lb_itu_cur_in      : int  0 0 0 0 0 0 0 0 0 0 ...
 $ lb_itu_file_in     : int  0 0 0 0 0 0 0 0 0 0 ...
 $ interfer_pend_in   : int  0 0 0 0 0 0 0 0 0 0 ...
 $ exm_office_cd      : chr  "" "" "" "" ...
```

[2] http://www.uspto.gov/learning-and-resources/electronic-data-products/trademark-case-files-dataset-0

```
$ file_location_dt   : chr  "" "" "" "" ...
$ mark_draw_cd       : chr  "" "" "" "" ...
$ mark_id_char       : chr  "" "" "" "" ...
$ opposit_pend_in    : int  0 0 0 0 0 0 0 0 0 0 ...
$ amend_principal_in : int  0 0 0 0 0 0 0 0 0 0 ...
$ concur_use_pub_in  : int  0 0 0 0 0 0 0 0 0 0 ...
$ publication_dt     : chr  "" "" "" "" ...
$ registration_dt    : chr  "1870-10-25" "1870-10-25" "1870-10-25" "1870-10-25" ...
$ renewal_dt         : chr  "" "" "" "" ...
$ renewal_file_in    : int  0 0 0 0 0 0 0 0 0 0 ...
$ repub_12c_dt       : chr  "" "" "" "" ...
$ repub_12c_in       : int  0 0 0 0 0 0 0 0 0 0 ...
$ incontest_ack_in   : int  0 0 0 0 0 0 0 0 0 0 ...
$ incontest_file_in  : int  0 0 0 0 0 0 0 0 0 0 ...
$ acq_dist_in        : int  0 0 0 0 0 0 0 0 0 0 ...
$ acq_dist_part_in   : int  0 0 0 0 0 0 0 0 0 0 ...
$ use_afdv_acc_in    : int  0 0 0 0 0 0 0 0 0 0 ...
$ use_afdv_file_in   : int  0 0 0 0 0 0 0 0 0 0 ...
$ use_afdv_par_acc_in: int  0 0 0 0 0 0 0 0 0 0 ...
$ serv_mark_in       : int  0 0 0 0 0 0 0 0 0 0 ...
$ std_char_claim_in  : int  0 0 0 0 0 0 0 0 0 0 ...
$ cfh_status_cd      : int  626 626 626 626 626 626 626 626 626 626 ...
$ cfh_status_dt      : chr  "2005-10-11" "2005-10-11" "2005-10-11" "2005-10-11" ...
$ amend_supp_reg_in  : int  0 0 0 0 0 0 0 0 0 0 ...
$ supp_reg_in        : int  0 0 0 0 0 0 0 0 0 0 ...
$ trade_mark_in      : int  0 0 0 0 0 0 0 0 0 0 ...
$ lb_use_cur_in      : int  0 0 0 0 0 0 0 0 0 0 ...
$ lb_none_file_in    : int  0 0 0 0 0 0 0 0 0 0 ...
$ ir_auto_reg_dt     : chr  "" "" "" "" ...
$ ir_first_refus_in  : int  NA NA NA NA NA NA NA NA NA NA ...
$ ir_death_dt        : chr  "" "" "" "" ...
$ ir_publication_dt  : chr  "" "" "" "" ...
$ ir_registration_dt : chr  "" "" "" "" ...
$ ir_registration_no : chr  "" "" "" "" ...
$ ir_renewal_dt      : chr  "" "" "" "" ...
$ ir_status_cd       : int  NA NA NA NA NA NA NA NA NA NA ...
$ ir_status_dt       : chr  "" "" "" "" ...
$ ir_priority_dt     : chr  "" "" "" "" ...
$ ir_priority_in     : int  NA NA NA NA NA NA NA NA NA NA ...
$ related_other_in   : int  NA NA NA NA NA NA NA NA NA NA ...
$ registration_no    : int  1 2 3 4 5 6 7 8 9 10 ...
$ tad_file_id        : int  1 1 1 1 1 1 1 1 1 1 ...
```

The data consists of over 79 observations consisting primarily of binary document submission variable (available or not available), dates delineating the processing schedule, and categorical status codes. The USPTO also provides general[3] and in-depth[4] descriptions of the variables in question.

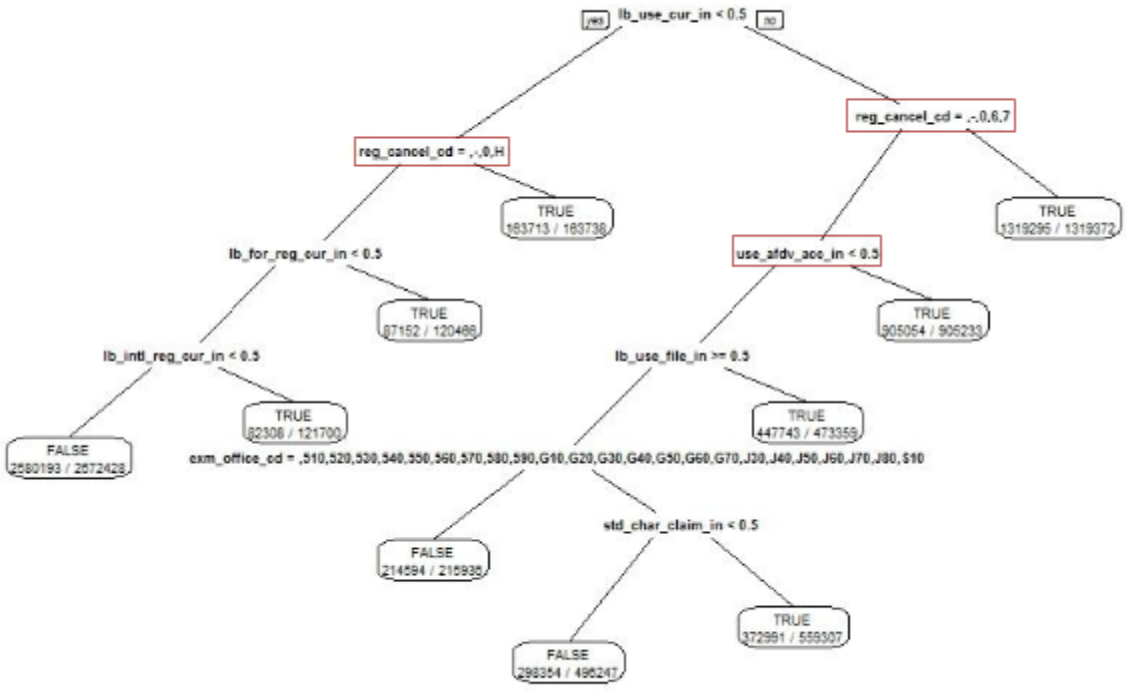
[3] http://www.uspto.gov/sites/default/files/documents/vartable_2015.pdf
[4] http://www.uspto.gov/sites/default/files/products/applications-documentation.pdf

General Methodology

The dataset does not provide a binary dependent variable indicating whether a trademark application has at any given time resulted in successful registration, but rather a numerical timeline code for the current case file status in the form of the `cfh_status_cd` variable. The in-depth description of this variable indicates 700 (Registered) as the initial threshold of registration, whereafter all subsequent codes indicate ongoing statuses including, but not limited to, cancellation and affirmation of the ongoing commercial use of trademarks in the form of a Statement of Use (SOU).

```
case2012$is_passed <- case2012$cfh_status_cd >= 700
```

This criterion for a dependent variable, however, introduces the problem that variables irrelevant to initial registration, such as registration cancel code (`reg_cancel_cd`), may be improperly introduced as predictors, as in the following example:



As these variables distort and overfit the models, preventing an effective prediction to test data, they are deliberately excluded via nullification to remain in the scope of this project, as in the following instance:

```
case2012$reg_cancel_cd = NULL
```

Furthermore, as many of the earlier trademark registrations have been backlogged as a result of only recently introduced status records and, presumably, legal procedure, all applications prior to 1980 have been excluded using the “lubridate” package, as below:

```
library(lubridate)
alldate <- ymd(case2012$filing_dt)
alldate <- year(alldate)
case2012$year = alldate
case2012 <- subset(case2012, year >= 1980)
case2012$year = NULL
```

All other dates and id-related categorical variables were removed for being irrelevent to initial registration, and in order to simplify the models:

```
case2012in <- select(case2012, ends_with("in"))
case2012cd <- select(case2012, ends_with("cd"))
case2012 <- bind_cols(case2012in, case2012cd)
rm(case2012cd)
rm(case2012in)
```

The removal and subsetting of the dataset results in an uneven distribution of TRUE and FALSE outcomes in the dependent `is_passed` variable. Distribution of the dependent variable is evened to reduce bias:

```
case2012TRUE <- subset(case2012, is_passed == TRUE)
case2012TRUE_EVEN <- sample_frac(case2012TRUE, size = 0.7971)
case2012FALSE <- subset(case2012, is_passed == FALSE)
case2012 <- bind_rows(case2012TRUE_EVEN, case2012FALSE)
rm(list = setdiff(ls(), "case2012"))
```

One of the four-character alpha-numeric variables--Trademark Drawing Code (`mark_draw_cd`)--required separation into two new distinct categorical variables, `draw_cd1` and `draw_cd2`, representing mark physical characteristics (1 character) and sizing codes (3 characters), respectively. NULL and single-character drawing codes occurred in the single digits, and were removed to avoid errors in the separate() function:

```
library(tidyr)
case2012d <- subset(case2012, mark_draw_cd != "")
case2012d <- subset(case2012d, mark_draw_cd != "1")
case2012d <- subset(case2012d, mark_draw_cd != "2")
case2012d <- subset(case2012d, mark_draw_cd != "3")
case2012d <- subset(case2012d, mark_draw_cd != "4")
draw_cd <- separate(case2012d,mark_draw_cd, c("draw_cd1","draw_cd2"), sep = 1)
case2012 <- draw_cd
rm(case2012d)
rm(draw_cd)
```

Finally, the 1980-2012 dataset is divided into a 75-25% Training-Test split for analysis via CART, RandomForest, and Logistic Regression:

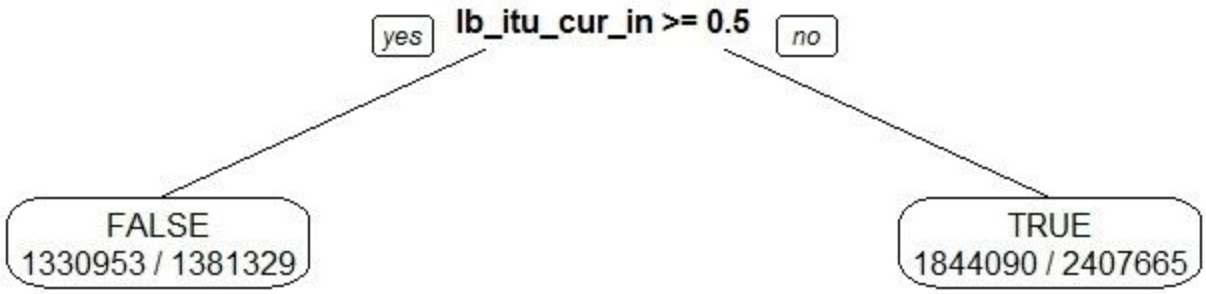
```
library(rpart)
library(rpart.plot)
library(caTools)
sampleSplit = sample.split(case2012$is_passed, SplitRatio = 0.75)
caseTrain = subset(case2012, sampleSplit == TRUE)
caseTest = subset(case2012, sampleSplit == FALSE)
rm(case2012)
rm(sampleSplit)
```

CART

Create a non-cross-validated decision tree using the training data:

```
caseTree = rpart(is_passed ~ ., data = caseTrain, method = "class", control =
rpart.control(minbucket = 25))
prp(caseTree, extra = 2, varlen = 0)
```

Resulting in:



The model trains exclusively to the "Intent to Use Application Currently" (`lb_itu_cur_in`) predictor, implying by fit-to-model that approximately 96% of all trademarks submitted under this filing basis will fail, whereas approximately 77% of all applications filed under the three other use bases will succeed.[5] Overall accuracy is approximately 83.8%.

The model is then tested against the training data. The 100+ level categorical variable `draw_cd2` has all new levels not present in the test data removed to avoid errors:

```
new <- which(!(caseTest$draw_cd2 %in% levels(caseTrain$draw_cd2)))
caseTest$draw_cd2[new] <- NA
predictCase = predict(caseTree, newdata = caseTest, type = "class")
```

[5] Other use bases include filing under "Use in Commerce," and international registration under either the Paris Convention or Madrid Protocol.

```
table(caseTest$is_passed, predictCase)
```

The results similarly demonstrate an overall accuracy of 83.729%, with a sensitivity of 97.326% and specificity of 70.133%:

```
> table(caseTest$is_passed, predictCase)
       predictCase
         FALSE   TRUE
  FALSE 442894 188615
  TRUE   16886 614603
> (614603 + 442894) / (614603 + 442894 + 16886 + 188615)
[1] 0.8372911
> (614603) / (614603 + 16886)
[1] 0.97326
> (442894) / (442894 + 188615)
[1] 0.7013265
```

The model is then refined according to a specific control parameter:

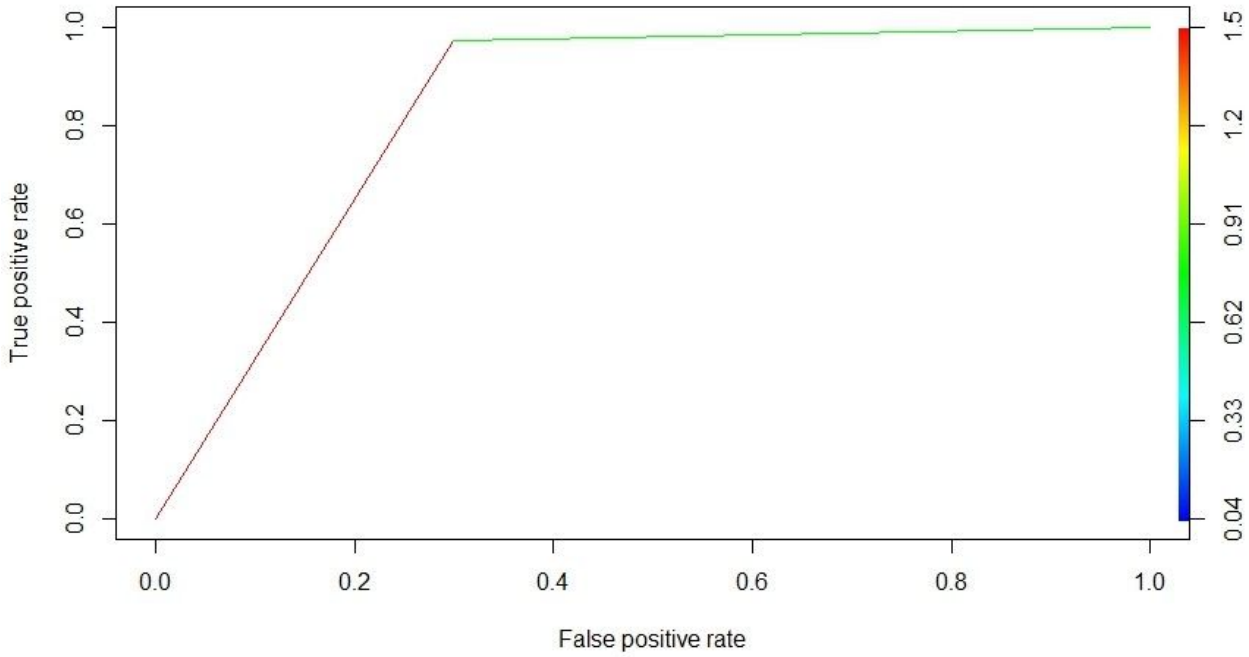
```
library(caret)
library(e1071)
fitControl = trainControl(method = "cv", number = 10)
cartGrid = expand.grid(.cp=seq(0,1,0.01))
caseTrain$is_passed = as.factor(caseTrain$is_passed)
caseTrain$draw_cd1 = as.factor(caseTrain$draw_cd1)
caseTrain$draw_cd2 = as.factor(caseTrain$draw_cd2)
train(is_passed ~ ., data = caseTrain, method = "rpart", trControl = fitControl, tuneGrid =
cartGrid) ## cp = 0.08
```

The results are slightly better, though not significantly different from the previous model, demonstrating an overall accuracy of 83.731%, with a sensitivity of 97.339% and specificity of 70.125%:

```
> table(caseTest$is_passed, predictCV)
       predictCV
         FALSE   TRUE
  FALSE 442845 188664
  TRUE   16806 614683
> (614683 + 442845) / (614683 + 442845 + 16806 + 188664)
[1] 0.8373157
> (614683) / (614683 + 16806)
[1] 0.9733867
> (442845) / (442845 + 188664)
[1] 0.7012489
```

Using the cross-validated model, an ROC curve with an AUC of approximately .837 is drawn:

```
> predictROCV = predict(caseTreeCV, newdata = caseTest)
> predCV = prediction(predictROCV[,2],caseTest$is_passed)
> perfcv = performance(predCV,"tpr","fpr")
> plot(perfcv, colorize = TRUE)
> performance(predCV, "auc")@y.values
[[1]]
[1] 0.8373178
```

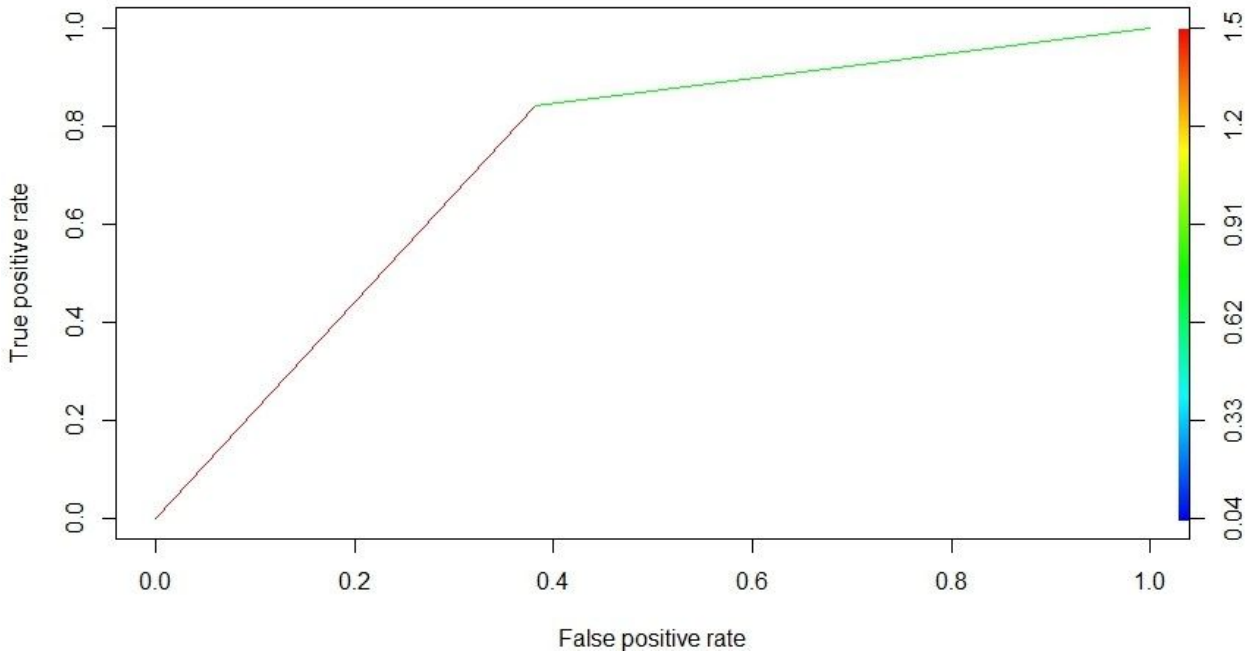


Performance of the decision-tree is then tested against the most recent trademark applications (2013-2015). The resulting overall accuracy is 71.743%, with a sensitivity of 84.098% and specificity of 61.859%:

```
> table(case2015d$is_passed, predtest)
       predtest
         FALSE   TRUE
  FALSE 357683 220536
  TRUE   73565 389037
> (389037 + 357683) / (389037 + 357683 + 73565 + 220536)
[1] 0.7174336
> (389037) / (389037 + 73565)
[1] 0.8409756
> (357683) / (357683 + 220536)
[1] 0.6185943
```

The AUC of the final model is approximately .730:

```
> predictROC2 = predict(caseTreeCV, newdata = case2015d)
> pred2 = prediction(predictROC2[,2],case2015d$is_passed)
> perf2 = performance(pred2,"tpr","fpr")
> plot(perf2, colorize = TRUE)
> performance(pred2, "auc")@y.values ## .730 AUC
[[1]]
[1] 0.729785
```



The CART investigation closes with a chart to predict variable importance in constructing the logistic regression model:

```
> varImp(caseTreeCV)
                      Overall
draw_cd2             63661.36
lb_itu_cur_in       934107.36
lb_itu_file_in      208478.81
lb_use_cur_in       732419.03
lb_use_file_in      178291.96
amend_lb_for_app_in      0.00
amend_lb_for_reg_in      0.00
amend_lb_itu_in          0.00
amend_lb_use_in          0.00
cancel_pend_in           0.00
cert_mark_in             0.00
chg_reg_in               0.00
coll_memb_mark_in        0.00
coll_serv_mark_in        0.00
coll_trade_mark_in       0.00
draw_color_cur_in        0.00
draw_color_file_in       0.00
concur_use_in            0.00
concur_use_pend_in       0.00
draw_3d_cur_in           0.00
draw_3d_file_in          0.00
lb_for_app_cur_in        0.00
lb_for_reg_cur_in        0.00
lb_intl_reg_cur_in       0.00
lb_for_app_file_in       0.00
lb_for_reg_file_in       0.00
lb_intl_reg_file_in      0.00
lb_none_cur_in           0.00
for_priority_in          0.00
interfer_pend_in         0.00
opposit_pend_in          0.00
amend_principal_in       0.00
concur_use_pub_in        0.00
renewal_file_in          0.00
repub_12c_in             0.00
incontest_file_in        0.00
acq_dist_in              0.00
acq_dist_part_in         0.00
use_afdv_file_in         0.00
use_afdv_par_acc_in      0.00
serv_mark_in             0.00
std_char_claim_in        0.00
amend_supp_reg_in        0.00
supp_reg_in              0.00
trade_mark_in            0.00
lb_none_file_in          0.00
ir_first_refus_in        0.00
ir_priority_in           0.00
related_other_in         0.00
draw_cd1                 0.00
ir_status_cd             0.00
```

Here, as before, the variable `lb_itu_cur_in` has the majority influence on the model, while the trademark sizing code (`draw_cd2`) has at least some marginal value as a predictor.

Random Forest

The Random Forest model requires virtually identical data preparation as that of CART, but as Random Forest is a significantly more RAM-intensive algorithm, it's advisable to use a significantly smaller sample from the raw data:

```
case2012TRUE <- subset(case2012, is_passed == TRUE)
case2012TRUE_EVEN <- sample_frac(case2012TRUE, size = 0.7971)
case2012FALSE <- subset(case2012, is_passed == FALSE)
case2012 <- bind_rows(case2012TRUE_EVEN, case2012FALSE) %>% sample_frac(size = 0.04)
rm(list = setdiff(ls(), "case2012"))
```

The randomForest package is loaded, and categorical variables are converted into factors to avoid errors:

```
library(randomForest)
caseTest$is_passed = as.factor(caseTest$is_passed)
caseTrain$is_passed = as.factor(caseTrain$is_passed)
caseTest$draw_cd1 = as.factor(caseTest$draw_cd1)
caseTrain$draw_cd1 = as.factor(caseTrain$draw_cd1)
caseTest$draw_cd2 = as.factor(caseTest$draw_cd2)
caseTrain$draw_cd2 = as.factor(caseTrain$draw_cd2)
```

As `draw_cd2` has 70 levels, however, the model returns the following error:

```
> caseTreeTrain = randomForest(is_passed ~ ., data = caseTrain, nodesize = 25, ntree = 200)
Error in randomForest.default(m, y, ...) :
  Can not handle categorical predictors with more than 53 categories.
```

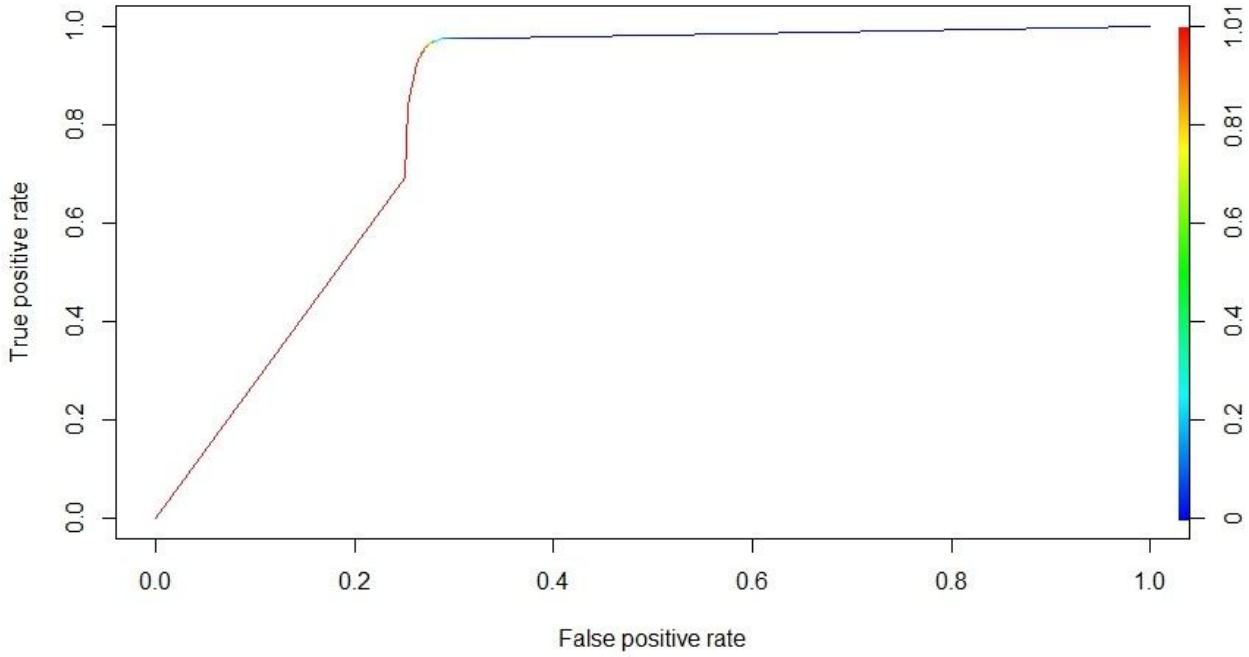
So, upon purging `draw_cd2` from the data set, the randomForest() algorithm returns an overall accuracy of 84.525%, with a sensitivity of 96.806% and specificity of 72.117%:

```
> table(caseTest$is_passed, predictForest)
       predictForest
        FALSE  TRUE
  FALSE 18123  7007
  TRUE    811 24579
> (24579 + 18123) / (24579 + 18123 + 811 + 7007)
[1] 0.8452494
> (24579) / (24579 + 811)
[1] 0.9680583
> (18123) / (18123 + 7007)
[1] 0.7211699
```

An ROC curve with an AUC curve of approximately .825 is generated from the test data:

```
library(ROCR)
predictROC = predict(caseTreeTrain, newdata = caseTest, type = "prob")
pred = prediction(predictROC[,2],caseTest$is_passed)
perf = performance(pred,"tpr","fpr")
plot(perf, colorize = TRUE)
```

```
> performance(pred, "auc")@y.values
[[1]]
[1] 0.8247508
```

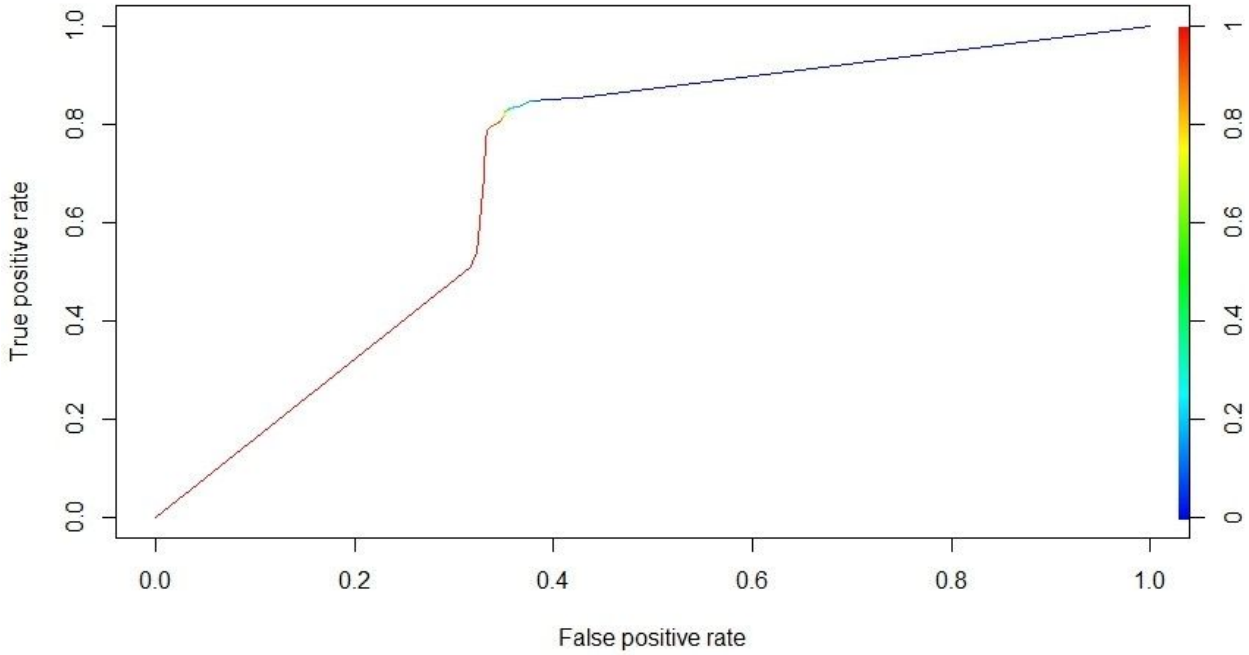


When applying the randomForest prediction to the 2013-2015 test data, the console returns an overall accuracy of 72.735%, a sensitivity of 82.870%, and a specificity of 64.625%:

```
> table(case2015d$is_passed, predForest2)
       predForest2
         FALSE   TRUE
  FALSE 373623 204518
  TRUE   79237 383333
> (383333 + 373623) / (383333 + 373623 + 79237 + 204518)
[1] 0.7273451
> (383333) / (383333 + 79237)
[1] 0.8287027
> (373623) / (373623 + 204518)
[1] 0.6462489
```

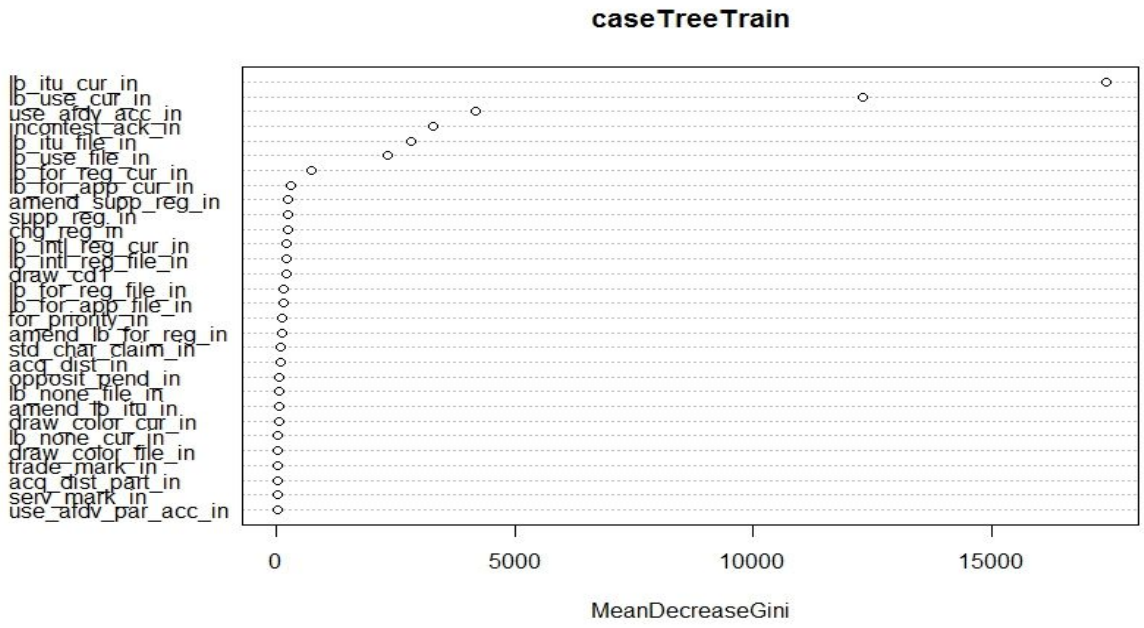
With an AUC of .701:

```
predictROC2 = predict(caseTreeCV, newdata = case2015d)
pred2 = prediction(predictROC2[,2],case2015d$is_passed)
perf2 = performance(pred2,"tpr","fpr")
plot(perf2, colorize = TRUE)
> performance(pred2, "auc")@y.values
[[1]]
[1] 0.7006521
```



Despite sampling only 4% of the raw data to produce a randomForest() model, the overall accuracy improved over CART by a little over 1%.

Plotting feature importance with varImpPlot() suggests similar features to consider for Logistic Regression. The variable `draw_cd2`, however, is absent because of incompatibility with the randomForest() function:



Logistic Regression

Referring to the earlier CART varImp() chart:

```
> varImp(caseTreeCV)
                     Overall
draw_cd2            63661.36
lb_itu_cur_in      934107.36
lb_itu_file_in     208478.81
lb_use_cur_in      732419.03
lb_use_file_in     178291.96
amend_lb_for_app_in        0.00
amend_lb_for_reg_in        0.00
amend_lb_itu_in            0.00
amend_lb_use_in            0.00
cancel_pend_in             0.00
cert_mark_in               0.00
```

Five variables appear to have significance on the model. Of these, four concern the status of application as it concerns “use bases,” or whether the trademark has previously been used for interstate or international commerce. The “Intent-to-Use” basis variable--indicating that marks have not been used prior to application--holds the preponderance of influence on the registration outcome of new, federally-filed trademark applications, with “Use in Commerce”--indicating prior use of trademarks in interstate commerce--holding secondary

influence. As any application can only be filed under one use basis at a time, only variables with the “Intent to Use” basis will be included in the model. The variables `lb_itu_cur_in` and `lb_itu_file_in` are complementary, as they concern whether applications have been filed under the “Intent-to-Use” basis, and whether said basis was submitted at time of filing, respectively. Combining the two variables into a new one, `combi`, may produce a more predictive model by transforming two distinct binary variables into a three-level factor:

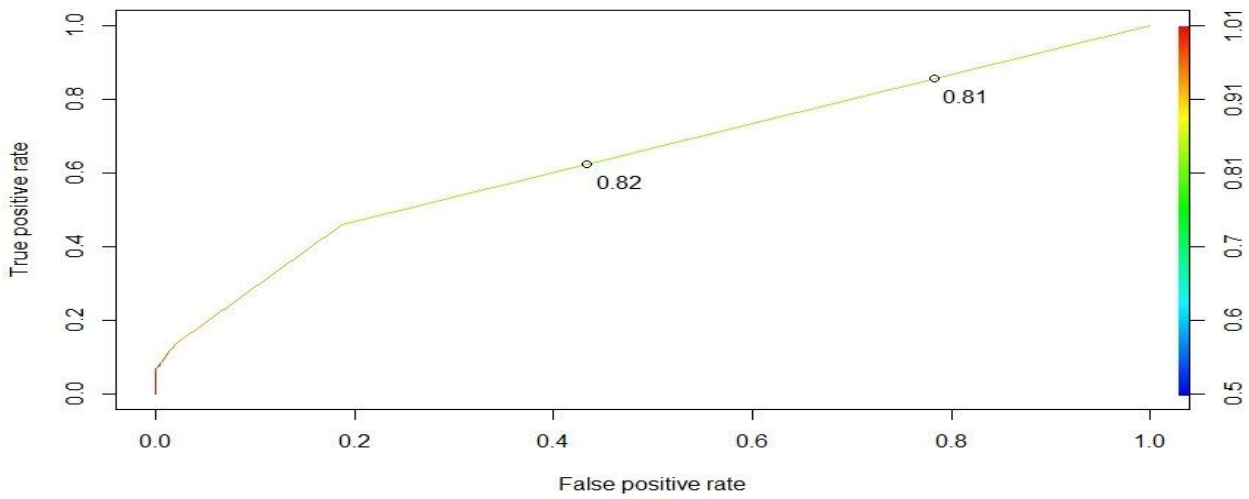
```
case2012 <- mutate(case2012, combi = lb_itu_cur_in + lb_itu_file_in)
```

Using this new variable, four logistic regression models are created and tested for fit to the training data:

```
logPassed1 <- glm(is_passed ~ lb_itu_cur_in + lb_itu_file_in, data = caseTrain, family =
binomial) ## AIC: 7556
logPassed2 <- glm(is_passed ~ combi, data = caseTrain, family = binomial) ## AIC: 7554
logPassed3 <- glm(is_passed ~ lb_itu_cur_in + lb_itu_file_in + draw_cd2, data = caseTrain,
family = binomial) ## AIC: 7391
logPassed4 <- glm(is_passed ~ combi + draw_cd2, data = caseTrain, family = binomial) ## AIC:
7390
```

Model	**AIC**	**Fit to Training Data (AUC)**
logPassed1 (control)	7556	0.5889312
logPassed2	7554	0.5889209
logPassed3	7391	0.6451761
logPassed4	7390	0.6451637

Feature engineering appears to negatively affect fit, so logPassed3 will be used to make predictions, with 0.82 as the prediction threshold:



The console returns an overall accuracy of 49.125%, a sensitivity of 54.080%, and a specificity of 18.699%:

```
> table(caseTrain$is_passed, PredictTrain < 0.82)

       FALSE TRUE
  FALSE   250 1087
  TRUE   3770 4440
> (4440 + 250) / (4440 + 250 + 3770 + 1087)
[1] 0.4912538
> (4440) / (4440 + 3770)
[1] 0.5408039
> (250) / (250 + 1087)
[1] 0.1869858
```

Fitting the model to the test data returns an overall accuracy of 85.998%, a sensitivity of 100.00%, and a specificity of 0.0000%:

```
> table(caseTest$is_passed, predtest < 0.82)

       FALSE
  FALSE   446
  TRUE   2737
> (2737) / (2737 + 446)
[1] 0.8598806
> (2737) / (2737)
[1] 1
> (0) / (446)
[1] 0
```

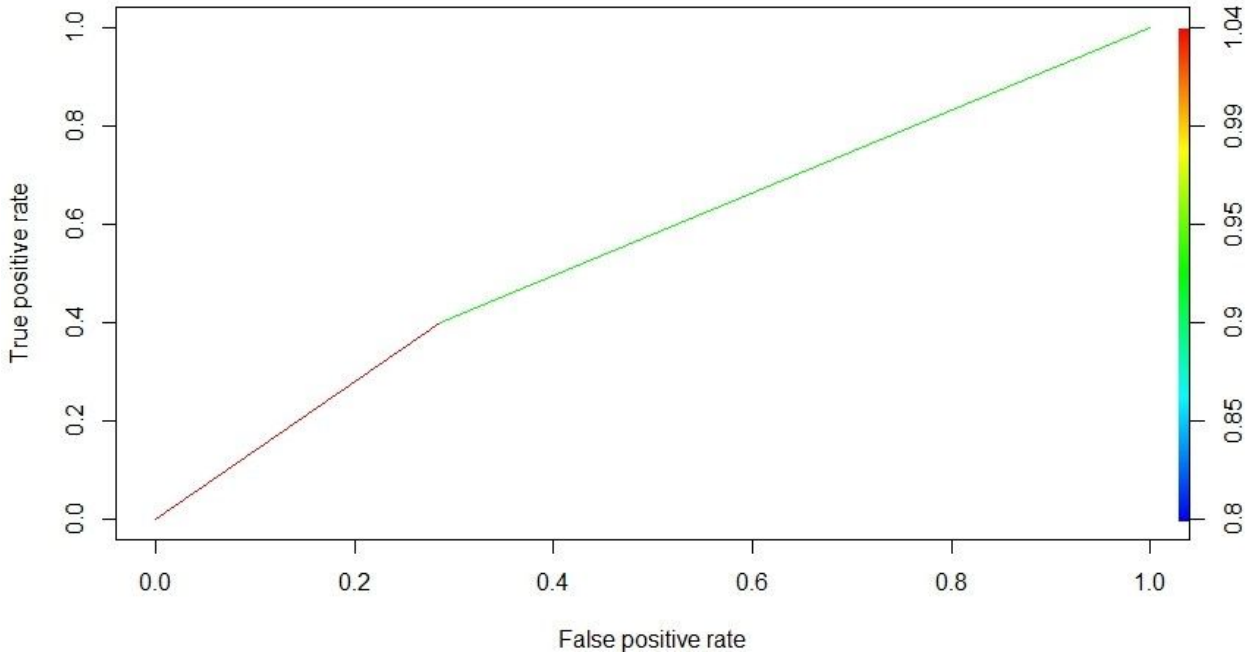
Fitting the model to the 2013-2015 test data returns an overall accuracy of 60.095%, a sensitivity of 60.131%, and a specificity of 28.571%:

```
> table(case2015d$is_passed, predtest2 < 0.82)

         FALSE   TRUE
  FALSE    200    500
  TRUE  245921 370901
> (370901 + 200) / (370901 + 200 + 245921 + 500)
[1] 0.6009519
> (370901) / (370901 + 245921)
[1] 0.6013096
> (200) / (200 + 500)
[1] 0.2857143
```

The ROC curve for the 2013-2015 test data returns an AUC of .556:

```
> performance(pred2, "auc")@y.values
[[1]]
[1] 0.556488
```



Overall Results

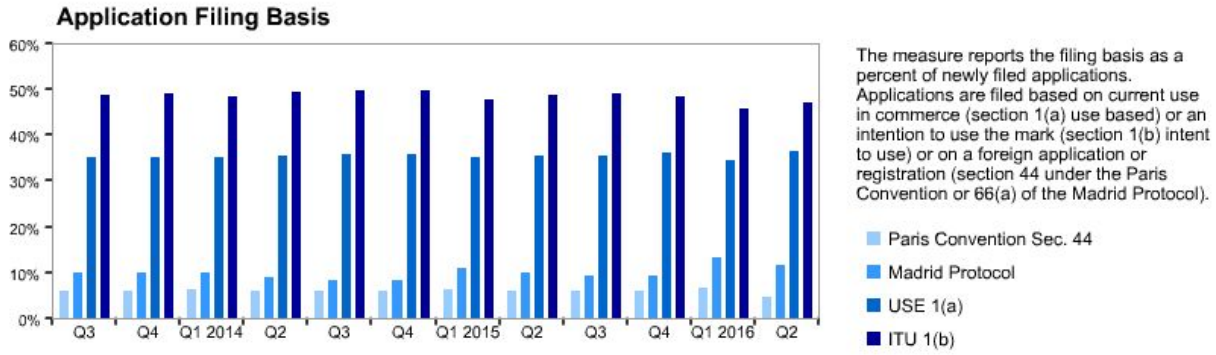
CART and Random Forest appear similarly effective in predicting registration outcomes for the 2013-2015 test data, with Random Forest generating the best overall accuracy of approximately

70%, whereas CART produces a slightly superior sensitivity to the test data of approximately 84%. Selecting an overall superior model is difficult due to the proximity of results and the slight possibility of sampling bias. There remains a strong likelihood, however, that Random Forest would have produced a comprehensively superior model had it accepted the trademark sizing code (`draw_cd2`) as a predictor.

Logistic Regression, on the other hand, produces overall results less than 10% as effective as those of CART and Random Forest. This appears to imply that the decision-oriented trademark approval process responds less effectively to linear models.

Interpreting the Results

The models imply a powerful negative correlation between successful registration of federal trademarks and not providing prior evidence of having used a trademark in interstate or international commerce, i.e., filing under “Intent-to-Use.” Inversely, demonstrating prior use of trademarks under “Use in Commerce” or other, foreign trademark application bases such as that under the Madrid Protocol correlates positively to successful trademark registration. The preponderance of “Intent-to-Use” over “Use in Commerce” as a predictor likely corresponds to the significantly larger annual pool of applications filed under the former, as shown by the following exploratory chart provided by the USPTO[6]:



Recommendations

Although the lengthier application process and more aggressive fee schedule of applications under the “Intent-to-Use” basis could precipitate higher rates of trademark abandonment on the part of the registrant, pending more detailed evidence to the contrary it is statistically reasonable to conclude that over 70% of applications filed under “Intent-to-Use” will fail under that basis alone. Unless a trademark registrant has compelling cause to assume his/her/their desired trademark will be pursued by a competitor to significant loss, it is to the registrant’s financial

[6] http://www.uspto.gov/dashboards/trademarks/main.dashxml

advantage to utilize the trademark in at least 2 states/countries prior to filing for federal registration via the USPTO.

Suggestions for Further Research

This project focused specifically on the likelihood of trademark registration based upon a limited set of binary and categorical variables, without delving into secondary predictors or dependent variables that could prove useful on a more granular level. Further research could include linear regression modeling to determine expected trademark application timeframes and their influence on trademark abandonment. Research may also wish to subset models by less common filing bases such as under filed under the Madrid Protocol and Paris Convention. Finally, text analytics based on trademark length, relevant industry, and other characteristics may also provide valuable insights for registrants pursuing successful trademarks.