Predicting Wine Quality Using Text Reviews

MSBA-8225: Analytical Methods for Text and Web Mining

Term Project | 12.14.2022

Savio Braganza | Nas Dowling | Jennie Franco
William Hillegas | Chris McMahon

Dr. Q B. Chung
Villanova University
Fall 2022

Page of

Contents

Project Summary

2 Significance of TM Project

Understanding the business need and research questions

The aim of this project is to develop a model which can effectively predict wine quality using attributes in the wine reviews and textual wine descriptions offered by experienced sommeliers. We apply duo-mining methods to maximize the predictive power of our model so as to extract every bit of value from our dataset. We focus on the following research questions in this study:

- What is the optimal model for predicting wine quality?
- What terms are typically indicative of a wine likely to be classed as "top quality"?
- How can we exploit sentiment polarity when selecting a fine wine?

3 Data Understanding

The data for this study is sourced from Kaggle, which was initially derived from the WineEnthusiast database in June, 2017. The dataset consists of the following attributes:

- Country - the country of origin of the wine
- Description - text review of the wine
- Designation - the vineyard within the winery from which the grapes are sourced
- Points - no. Of points WineEnthusiasts rated the wine on a scale of 1 to 100
- Price - the price given to the wine bottle
- Province - the province or state where the wine is produced
- Regions (region 1 & region 2) - refers to the wine-growing regions or areas
- Taster Name - the name of the wine reviewer
- Taster Twitter Handle - Twitter username for wine reviewer

In the data understanding phase we evaluated the entire dataset for inclusion in our model development but ended up proceeding with a sample of 7,500 records for two reasons:

1. Dropped records containing Null values, reducing the data size to around 20K.
2. Limitations of RapidMiner Academic License. When dealing with records >10K we frequently encountered memory-related issues, specifically for Process Documents & Model Development operators.

We removed attributes that we deemed were extraneous. These attributes included country (the only country left in our dataset after reducing the size and removing the null values was the US), id, region_2, and taster_twitter_handle. Since we are interested in predicting wine quality, we used R to derive a label titled 'PointsBin' which classifies wines into two groups (High and Low-Med) according to their point values. Specifically, a wine is classified as "High", if its points are above the 75th percentile; all others are classified as "Low-Med". Sumtable illustrated that 92 for points was the 75th percentile within our 7,500 row data set, so we utilized this finding to inform our bucketing logic. (R code for these steps available in Appendix)

Data Processing

4 Data Preparation

Establish corpus and perform the preprocessing steps

In this section, we imported the dataset into RapidMiner for preprocessing and model development. When reading the files as a CSV, we dropped points from the dataset due to collinearity with 'points' which was used to derive 'PointsBin'. The next step was preprocessing, where we utilized the Process Documents from Data operator, to tokenize, transform to lower case, and filter stopwords. (Figure 2)

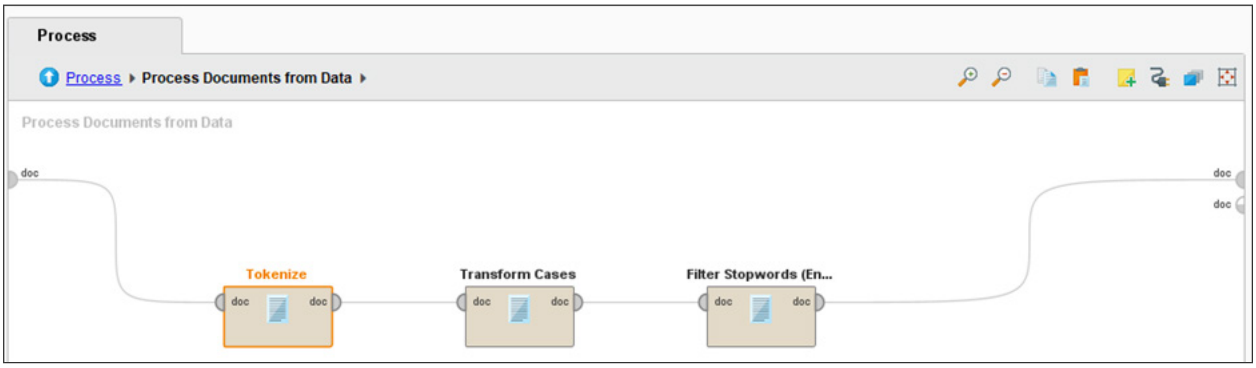

Process
Process ▸ Process Documents from Data ▸
Process Documents from Data
doc
Tokenize
Transform Cases
Filter Stopwords (En...
doc
doc

Figure 2: Process Documents From Data Operator

After processing, we created a TF-IDF (Term Frequency-Inverse Document Frequency) to use for model creation (screenshot of TF-IDF in Appendix).

Model Development

5 Create Models

For the development of the models, we ran the following algorithms:

1. Naive Bayes
2. Decision Tree
3. Gradient Boost

The confusion matrix for each model is shown in the Appendix, while the precision, recall and F1 score are summarized in Figure 3.

Pred High Quality	Naive Bayes	Decision Tree	Gradient Boost
Class Recall	63.3%	2.0%	44.7%
Class Precision	30.4%	56.1%	56.6%
F1 Score	41.4%	3.8%	50.0%

Figure 3: Model Comparison Chart

The extremely low class recall ruled out the Decision Tree model, as this indicated an overly conservative model that would result in numerous false negatives. We attempted multiple confidence levels for the decision tree to refine and correct for this, however RapidMiner failed to run any confidence adjustments for the Decision Tree. Naives Bayes had the best class recall, suggesting that this model would be optimal if our goal is to correctly classify high quality wines. Gradient boost had an average recall but the best precision, meaning that we have the highest likelihood of accurately predicting high quality wine based on text review. Calculating the F1 score assisted in validating our decision that Gradient Boost was the optimal model for this project.

6 Insights

The Gradient Boost model has the best performance among the models. It has decent recall, the best precision, and best F1 score of the three. It should be pointed out that these numbers aren't particularly high when one thinks of typical precision, recall, and F1 scores. This may be due to the quality of the description attribute when pressed into service for text analytics.
Prior to building our final model, we ran a process in RapidMiner without designating the description as a text attribute and saw much higher numbers for precision and recall, as well as accuracy.

After changing the description to text, these numbers dropped. We theorize that because a lot of the same words/tokens appear in both high and low-medium quality wine descriptions, the model is not sensitive enough to separate the signal from the noise. Usually text analytics is supposed to augment data mining to improve lift, but unfortunately we did not find that to be the case with this dataset. It was a good learning experience though, and we judge it to have been worth the effort nonetheless.

Next, we address the second question we posed regarding terms that may be associated with high quality wine.

Despite not selecting the Decision Tree model, it did have high class precision relative to the other models, allowing us to glean insights from the classifiers that were deemed strong indicators of “high quality” wine. Figure 4 highlights some of these terms which included “dewy”, “riveting, “wondrous”, “exceptional” & “chopped”.

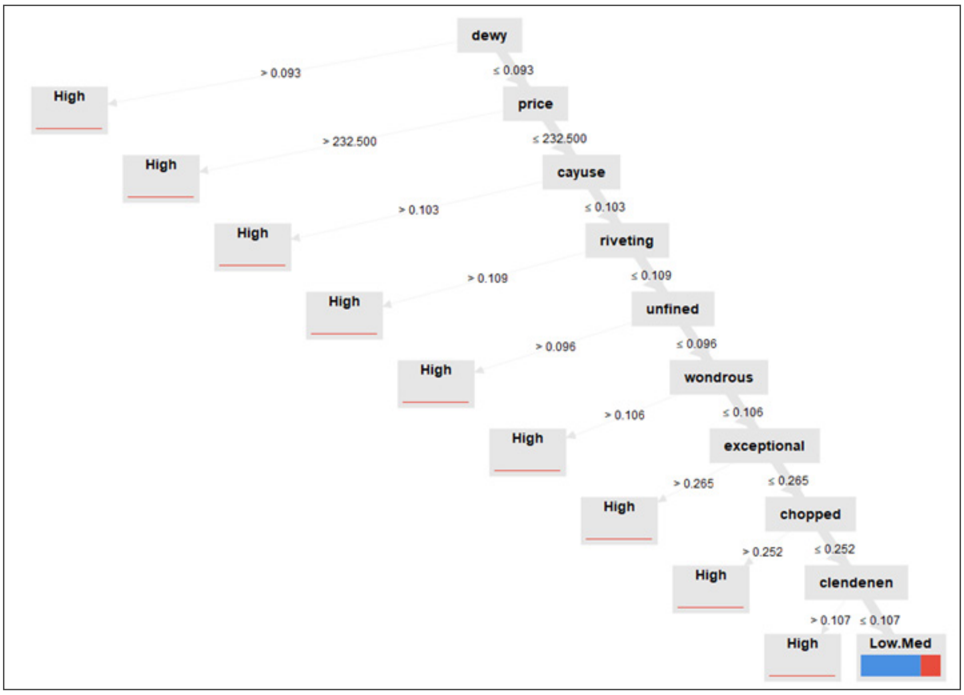


Figure 4: Decision Tree Model Visualization

To augment our analysis and determine which terms appeared most frequently across all of the wine descriptions, we created a word cloud using Python and a word frequency chart in R across both high and low-medium grade wine, both of which are displayed in Figure 5.

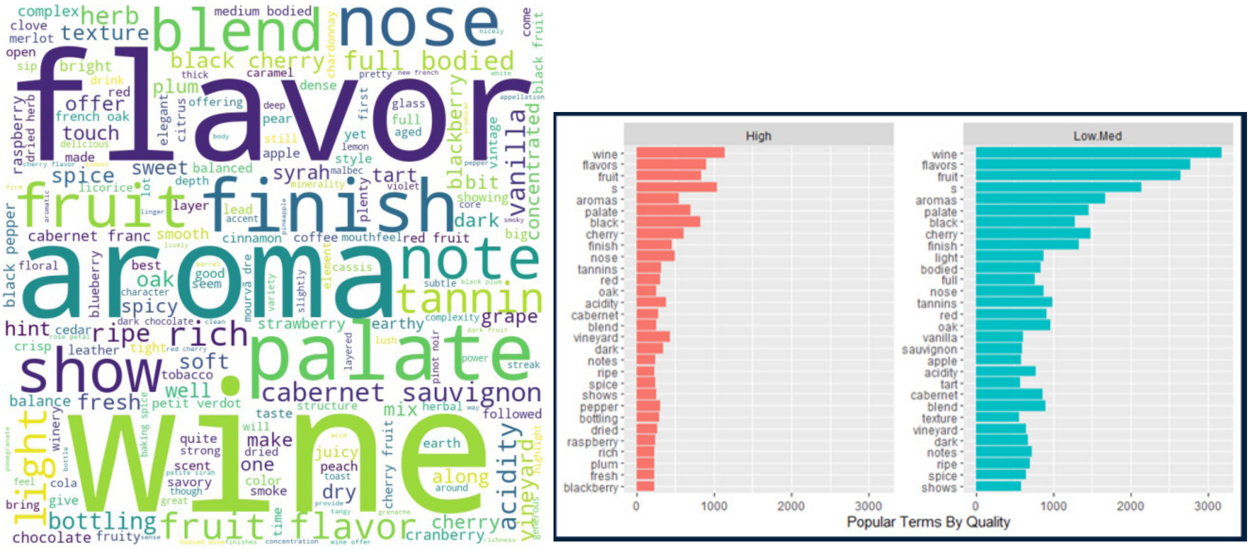


Figure 5: WordCloud & Popular Review Terms By Quality

As suspected, the terms with the highest frequency count shown in large text in the WordCloud are common wine description terms, such as wine, aroma, and flavor. The less frequent terms shown in smaller print include technical jargon or flavors, such as french oak, tobacco, and full bodied.

The word frequency chart reinforces the word cloud. The biggest word, wine, is the term that appears most frequently atop the list for both high and low-medium quality wines. At the bottom of this chart are more specialized terms and flavors (the same words which appear small in the word cloud).

For deeper analysis we decided to use sentiment analysis to extract further insights. This was implemented using the Operator Toolbox Extract Sentiment operator in RapidMiner (this specific operator was chosen since some of the other choices used to extract sentiment were unavailable or had been removed from the RapidMiner Marketplace).

A score of zero indicates neutrality, whereas a value above zero suggests positive sentiment and anything below zero suggests negative sentiment. A histogram of the polarity is shown in Figure 6.

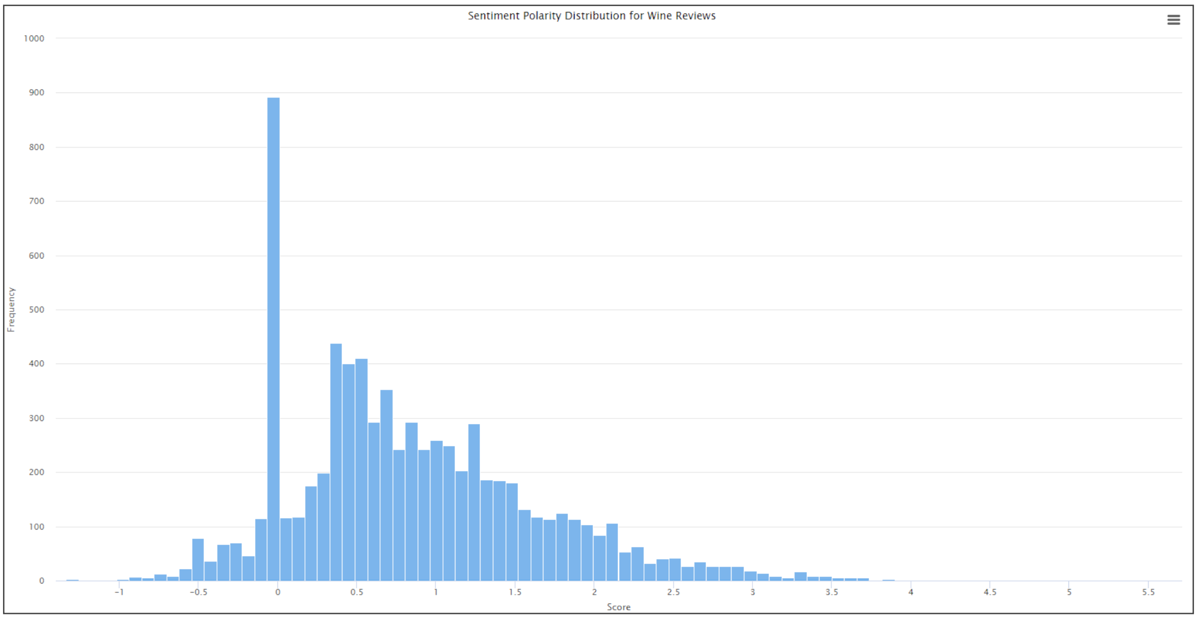


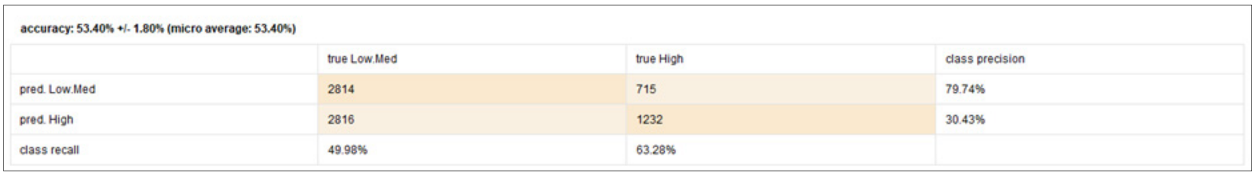
Figure 6: Sentiment Polarity Distribution by Wine Reviews

The sentiment appears to be positive-skewed across all wine qualities.

Conclusion

Wine quality can be effectively analyzed using machine learning algorithms like Gradient Boosting. Text analytics can glean supplemental insights into wine quality. With careful effort, it is definitely possible to fine-tune a model that enables one to maximize profits by selecting high quality wines to charge higher prices if one is a wine-producer or to be a more discerning buyer if one is a wine-consumer.

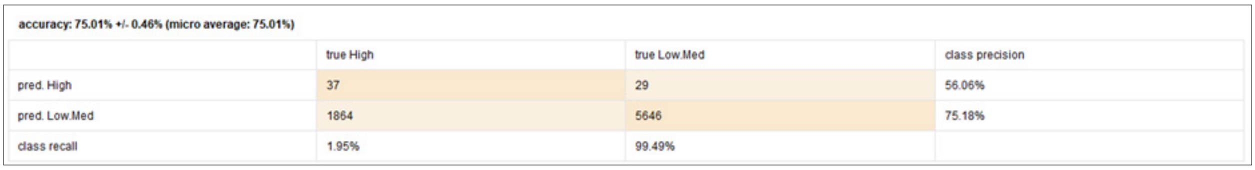
APPENDIX



accuracy: 53.40% +/- 1.80% (micro average: 53.40%)

	true Low.Med	true High	class precision
pred. Low.Med	2814	715	79.74%
pred. High	2816	1232	30.43%
class recall	49.98%	63.28%	

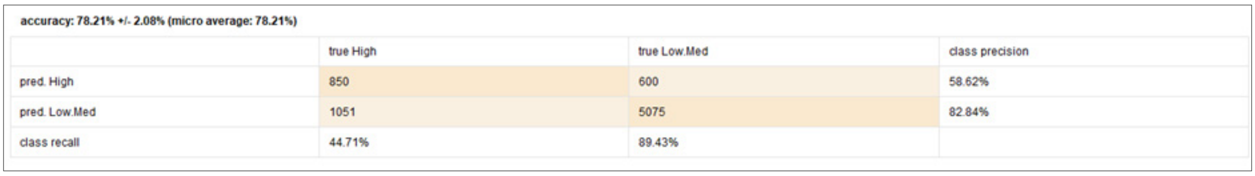
Figure 7: Naive Bayes Confusion Matrix



accuracy: 75.01% +/- 0.46% (micro average: 75.01%)

	true High	true Low.Med	class precision
pred. High	37	29	56.06%
pred. Low.Med	1864	5646	75.18%
class recall	1.95%	99.49%	

Figure 8: Decision Tree Confusion Matrix



accuracy: 78.21% +/- 2.08% (micro average: 78.21%)

	true High	true Low.Med	class precision
pred. High	850	600	58.62%
pred. Low.Med	1051	5075	82.84%
class recall	44.71%	89.43%	

Figure 9: Gradient Boost Confusion Matrix

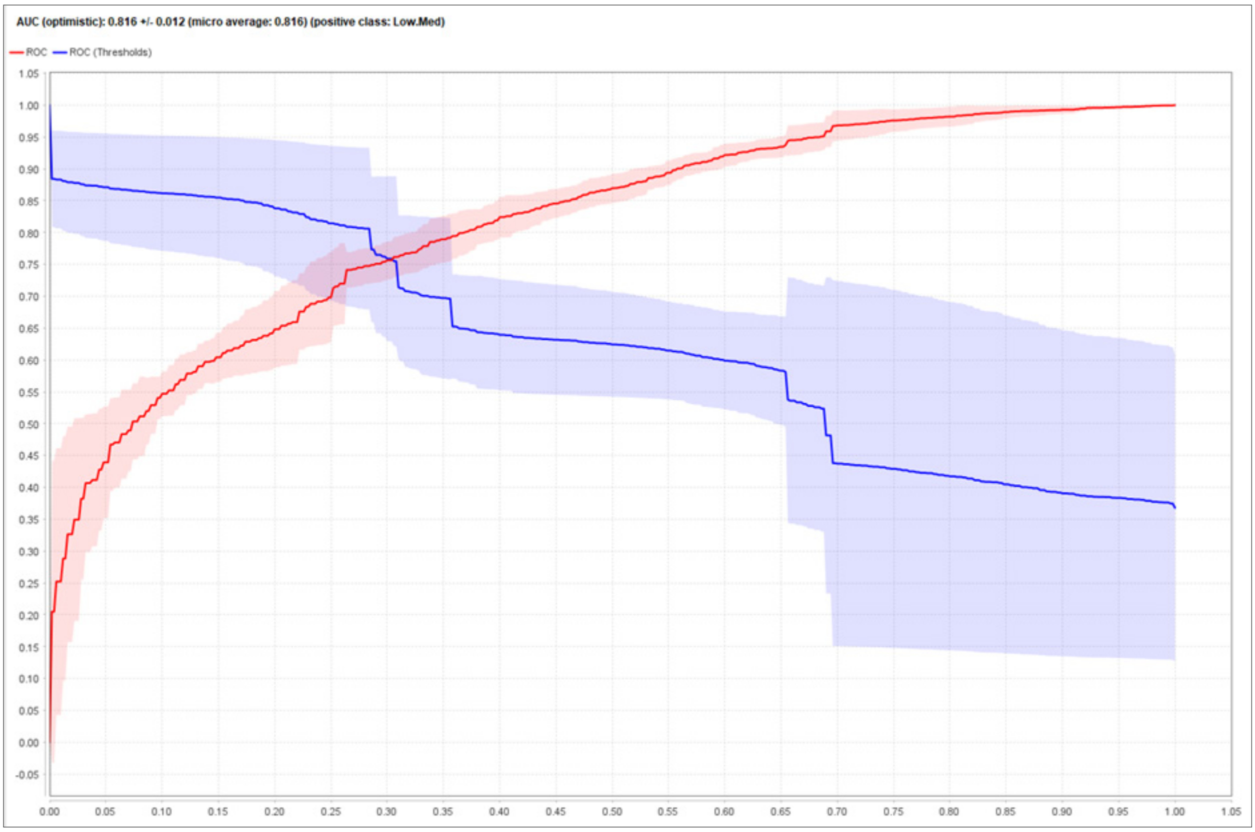
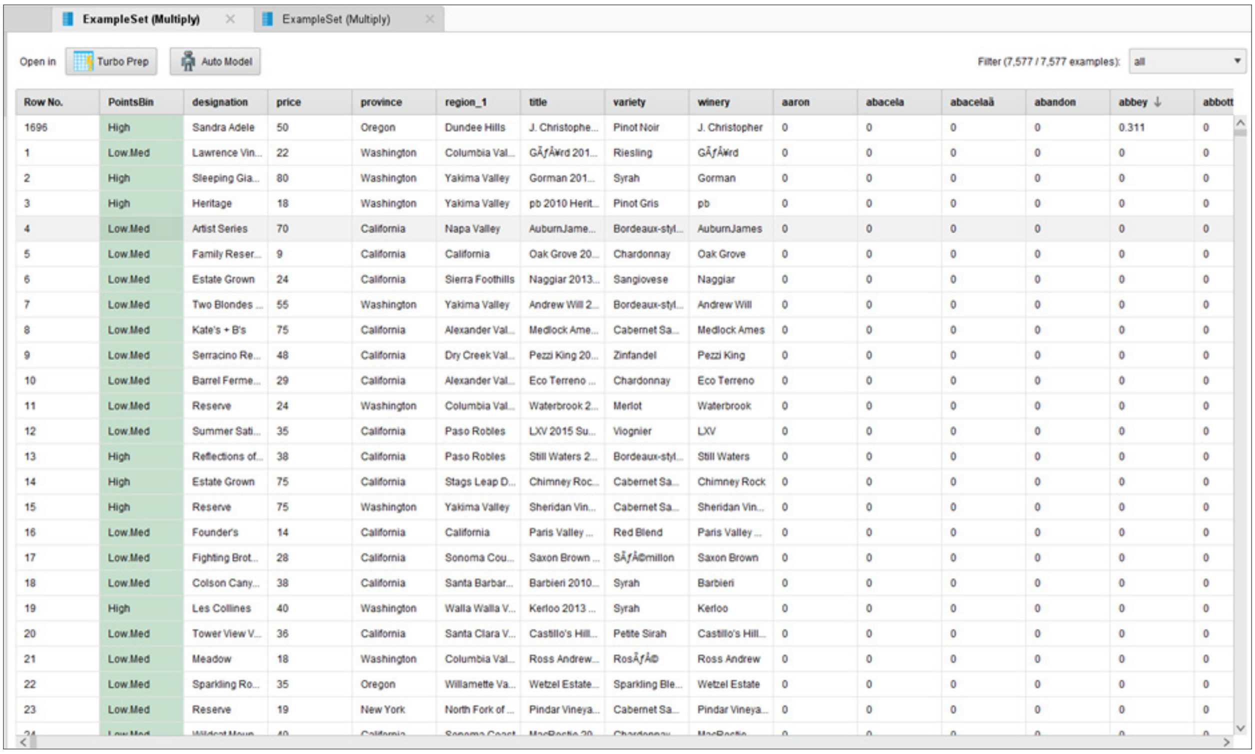


Figure 10: AUC for Gradient Boost



Filter (7,577 / 7,577 examples): all

Row No.	PointsBin	designation	price	province	region_1	title	variety	winery	aaron	abacela	abacelaã	abandon	abbey ↓	abbott
1696	High	Sandra Adele	50	Oregon	Dundee Hills	J. Christophe...	Pinot Noir	J. Christopher	0	0	0	0	0.311	0
1	Low.Med	Lawrence Vin...	22	Washington	Columbia Val...	GÃƒÂ¥rd 201...	Riesling	GÃƒÂ¥rd	0	0	0	0	0	0
2	High	Sleeping Gia...	80	Washington	Yakima Valley	Gorman 201...	Syrah	Gorman	0	0	0	0	0	0
3	High	Heritage	18	Washington	Yakima Valley	pb 2010 Herit...	Pinot Gris	pb	0	0	0	0	0	0
4	Low.Med	Artist Series	70	California	Napa Valley	AuburnJame...	Bordeaux-styl...	AuburnJames	0	0	0	0	0	0
5	Low.Med	Family Reser...	9	California	California	Oak Grove 20...	Chardonnay	Oak Grove	0	0	0	0	0	0
6	Low.Med	Estate Grown	24	California	Sierra Foothills	Naggiar 2013...	Sangiovese	Naggiar	0	0	0	0	0	0
7	Low.Med	Two Blondes ...	55	Washington	Yakima Valley	Andrew Will 2...	Bordeaux-styl...	Andrew Will	0	0	0	0	0	0
8	Low.Med	Kate's + B's	75	California	Alexander Val...	Medlock Ame...	Cabernet Sa...	Medlock Ames	0	0	0	0	0	0
9	Low.Med	Serracino Re...	48	California	Dry Creek Val...	Pezzi King 20...	Zinfandel	Pezzi King	0	0	0	0	0	0
10	Low.Med	Barrel Ferme...	29	California	Alexander Val...	Eco Terreno ...	Chardonnay	Eco Terreno	0	0	0	0	0	0
11	Low.Med	Reserve	24	Washington	Columbia Val...	Waterbrook 2...	Merlot	Waterbrook	0	0	0	0	0	0
12	Low.Med	Summer Sati...	35	California	Paso Robles	LXV 2015 Su...	Viognier	LXV	0	0	0	0	0	0
13	High	Reflections of...	38	California	Paso Robles	Still Waters 2...	Bordeaux-styl...	Still Waters	0	0	0	0	0	0
14	High	Estate Grown	75	California	Stags Leap D...	Chimney Roc...	Cabernet Sa...	Chimney Rock	0	0	0	0	0	0
15	High	Reserve	75	Washington	Yakima Valley	Sheridan Vin...	Cabernet Sa...	Sheridan Vin...	0	0	0	0	0	0
16	Low.Med	Founder's	14	California	California	Paris Valley ...	Red Blend	Paris Valley ...	0	0	0	0	0	0
17	Low.Med	Fighting Brot...	28	California	Sonoma Cou...	Saxon Brown ...	SÃƒÂ©millon	Saxon Brown	0	0	0	0	0	0
18	Low.Med	Colson Cany...	38	California	Santa Barbar...	Barbieri 2010...	Syrah	Barbieri	0	0	0	0	0	0
19	High	Les Collines	40	Washington	Walla Walla V...	Kerloo 2013 ...	Syrah	Kerloo	0	0	0	0	0	0
20	Low.Med	Tower View V...	36	California	Santa Clara V...	Castillo's Hill...	Petite Sirah	Castillo's Hill...	0	0	0	0	0	0
21	Low.Med	Meadow	18	Washington	Columbia Val...	Ross Andrew...	RosÃƒÂ©	Ross Andrew	0	0	0	0	0	0
22	Low.Med	Sparkling Ro...	35	Oregon	Willamette Va...	Wetzel Estate...	Sparkling Ble...	Wetzel Estate	0	0	0	0	0	0
23	Low.Med	Reserve	19	New York	North Fork of ...	Pindar Vineya...	Cabernet Sa...	Pindar Vineya...	0	0	0	0	0	0
24	Low.Med	Wildcat Moun...	40	California	Sonoma Coast	MacRostie 20...	Chardonnay	MacRostie	0	0	0	0	0	0

Figure 11: Term Frequency-Inverse Document Frequency

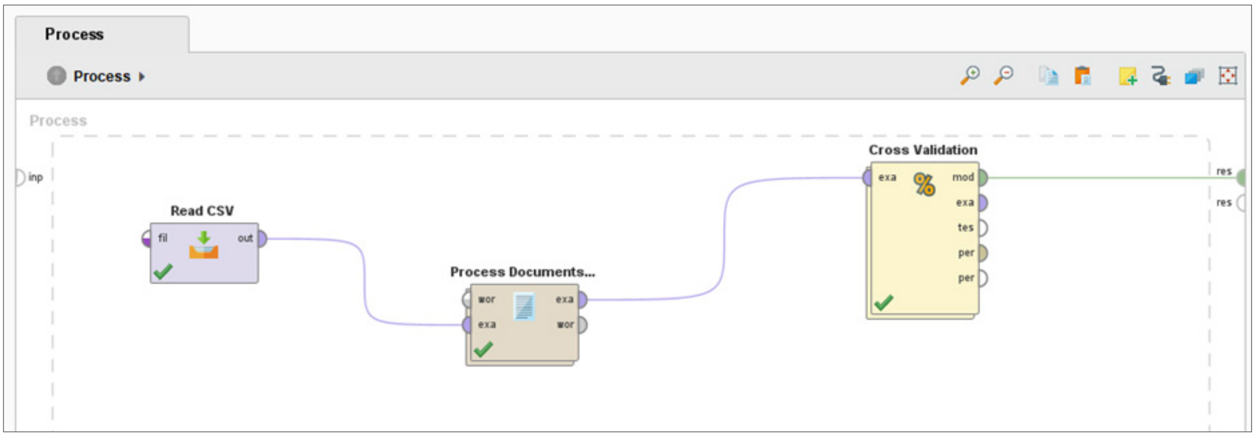


Figure 12: RapidMiner Model Development Process

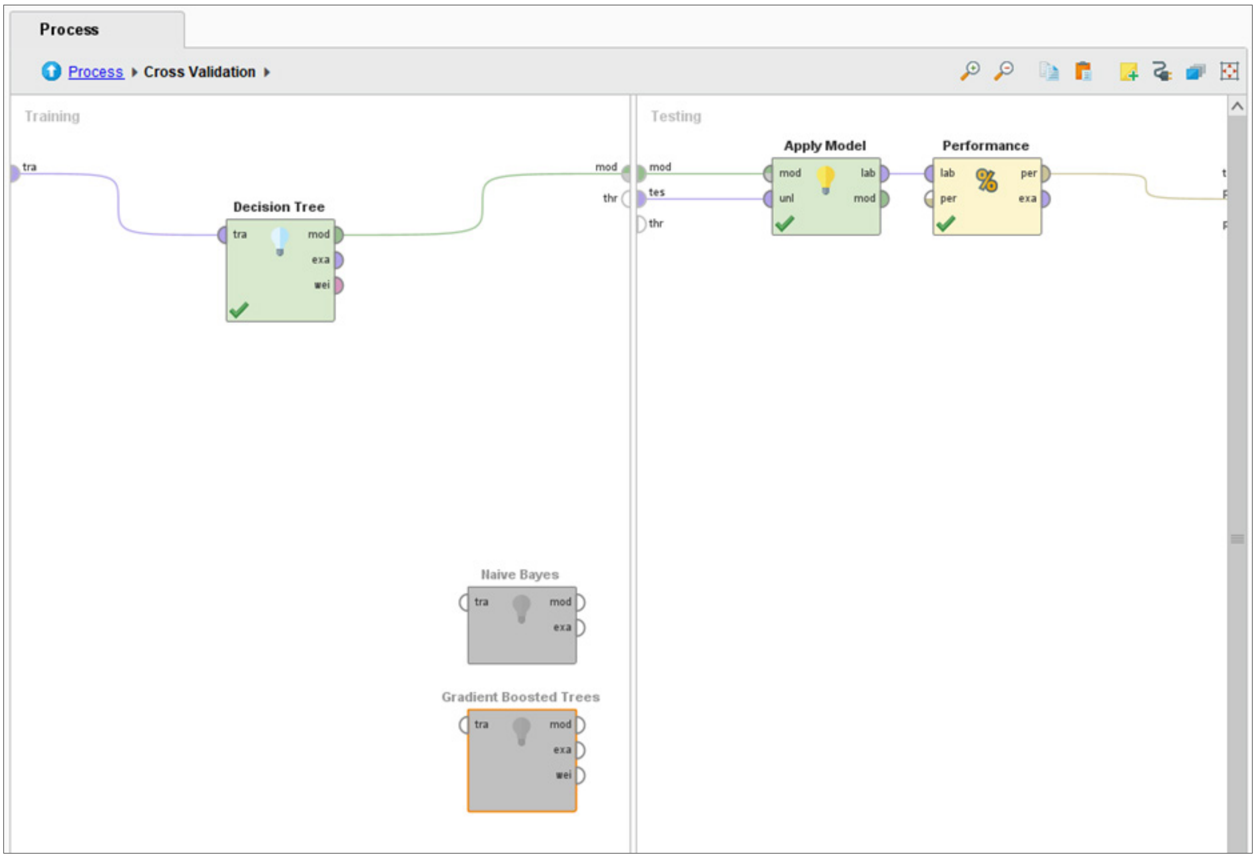


Figure 13: RapidMiner Model Development Process - Cross Validation with Decision Tree

R Code for Data Preparation & Formatting

~~~
---
title: "Fall 2022 MSA 8225: Analytical Methods for Text/Web Mining - Term Project"
12/13/2022
---

Import & Install Packages
```{r}

library(reshape2)
library(textdata)
library(stringi)
library(sentimentr)
library(readr)
library(wordcloud)
library(lubridate)
library(ggplot2)
library(ggraph)
library(igraph)
library(plotrix)
library(bitops)
library(httr)
library(NLP)
library(RCurl)
library(tm)
library(twitteR)
library(XML)
library(vroom)
library(readtext)
library(dplyr)
library(readxl)
library(tidyverse)
library(quanteda)
library(tokenizers)
library(textstem)
library(knitr)
library(data.table)
library(ggplot2)
library(wordcloud2)
library(tidytext)
library(magrittr)
library(vtable)
library(writexl)
library(gofastr)
~~~

~~~
library(topicmodels)
library(devtools)
library(LDAvis)
library(tidyr)
library(vlad)
library(class)
library(caret)
library(rpart)
library(rpart.plot)
library(randomForest)

```


Import Data
```{r}
#Download Excels

data <- read_excel("C:/Users/C-McM/OneDrive/Desktop/Web & Text Mining/winemag-data.xlsx")

head(data)
str(data)
summary(data)

#Drop Taster Name, Taster Twitter,  Title from Data, Region 2
Corpus1 = subset(data, select = -c(att1,region_2, region_2,taster_twitter_handle))

Corpus2 <- Corpus1 %>% mutate(id = row_number())

#Evaluate Rows with missing data
sapply(Corpus2, function(x) sum(is.na(x)))

```


Bucket Wine reviews into buckets
```{r}
sumtable(Corpus2)
summary(Corpus2)
head(Corpus2,50)

CorpusSmall <- Corpus2 %>% sample_frac(0.339)

sumtable(CorpusSmall)
~~~

~~~
CorpusBins <- CorpusSmall %>% mutate(PointsBin = case_when(points < 92 ~ 'Low.Med',
                                                          points >= 92 ~ 'High'))
CorpusBins %>% select(PointsBin) %>% table


CorpusBins2 <- CorpusBins %>%  group_by(PointsBin) %>%
  unnest_tokens(word,description) %>% ungroup()

CorpusBins2

#create word visual

CorpusWords <- read_csv("C:/Users/C-McM/OneDrive/Desktop/Web & Text Mining/CorpusWords.csv")

CorpusWords2 <- CorpusWords %>%  group_by(PointsBin) %>%
  unnest_tokens(word,text) %>% ungroup()

CorpusWords2

bingwordcounts <- CorpusWords2 %>%
  count(word, PointsBin, sort = TRUE) %>%
  ungroup()

bingwordcounts

bingwordcounts %>%
  group_by(PointsBin) %>%
  top_n(30) %>%
  ungroup() %>%
  mutate (word = reorder(word, n)) %>%
  ggplot(aes(word, n, fill = PointsBin)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~PointsBin, scales = "free_y") +
  labs(y = "Popular Terms By Quality",
       x = NULL) +
  coord_flip()

#Export Corpus
write.table(CorpusBins, "C:/Users/C-McM/OneDrive/Desktop/Web & Text Mining/CorpusBins2.csv", row.names=F, sep=",")


```
~~~

Python Code for WordCloud

```
# Install  the following packages in the python CMD.exe prompt

from wordcloud import WordCloud, STOPWORDS
import matplotlib.pyplot as plt
import pandas as pd

# Script to generate Wordcloud

df = pd.read_csv("CorpusBins2.csv")

print(df)

# Build WordCloud

comment_words = ''
stopwords = set(STOPWORDS)

#iterate through the excel file
for val in df.description:

    # typecaset each val to string
    val = str(val)

    # split the value
    tokens = val.split()

   # converts each token into lowercase
    for i in range(len(tokens)):
        tokens[i] = tokens[i].lower()

    comment_words += " ".join(tokens)+" "


wordcloud = WordCloud(width = 800, height = 800,
            background_color ='white',
            stopwords = stopwords,
            min_font_size = 10).generate(comment_words)
```

```
# plot the Worcloud image

wcFig = plt.figure(figsize = (8,8), facecolor = None)
plt.imshow(wordcloud)
plt.axis("off")
plt.tight_layout(pad=0)

# Saving the plot as an image
wcFig.savefig('00_Term Project/WordCloud.jpg')

# Show plot
plt.show()
```