CAPSTONE FINAL

Michael Bubulka

1/2/2021

Executive Summary

This project was my attempt to learn more about various models. I explored a smaller dataset than the MovieLens which allowed me to use the caret model to try various models. I chose the metric RMSE as it was familiar but other metrics could have been chosen.

The first step was to setup the required packages.

```
######  install all required packages for this project
install.packages("caret")

library(caret)

if(!require(tidyverse)) install.packages("tidyverse", repos = "http://cran.us
.r-project.org")if(!require(caret)) install.packages("caret", repos = "http:/
/cran.us.r-project.org")
if(!require(data.table)) install.packages("data.table", repos = "http://cran.
us.r-project.org")library(tidyverse)
library(caret)
library(data.table)
# Adding Additional Packages
library (broom)
library(lubridate)library(tibble)
install.packages("randomForest")

library(randomForest)

install.packages("matrixStats")

library(matrixStats)

library(purrr)
install.packages("AppliedPredictiveModeling")

library(AppliedPredictiveModeling)
install.packages("e1071")

library(e1071)
library(readr)
library(readxl)
library(ggplot2)
install.packages("caretEnsemble")
```

```r
library(caretEnsemble)

install.packages("RANN")

install.packages("arm")

library(arm)

install.packages("penalized")

library(penalized)

install.packages("pls")

library(pls)

install.packages("quantregForest")

library(quantregForest)

library(dplyr)
```

I chose a dataset from Kaggle which was a csv file. I converted it to an excel file and loaded it into Rstudio () Reference: kaggle datasets download -d sootersaalu/amazon-top-50-bestselling-books-2009-2019)

I chose a smaller dataset that would be easier to run. This is a look at Amazons top 50 best selling books from 2009 to 2019. It maybe necessary to download the excel file which will be with the uploads. I also adapted the excel file to remove the Name column, it causes the models to crash or run slowly.

```r
##Data was downloaded as a CSV and converted into excel file.   Excel  file w
ill be attached separately.
dataset <- read_xlsx("dataset_1.xlsx")

Data pulled from this source:  Will include a copy of excel file.
```

@misc{sooter saalu_2020, title={Amazon Top 50 Bestselling Books 2009 - 2019}, url={https://www.kaggle.com/dsv/1556647}, DOI={10.34740/KAGGLE/DSV/1556647}, publisher={Kaggle}, author={Sooter Saalu}, year={2020} }

I need to clean the data to make it useful for the various models. Part of that was converting the Genre from a character to factor based on two categories of Fiction and Non Fiction. Also I converted Author from character to Factor because I figured various user ratings could be influence by the author. I then set up a training set and test set.

```r
### Cleaning the data and creating the trainset and test set.
dataset_tidy <- as.data.frame(dataset)
dataset_tidy$Genre <- as.factor(dataset_tidy$Genre)
dataset_tidy$Author <- as.factor(dataset_tidy$Author)
set.seed(1, sample.kind ="Rounding")
```

```
## Warning in set.seed(1, sample.kind = "Rounding"): non-uniform 'Rounding' s
ampler
## used
```

```r
y <- dataset_tidy$`User Rating`

test_index <- createDataPartition(y, times = 1, p = 0.7, list = FALSE)

train_set <- dataset_tidy%>% slice(test_index)
test_set <- dataset_tidy %>% slice(-test_index)
```

I explored the data to see which variables might be important. The preProcess function used as a stand alone was an interesting method to explore the data. The range of the User Rating goes from 3.3 to 4.9 with an average of 4.618. I will use other preprocessing methods to explore the dataset. Here is with it set to scale, which divides values by standard deviation.

```r
##Sumarize the data with Scale
summary(dataset_tidy[,1:6])
```

```
##                                    Author      User Rating       Reviews    
##  Jeff Kinney                          : 12   Min.   :3.300   Min.   :   37  
##  Gary Chapman                         : 11   1st Qu.:4.500   1st Qu.: 4058  
##  Rick Riordan                         : 11   Median :4.700   Median : 8580  
##  Suzanne Collins                      : 11   Mean   :4.618   Mean   :11953  
##  American Psychological Association: 10   3rd Qu.:4.800   3rd Qu.:17253  
##  Dr. Seuss                            :  9   Max.   :4.900   Max.   :87841  
##  (Other)                              :486                                  
##      Price             Year                 Genre    
##  Min.   :  0.0   Min.   :2009   Fiction    :240  
##  1st Qu.:  7.0   1st Qu.:2011   Non Fiction:310  
##  Median : 11.0   Median :2014                    
##  Mean   : 13.1   Mean   :2014                    
##  3rd Qu.: 16.0   3rd Qu.:2017                    
##  Max.   :105.0   Max.   :2019                    
## 
```

```r
#### calculate the pre-process parameters from the dataset
preprocessParams <- preProcess(dataset_tidy[,1:6], method=c("scale"))
preprocessParams
```

```
## Created from 550 samples and 6 variables
## 
## Pre-processing:
##   - ignored (2)
##   - scaled (4)
```

```r
##Sumarize the data with Center and Scale
summary(dataset_tidy[,1:6])
```

```
##                                  Author     User Rating       Reviews
##  Jeff Kinney                       : 12   Min.   :3.300   Min.   :   37
##  Gary Chapman                      : 11   1st Qu.:4.500   1st Qu.: 4058
##  Rick Riordan                      : 11   Median :4.700   Median : 8580
##  Suzanne Collins                   : 11   Mean   :4.618   Mean   :11953
##  American Psychological Association: 10   3rd Qu.:4.800   3rd Qu.:17253
##  Dr. Seuss                         :  9   Max.   :4.900   Max.   :87841
##  (Other)                           :486
##      Price             Year              Genre
##  Min.   :  0.0   Min.   :2009   Fiction    :240
##  1st Qu.:  7.0   1st Qu.:2011   Non Fiction:310
##  Median : 11.0   Median :2014
##  Mean   : 13.1   Mean   :2014
##  3rd Qu.: 16.0   3rd Qu.:2017
##  Max.   :105.0   Max.   :2019
##
```

```
#### calculate the pre-process parameters from the dataset
preprocessParams_1 <- preProcess(dataset_tidy[,1:6], method=c("scale","center
"))
preprocessParams_1
```

```
## Created from 550 samples and 6 variables
##
## Pre-processing:
##   - centered (4)
##   - ignored (2)
##   - scaled (4)
```

```
## Viewing the training and test dataset
str(test_set)
```

```
## 'data.frame':    163 obs. of  6 variables:
##  $ Author     : Factor w/ 248 levels "Abraham Verghese",..: 125 220 96 175
97 13 90 144 49 205 ...
##  $ User Rating: num  4.7 4.6 4.7 4.8 4.4 4.7 4.6 4.6 4.5 4.8 ...
##  $ Reviews    : num  17350 2052 21424 7665 12643 ...
##  $ Price      : num  8 22 6 12 11 15 8 2 8 13 ...
##  $ Year       : num  2016 2011 2017 2019 2011 ...
##  $ Genre      : Factor w/ 2 levels "Fiction","Non Fiction": 2 1 1 2 1 1 1
2 2 2 ...
```

```
head(test_set)
```

```
##                      Author User Rating Reviews Price Year       Genre
## 1                  JJ Smith         4.7   17350     8 2016 Non Fiction
## 2              Stephen King         4.6    2052    22 2011     Fiction
## 3             George Orwell         4.7   21424     6 2017     Fiction
## 4 National Geographic Kids          4.8    7665    12 2019 Non Fiction
## 5       George R. R. Martin         4.4   12643    11 2011     Fiction
## 6               Amor Towles         4.7   19699    15 2017     Fiction
```

```
str(train_set)
```

```
## 'data.frame':    387 obs. of  6 variables:
##  $ Author     : Factor w/ 248 levels "Abraham Verghese",..: 135 97 115 90
119 150 223 6 30 30 ...
##  $ User Rating: num  4.7 4.7 4.7 4.6 4.6 4.5 4.6 4.5 4.6 4.4 ...
##  $ Reviews    : num  18979 19735 5983 23848 4149 ...
##  $ Price      : num  15 30 3 8 32 5 17 4 6 6 ...
##  $ Year       : num  2018 2014 2018 2016 2011 ...
##  $ Genre      : Factor w/ 2 levels "Fiction","Non Fiction": 2 1 2 1 2 1 2
2 2 2 ...
```

```
head(train_set)
```

```
##                 Author User Rating Reviews Price Year       Genre
## 1  Jordan B. Peterson         4.7   18979    15 2018 Non Fiction
## 2 George R. R. Martin         4.7   19735    30 2014     Fiction
## 3         James Comey         4.7    5983     3 2018 Non Fiction
## 4     Fredrik Backman         4.6   23848     8 2016     Fiction
## 5       Jaycee Dugard         4.6    4149    32 2011 Non Fiction
## 6   Madeleine L'Engle         4.5    5153     5 2018     Fiction
```

```
## Looking at the average user
avg_user_rating <-  mean(train_set$`User Rating`)
avg_user_rating
```

```
## [1] 4.617313
```

This should match the preprocessing values and it does.

Step 1 was to set up a linear regression to see what impacts of the different variables might be. I did not use caret for this portion but will use it later.

```
## LM model without using caret
fit_lm <- lm(train_set$`User Rating` ~ Reviews + Year +Price + Genre, data =
train_set)
fit_lm$coeff
```

```
##     (Intercept)         Reviews            Year           Price
##   -2.636046e+01   -2.608245e-06    1.542908e-02   -1.775137e-03
## GenreNon Fiction
##   -7.133556e-02
```

```
y_hat <- predict(fit_lm, test_set)
```

```
rmse_lm_wo <- RMSE(y_hat, test_set$`User Rating`)
rmse_lm_wo
```

```
## [1] 0.2082181
```

The output with all variables(# of Reviews, Year, Price, Genre) was 0.2082. Next I will remove Genre to see if it impacts the predictions.

```
## a look a LM with out caret and ingoring genre
fit_lm_genre <- lm(`User Rating` ~ Reviews + Year +Price , data = train_set)
fit_lm_genre$coeff

##   (Intercept)       Reviews          Year         Price 
## -2.452256e+01 -1.785370e-06  1.449398e-02 -2.250584e-03

y_hat_genre <- predict(fit_lm_genre, test_set)


rmse_lm_wo_genre <- RMSE(y_hat_genre, test_set$`User Rating`)
rmse_lm_wo_genre

## [1] 0.208937
```

Next I will remove Price and Genre to see what impacts those variables had.

```
### A look at Reviews and Year only on a LM model
fit_lm_genre_price <- lm(`User Rating` ~ Reviews + Year  , data = train_set)
fit_lm_genre_price$coeff

##   (Intercept)       Reviews          Year 
## -2.708066e+01 -1.598561e-06  1.574802e-02

y_hat_genre_price <- predict(fit_lm_genre_price, test_set)


rmse_lm_wo_genre_price <- RMSE(y_hat_genre_price, test_set$`User Rating`)
rmse_lm_wo_genre_price

## [1] 0.2093991



Last I look at just User Rating compared with the number of Reviews.


## LM model of Reviews only
fit_lm_genre_price_year <- lm(`User Rating` ~ Reviews   , data = train_set)
fit_lm_genre_price_year$coeff

##   (Intercept)       Reviews 
##  4.625023e+00 -6.238977e-07

y_hat_genre_price_year <- predict(fit_lm_genre_price_year, test_set)


rmse_lm_wo_genre_price_year <- RMSE(y_hat_genre_price_year, test_set$`User Ra
ting`)
rmse_lm_wo_genre_price_year

## [1] 0.2218029
```

```
To see all the results.  I put them in the table below.

## results of LM models
results_lm_wo_caret <- data.frame(Method =c( "Linear Regression with Reviews,
Year, Price, Genre", "Linear Regression with Reviews, Year, Price", "Linear R
egression with Reviews, Year","Linear Regression with Reviews Only"), RMSE =c
(rmse_lm_wo, rmse_lm_wo_genre,rmse_lm_wo_genre_price, rmse_lm_wo_genre_price_
year))
results_lm_wo_caret

##                                                Method      RMSE
## 1 Linear Regression with Reviews, Year, Price, Genre 0.2082181
## 2        Linear Regression with Reviews, Year, Price 0.2089370
## 3               Linear Regression with Reviews, Year 0.2093991
## 4                Linear Regression with Reviews Only 0.2218029
```

The more variables you use in the model the less your RMSE is but the range is .2218 to .2082. I was not able to use the Authors variable in the LM model as it created an error. Next I will compare these model runs with similiar set up but using caret. I will explore other models beyond Linear Regression later.

```
### Using Caret to build the LM models
### Genre, year, price, reviews
set.seed(1, sample.kind = "Rounding")

## Warning in set.seed(1, sample.kind = "Rounding"): non-uniform 'Rounding' s
ampler
## used

train_lm <- train(`User Rating` ~ Reviews + Year+ Price + Genre , method = "l
m", data = train_set)
y_hat_lm <- predict(train_lm, test_set, type ="raw")
rmse_lm <- RMSE(y_hat_lm, test_set$`User Rating`)
rmse_lm

## [1] 0.2082181

## Using Caret, LM Model Year, Price, Reviews
set.seed(1, sample.kind = "Rounding")

## Warning in set.seed(1, sample.kind = "Rounding"): non-uniform 'Rounding' s
ampler
## used

train_lm_genre <- train(`User Rating` ~ Reviews + Year+ Price  , method = "lm
", data = train_set)
y_hat_lm_genre <- predict(train_lm_genre, test_set, type ="raw")
rmse_lm_genre <- RMSE(y_hat_lm_genre, test_set$`User Rating`)
rmse_lm_genre
```

```
## [1] 0.208937
```

```r
##Using Caret, LM model  of Reviews, year
set.seed(1, sample.kind = "Rounding")
```

```
## Warning in set.seed(1, sample.kind = "Rounding"): non-uniform 'Rounding' s
ampler
## used
```

```r
train_lm_genre_price <- train(`User Rating` ~ Reviews + Year  , method = "lm"
, data = train_set)
y_hat_lm_genre_price <- predict(train_lm_genre_price, test_set, type ="raw")
rmse_lm_genre_price <- RMSE(y_hat_lm_genre_price, test_set$`User Rating`)
rmse_lm_genre_price
```

```
## [1] 0.2093991
```

```r
## using caret, LM model of Reviews only
train_lm_genre_price_year <- train(`User Rating` ~ Reviews   , method = "lm",
data = train_set)
y_hat_lm_genre_price_year <- predict(train_lm_genre_price_year, test_set, typ
e ="raw")
rmse_lm_genre_price_year <- RMSE(y_hat_lm_genre_price_year, test_set$`User Ra
ting`)
rmse_lm_genre_price_year
```

```
## [1] 0.2218029
```

```r
## results of Caret LM models
results_lm_caret <- data.frame(Method=c("LM w/ Caret and all variables","LM w
/ Caret and Reviews,Year, Price","LM w/ Caret and Reviews, Year","LM w/ Caret
and Reviews"), RMSE = c(rmse_lm,rmse_lm_genre,rmse_lm_genre_price,rmse_lm_gen
re_price_year))
results_lm_caret
```

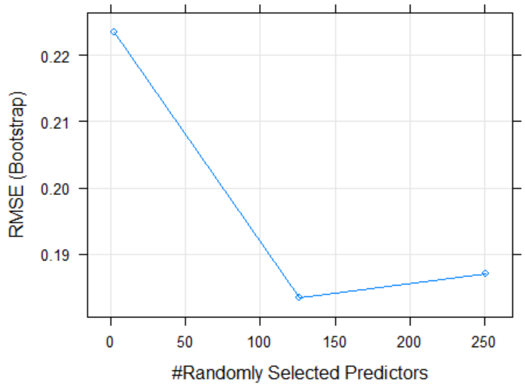
```
##                                 Method      RMSE
## 1       LM w/ Caret and all variables 0.2082181
## 2 LM w/ Caret and Reviews,Year, Price 0.2089370
## 3       LM w/ Caret and Reviews, Year 0.2093991
## 4             LM w/ Caret and Reviews 0.2218029
```

```r
## comparison of simple model and caret models
comparison_results <- data.frame(results_lm_caret, results_lm_wo_caret)
comparison_results
```

```
##                                 Method      RMSE
## 1       LM w/ Caret and all variables 0.2082181
## 2 LM w/ Caret and Reviews,Year, Price 0.2089370
## 3       LM w/ Caret and Reviews, Year 0.2093991
## 4             LM w/ Caret and Reviews 0.2218029
##                                              Method.1    RMSE.1
## 1 Linear Regression with Reviews, Year, Price, Genre 0.2082181
## 2        Linear Regression with Reviews, Year, Price 0.2089370
## 3               Linear Regression with Reviews, Year 0.2093991
## 4                Linear Regression with Reviews Only 0.2218029
```

As it should be the results are the same but there are other types of modeling. Can I get the RMSE to be lower than the Linear Regression. The remainder of the models will be using Caret only.

```
## using caret to explore other models.
### Randomforest
set.seed(1, sample.kind = "Rounding")

## Warning in set.seed(1, sample.kind = "Rounding"): non-uniform 'Rounding' s
ampler
## used

train_rf <- train(`User Rating`~ . , method = "rf", data = train_set, metric
="RMSE")
y_hat_rf <- predict(train_rf, test_set, type = "raw")
rmse_rf <- RMSE(y_hat_rf,test_set$`User Rating`)


rmse_rf

## [1] 0.1753545

plot(train_rf)
```



I added the metric of RMSE to the model and allowed it to pick the variables. Authors would have been included in that and the RMSE from the Random Forest was 0.1753.

Now to look at it using KNN. Additional tuning was added such as cross validation

```
## using caret to explore KNN
set.seed(1, sample.kind = "Rounding")
```

```
## Warning in set.seed(1, sample.kind = "Rounding"): non-uniform 'Rounding' s
ampler
## used
```

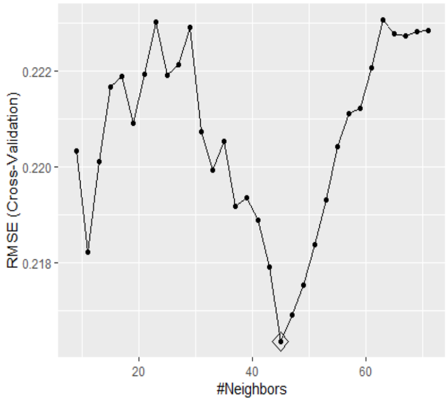
```
control <- trainControl(method = "cv", number = 10, p = .9)
train_knn <- train(`User Rating` ~ ., method= "knn", data = train_set, tuneGr
id = data.frame(k=seq(9,71,2)), trControl = control, metric ="RMSE")
train_knn$bestTune
```

```
##      k
## 19 45
```

```
y_hat_knn <- predict(train_knn, test_set, type ="raw")
rmse_knn <- RMSE(y_hat_knn, test_set$`User Rating`)
rmse_knn
```

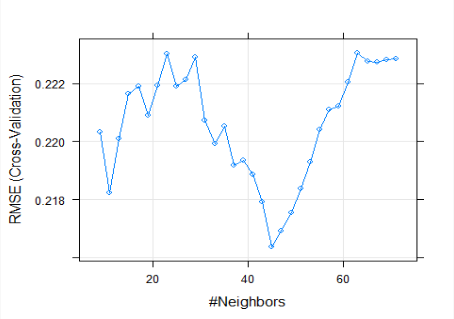
```
## [1] 0.2131459
```

```
ggplot(train_knn, highlight = TRUE)
```



```
train_knn$finalModel
```

```
## 45-nearest neighbor regression model
```



```
plot(train_knn)
```

```
## using caret to
train_penalized <- train(`User Rating` ~ ., method= "penalized", data= train_
set, metric= "RMSE")
```

```
y_hat_pen <- predict(train_penalized, test_set, type ="raw")
rmse_pen <- RMSE(y_hat_pen, test_set$`User Rating`)
rmse_pen

## [1] 0.2009099

set.seed(1, sample.kind ="Rounding")

## Warning in set.seed(1, sample.kind = "Rounding"): non-uniform 'Rounding' s
ampler
## used

train_glm <- train(`User Rating`~., method = "bayesglm", data =train_set, met
ric ="RMSE")
y_hat_glm <- predict(train_glm, test_set, type ="raw")
rmse_glm <- RMSE(y_hat_glm, test_set$`User Rating`)
rmse_glm

## [1] 0.1703655
```

Below are the results of all the Caret Models that were run. The models include Linear Regression, Bayes GLM, KNN, RandomForest, and Penalized. RMSE was the chosen metric for all models.

```
results <- data.frame(Method =c( "Linear Regresion with Caret"," Bayes GLM","
KNN","RandomForest", "Penalized"), RMSE =c(rmse_lm, rmse_glm,rmse_knn,rmse_rf
, rmse_pen))
results

##                        Method      RMSE
## 1 Linear Regresion with Caret 0.2082181
## 2                   Bayes GLM 0.1703655
## 3                         KNN 0.2131459
## 4                RandomForest 0.1753545
## 5                   Penalized 0.2009099
```

As you can see from the results above. The lowest RMSE produced was from the Bayes GLM. Each of these models all the model to pick the appropriate variables, except Linear Regression. This project allowed me to become familiar with the various parameters of CARET. Metrics, Tuning, etc… I also tried to learn the preProcess parameter but with less success.

Its easy to see that some variables are important to the overall sucess. A well known author migh influence the number of sales which could lead to higher number of reviews. Price also influences the number of sales. The year and genre have some influence but it appears to a lessor extent. These models could be used to help Amazon determine how sucessful a book might be.

Future work would be to learn how to incorporate the confusion matrix into this as well as work with Classification models.