Lending Club

Prashan A. Welipitiya

3/14/2021

In this project we use the LendingClub data from 2012 to 2014 to predict the loan status classification of an individual based on certain predictors.

After getting the data the predictors we use are annual income, fico score range, funded amount, last payment, interest rate and average current balance. We will run logistic regression, decision trees and random forest. Knn was not an option due to the size of data.

```
library(pacman)
p_load(tidyverse, tidymodels, lubridate, janitor, rpart, rpart.plot, C50)
```

```
data <- read.csv("data/lending_club_data_2012_2014.csv")
```

```
ls <- data %>%
  dplyr::select(-id, -member_id, -url) %>%
  dplyr::select(loan_status, annual_inc,fico_range_low, fico_range_high, funded_amnt, last_pymnt_amnt, 
  drop_na(loan_status) %>%
  dplyr::filter(loan_status=="Charged Off"|loan_status=="Fully Paid") %>%
  remove_empty(which = c("rows", "cols"), quiet = TRUE)
```

```
head(ls)
```

```
##   loan_status annual_inc fico_range_low fico_range_high funded_amnt
## 1 Charged Off      58000            710             714       10400
## 2  Fully Paid      78000            750             754       15000
## 3  Fully Paid      69000            680             684        9600
## 4 Charged Off      50000            685             689        7650
## 5  Fully Paid      63800            685             689       21425
## 6  Fully Paid      75000            675             679       17000
##   last_pymnt_amnt int_rate avg_cur_bal
## 1          321.08     6.99        9536
## 2        12017.81    12.39       29828
## 3         9338.58    13.66        3214
## 4           17.70    13.66        5857
## 5        17813.19    15.59        4232
## 6        10888.01    13.66       17456
```

```
split <- initial_split(ls, prop = 0.75)

ls_recipe <- training(split) %>%
  recipe(loan_status ~ .) %>%
```

```
  step_nzv(all_predictors()) %>%
  step_medianimpute(all_numeric()) %>%
  step_center(all_numeric(), -all_outcomes()) %>%
  step_scale(all_numeric(), -all_outcomes()) %>%
  prep()
```

```
testing <- ls_recipe %>%
  bake(testing(split))

training <- juice(ls_recipe)
```

```
samp <- sample_n(training, size = 62000, replace = FALSE)
```

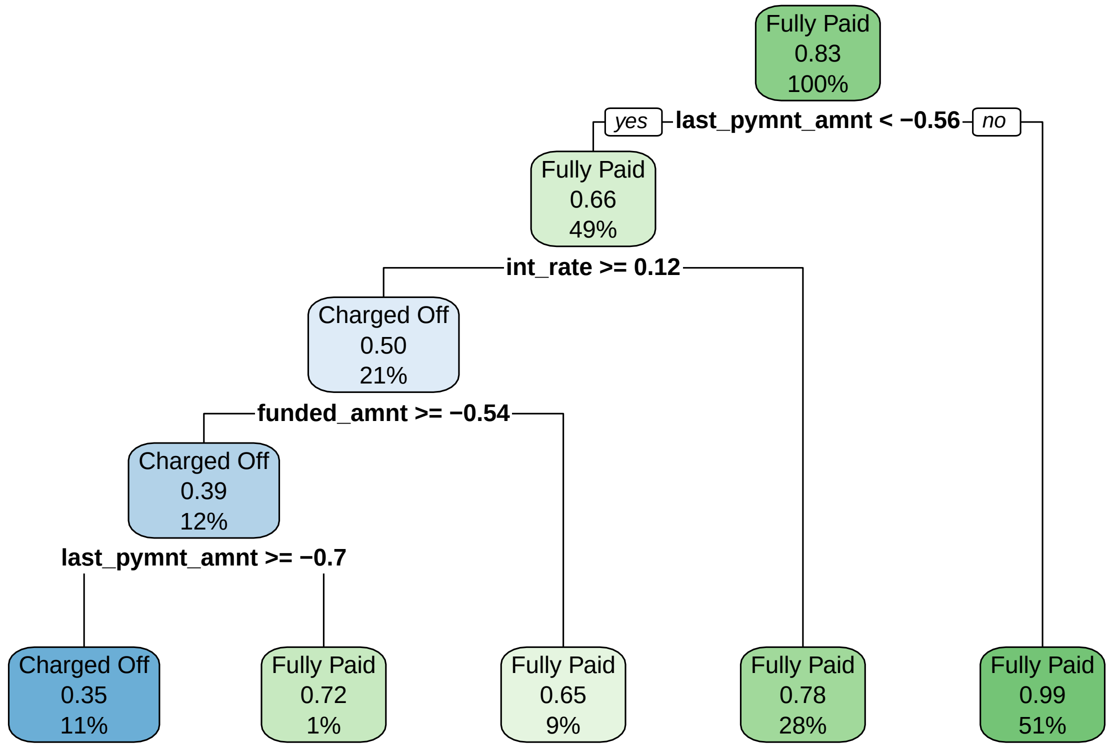
Fitting Models

Null Model

```
mod_null <- glm(loan_status ~ 1,data=training, family=binomial)
```

Decision Tree

```
mod_tree <- rpart(loan_status ~ ., data = training)
rpart.plot(mod_tree)
```




```r
p.rpart <- predict(mod_tree, testing)
summary(p.rpart)
```

```
##   Charged Off        Fully Paid
##  Min.   :0.01157   Min.   :0.3546
##  1st Qu.:0.01157   1st Qu.:0.7786
##  Median :0.01157   Median :0.9884
##  Mean   :0.17210   Mean   :0.8279
##  3rd Qu.:0.22140   3rd Qu.:0.9884
##  Max.   :0.64541   Max.   :0.9884
```

```r
summary(as.numeric(testing$loan_status))
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   2.000   2.000   1.828   2.000   2.000
```

```r
cor(p.rpart, as.numeric(testing$loan_status))
```

```
##                   [,1]
## Charged Off -0.5458255
## Fully Paid   0.5458255
```

Random Forest

```
mod_rf <- rand_forest(trees = 100) %>%
  set_engine("randomForest") %>%
  set_mode("classification") %>%
  fit(loan_status ~ ., data = training)
```

Logistic Regression

```
mod_glm <- logistic_reg() %>%
  set_engine("glm") %>%
  set_mode("classification") %>%
  fit(loan_status ~ ., data = training)
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

Evaluations

```
mod_glm %>%
  predict(testing) %>%
  bind_cols(testing) %>%
  metrics(truth = loan_status, estimate = .pred_class)
```

```
## # A tibble: 2 x 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy binary         0.851
## 2 kap      binary         0.357
```

```
mod_rf %>%
  predict(testing) %>%
  bind_cols(testing) %>%
  metrics(truth = loan_status, estimate = .pred_class)
```

```
## # A tibble: 2 x 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy binary         0.876
## 2 kap      binary         0.509
```

```
mod_tree %>%
  predict(testing, type = "class") %>%
  bind_cols(testing) %>%
  metrics(truth = loan_status, estimate = ...1)
```

```
## New names:
## * NA -> ...1
```

```
## # A tibble: 2 x 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy binary         0.863
## 2 kap      binary         0.438
```

Improvements

```
mod_glm2 <- glm(loan_status ~ ., data = training, family = binomial)
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
summary(mod_glm2)
```

```
##
## Call:
## glm(formula = loan_status ~ ., family = binomial, data = training)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -8.4904   0.0000   0.0356   0.6038   2.5945
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)      6.760821   0.059992 112.696   <2e-16 ***
## annual_inc       0.202031   0.009650  20.935   <2e-16 ***
## fico_range_low   9.767727  24.446033   0.400    0.689
## fico_range_high -9.686805  24.446052  -0.396    0.692
## funded_amnt     -0.693925   0.007334 -94.616   <2e-16 ***
## last_pymnt_amnt  9.058842   0.091716  98.770   <2e-16 ***
## int_rate        -0.665402   0.006778 -98.170   <2e-16 ***
## avg_cur_bal      0.071962   0.007467   9.638   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 283552  on 308454  degrees of freedom
## Residual deviance: 183941  on 308447  degrees of freedom
## AIC: 183957
##
## Number of Fisher Scoring iterations: 10
```

```
mod_glm2 <- logistic_reg(penalty = 0.001, mixture = 0.5) %>%
  set_engine("glmnet") %>%
  set_mode("classification") %>%
  fit(loan_status ~ annual_inc + funded_amnt + last_pymnt_amnt + int_rate + avg_cur_bal, data = trainin

mod_glm2 %>%
  predict(testing) %>%
  bind_cols(testing) %>%
  metrics(truth = loan_status, estimate = .pred_class)
```

```
## # A tibble: 2 x 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy binary         0.850
## 2 kap      binary         0.329
```

```
mod_rf2 <- rand_forest(trees = 100) %>%
  set_engine("ranger") %>%
  set_mode("classification") %>%
  fit(loan_status ~ annual_inc + funded_amnt + last_pymnt_amnt + int_rate + avg_cur_bal, data = training

mod_rf2 %>%
  predict(testing) %>%
  bind_cols(testing) %>%
  metrics(truth = loan_status, estimate = .pred_class)
```

```
## # A tibble: 2 x 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy binary         0.889
## 2 kap      binary         0.568
```

```
mod_tree2 <- C5.0(loan_status ~ annual_inc + funded_amnt + last_pymnt_amnt + int_rate + avg_cur_bal, da

mod_tree2 %>%
  predict(testing, type = "class") %>%
  bind_cols(testing) %>%
  metrics(truth = loan_status, estimate = ...1)
```

```
## New names:
## * NA -> ...1
```

```
## # A tibble: 2 x 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy binary         0.887
## 2 kap      binary         0.531
```

We found that random forest was the effective model in classifying Loan status with an accuracy of 88.75. Decision tree with C5.0 was highly effective also.