HarvardX Data Science Capstone Wine Ratings Report

Sarat Sarngadharan

1/5/2020

Executive Summary

As part of this project, I have trained a machine learning algorithm using the inputs in one subset to predict wine points ratings in the validation set. A simple guess (based on the average) on score given to a wine bottle taster would be 88.41 .This would have a Root Mean Square Error of 3.06. Similar to the MovieLens Capstone, we could improve the prediction accuracy by 20% by considering the below

- Geographical effect (Country of Origin /Province)
- Variety effect
- Vintage effect (Year of Make)
- Winery effect
- Taster effect

Introduction

Wine Dataset

For this project, I have used the Wine review dataset from kaggle. The data was scraped by user called **zackthoutt** from WineEnthusiast [Dataset source:] (https://www.kaggle.com/zynicide/wine-reviews#winemag-data-130k-v2.csv)

The dataset has **129,971** entries and 14 total columns. The major fields in the table are shown below

- **country** - Country of Origin of wine
- **description** - Review description from the taster
- **designation** - The vineyard within the winery where the grapes that made the wine are from
- **points** - The number of points Wine Enthusiast rated the wine on a scale of 1-100
- **price** - The cost for a bottle of the wine
- **province** - The province or state that the wine is from
- **taster_name** - Namer of the taster
- **taster_twitter_handle** - Twitter Handle of the reviere
- **title** - The title of the wine review, which often contains the vintage (year of make)
- **variety** - The type of grapes used to make the wine (ie Pinot Noir)
- **winery** - The winery that made the wine

Goal

This project aims to create a machine learning model that minimizes the **RMSE** of the predictions.

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{g,va,vi,w,t}\left(\hat{y}_{g,va,vi,w,t} - y_{g,va,vi,w,t}\right)^2}$$

$y_{g,va,vi,w,t}$ - true rating $\hat{y}_{g,va,vi,w,t}$ - predicted rating (g - geographical location (country / province), va from grape variety, vi - vintage w - winery t - taster) N -Number of Predictions

Initial Exploration

I have stored the Kaggle dataset under my Github capstone library to ensure that the code runs standalone Github Capstone:. First we load the required libraries and then file was downloaded

```
summary(ratings_wine)
```

```
##    country          description        designation            points
##  Length:129971      Length:129971      Length:129971      Min.   : 80.00
##  Class :character   Class :character   Class :character   1st Qu.: 86.00
##  Mode  :character   Mode  :character   Mode  :character   Median : 88.00
##                                                           Mean   : 88.45
##                                                           3rd Qu.: 91.00
##                                                           Max.   :100.00
##
##      price            province          region_1           region_2
##  Min.   :   4.00   Length:129971      Length:129971      Length:129971
##  1st Qu.:  17.00   Class :character   Class :character   Class :character
##  Median :  25.00   Mode  :character   Mode  :character   Mode  :character
##  Mean   :  35.36
##  3rd Qu.:  42.00
##  Max.   :3300.00
##  NA's   :8996
##  taster_name        taster_twitter_handle    title              variety
##  Length:129971      Length:129971         Length:129971      Length:129971
##  Class :character   Class :character      Class :character   Class :character
##  Mode  :character   Mode  :character      Mode  :character   Mode  :character
##
##
##
##
##     winery
##  Length:129971
##  Class :character
##  Mode  :character
##
##
##
##
```

There are 129971 rows in the dataset with 14 fields

Geographical Analysis

In this section we would evaluate the orgin of the wine (Country / Province)

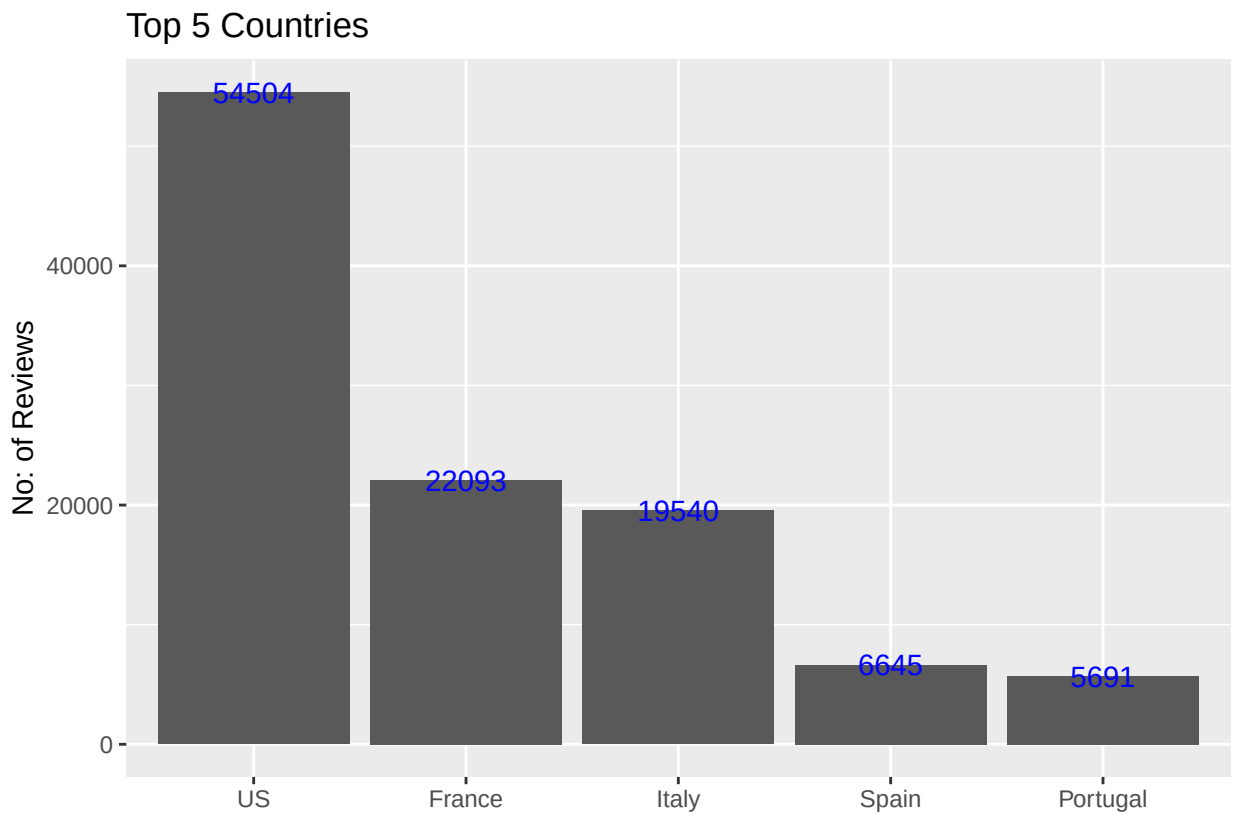
Country

```
unique(ratings_wine$country)
```

```
##  [1] "Italy"                "Portugal"             "US"
##  [4] "Spain"                "France"               "Germany"
##  [7] "Argentina"            "Chile"                "Australia"
## [10] "Austria"              "South Africa"         "New Zealand"
## [13] "Israel"               "Hungary"              "Greece"
## [16] "Romania"              "Mexico"               "Canada"
## [19] ""                     "Turkey"               "Czech Republic"
## [22] "Slovenia"             "Luxembourg"           "Croatia"
## [25] "Georgia"              "Uruguay"              "England"
## [28] "Lebanon"              "Serbia"               "Brazil"
## [31] "Moldova"              "Morocco"              "Peru"
## [34] "India"                "Bulgaria"             "Cyprus"
## [37] "Armenia"              "Switzerland"          "Bosnia and Herzegovina"
## [40] "Ukraine"              "Slovakia"             "Macedonia"
## [43] "China"                "Egypt"
```

We could observe that there wines from 44 different countries evaluated by various tasters. Bar plot showing the number of reviews by country (top 5) is generated below.

```
ratings_wine %>% group_by(country) %>%
  summarize(n=n()) %>% top_n(5,wt=n) %>%
  ggplot(aes(x=reorder(country,-n),y=n)) +
  geom_bar(stat="identity")+
  geom_text(aes(label=n),color='blue')+
  labs(title="Top 5 Countries", y="No: of Reviews", x="")
```



One could see most reviews were for wines from US followed by France and Italy

Province

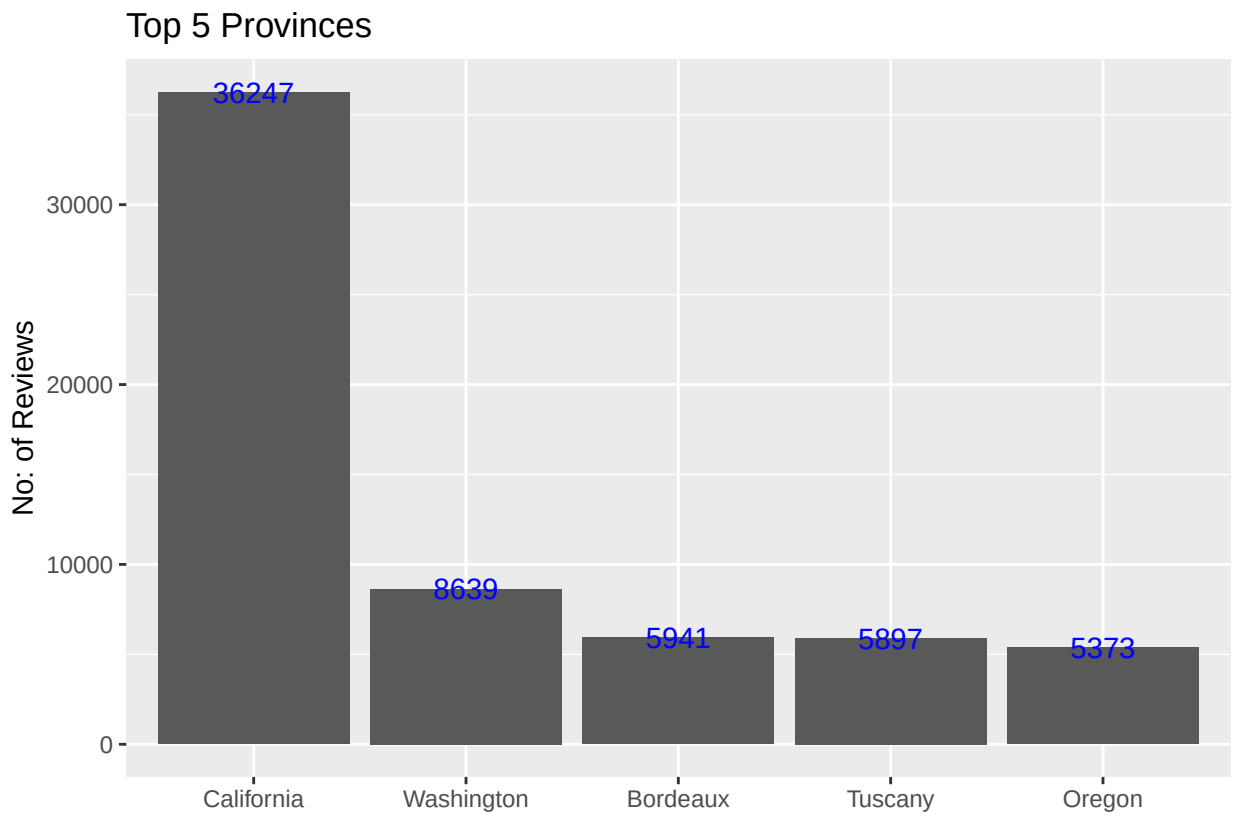
As Country alone may be the only indicator to identify origin of wine, we are doing a similar analyis at a province level

```
length(unique(ratings_wine$province))
```

```
## [1] 426
```

We could observe that there wines from 426 different provinces evaluated by various testers. Bar plot showing the number of reviews by provinces (top 5) is generated below.

```
ratings_wine %>% group_by(province) %>%
  summarize(n=n()) %>% top_n(5,wt=n) %>%
  ggplot(aes(x=reorder(province,-n),y=n)) +
  geom_bar(stat="identity")+
  geom_text(aes(label=n),color='blue')+
  labs(title="Top 5 Provinces", y="No: of Reviews", x="")
```



Consistent to the country analysis, we could see that dataset contains mainly reviews for wine from US states

Points Analysis

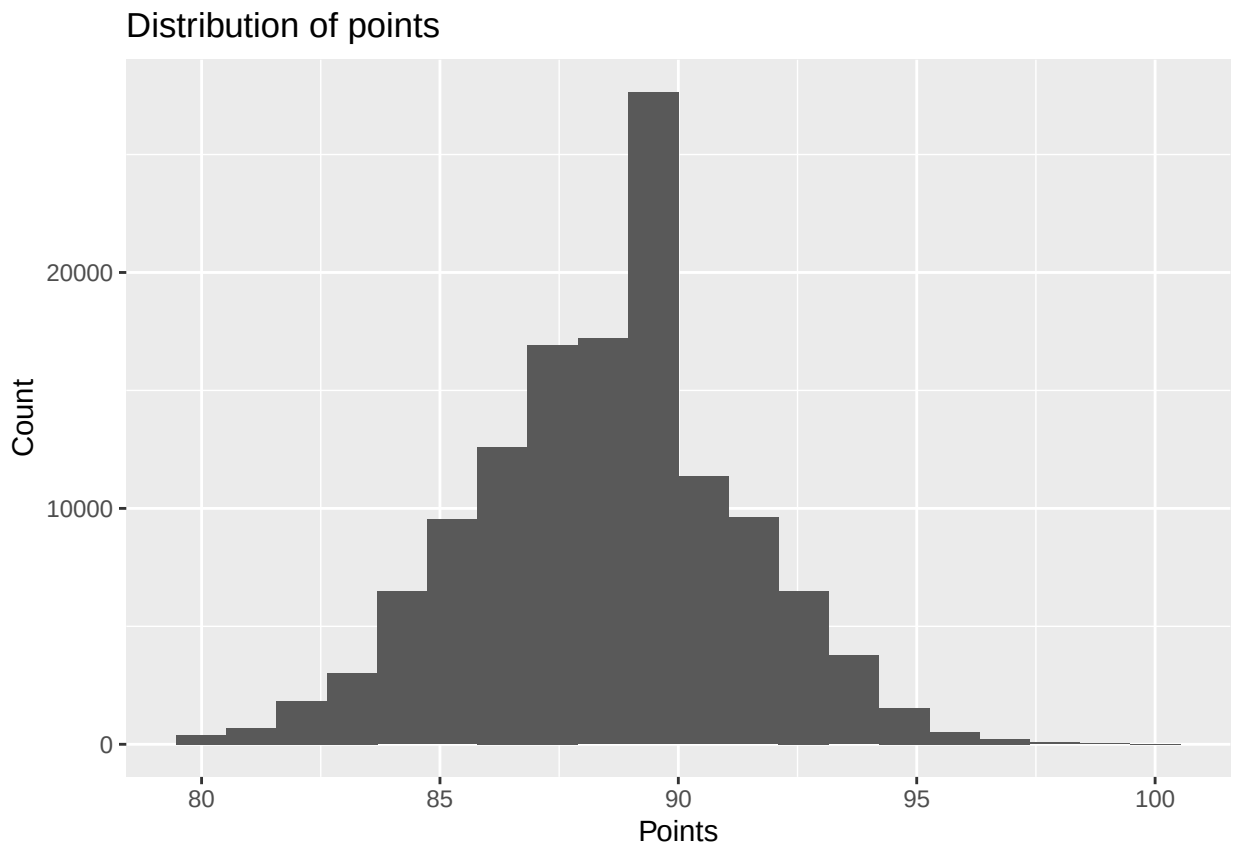
As we are looking to predict the points given for a wine, let us examine points data.

```
ratings_wine %>% select(points) %>%
 summarize(Min_Points=min(points),
           Max_Points=max(points),
           Average_Points=mean(points),
           SD =sd(points))
```

```
##   Min_Points Max_Points Average_Points      SD
## 1         80        100       88.44714 3.03973
```

We could observe that the dataset contains data with minimum rating of 80 and maximum rating of 100 with an average of 88.44714

```
ratings_wine %>% select(points) %>%
  ggplot(aes(points)) +
  geom_histogram(bins = 20)+
  labs(title="Distribution of points", x="Points", y="Count")
```



As observed earlier, we could see that the distribution has a minimum of 80 points and maximum of 100 points. The data is fairly normally distributed.

Price Analysis

In our summary analysis earlier we observed that there are more than 8000 entries without any price data. We would exclude those rows from our analysis.

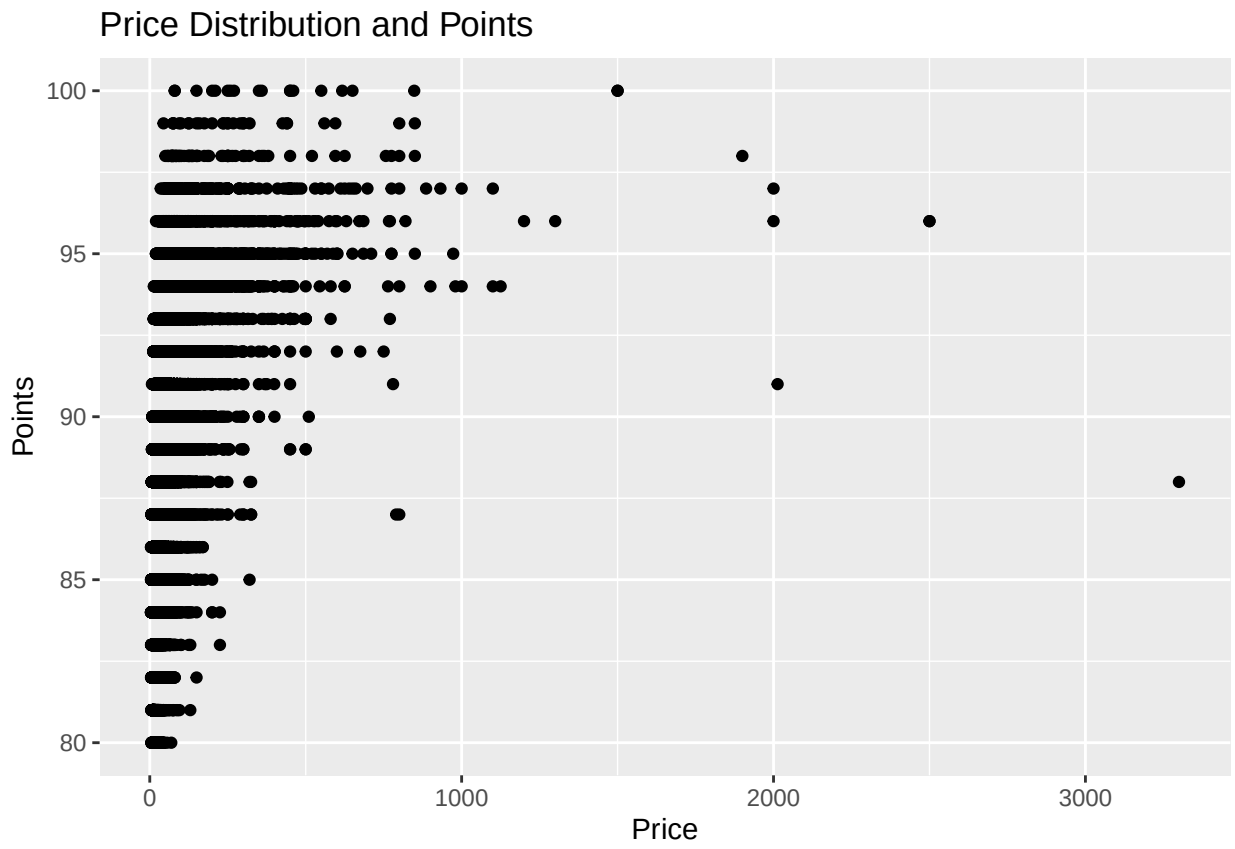
```
#Removing bottles for which price is not available

No_Price <- which(is.na(ratings_wine$price))
ratings_wine <- ratings_wine[-No_Price ,]
```

```
ratings_wine %>% filter(price != "") %>%
  summarize(Min_Price=min(price),
            Max_Price=max(price),
            Average_Price=mean(price),
            SD=round(sd(price),2))
```

```
##   Min_Price Max_Price Average_Price    SD
## 1         4      3300      35.36339 41.02
```

Price of wine varies from 4 to 3300 with an average Price of 35.3. Now let us review the a correlation between price and quality

```
ratings_wine %>% filter(price !="") %>%
  ggplot(aes(x=price,y=points))+
  geom_point()+
  labs(title="Price Distribution and Points", y="Points", x="Price")
```




```
ratings_winep <- ratings_wine %>% filter(price !="")
cor(ratings_winep$price, ratings_winep$points, method = c("pearson"))
```

```
## [1] 0.4161667
```

We could see that there is slight positive corelation between price and points awarded to the wine

Variety Analysis

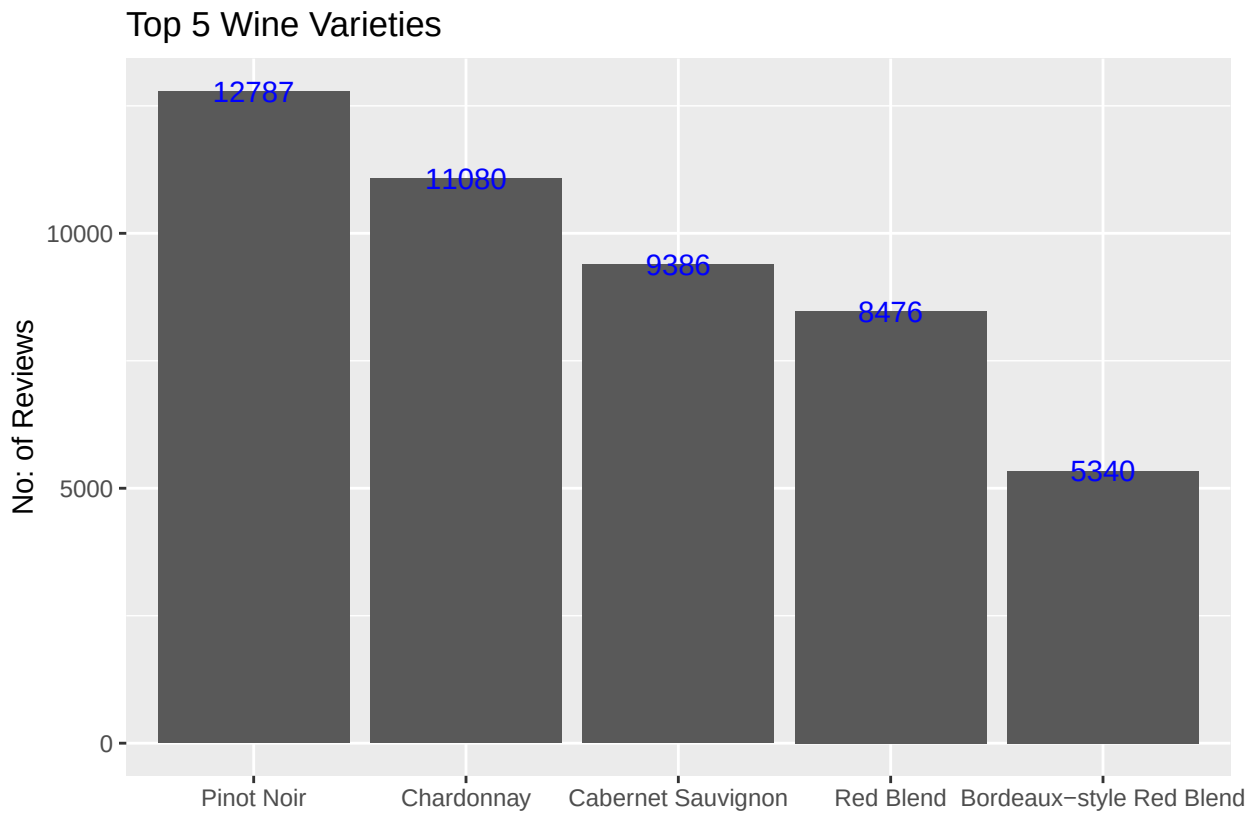
Let us examine the unique variety wines that were rated by tasters

```
length(unique(ratings_wine$variety))
```

```
## [1] 698
```

We could observe that there 708 varieties of wine.

```
ratings_wine %>% group_by(variety) %>%
  summarize(n=n()) %>% top_n(5,wt=n) %>%
  ggplot(aes(x=reorder(variety,-n),y=n)) +
  geom_bar(stat="identity")+
  geom_text(aes(label=n),color='blue')+
  labs(title="Top 5 Wine Varieties", y="No: of Reviews", x="")
```



Pinot Noir and Chardonnay were rated the most by tasters

Vintage Analysis

While evaluating the dataset you could see that the title contains the year of make in most cases. Let us analyze if vintage series would be rated higher than the newer ones.

```
#split the string  to remove initial numeric part from winery details for pattern extraction of year
library(stringr)
title_modified<-str_split_fixed(ratings_wine$title," ", 2)

#extract year of make
year_format <- "\\d\\d\\d\\d"
ratings_wine <- ratings_wine %>% mutate(Year_Make = as.numeric(str_extract(title_modified[,2],year_forma
```

```
ratings_wine %>% filter(Year_Make >"1900" & Year_Make < "2020") %>%
  ggplot(aes(x=Year_Make,y=points))+
  geom_point()+
  labs(title="Price Distribution and Wine Age", y="Points", x="Year of Make")
```

You could see most of the sample data is for wines that were made after 2000. Some Vintage wines are rated higher than the newer ones.

Winery Analysis

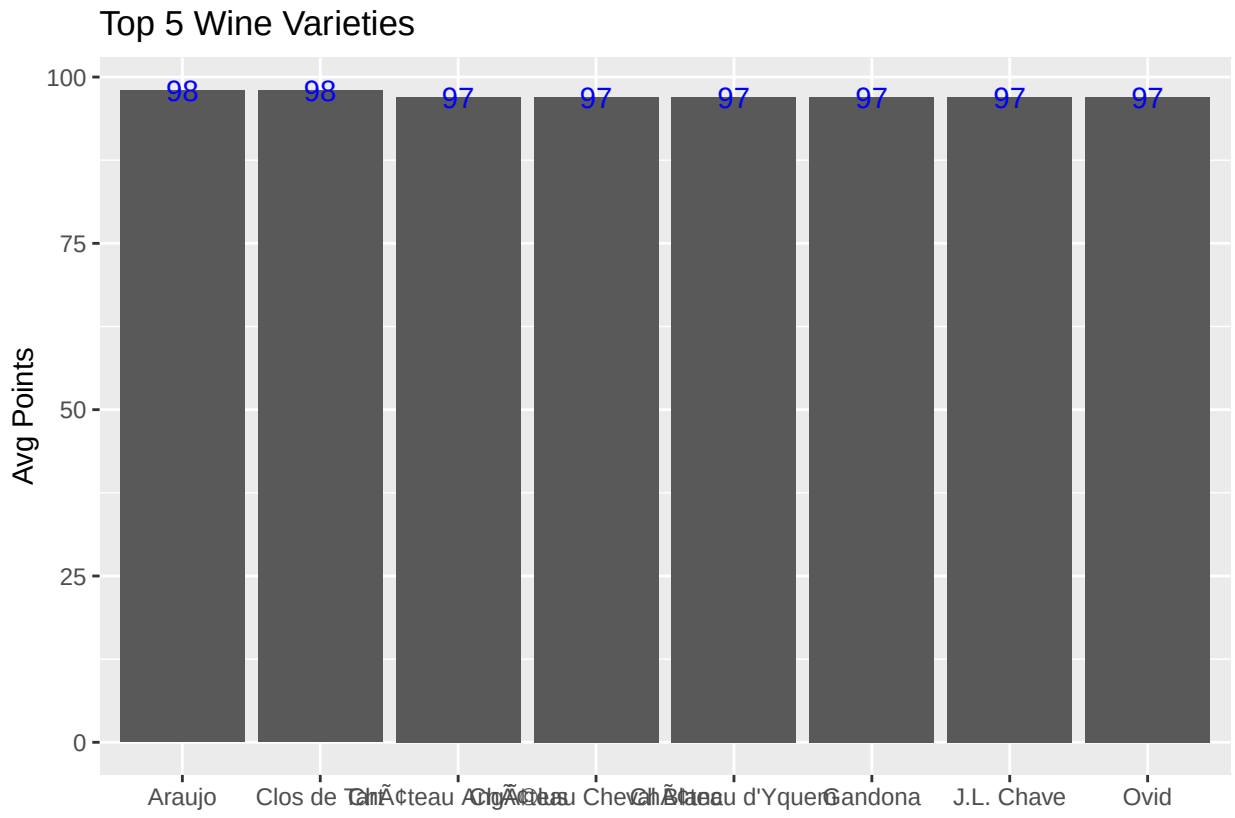
Let us analyze how many unique wineries were analyzed

```
length(unique(ratings_wine$winery))
```

```
## [1] 15855
```

We could see that there are 16757 wineries whose wines were rated by tasters. Let us verify the assumption whether a better brand winery would attract more points.

```
ratings_wine %>% group_by(winery) %>%
  summarize(n=mean(points))  %>% top_n(5,wt=n) %>%
  ggplot(aes(x=reorder(winery,-n),y=n)) +
  geom_bar(stat="identity")+
  geom_text(aes(label=n),color='blue')+
  labs(title="Top 5 Wine Varieties", y="Avg Points", x="")
```



We could see that wines from some wineries such a Araujo and Gandona are rated very highly

Taster Analysis

Taster rating behavior would impact the ratings given. Some testers be very critical and some may be very generous in their scoring

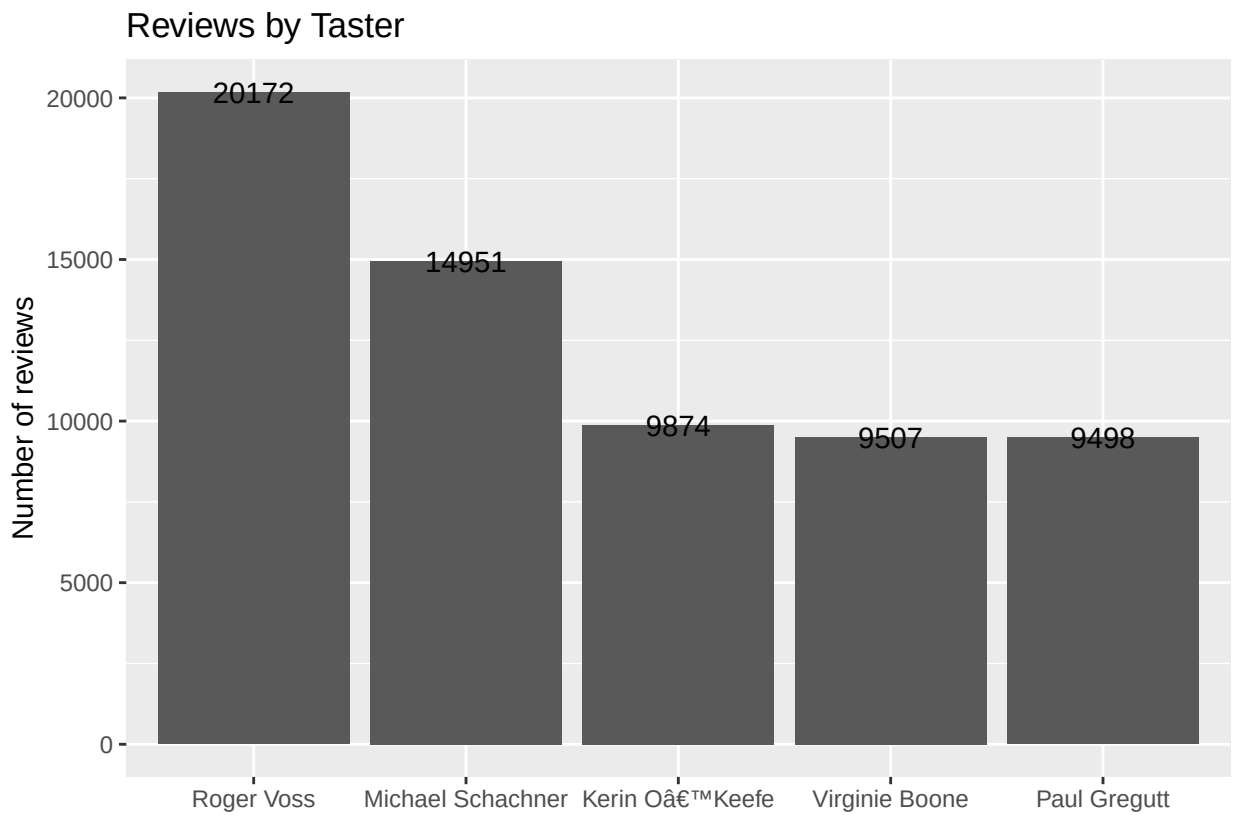
```
(unique(ratings_wine$taster_name))
```

```
##  [1] "Roger Voss"          "Paul Gregutt"        "Alexander Peartree"
##  [4] "Michael Schachner"   "Kerin Oâ\200\231Keefe"    "Anna Lee C. Iijima"
##  [7] "Virginie Boone"      "Matt Kettmann"       ""
## [10] "Sean P. Sullivan"    "Jim Gordon"          "Joe Czerwinski"
## [13] "Anne KrebiehlÂ MW"  "Lauren Buzzeo"       "Mike DeSimone"
## [16] "Jeff Jenssen"        "Susan Kostrzewa"     "Carrie Dykes"
## [19] "Fiona Adams"         "Christina Pickard"
```

We could see that there are 19 unique tasters as some rows do not have taster name populated

Similar to earlier analysis, let us review tasters who reviewed most. A

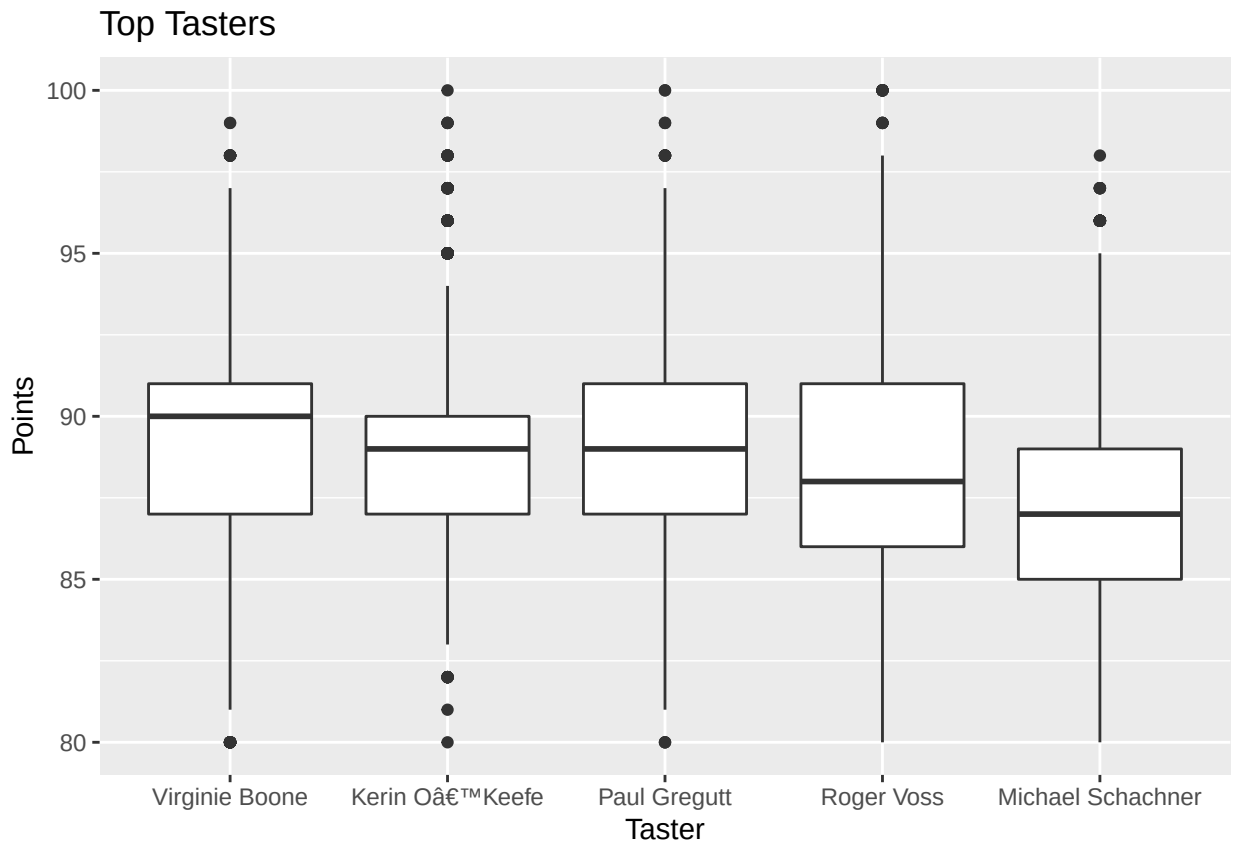
```
ratings_wine %>% filter(taster_name != "") %>%
  group_by(taster_name) %>%
  summarize(n=n()) %>%top_n(5,wt=n) %>%
  ggplot(aes(x=reorder(taster_name,-n),y=n)) +
  geom_bar(stat="identity")+
  geom_text(aes(label=n))+
  labs(title="Reviews by Taster", y="Number of reviews", x="")
```



Roger Voss reviewed most number of wines.

Let us look now at a distribution of score by taster for the top 5 tasters.

```
top_tasters <- ratings_wine %>%
  filter(taster_name!="") %>% group_by(taster_name)%>%
  summarize(n=n()) %>% top_n(5,wt=n) %>% .$taster_name

ratings_wine %>% filter(taster_name %in% top_tasters) %>%
  ggplot(aes(x=reorder(taster_name,-points,FUN = median), y=points)) +
  geom_boxplot()+
  labs(title="Top Tasters", y="Points", x="Taster")
```



Like we expected we could see that some tasters like Schachner are more critical than some other tasters.

NLP on Taster's descriptive feedback

NLP on description field Reference used: Basic Test mining in R (https://rstudio-pubs-static.s3.amazonaws.com/132792_864e3813b0ec47cb95c7e1e2e2ad83e7.html)

At first we use the tm package to transfrom the dataset to a corpus: Next we normalize the texts in the reviews using a series of pre-processing steps: * lower case conversion * Removing numbers * Removing punctuation marks, white spaces and stopwords

To analyze the textual data, we use a Document-Term Matrix (DTM) representation and remove the less frequent terms and generate a word cloud

```
library(tm)
library(SnowballC)
library(wordcloud)

wine_desc_corpus = Corpus(VectorSource(ratings_wine$description))

#preprocessing steps
#  a) change to lower case
#  b) remove punctuation points
#  c) remove stopwords
#  d) remove addition white spaces
#  e) remove numbers
```

```
wine_desc_corpus = tm_map(wine_desc_corpus, content_transformer(tolower))
wine_desc_corpus = tm_map(wine_desc_corpus, removePunctuation)
wine_desc_corpus = tm_map(wine_desc_corpus, removeWords, c("the", "and", stopwords("english")))
wine_desc_corpus = tm_map(wine_desc_corpus, stripWhitespace)
wine_desc_corpus = tm_map(wine_desc_corpus, removeNumbers)

#To analyze the textual data, use a Document-Term Matrix (DTM) representation

wine_desc_dtm <- DocumentTermMatrix(wine_desc_corpus)
#To reduce the dimension of the DTM, we can emove the less frequent terms
wine_desc_dtm = removeSparseTerms(wine_desc_dtm, 0.99)


#Generate word cloud
freq = data.frame(sort(colSums(as.matrix(wine_desc_dtm)), decreasing=TRUE))
wordcloud(rownames(freq), freq[,1], max.words=50, colors=brewer.pal(9, "Dark2"))
```

We could see that flavors, aroma, finish, fruit, ripe and acidity are frequently used.

Prediction Modelling

For Machine Learning purposes, we would split our ratings_wine dataset into a train_set and validation set.

train_set: This is the bulk of the data on which we will train our model **validation**: This will be our validation set (10% of data) where in which we will validate our model and report our final RMSE

```
#Create the training / testing dataset. Validation set would be 10% of the data
set.seed(1)
test_index <- createDataPartition(y = ratings_wine$points, times = 1, p = 0.1, list = FALSE)
train_set <- ratings_wine[-test_index,]
temp <- ratings_wine[test_index,]

validation <- temp %>%
  semi_join(train_set, by = "country") %>%
  semi_join(train_set, by = "province") %>%
  semi_join(train_set, by = "taster_name") %>%
  semi_join(train_set, by = "variety") %>%
  semi_join(train_set, by = "winery") %>%
  semi_join(train_set, by = "Year_Make")


# Add rows removed from validation set back into train set

removed <- anti_join(temp, validation)
train_set <- rbind(train_set, removed)

rm(removed, temp, test_index)
```

Define the RMSE Function that would be used to calculate RMSE for the various prediction models that we discuss in the below section

```
## RMSE Function
RMSE <- function(true_ratings, predicted_ratings){
  sqrt(mean((true_ratings - predicted_ratings)^2))
}
```

Naive Implementation

In this prediction model, we use the mean of the dataset to predict the points and any difference is attributed to a random error

```
#Naive Implementation with mean points
mean_points_rating<- mean(train_set$points)
#calculate rmse_naive in validation
rmse_naive <- RMSE(validation$points,mean_points_rating)
#store the RMSE data
rmse_data<- bind_rows(data_frame(Method = "Mean Points", RMSE = rmse_naive))
```

```
## Warning: `data_frame()` is deprecated, use `tibble()`.
## This warning is displayed once per session.
```

```
rmse_data
```

```
## # A tibble: 1 x 2
```

```
##   Method       RMSE
##   <chr>       <dbl>
## 1 Mean Points  3.04
```

We could observe that the RMSE is 3.04. We would build model considering the Geographical impact next and see who the RMSE improves

Geographical Effect

As we noticed in the earlier section, most of the reviews were from US and from a province standpoint from California. Let us build a model that consider the geographical effect on the ratings

First the country effect

```
#Geographical effects - Country
country_average <- train_set %>%
  group_by(country) %>%
  summarize(country_effect = mean(points - mean_points_rating))

predicted_ratings <- mean_points_rating + validation %>%
  left_join(country_average, by='country') %>%
  pull(country_effect)

rmse_country_effect <- RMSE(predicted_ratings, validation$points)
#store the RMSE data
rmse_data <- bind_rows(rmse_data,
                           data_frame(Method= "Country Effect",
                                      RMSE=rmse_country_effect))
rmse_data
```

```
## # A tibble: 2 x 2
##   Method          RMSE
##   <chr>          <dbl>
## 1 Mean Points     3.04
## 2 Country Effect  2.95
```

We can observe that the RMSE reduced to 2.95 when we considered the country effect. Now let us examine if considering the province would improve this RMSE further

```
##Geographical effect - Province

province_average <- train_set %>%
  left_join(country_average, by='country') %>%
  group_by(province) %>%
  summarize(province_effect = mean(points - mean_points_rating- country_effect))

predicted_ratings <- validation %>%
  left_join(country_average, by='country') %>%
  left_join(province_average, by='province') %>%
  mutate(province_effect = mean_points_rating + country_effect + province_effect) %>%
  pull(province_effect)
```

```
rmse_province_effect <- RMSE(predicted_ratings, validation$points)
#store the RMSE data
rmse_data <- bind_rows(rmse_data,
                       data_frame(Method= "Country/Province Effect",
                                  RMSE=rmse_province_effect))
rmse_data
```

```
## # A tibble: 3 x 2
##   Method                    RMSE
##   <chr>                    <dbl>
## 1 Mean Points               3.04
## 2 Country Effect            2.95
## 3 Country/Province Effect   2.85
```

Considering the province effect we can see that the RMSE reduced 2.85

Variety Effect

In the below section we would consider the impact of variety of grape used for wine. Varieties which are common in US Pinot Noir, Chardonnay were reviewed more.

```
#variety effect
variety_average <- train_set %>%
  left_join(country_average, by='country') %>%
  left_join(province_average, by='province') %>%
  group_by(variety) %>%
  summarize(variety_effect = mean(points - mean_points_rating - country_effect - province_effect))

predicted_ratings <- validation %>%
  left_join(country_average, by='country') %>%
  left_join(province_average, by='province') %>%
  left_join(variety_average, by='variety') %>%
  mutate(variety_effect = mean_points_rating + country_effect + province_effect + variety_effect) %>%
  pull(variety_effect)

rmse_variety_effect <- RMSE(predicted_ratings, validation$points)
#store the RMSE data
rmse_data <- bind_rows(rmse_data,
                       data_frame(Method= "Country/Province/Variety Effect",
                                  RMSE=rmse_variety_effect))
rmse_data
```

```
## # A tibble: 4 x 2
##   Method                           RMSE
##   <chr>                           <dbl>
## 1 Mean Points                      3.04
## 2 Country Effect                   2.95
## 3 Country/Province Effect          2.85
## 4 Country/Province/Variety Effect  2.76
```

We could see that the RMSE reduced to 2.76 while we considered the variety effect

Vintage Effect

In the earlier section we extracted the Year of make from the Title column. We are considering the vintage effect as well on the below model

```
#vintage effect
vintage_average <- train_set %>%
  left_join(country_average, by='country') %>%
  left_join(province_average, by='province') %>%
  left_join(variety_average,by='variety') %>%
  group_by(Year_Make) %>%
  summarize(vintage_effect = mean(points - mean_points_rating - country_effect - province_effect - vari


predicted_ratings <- validation %>%
  left_join(country_average, by='country') %>%
  left_join(province_average, by='province') %>%
  left_join(variety_average, by='variety') %>%
  left_join(vintage_average, by='Year_Make') %>%
  mutate(vintage_effect = mean_points_rating + country_effect + province_effect + variety_effect +vintag
  pull(vintage_effect)

rmse_vintage_effect <- RMSE(predicted_ratings, validation$points)
#store the RMSE data
rmse_data <- bind_rows(rmse_data,
                       data_frame(Method= "Country/Province/Variety/Vintage Effect",
                                  RMSE=rmse_vintage_effect))
rmse_data
```

```
## # A tibble: 5 x 2
##   Method                                  RMSE
##   <chr>                                  <dbl>
## 1 Mean Points                             3.04
## 2 Country Effect                          2.95
## 3 Country/Province Effect                 2.85
## 4 Country/Province/Variety Effect         2.76
## 5 Country/Province/Variety/Vintage Effect  2.74
```

With Vintage effect we notice that RMSE reduced to 2.74

Winery Effect

Some Wineries would have a strong brand value and might attract higher points for the wines.

```
#winery effect
winery_average <- train_set %>%
  left_join(country_average, by='country') %>%
  left_join(province_average, by='province') %>%
  left_join(variety_average,by='variety') %>%
  left_join(vintage_average,by='Year_Make') %>%
  group_by(winery) %>%
  summarize(winery_effect = mean(points - mean_points_rating - country_effect - variety_effect - vintag
```

```
predicted_ratings <- validation %>%
  left_join(country_average, by='country') %>%
  left_join(province_average, by='province') %>%
  left_join(variety_average, by='variety') %>%
  left_join(vintage_average, by='Year_Make') %>%
  left_join(winery_average, by='winery') %>%
  mutate(winery_effect = mean_points_rating + country_effect + province_effect + variety_effect + vintag
  pull(winery_effect)

rmse_winery_effect <- RMSE(predicted_ratings, validation$points)
#store the RMSE data
rmse_data <- bind_rows(rmse_data,
                       data_frame(Method= "Country/Province/Variety/Vintage/Winery Effect",
                                  RMSE=rmse_winery_effect))
rmse_data
```

```
## # A tibble: 6 x 2
##   Method                                          RMSE
##   <chr>                                          <dbl>
## 1 Mean Points                                     3.04
## 2 Country Effect                                  2.95
## 3 Country/Province Effect                         2.85
## 4 Country/Province/Variety Effect                 2.76
## 5 Country/Province/Variety/Vintage Effect         2.74
## 6 Country/Province/Variety/Vintage/Winery Effect  2.44
```

We can see that the considering the Winery impact reduced the RMSE to 2.44

Taster Effect

As we observed earlier some tasters are more critical than some generous tasters. Let us consider the taster bias as well

$$Y_{g,va,vi,w,t,i} = \mu + b_g + b_v a + b_v i + bw + bt + \epsilon_{g,va,vi,w,t,i}$$

$Y_{g,va,vi,w,t,i}$ - Predicted value μ -Average points b_g - Geographical Bias $b_v a$ - Variety Bias $b_v i$ - Vintage Bias b_w - Winery Bias b_t - taster Bias $\epsilon_{g,va,vi,w,t,i}$ - independent error

```
#taster effect
taster_average <- train_set %>%
  left_join(country_average, by='country') %>%
  left_join(province_average, by='province') %>%
  left_join(variety_average,by='variety') %>%
  left_join(vintage_average,by='Year_Make') %>%
  left_join(winery_average,by='winery') %>%
  group_by(taster_name) %>%
  summarize(taster_effect = mean(points - mean_points_rating - country_effect - variety_effect - vintag

predicted_ratings <- validation %>%
  left_join(country_average, by='country') %>%
  left_join(province_average, by='province') %>%
```

```
  left_join(variety_average, by='variety') %>%
  left_join(vintage_average, by='Year_Make') %>%
  left_join(winery_average, by='winery') %>%
  left_join(taster_average, by='taster_name') %>%
  mutate(taster_effect = mean_points_rating + country_effect + province_effect + variety_effect + vintag
  pull(taster_effect)

rmse_taster_effect <- RMSE(predicted_ratings, validation$points)
#store the RMSE data
rmse_data <- bind_rows(rmse_data,
                       data_frame(Method= "Country/Province/Variety/Vintage/Winery/Taster Effect",
                                  RMSE=rmse_taster_effect))
rmse_data
```

```
## # A tibble: 7 x 2
##   Method                                                RMSE
##   <chr>                                                <dbl>
## 1 Mean Points                                           3.04
## 2 Country Effect                                        2.95
## 3 Country/Province Effect                               2.85
## 4 Country/Province/Variety Effect                       2.76
## 5 Country/Province/Variety/Vintage Effect               2.74
## 6 Country/Province/Variety/Vintage/Winery Effect        2.44
## 7 Country/Province/Variety/Vintage/Winery/Taster Effect  2.43
```

RMSE reduced to 2.43 which is considerable improvement from the Naive model

Results and Discussion

In this project, I have trained machine learning algorithm using the inputs in one subset to predict wine points ratings in the validation set. A naive implementation had a Root Mean Square Error of 3.06. We have improve the prediction accuracy by 20% by considering the below effects.

- Geographical effect (Country of Origin /Province)
- Variety effect
- Vintage effect (Year of Make)
- Winery effect
- Taster effect

```
rmse_data
```

```
## # A tibble: 7 x 2
##   Method                                                RMSE
##   <chr>                                                <dbl>
## 1 Mean Points                                           3.04
## 2 Country Effect                                        2.95
## 3 Country/Province Effect                               2.85
## 4 Country/Province/Variety Effect                       2.76
## 5 Country/Province/Variety/Vintage Effect               2.74
## 6 Country/Province/Variety/Vintage/Winery Effect        2.44
## 7 Country/Province/Variety/Vintage/Winery/Taster Effect  2.43
```

The lowest RMSE (2.43) acheived is when we considered all the effects together

Conclusion

A prediction model that we created for wine ratings improved the prediction accuracy by 20%. Further analysis could be done on to predict the sentiment of a review using the words used in the taster feedback for the wine. Also further analysis can be done by experimenting with various machine learning algorithms to see if they perform better than linear regression

Environment

```
version
```

```
##                _
## platform       x86_64-w64-mingw32
## arch           x86_64
## os             mingw32
## system         x86_64, mingw32
## status
## major          3
## minor          6.2
## year           2019
## month          12
## day            12
## svn rev        77560
## language       R
## version.string R version 3.6.2 (2019-12-12)
## nickname       Dark and Stormy Night
```

#References

- https://rafalab.github.io/dsbook/**
- [Dataset source & description] (https://www.kaggle.com/zynicide/wine-reviews#winemag-data-130k-v2.csv)
- [NLP] (https://rstudio-pubs-static.s3.amazonaws.com/132792_864e3813b0ec47cb95c7e1e2e2ad83e7.html)