RAMEN_CYO_Project_Report

Mangalam Khare

25th July 2021

Introduction

This project is part of Choose your own (CYO) project of the HarvardX course Capstone project. The objective of this project is to develop Machine learning algorithm using the Ramen Rating data set.This data set is downloaded from Kaggle.Several machine learning algorithm has been used and results have been compared to get the smallest RMSE possible as a measure of evaluating model performance

RMSE, Root Mean Square Error is the measure of the differences between predicted values and actual/observed values.

This Report has a problem statement section, data set preparation, Data pre-processing and exploratory analysis, Modelling and analysis of various models, results and conclusion

Problem Statement

The objective of this project is to use machine learning algorithms that predicts Ramen Ratings (Stars) using the inputs/ features present in the Ramen Ratings dataset. This dataset is split into Train (df_ramen_trian) and test(df_ramen_test) data. The algorithms are trained with train set and validated with test set As mentioned in the Introduction section the aim is to get the smallest RMSE possible

data can be downloaded from kaggle https://www.kaggle.com/residentmario/ramen-ratings)

OR GitHub

https://raw.githubusercontent.com/mangalamkhare/HarvardX_Data_Science/main/ramen-ratings.csv

Dataset Preparation

```
############################################################
# Create Ramen data set
############################################################

if(!require(tidyverse)) install.packages("tidyverse")
if(!require(caret)) install.packages("caret")
if(!require(data.table)) install.packages("data.table")

if(!require(FNN)) install.packages("FNN")
if(!require(mltools)) install.packages("mltools")
```

```r
if(!require(plyr)) install.packages("plyr")
if(!require(ggpubr)) install.packages("ggpubr")


library(tidyverse)
library(caret)
library(data.table)
library(FNN)
library(ggplot2)
library(lubridate)
library(dslabs)
library(plyr)
library(ggpubr)
library(class)
library(rpart)
library(randomForest)

# Ramen Ratings dataset:

link<-'https://raw.githubusercontent.com/mangalamkhare/HarvardX_Data_Science/main/ramen-ratings.csv'

df <- read.csv(file =link)

head(df)
```

```
##   Review..          Brand
## 1     2580      New Touch
## 2     2579       Just Way
## 3     2578         Nissin
## 4     2577        Wei Lih
## 5     2576 Ching’s Secret
## 6     2575  Samyang Foods
##                                                        Variety Style     Country
## 1                                   T’s Restaurant Tantanmen    Cup       Japan
## 2 Noodles Spicy Hot Sesame Spicy Hot Sesame Guan-miao Noodles  Pack      Taiwan
## 3                               Cup Noodles Chicken Vegetable   Cup         USA
## 4                               GGE Ramen Snack Tomato Flavor  Pack      Taiwan
## 5                                             Singapore Curry  Pack       India
## 6                                      Kimchi song Song Ramen  Pack South Korea
##   Stars Top.Ten
## 1  3.75
## 2     1
## 3  2.25
## 4  2.75
## 5  3.75
## 6  4.75
```

```r
df_ramen <- as.data.frame(df) %>% mutate(
                             reviewId = 'Review #' ,
                             topten = as.character("Top Ten"),
                             brand = as.character(Brand),
                             variety = as.character(Variety),
                             style = as.character(Style),
                             country = as.character(Country),
                             stars= as.numeric(Stars))
head(df_ramen)
```

```
##   Review..          Brand
## 1     2580      New Touch
## 2     2579       Just Way
## 3     2578         Nissin
## 4     2577        Wei Lih
## 5     2576 Ching’s Secret
## 6     2575  Samyang Foods
##                                                        Variety Style     Country
## 1                                   T’s Restaurant Tantanmen    Cup       Japan
## 2 Noodles Spicy Hot Sesame Spicy Hot Sesame Guan-miao Noodles  Pack      Taiwan
## 3                               Cup Noodles Chicken Vegetable   Cup         USA
## 4                               GGE Ramen Snack Tomato Flavor  Pack      Taiwan
## 5                                             Singapore Curry  Pack       India
## 6                                      Kimchi song Song Ramen  Pack South Korea
##   Stars Top.Ten reviewId  topten          brand
## 1  3.75         Review # Top Ten      New Touch
## 2     1         Review # Top Ten       Just Way
## 3  2.25         Review # Top Ten         Nissin
## 4  2.75         Review # Top Ten        Wei Lih
## 5  3.75         Review # Top Ten Ching’s Secret
## 6  4.75         Review # Top Ten  Samyang Foods
##                                                        variety style     country
## 1                                   T’s Restaurant Tantanmen    Cup       Japan
## 2 Noodles Spicy Hot Sesame Spicy Hot Sesame Guan-miao Noodles  Pack      Taiwan
## 3                               Cup Noodles Chicken Vegetable   Cup         USA
## 4                               GGE Ramen Snack Tomato Flavor  Pack      Taiwan
## 5                                             Singapore Curry  Pack       India
## 6                                      Kimchi song Song Ramen  Pack South Korea
##   stars
## 1  3.75
## 2  1.00
## 3  2.25
## 4  2.75
## 5  3.75
## 6  4.75
```

Data pre-processing and exploratory analysis

Check few rows of the ramen data set to get familiar with the data It contains 7 columns reviewId, topten,brand, variety,style, country and stars which represents rating. Each row represents data for a single product review.

```
head(df_ramen)
```

```
##   Review..          Brand
## 1     2580      New Touch
## 2     2579       Just Way
## 3     2578         Nissin
## 4     2577        Wei Lih
## 5     2576 Ching’s Secret
## 6     2575  Samyang Foods
##                                                        Variety Style     Country
## 1                                     T’s Restaurant Tantanmen    Cup       Japan
## 2 Noodles Spicy Hot Sesame Spicy Hot Sesame Guan-miao Noodles  Pack      Taiwan
## 3                              Cup Noodles Chicken Vegetable   Cup         USA
## 4                              GGE Ramen Snack Tomato Flavor  Pack      Taiwan
## 5                                            Singapore Curry  Pack       India
## 6                                     Kimchi song Song Ramen  Pack South Korea
##   Stars Top.Ten reviewId  topten          brand
## 1  3.75         Review # Top Ten      New Touch
## 2     1         Review # Top Ten       Just Way
## 3  2.25         Review # Top Ten         Nissin
## 4  2.75         Review # Top Ten        Wei Lih
## 5  3.75         Review # Top Ten Ching’s Secret
## 6  4.75         Review # Top Ten  Samyang Foods
##                                                        variety style     country
## 1                                     T’s Restaurant Tantanmen    Cup       Japan
## 2 Noodles Spicy Hot Sesame Spicy Hot Sesame Guan-miao Noodles  Pack      Taiwan
## 3                              Cup Noodles Chicken Vegetable   Cup         USA
## 4                              GGE Ramen Snack Tomato Flavor  Pack      Taiwan
## 5                                            Singapore Curry  Pack       India
## 6                                     Kimchi song Song Ramen  Pack South Korea
##   stars
## 1  3.75
## 2  1.00
## 3  2.25
## 4  2.75
## 5  3.75
## 6  4.75
```

Check Dimensions and Summary stats

Check for the dimensions of the data set to get total no of rows and columns and Summary stats

```
# Rows Columns

dim(df_ramen)
```

```
## [1] 2580   14
```

```
# Data set Summary

summary(df_ramen)
```

```
##     Review..         Brand             Variety             Style
##  Min.   :   1.0   Length:2580        Length:2580        Length:2580
##  1st Qu.: 645.8   Class :character   Class :character   Class :character
##  Median :1290.5   Mode  :character   Mode  :character   Mode  :character
##  Mean   :1290.5
##  3rd Qu.:1935.2
##  Max.   :2580.0
##
##    Country             Stars             Top.Ten            reviewId
##  Length:2580        Length:2580        Length:2580        Length:2580
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##     topten             brand             variety             style
##  Length:2580        Length:2580        Length:2580        Length:2580
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##    country              stars
##  Length:2580        Min.   :0.000
##  Class :character   1st Qu.:3.250
##  Mode  :character   Median :3.750
##                     Mean   :3.655
##                     3rd Qu.:4.250
##                     Max.   :5.000
##                     NA’s   :3
```

```
# check for the number of unique brand, style, variety and country in the ramen dataset

# Unique number of Style and Country

df_ramen %>%
summarize(n_style = n_distinct(style),
n_country = n_distinct(country))
```

```
##   n_style n_country
## 1       8        38
```

```
# unique number of brand and variety

df_ramen %>%
summarize(n_brand = n_distinct(brand),
n_variety = n_distinct(variety))
```

```
##   n_brand n_variety
## 1     355      2413
```

Since reviewId represents unique review it will not be used for modelling, we will drop it.

```
# check topten column

n_topten <- unique(df_ramen$topten)

n_topten
```

```
## [1] "Top Ten"
```

Since top10 does not have any useful data we will drop this as well

```r
df_ramen <- df_ramen %>%  select(brand , variety, style,country, stars)

head(df_ramen)
```

```
##           brand                                                     variety
## 1      New Touch                                  T's Restaurant Tantanmen 
## 2       Just Way Noodles Spicy Hot Sesame Spicy Hot Sesame Guan-miao Noodles
## 3         Nissin                                Cup Noodles Chicken Vegetable
## 4        Wei Lih                                GGE Ramen Snack Tomato Flavor
## 5 Ching's Secret                                              Singapore Curry
## 6  Samyang Foods                                       Kimchi song Song Ramen
##   style     country stars
## 1   Cup       Japan  3.75
## 2  Pack      Taiwan  1.00
## 3   Cup         USA  2.25
## 4  Pack      Taiwan  2.75
## 5  Pack       India  3.75
## 6  Pack South Korea  4.75
```

```r
# Clean the data set

df_ramen [df_ramen == "Unrated"] <- "0"
df_ramen <- df_ramen  %>%na.omit()
```

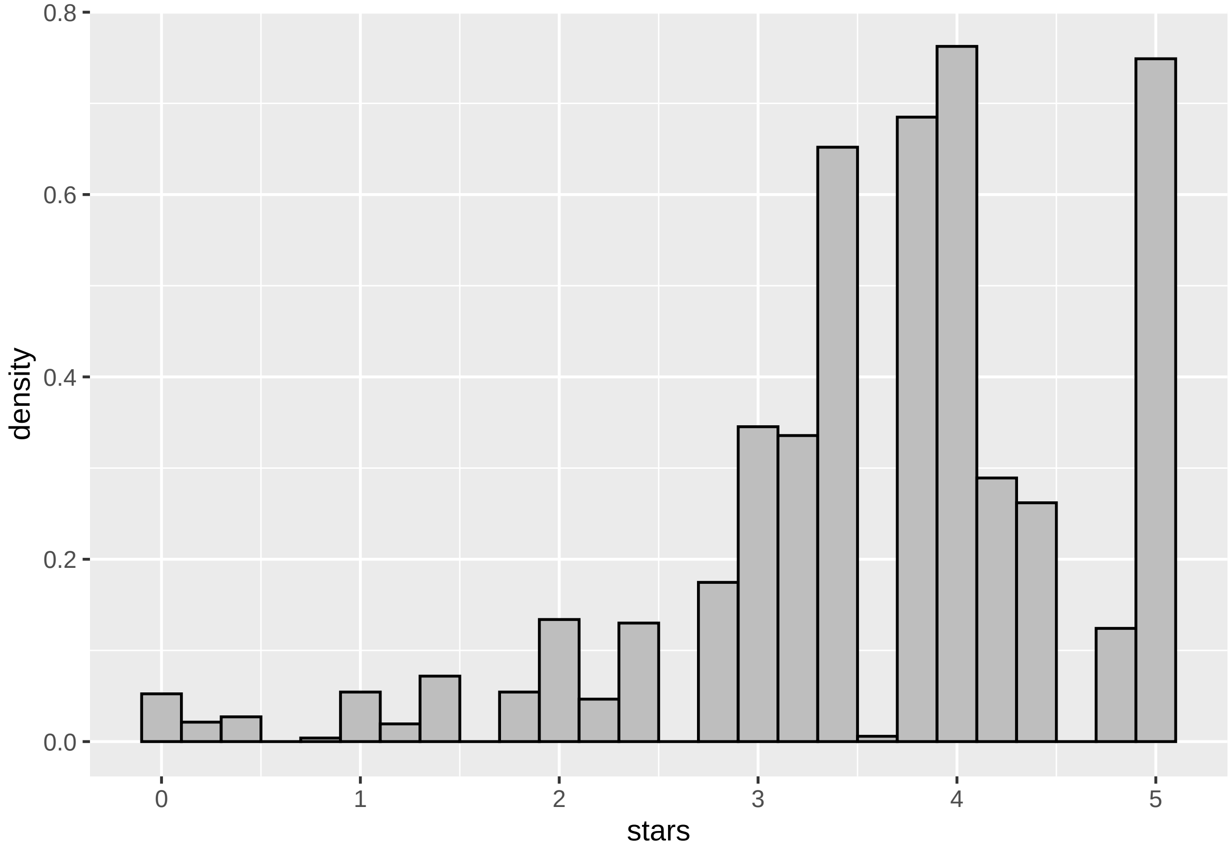
Define the function for RMSE

RMSE <- function(true__ratings, predicted__ratings){ sqrt(mean((true__ratings-predicted__ratings)^2)) }

Stars/Ratings distribution

Indicates that most of the ramen are rated between 3 and 5, we will also check the distributions of other features and decide on features to be included for prediction

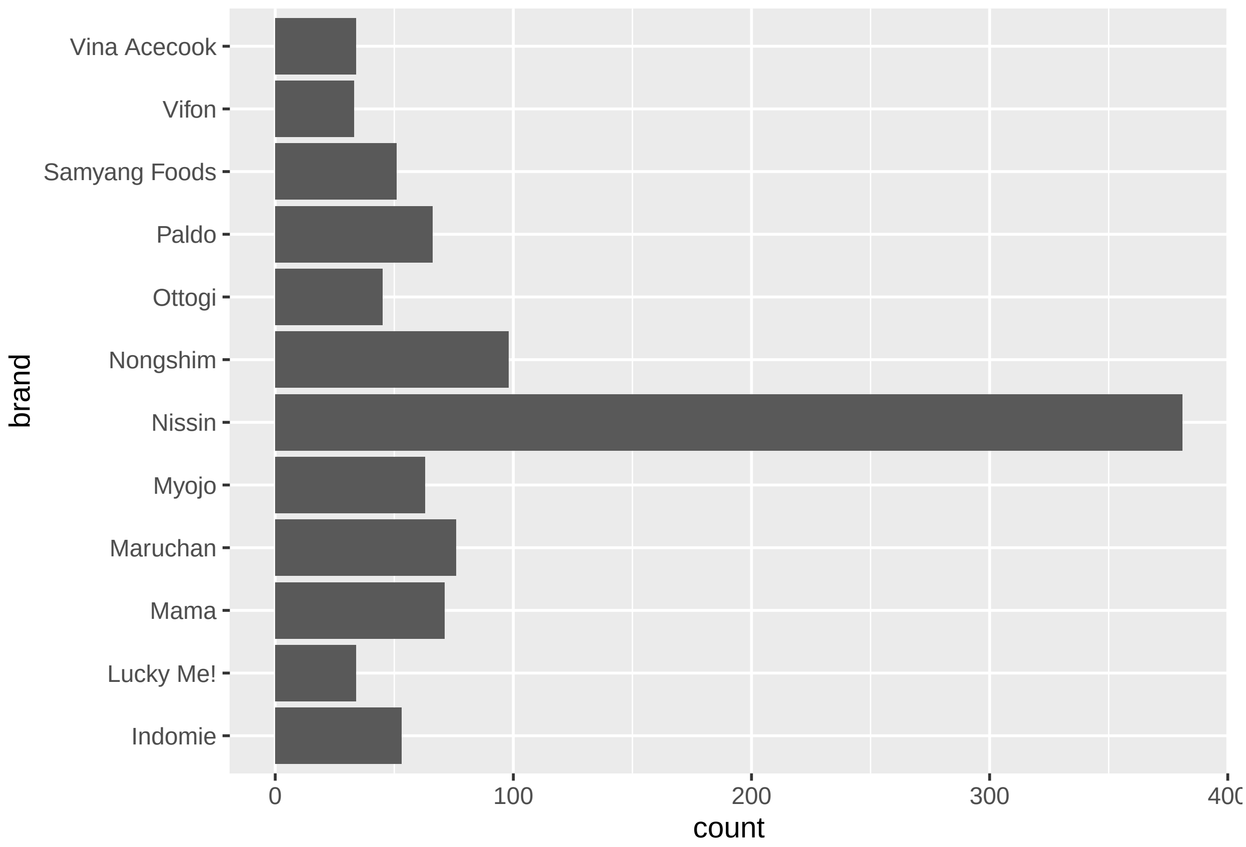
```r
# distribution of stars

ggplot(df_ramen, aes(x=stars)) +
geom_histogram(aes(y=..density..),
      binwidth=0.2,
      colour="black", fill="grey")
```



Distribution of Brands

There are 355 brands. If we try to show all the plot may become difficult to read hence we will just include brands where frequency is > 30 We can see that the brand Nissan is the top most with a very large difference with remaining brands

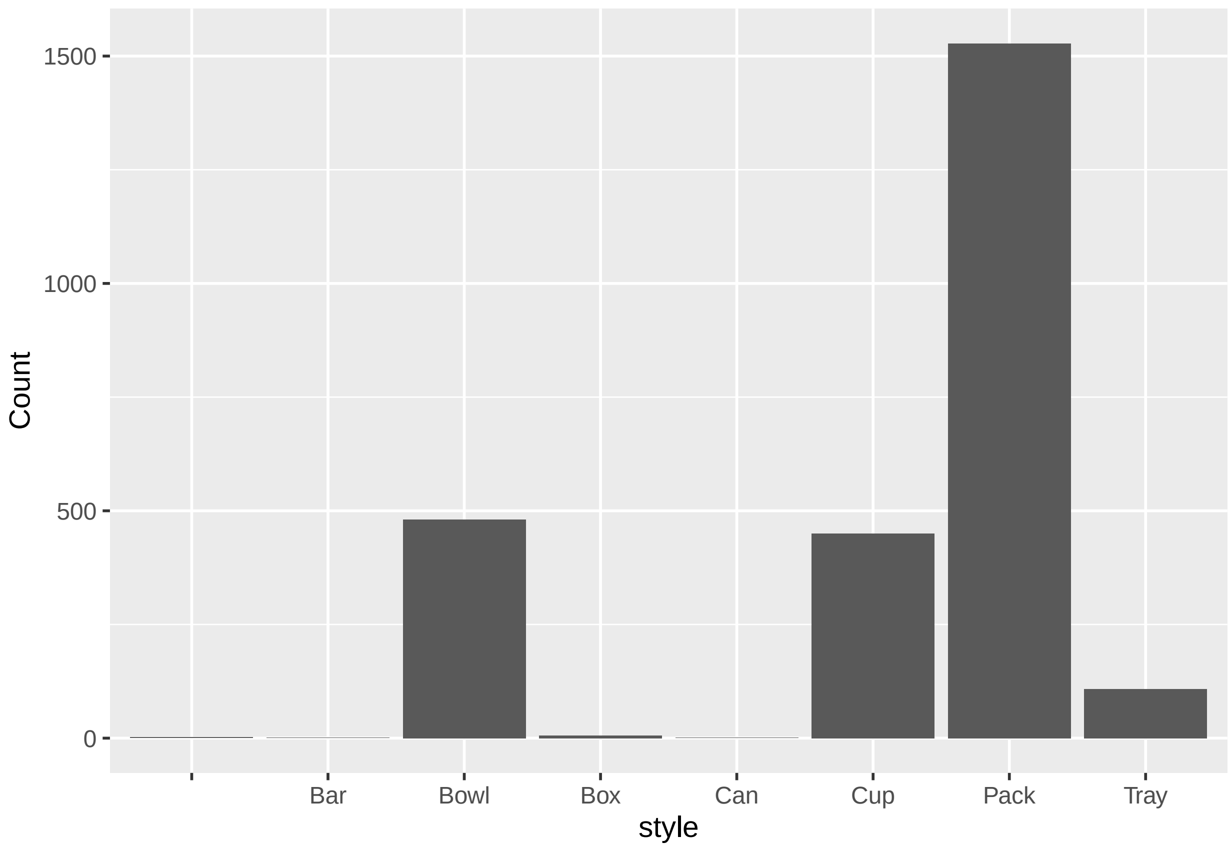
```
# Brands distribution
df_ramen %>% group_by(brand) %>% filter(n() > 30) %>%
ggplot(aes(x = brand) )+ geom_bar() + coord_flip()
```



Distribution of Style

We can see that some of the styles like Box, Bar,can have very very less number of reviews we will drop them from our data set

```
# stars distributions
df_ramen %>%
ggplot(aes(style)) +
geom_bar() +
ylab("Count")
```

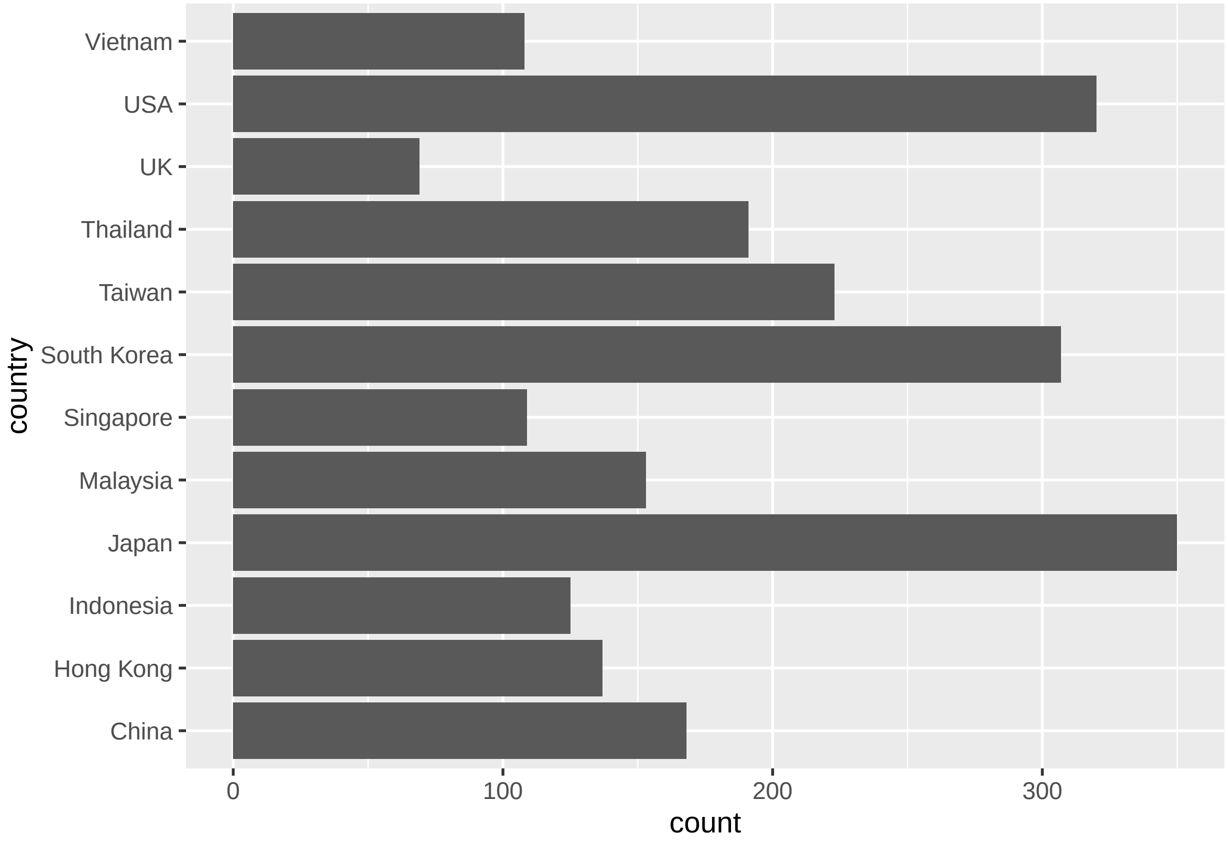


```
df_ramen<- df_ramen %>%
filter(style %in% c("Pack", "Bowl", "Cup", "Tray"))
```

Distribution of Country

As we have seen earlier there are 38 unique counytries the plot may become difficult to read hence we will just include countries where frequency is > 50. We will also club remaining countries with low frequency into Others

```
# Country Distribution

df_ramen %>% group_by(country) %>% filter(n() > 50) %>%
ggplot(aes(x = country)) + geom_bar() + coord_flip()
```



```
y <- count(df_ramen, 'country')
df_country <-  y[order(-y$freq),]
df_country
```

```
##          country freq
## 19         Japan  350
## 37           USA  320
## 31   South Korea  307
## 33        Taiwan  223
## 34      Thailand  191
## 6          China  168
## 20      Malaysia  153
## 15     Hong Kong  137
## 18     Indonesia  125
## 30     Singapore  109
## 38       Vietnam  108
## 35            UK   69
## 27   Philippines   47
## 5         Canada   41
## 17         India   31
## 12       Germany   27
## 21        Mexico   25
## 1      Australia   22
## 24   Netherlands   15
## 22       Myanmar   14
## 23         Nepal   14
## 16       Hungary    9
## 26      Pakistan    9
## 2     Bangladesh    7
## 7       Colombia    6
## 3         Brazil    5
## 4       Cambodia    5
## 10          Fiji    4
## 14       Holland    4
## 28        Poland    4
## 8          Dubai    3
## 11       Finland    3
## 29       Sarawak    3
## 32        Sweden    3
## 9        Estonia    2
## 13         Ghana    2
## 25       Nigeria    1
## 36 United States    1
```

```
df_country <- df_country %>%
filter(freq > 100)

df_country
```

```
##        country freq
## 1        Japan  350
## 2          USA  320
## 3  South Korea  307
## 4       Taiwan  223
## 5     Thailand  191
## 6        China  168
## 7     Malaysia  153
## 8    Hong Kong  137
## 9    Indonesia  125
## 10   Singapore  109
## 11     Vietnam  108
```

```
unique(df_ramen$country)
```

```
##  [1] "Japan"     "Taiwan"    "USA"       "Others"    "Singapore" "Thailand"
##  [7] "Hong Kong" "Vietnam"   "Malaysia"  "Indonesia" "China"
```

Data Preparation

We will Start with Simple RMSE using mean as base and then use Style and country as our features for further modelling and analysis Since style and country features have categorical data, we will create columns for binary variables(dummy data). As we are not using variey and brand features for our modelling we will remove them from our data set and will only keep features used for analysis

```
# First take backup of the entire data set
```

```r
ramen <- df_ramen

df_ramen <- df_ramen %>% select(style, country, stars)

# Create dummy variables

dummy <- dummyVars(" ~ .", data = df_ramen)

df_ramen_dummy <- data.frame(predict(dummy, newdata = df_ramen))

# split the data set into training and test sets
# train set- 80%, test set/validation set - 20%

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

test_index <- createDataPartition(y = df_ramen_dummy$stars, times = 1, p = 0.2, list = FALSE)

df_ramen_train<- df_ramen_dummy[-test_index,]
df_ramen_test <- df_ramen_dummy[test_index,]
```

Modelling and analysis

Base : Average Stars/Rating model

In this model we will Compute the mean stars/rating from the ramen train data set mean rating is used to predict the same rating for all types, regardless of any other feature. This simple model assumes that all the diferences in Stars are explained by the random variable alone.

```r
#Base : Average Ramen star/rating model

# Compute the mean rating from the ramen train data set

mu <- mean(df_ramen_train$stars)
mu
```

```
## [1] 3.648539
```

```r
# Test Results based on base prediction

base_rmse <- RMSE(df_ramen_test$stars, mu)
base_rmse
```

```
## [1] 1.004377
```

```r
# Check results save prediction in dataframe

rmse_results <- data_frame(Method = " Average Stars/Rating model",
                           RMSE = base_rmse)
rmse_results %>% knitr::kable()
```

Method	RMSE
Average Stars/Rating model	1.004377

This will serve as base RMSE. We will now apply Machine Learning algorithms to improve it further.lets start will Liner regression first to establish a base algorithm and then move up to Decision tree, random forest and KNN Regression

Linear Regression

```
# Liner Regression

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

train_lm <- lm(stars ~ ., data = df_ramen_train)

summary(train_lm)
```

```
##
## Call:
## lm(formula = stars ~ ., data = df_ramen_train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.8389 -0.4585  0.1454  0.6123  1.7788
##
## Coefficients: (2 not defined because of singularities)
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)        3.21616    0.15178  21.190  < 2e-16 ***
## styleBowl          0.03420    0.11811   0.290  0.77220
## styleCup          -0.12593    0.11988  -1.051  0.29361
## stylePack          0.04057    0.11244   0.361  0.71828
## styleTray               NA         NA      NA       NA
## countryChina       0.23010    0.13766   1.671  0.09478 .
## countryHong.Kong   0.60420    0.14207   4.253 2.21e-05 ***
## countryIndonesia   0.80616    0.14455   5.577 2.78e-08 ***
## countryJapan       0.74864    0.12253   6.110 1.19e-09 ***
## countryMalaysia    0.98541    0.13845   7.117 1.52e-12 ***
## countryOthers      0.24519    0.11486   2.135  0.03290 *
## countrySingapore   0.85307    0.14950   5.706 1.32e-08 ***
## countryTaiwan      0.39636    0.12960   3.058  0.00225 **
## countryThailand    0.13099    0.13276   0.987  0.32394
## countryUSA         0.24236    0.12375   1.958  0.05031 .
## countryVietnam          NA         NA      NA       NA
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 0.9779 on 2039 degrees of freedom
## Multiple R-squared:  0.08132,    Adjusted R-squared:  0.07547
## F-statistic: 13.88 on 13 and 2039 DF,  p-value: < 2.2e-16
```

```
prediction <- predict(train_lm,df_ramen_test)

#prediction

rmse <- RMSE(prediction, df_ramen_test$stars)

rmse
```

```
## [1] 0.9698033
```

```
rmse_results <- bind_rows(rmse_results,
                  data_frame(Method="Linear Regression model",
                             RMSE = rmse))
rmse_results %>% knitr::kable()
```

Method	RMSE
Average Stars/Rating model	1.0043770
Linear Regression model	0.9698033

As we can see RMSE value of 0.9805611 its improved from our base model

Decision Tree

```
# Decision Tree
set.seed(1, sample.kind = "Rounding")
train_rpart <- rpart(stars~., method = "anova", data = df_ramen_train)

train_rpart
```

```
## n= 2053
##
## node), split, n, deviance, yval
##       * denotes terminal node
##
##  1) root 2053 2122.61300 3.648539
##    2) countryMalaysia< 0.5 1929 2004.09600 3.611871
##      4) countryJapan< 0.5 1655 1726.61700 3.551133
##        8) countryIndonesia< 0.5 1553 1641.80700 3.519237
##         16) countrySingapore< 0.5 1465 1561.05000 3.486433 *
##         17) countrySingapore>=0.5 88   52.93679 4.065341 *
##        9) countryIndonesia>=0.5 102   59.17463 4.036765 *
##      5) countryJapan>=0.5 274  234.49430 3.978741 *
##    3) countryMalaysia>=0.5 124   75.57796 4.218952 *
```

```
prediction <- predict(train_rpart, df_ramen_test)

rmse <- RMSE(prediction, df_ramen_test$stars)
```

```
rmse
```

```
## [1] 0.9803636
```

```
rmse_results <- bind_rows(rmse_results,
                 data_frame(Method="Decision Tree Regression model",
                            RMSE = rmse))
rmse_results %>% knitr::kable()
```

Method	RMSE
Average Stars/Rating model	1.0043770
Linear Regression model	0.9698033
Decision Tree Regression model	0.9803636

Random Forest

```
# Random Forest

set.seed(1, sample.kind = "Rounding")
train_rf <- randomForest(stars~., data = df_ramen_train, mtry = 2,
#train_rf <- randomForest(stars~., data = df_ramen_train, mtry = seq(1:7),
             importance = TRUE )

train_rf
```

```
##
## Call:
##  randomForest(formula = stars ~ ., data = df_ramen_train, mtry = 2,      importance = TRUE)
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 2
##
##           Mean of squared residuals: 0.9625267
##                     % Var explained: 6.9
```

```
prediction <- predict(train_rf, data = df_ramen_test)

rmse <- RMSE(prediction, df_ramen_test$stars)

rmse
```

```
## [1] 1.022224
```

```
rmse_results <- bind_rows(rmse_results,
              data_frame(Method="Random Forest Regression model",
                         RMSE = rmse))
rmse_results %>% knitr::kable()
```

Method	RMSE
Average Stars/Rating model	1.0043770
Linear Regression model	0.9698033
Decision Tree Regression model	0.9803636
Random Forest Regression model	1.0222242

Knn Regression

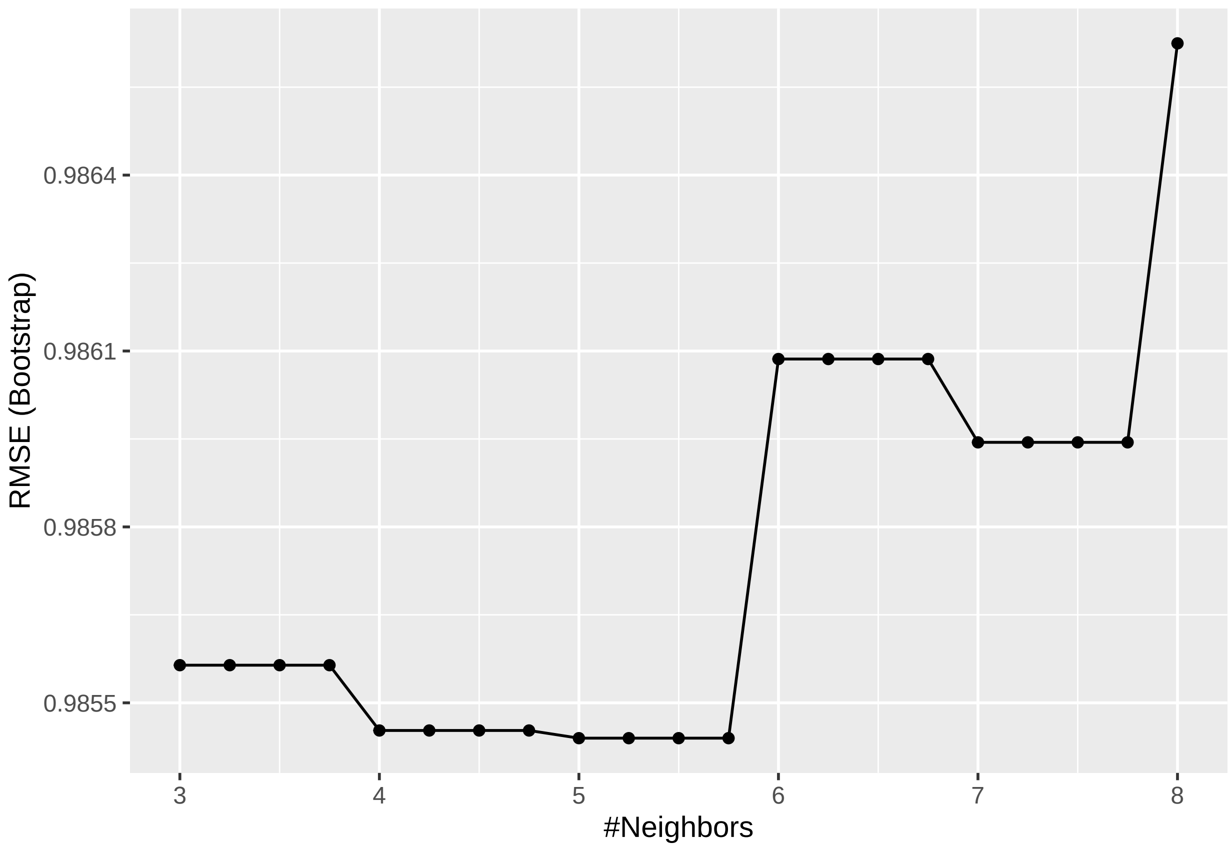
```
# Knn Regression

set.seed(1, sample.kind = "Rounding")
train_knn <- train(stars ~ .,  method = "knn",
       #tuneGrid = data.frame(k = seq(3, 5, 0.25)),
       tuneGrid = data.frame(k = seq(3, 8, 0.25)),
       data = df_ramen_train)


prediction <- predict(train_knn,df_ramen_test)

#prediction

rmse<-RMSE(prediction, df_ramen_test$stars)

rmse
```

```
## [1] 0.9651393
```

```
rmse_results <- bind_rows(rmse_results,
              data_frame(Method="Knn Regression model",
                         RMSE = rmse))
rmse_results %>% knitr::kable()
```

Method	RMSE
Average Stars/Rating model	1.0043770
Linear Regression model	0.9698033
Decision Tree Regression model	0.9803636
Random Forest Regression model	1.0222242
Knn Regression model	0.9651393

```
ggplot(train_knn)
```



```
train_knn$bestTune
```

```
##       k
## 12 5.75
```

In Regressions models KNN performed the best. Since we are not able to Improve RMSE's further lets try to convert regression into classification, predicting if Ramen noodles are good or not we will consider stars>3.75 as 1 (Good) and < as 0 (Not Good)

Data Preparation for Classification models

```
df_ramen_train_cl <- df_ramen_train

df_ramen_test_cl <- df_ramen_test

head(df_ramen_train_cl)
```

```
##   styleBowl styleCup stylePack styleTray countryChina countryHong.Kong
## 1         0        1         0         0            0                0
## 2         0        0         1         0            0                0
## 3         0        1         0         0            0                0
## 4         0        0         1         0            0                0
## 5         0        0         1         0            0                0
## 7         0        1         0         0            0                0
##   countryIndonesia countryJapan countryMalaysia countryOthers countrySingapore
## 1                0            1               0             0                0
## 2                0            0               0             0                0
## 3                0            0               0             0                0
## 4                0            0               0             0                0
## 5                0            0               0             1                0
## 7                0            1               0             0                0
##   countryTaiwan countryThailand countryUSA countryVietnam stars
## 1             0               0          0              0  3.75
## 2             1               0          0              0  1.00
## 3             0               0          1              0  2.25
## 4             1               0          0              0  2.75
## 5             0               0          0              0  3.75
## 7             0               0          0              0  4.00
```

```
head(df_ramen_test_cl)
```

```
##    styleBowl styleCup stylePack styleTray countryChina countryHong.Kong
## 6          0        0         1         0            0                0
## 11         0        0         1         0            0                0
## 14         1        0         0         0            0                0
## 22         0        0         1         0            0                0
## 31         0        0         1         0            0                0
## 51         0        0         1         0            0                0
##    countryIndonesia countryJapan countryMalaysia countryOthers countrySingapore
## 6                 0            0               0             1                0
## 11                0            0               0             0                0
## 14                0            1               0             0                0
## 22                0            0               0             0                0
## 31                0            0               0             1                0
## 51                0            0               0             0                1
##    countryTaiwan countryThailand countryUSA countryVietnam stars
## 6              0               0          0              0  4.75
## 11             0               1          0              0  5.00
## 14             0               0          0              0  4.50
## 22             0               0          1              0  5.00
## 31             0               0          0              0  5.00
## 51             0               0          0              0  5.00
```

```
df_ramen_train_cl <- mutate(df_ramen_train_cl , isGood = ifelse(df_ramen_train_cl$stars > 3.75, 1, 0))
df_ramen_test_cl <- mutate(df_ramen_test_cl , isGood = ifelse(df_ramen_test_cl$stars > 3.75, 1, 0))

# we will drop stars column

df_ramen_train_cl <- df_ramen_train_cl %>%  select(-stars)
df_ramen_test_cl <- df_ramen_test_cl %>%  select(-stars)

head(df_ramen_train_cl)
```

```
##   styleBowl styleCup stylePack styleTray countryChina countryHong.Kong
## 1         0        1         0         0            0                0
## 2         0        0         1         0            0                0
## 3         0        1         0         0            0                0
## 4         0        0         1         0            0                0
## 5         0        0         1         0            0                0
## 7         0        1         0         0            0                0
##   countryIndonesia countryJapan countryMalaysia countryOthers countrySingapore
## 1                0            1               0             0                0
## 2                0            0               0             0                0
## 3                0            0               0             0                0
## 4                0            0               0             0                0
## 5                0            0               0             1                0
## 7                0            1               0             0                0
##   countryTaiwan countryThailand countryUSA countryVietnam isGood
## 1             0               0          0              0      0
## 2             1               0          0              0      0
## 3             0               0          1              0      0
## 4             1               0          0              0      0
## 5             0               0          0              0      0
## 7             0               0          0              0      1
```

```
head(df_ramen_test_cl)
```

```
##    styleBowl styleCup stylePack styleTray countryChina countryHong.Kong
## 6          0        0         1         0            0                0
## 11         0        0         1         0            0                0
## 14         1        0         0         0            0                0
## 22         0        0         1         0            0                0
## 31         0        0         1         0            0                0
## 51         0        0         1         0            0                0
##    countryIndonesia countryJapan countryMalaysia countryOthers countrySingapore
## 6                 0            0               0             1                0
## 11                0            0               0             0                0
## 14                0            1               0             0                0
## 22                0            0               0             0                0
## 31                0            0               0             1                0
## 51                0            0               0             0                1
##    countryTaiwan countryThailand countryUSA countryVietnam isGood
## 6              0               0          0              0      1
## 11             0               1          0              0      1
## 14             0               0          0              0      1
## 22             0               0          1              0      1
## 31             0               0          0              0      1
## 51             0               0          0              0      1
```

Classification Models

LDA Model

```
# LDA
set.seed(1, sample.kind = "Rounding")
train_lm <- lm(isGood ~ ., data = df_ramen_train_cl)
```

```
summary(train_lm)
```

```
##
## Call:
## lm(formula = isGood ~ ., data = df_ramen_train_cl)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.6564 -0.3743 -0.2759  0.4492  0.8512
##
## Coefficients: (2 not defined because of singularities)
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)       0.17060    0.07397   2.306 0.021183 *
## styleBowl        -0.01395    0.05756  -0.242 0.808542
## styleCup         -0.10562    0.05842  -1.808 0.070759 .
## stylePack        -0.02177    0.05479  -0.397 0.691211
## styleTray              NA         NA      NA       NA
## countryChina      0.25534    0.06709   3.806 0.000145 ***
## countryHong.Kong  0.45215    0.06924   6.531 8.25e-11 ***
## countryIndonesia  0.50157    0.07045   7.120 1.49e-12 ***
## countryJapan      0.48583    0.05971   8.136 7.02e-16 ***
## countryMalaysia   0.49932    0.06747   7.400 1.98e-13 ***
## countryOthers     0.21092    0.05597   3.768 0.000169 ***
## countrySingapore  0.42874    0.07285   5.885 4.64e-09 ***
## countryTaiwan     0.30003    0.06316   4.751 2.17e-06 ***
## countryThailand   0.15341    0.06470   2.371 0.017826 *
## countryUSA        0.22549    0.06031   3.739 0.000190 ***
## countryVietnam         NA         NA      NA       NA
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 0.4766 on 2039 degrees of freedom
## Multiple R-squared:  0.08357,    Adjusted R-squared:  0.07773
## F-statistic:  14.3 on 13 and 2039 DF,  p-value: < 2.2e-16
```

```
prediction <- predict(train_lm,df_ramen_test_cl)

# prediction

rmse <- RMSE(prediction, df_ramen_test_cl$isGood)

rmse
```

```
## [1] 0.4798753
```

```
rmse_results <- bind_rows(rmse_results,
                 data_frame(Method="LDA Classification model",
                            RMSE = rmse))
rmse_results %>% knitr::kable()
```

Method	RMSE
Average Stars/Rating model	1.0043770
Linear Regression model	0.9698033
Decision Tree Regression model	0.9803636
Random Forest Regression model	1.0222242
Knn Regression model	0.9651393
LDA Classification model	0.4798753

As we can see the accuracy is improved to a greater extent. Now we will run different classification models to check if it improves further

Knn Classification Model

```
# Knn Classification
set.seed(1, sample.kind = "Rounding")
train_knn_cl <- train(isGood ~ .,
          method = "knn",
          data = df_ramen_train_cl,
          #tuneGrid = data.frame(k = seq(1, 7, 0.25)))
          #tuneGrid = data.frame(k = seq(3, 5, 0.25)))
          tuneGrid = data.frame(k = seq(3, 8, 0.25)))

prediction <- predict(train_knn_cl, df_ramen_test_cl)

rmse <- RMSE(prediction, df_ramen_test_cl$isGood)

rmse
```

```
## [1] 0.4773028
```

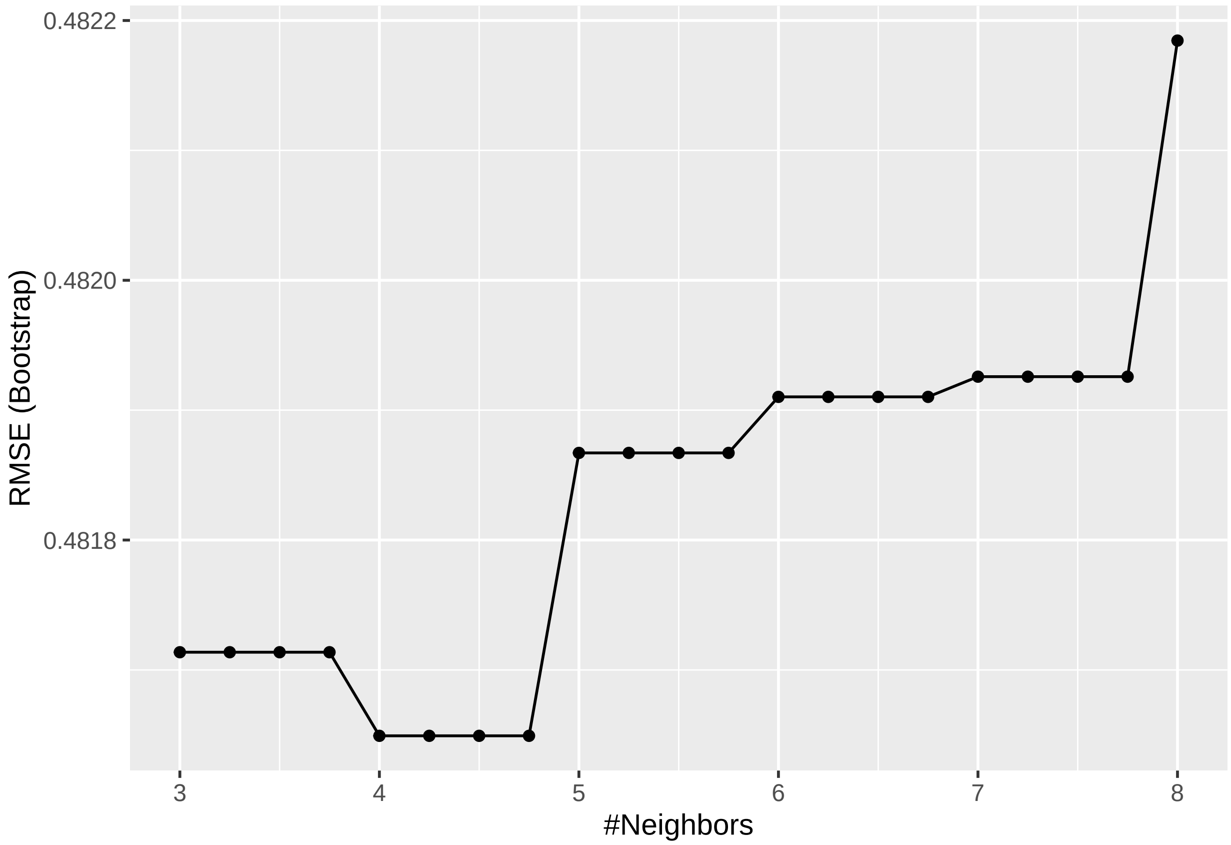
```
rmse_results <- bind_rows(rmse_results,
                 data_frame(Method="Knn Classification model",
                            RMSE = rmse))
rmse_results %>% knitr::kable()
```

Method	RMSE
Average Stars/Rating model	1.0043770
Linear Regression model	0.9698033
Decision Tree Regression model	0.9803636
Random Forest Regression model	1.0222242
Knn Regression model	0.9651393
LDA Classification model	0.4798753
Knn Classification model	0.4773028

```
train_knn_cl$bestTune
```

```
##      k
## 8 4.75
```

```
ggplot(train_knn_cl)
```



Cross Validation

```
# Cross Validation
set.seed(1, sample.kind = "Rounding")
train_knn_cv_cl <- train(isGood ~ .,
             method = "knn",
             data = df_ramen_train_cl,
             #tuneGrid = data.frame(k = seq(1, 7, 0.25)),
             tuneGrid = data.frame(k = seq(3, 8, 0.25)),
             trControl = trainControl(method = "cv", number = 10, p = 0.9))

prediction <- predict(train_knn_cv_cl, df_ramen_test_cl)

rmse <- RMSE(prediction, df_ramen_test_cl$isGood)

rmse
```

```
## [1] 0.4773028
```

```
rmse_results <- bind_rows(rmse_results,
                    data_frame(Method="Cross validation Classification model",
                               RMSE = rmse))
rmse_results %>% knitr::kable()
```

Method	RMSE
Average Stars/Rating model	1.0043770
Linear Regression model	0.9698033
Decision Tree Regression model	0.9803636
Random Forest Regression model	1.0222242
Knn Regression model	0.9651393
LDA Classification model	0.4798753
Knn Classification model	0.4773028
Cross validation Classification model	0.4773028

```
train_knn_cv_cl$bestTune
```

```
##       k
## 12 5.75
```

I Tried Knn and cross validation for multiple tunegrid but the Results remained same. Last we will try to see if Classification Tree Model Improves the results or not

Classification Tree

```
# Classification Tree
set.seed(1, sample.kind = "Rounding")
train_rpart_cl <- train(isGood ~ .,
            method = "rpart",
            tuneGrid = data.frame(cp = seq(0, 0.5, 0.02)),
            data = df_ramen_train_cl)

prediction <- predict(train_rpart_cl, df_ramen_test_cl)

rmse <- RMSE(prediction, df_ramen_test_cl$isGood)

rmse
```

```
## [1] 0.4770264
```

```
rmse_results <- bind_rows(rmse_results,
                    data_frame(Method="Classification tree model",
                               RMSE = rmse))
rmse_results %>% knitr::kable()
```

Method	RMSE
Average Stars/Rating model	1.0043770
Linear Regression model	0.9698033
Decision Tree Regression model	0.9803636
Random Forest Regression model	1.0222242
Knn Regression model	0.9651393
LDA Classification model	0.4798753
Knn Classification model	0.4773028
Cross validation Classification model	0.4773028
Classification tree model	0.4770264

```
train_rpart_cl$bestTune
```

```
##   cp
## 1  0
```

We have trained multiple classification models but could not improve the accuracy further The best accuracy was obtained with Knn Classification: RMSE -

```
rmse
```

```
## [1] 0.4770264
```

Results

The RMSE values of all the represented models are the following:

```
rmse_results %>% knitr::kable()
```

Method	RMSE
Average Stars/Rating model	1.0043770
Linear Regression model	0.9698033
Decision Tree Regression model	0.9803636
Random Forest Regression model	1.0222242
Knn Regression model	0.9651393
LDA Classification model	0.4798753
Knn Classification model	0.4773028
Cross validation Classification model	0.4773028
Classification tree model	0.4770264

Conclusion

Based on various models as explained in the Modeling section we have developed various machine learning algorithms regression and classification to predict ratings using Ramen dataset.

The Final RMSE is

```
rmse
```

```
## [1] 0.4770264
```

Future work

In this Analysis we ran Machine learning algorithms on Style and Country features. we could may further improve by using other 2 features (variety and brand). different combinations or all features. This is a small data set we can try to get bigger data set and do analysis