

732A92 Text mining project

Evaluation wine based on descriptions

*Student: Duc Duong (mindu931)*

*Instructor: prof. Marco Kuhlmann*

Linköping, 16th January 2019

**Contents**

[Abstract 3](#_Toc535804713)

[1. Introduction 3](#_Toc535804714)

[1.1 Motivation 3](#_Toc535804715)

[1.2 Aim 3](#_Toc535804716)

[1.3 Outline 3](#_Toc535804717)

[2. Data 3](#_Toc535804718)

[3. Theory 6](#_Toc535804719)

[4. Method 9](#_Toc535804720)

[5. Result and explain 11](#_Toc535804721)

[5.1 Preprocess text 11](#_Toc535804722)

[5.2 Compare kernels 22](#_Toc535804723)

[5.3 Improve the model 25](#_Toc535804724)

[6. Second aspect: High value wine 49](#_Toc535804725)

[6.1 Motivation 49](#_Toc535804726)

[6.2 Method and result 50](#_Toc535804727)

[7. Further discussion 55](#_Toc535804728)

[7.1 Suggestions for choosing wine 55](#_Toc535804729)

[7.2 Future work: 58](#_Toc535804730)

[8. Conclusion 58](#_Toc535804731)

[References 58](#_Toc535804732)

[Appendix 59](#_Toc535804733)

# Abstract

Wine sensory examination and assessment is not easy, even with a wine specialist. However, the description printed on each bottle can bring us some helpful information. Based on that ideal, I decided to analyze the data about wine review (taken form Kaggle) and help the consumers make the decision on two different aspects. The first one is distinguish good and excellent wine. The second aspect is investigating the finance beneficial bottle (high value with an acceptable amount of money). Different natural language processing techniques are applied to process the text data. Then I compare Naive Bayes, support vector machine (SVM) Linear and SVM RBF kernels when building the best model for predicting. After the best kernel is selected, n-gram and Part-Of-Speech Tagger are used to improve the model. The final results are quite impressive with accuracy 78.69% for the first issue and 77.09% for the second one. Based on these models, some suggestions are provided for the consumers to choose satisfied bottles.

# Introduction

## 1.1 Motivation

While standing in a wine cellar, I was always confusing with a lot of questions: How can I choose an excellent wine for my family? What is the best wine fitted with my budget? The problems could be solved easier if I had a good sense. But, sadly, like many people in the world, I don’t. The problems still continue when I listen to the advice of the salesman and even taste some wine. So, I usually choose randomly a bottle based on this advice for my events. Unluckily, sometimes my relatives didn’t think that it was an excellent wine.

## 1.2 Aim

I guess that the information from the provider (description, country, region, designation...) should bring some important clues. So, I try to solve the issues based on text mining idea: Take the wine data which is already reviewed and marked by some specialists, then process the data together with the points and build a model to prediction in order to figure out which bottle is good or excellent, which one have high benefit or just medium.

## 1.3 Outline

In this report, Firstly, I will provide information about the data. Then, I briefly summary relevant theories. The method I used will be presented in details. It includes the steps for preprocessing, techniques for comparing and choosing the best model. Finally, there are some suggestions for choosing wine, some discussions and my conclusion.

# Data

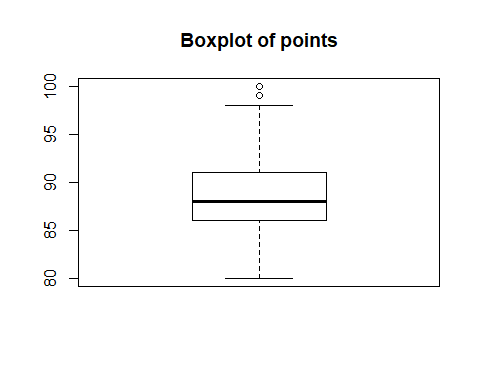
Wine review data is taken form Kaggle ([link](https://www.kaggle.com/zynicide/wine-reviews)). I choose the second .csv version of this dataset. In this version, duplicated data is removed. The data include 120975 samples about wine. Each sample contains the following information: country, description, designation, points, price, province, region, more specific region (region2), tester name, taster\_twitter handle, title, variety, and winery. Here is the overlook of the data:

head(wine)

## X country  
## 1 0 Italy  
## 2 1 Portugal  
## 3 2 US  
## 4 3 US  
## 5 4 US  
## 6 5 Spain  
## description  
## 1 Aromas include tropical fruit, broom, brimstone and dried herb. The palate isn't overly expressive, offering unripened apple, citrus and dried sage alongside brisk acidity.  
## 2 This is ripe and fruity, a wine that is smooth while still structured. Firm tannins are filled out with juicy red berry fruits and freshened with acidity. It's already drinkable, although it will certainly be better from 2016.  
## 3 Tart and snappy, the flavors of lime flesh and rind dominate. Some green pineapple pokes through, with crisp acidity underscoring the flavors. The wine was all stainless-steel fermented.  
## 4 Pineapple rind, lemon pith and orange blossom start off the aromas. The palate is a bit more opulent, with notes of honey-drizzled guava and mango giving way to a slightly astringent, semidry finish.  
## 5 Much like the regular bottling from 2012, this comes across as rather rough and tannic, with rustic, earthy, herbal characteristics. Nonetheless, if you think of it as a pleasantly unfussy country wine, it's a good companion to a hearty winter stew.  
## 6 Blackberry and raspberry aromas show a typical Navarran whiff of green herbs and, in this case, horseradish. In the mouth, this is fairly full bodied, with tomatoey acidity. Spicy, herbal flavors complement dark plum fruit, while the finish is fresh but grabby.  
## designation points price province  
## 1 VulkÃ  Bianco 87 NA Sicily & Sardinia  
## 2 Avidagos 87 15 Douro  
## 3 87 14 Oregon  
## 4 Reserve Late Harvest 87 13 Michigan  
## 5 Vintner's Reserve Wild Child Block 87 65 Oregon  
## 6 Ars In Vitro 87 15 Northern Spain  
## region\_1 region\_2 taster\_name  
## 1 Etna Kerin Oâ\200\231Keefe  
## 2 Roger Voss  
## 3 Willamette Valley Willamette Valley Paul Gregutt  
## 4 Lake Michigan Shore Alexander Peartree  
## 5 Willamette Valley Willamette Valley Paul Gregutt  
## 6 Navarra Michael Schachner  
## taster\_twitter\_handle  
## 1 @kerinokeefe  
## 2 @vossroger  
## 3 @paulgwineÂ   
## 4   
## 5 @paulgwineÂ   
## 6 @wineschach  
## title  
## 1 Nicosia 2013 VulkÃ  Bianco (Etna)  
## 2 Quinta dos Avidagos 2011 Avidagos Red (Douro)  
## 3 Rainstorm 2013 Pinot Gris (Willamette Valley)  
## 4 St. Julian 2013 Reserve Late Harvest Riesling (Lake Michigan Shore)  
## 5 Sweet Cheeks 2012 Vintner's Reserve Wild Child Block Pinot Noir (Willamette Valley)  
## 6 Tandem 2011 Ars In Vitro Tempranillo-Merlot (Navarra)  
## variety winery  
## 1 White Blend Nicosia  
## 2 Portuguese Red Quinta dos Avidagos  
## 3 Pinot Gris Rainstorm  
## 4 Riesling St. Julian  
## 5 Pinot Noir Sweet Cheeks  
## 6 Tempranillo-Merlot Tandem

Points are important information in this data. Points are marked by wine specialists, which make it reliable. The owner only public data which have at least 80 points (out of 100) which means the data contain only good and excellent wine. Here is the distribution of the points:

boxplot(wine$points, main="Boxplot of points")



# Theory

In this section, I will briefly talk about some theory in natural language processing (NLP) and machine learning (ML) in a simple way of reviewing. For some terms, only the aspect that relevant to this project is presented. I assume that the reader already has some basic knowledge about NLP and ML before (this part is not a guide for a person with a blank background). I also noticed that we are processing the text written in English.

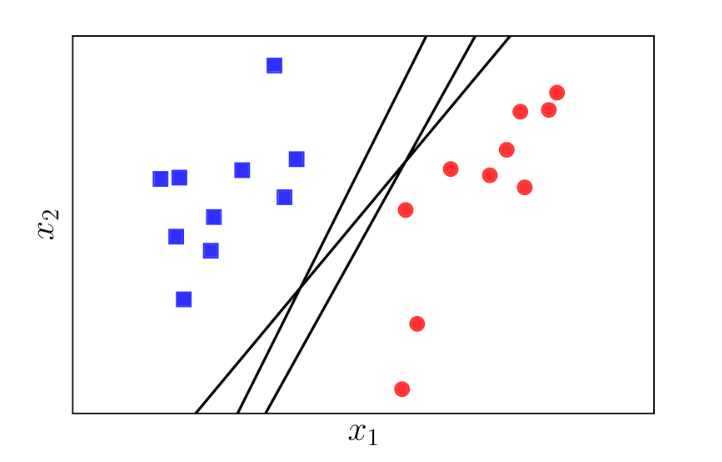
* Terms in text preprocessing:
* Stop words: This is the term for useless words in the text. To be specific, these words usually have no (or very low) meaning but represent a lot. For example: a, an, the… When processing text data. We usually remove these words.
* Steaming: This is the process of reducing a word to its root form. For example: processes, processing and processed are 3 different words, but it just 3 different representatives of the word “process”. Another example is evaluated and evaluation.
* Tokenization: the process of splitting text into smaller parts. Each part can consider as a feature when training in machine learning kernels. If the smaller part here is the single word (split the text to single words) then all words we have will become the bag of words.
* Corpus: You can understand simply that corpus is a collection of all features in NLP, used for training. For example, all words in the bag of words counts as a corpus. Each element in a corpus can be considered as a feature.
* Document term matrix (DTM) is a matrix, which has columns are all features of the corpus and rows is the document ID. The cross between row and column is the point for that word in that document. Points can be calculated in some different ways. In this project, I use tf-idf.
* Tf-idf: Stand for term frequency-inverse document frequency. This is a statistic method for calculating the importance of each word following this formula:

number of documents where the term t appears. If this value equal 0, it will be adjust to 1.   
D: total number of documents.

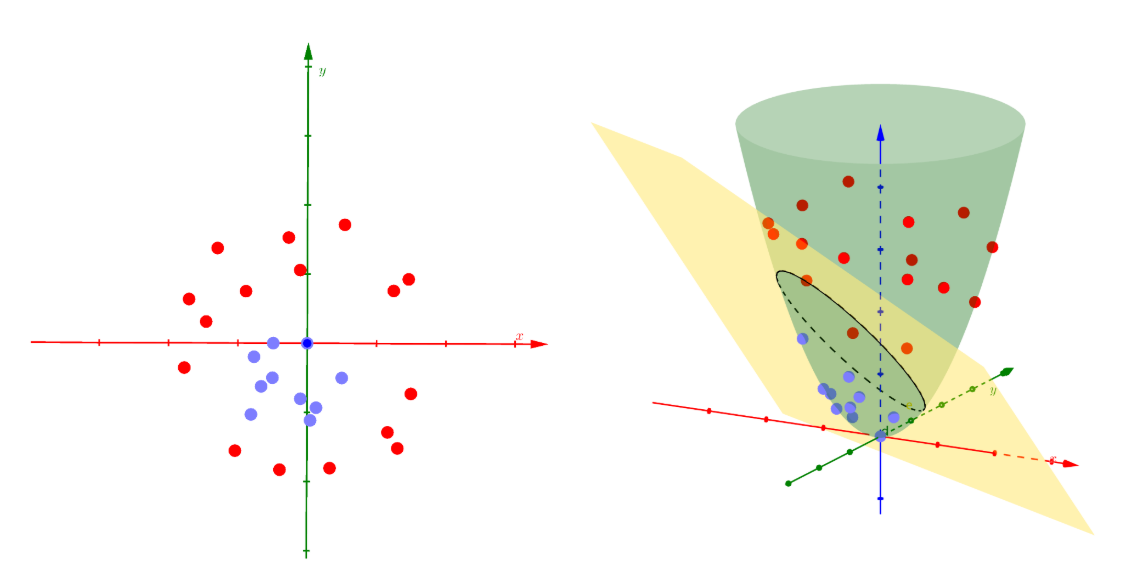
* Machine learning kernels: a kernel can understand as a core algorithm used when training a model in ML. Here are the three kernels that I use in this report:
* Naive Bayes (NB): Let initial with we have a vector X of a document, which built in document-term matrix. The probability of assign a class label y to that vector is:

Naive-Bayes theory make an assumption that all features are independent. The assumption is not realistic. Somehow, it works well.

* SVM: Stands for support vector machine. In the previous kernel, we have vector X for each document. Image that each document is represent by vector X in a multi-dimension space. (The number of dimension is the number of features). The aim is finding a hyperplane to seperate the space into two classes.
* SVM linear and RBF kenels: SVM Linear will decide the hyperplane as a flat. SVM RBF will use an other dimension to solve the problem. Let’s see an example of how linear and RBF kerels work in some pictures:



SVM linear kernel [5]



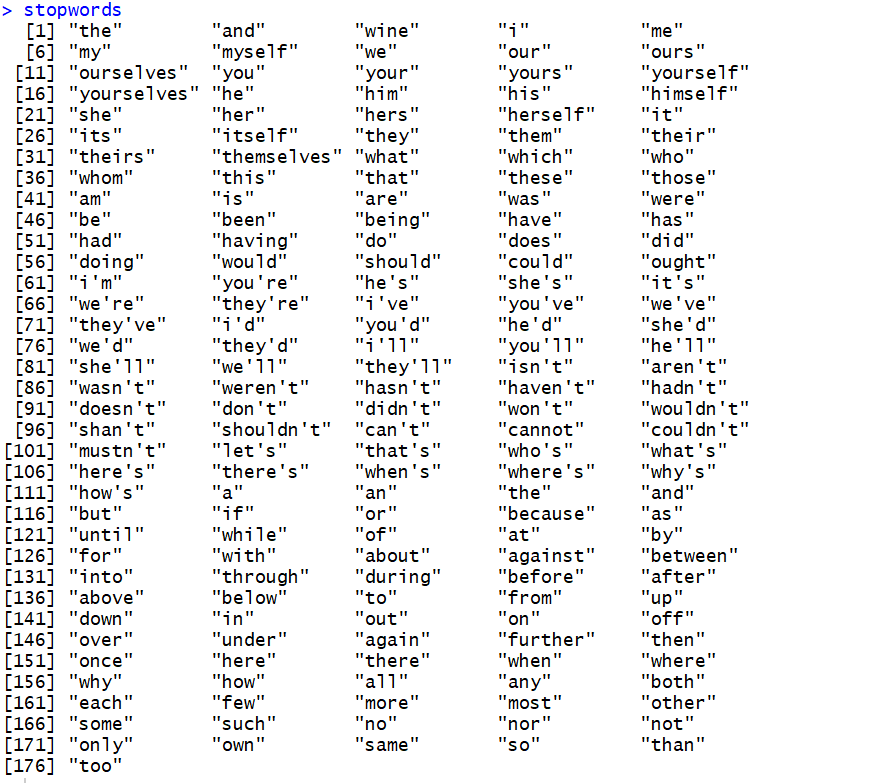
SVM RBF kernel [6]

* Ngram: n-gram is n continue sequence words. In this project, n-gram model is used for making corpus, which means features are sequence of n words. An example is n=2 continue words will be features. So, the sentence: “this is a sentence” will have “this\_is”, “is\_a”, “a\_sentence” as features.
* Part-Of-Speech Tagger (POS Tagger): This is the method to process text and decided each word is nouns, ajdtive, adverb…

# Method

In this part, I would like to discuss about the method for the first problem of this project: Which is the “good” and “excellent” wine. The second problem about wine that has “high” and “medium” value is solved using a similar method but will be discussed in details in a different part.

* Define the class: The first step is to define the class for each document. Base on the distribution of the point, I will take the point 88 as the boundary because it is the median value, which will make the data become balance. So, the wine that has higher than 88 will be the excellent wine, while the rest is the good wine.
* Preprocess data: I use 80% of the data as the training set and 20% as the testing set. The training data is processed as follow:
* Make the text ready: I decided that the information about the country, designation, province, region, and wine variety is important. So, I decided to merge that information into the description as a paragraph. That paragraph is the one that I will process.
* Language convert[3]: In the data set. There is some variety that is not unified (same type but written in the different language). So I convert it to the most famous words as follow:
* ++ Replace German names with English names: “weissburgunder” is replaced as “chardonnay”. “spatburgunder” is replaced as “pinot noir”. “grauburgunder” is replaced as “pinot gris”.   
  ++ Replace the Spanish “garnacha”" with the french “grenache”.   
  ++ Replace the Italian “pinot nero”" with the french “pinot noir”.   
  ++ Replace the Portugues “alvarinho”" with the spanish “albarino”.
* Remove non-ASCII characters.
* Remove punction .
* Make words to lower from.
* Remove number.
* Remove stop words: I use the default stop words lists in the *tm* library and then adjust it. Firstly, the lists have the word “very”. In my opinion, very is a valuable word in this data. Because in the descriptions, for example, “sweet” and “very sweet” are different levels of flavor. So, I remove that word on the list. Then, I add three words “the”, “and” and “wine” to the stop word list because, in this data set, it doesn’t have any meaning. Here is the final stop word list:



* Steamming: I steam the word by using the package SnowballC.
* Tokenizing and making the bag of words as the corpus. Then I only keep 99% spare term, which mean that only the words that appear in at least 1% of documents are kept. It makes sense because there are some rare words which only appear in one or a small number of documents. That words are not helpful for training because maybe we never meet it again in the testing set.
* Create the document terms matrix base on tf-idf.
* Now the data is ready for training. For testing data, the preprocess steps is similar, except the DTM. DTM of testing data will be create based on the corpus of training data (which means some words that don’t appear in the training data will be drop)
* Compare kernels: At this step, I’ll use the ready DTM for training model, then test with the testing DTM. Three kernel Naive-Bayes, SVM linear and SVM RBF are used to compare. Because of the limitation in my resource (Laptop core i5 7th gen, 8GB) and time. I only use the sample 20% of the data (16% - 19356 samples for training and 4% - 4839 samples for testing) to compute. Different parameters used for training is also reported. Notice that the time for training NB and SVM linear is quite fast (a few minutes) but It takes a long time for training SVM RBF (~14 hours, for 505 terms in DTM, include time for tunning model)
* Improve the best model: After comparing, the best model is selected. Then I continue to improve the model by using two methods:
* Using n-gram model.
* Adjust the weight for some terms using Part-Of-Speech Tagger (POS Tagger).

# Result and explain

## 5.1 Preprocess text

Firstly, let’s take a look at the first original paragraph and the paragraphs after preprocessing to see how it works:

train$description[1]

## [1] "This smells mostly of oak and barrel spice, with barely any fruit seeping through all the wood grain that's on display. Plump on the palate, this overoaked Chardonnay tastes almost entirely of resin and spice. Throw in clove and vanilla flavors on the finish, and you get the picture. Chile Duette Casablanca Valley Chardonnay"

wine\_train\_set[[1]]$content

## [1] "smell most oak barrel spice bare fruit seep wood grain that display plump palat overoak chardonnay tast almost entir resin spice throw clove vanilla flavor finish get pictur chile duett casablanca valley chardonnay"

Then, Here is the DTM for the first document of training set:

wine\_train\_set[1,]

## accent acid across add   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## africa aftertast age alcohol   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## almond almost along alongsid   
## 0.00000000 0.15504157 0.00000000 0.00000000   
## alreadi alsac also although   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## ampl anis anoth appeal   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## appl approach apricot argentina   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## aroma aromat around astring   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## attract australia austria back   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## bake balanc barbara barolo   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## barrel beauti berri best   
## 0.15305057 0.00000000 0.00000000 0.00000000   
## better big bit bitter   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## black blackberri blanc blend   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## blossom blue blueberri bodi   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## bold bordeaux bordeauxstyl bottl   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## bouquet boysenberri brambl bright   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## bring brisk brut burgundi   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## butter cab cabernet california   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## can candi caramel carnero   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## carri cassi catalonia cedar   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## cellar central champagn char   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## charact chardonnay cherri chewi   
## 0.00000000 0.19943433 0.00000000 0.00000000   
## chile chocol chunki cinnamon   
## 0.14711002 0.00000000 0.00000000 0.00000000   
## citrus citrusi classic classico   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## clean close clove coast   
## 0.00000000 0.00000000 0.15208806 0.00000000   
## cocoa coffe cola color   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## columbia combin come complex   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## concentr cool core counti   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## cranberri creami creek crisp   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## cru crush ctes currant   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## cut cuve dark deep   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## del delic delici deliv   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## dens depth despit develop   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## doesnt domin dri drink   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## dusti earth earthi easi   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## edg eleg element end   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## enjoy enough espresso estat   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## even excel exot express   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## extra extract fair famili   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## featur feel ferment fill   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## find fine finger finish   
## 0.00000000 0.00000000 0.00000000 0.04989804   
## firm first flavor fleshi   
## 0.00000000 0.00000000 0.02956396 0.00000000   
## floral flower focus follow   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## food foothil forest forward   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## fragrant frame franc french   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## fresh front fruit fruiti   
## 0.00000000 0.00000000 0.04106600 0.00000000   
## full fullbodi generous gentl   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## germani get give glass   
## 0.00000000 0.18456806 0.00000000 0.00000000   
## good grand grape grapefruit   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## graphit great green grenach   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## grill grip gris grown   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## hard heavi herb herbal   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## high highlight hill hint   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## hold honey honeysuckl hot   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## impress includ integr intens   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## interest intrigu invit itali   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## jam jammi juic juici   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## just keep lack lake   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## last layer lead leaf   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## lean least leather leav   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## lemon lend length les   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## licoric lift light like   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## lime linger littl live   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## load loir long lot   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## love lush made make   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## malbec mango mani mark   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## matur meat medium mediumbodi   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## melon mendoza merlot midpal   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## mild miner mint mix   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## mocha moder montalcino month   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## mountain mourvdr mouth mouthfeel   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## much napa napasonoma natur   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## nebbiolo nectarin need new   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## next nice noir north   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## northeastern northern nose note   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## now nuanc oak oaki   
## 0.00000000 0.00000000 0.08861248 0.00000000   
## offer old oliv one   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## open opul orang oregon   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## overal pack pair palat   
## 0.00000000 0.00000000 0.00000000 0.05340088   
## paso peach pear peel   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## pepper pepperi perfect perfum   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## persist petit pie piedmont   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## pineappl pinot play pleasant   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## plenti plum plump polish   
## 0.00000000 0.00000000 0.19247727 0.00000000   
## pomegran portug portugues potenti   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## power present pretti price   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## produc provid provinc prune   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## pure purpl qualiti quit   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## raci raisin ranch raspberri   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## rather readi red refresh   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## region remain reserv reserva   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## reveal rhnestyl rich riesl   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## right rioja ripe riserva   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## river roast robl ros   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## rose round russian rustic   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## sage sangioves santa sardinia   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## sauvignon savori scent seem   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## select sens set sharp   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## show sicili side sierra   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## silki simpl sip sirah   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## skin slight smell smoke   
## 0.00000000 0.00000000 0.17657168 0.00000000   
## smoki smooth soft soften   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## soil solid somewhat sonoma   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## soon sour sourc south   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## southern southwest spain sparkl   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## spice spici start still   
## 0.15574770 0.00000000 0.00000000 0.00000000   
## stone straightforward strawberri streak   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## strong structur style subtl   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## sugar suggest superior suppl   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## support sweet syrah take   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## tangerin tangi tannic tannin   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## tart tast tea tempranillo   
## 0.00000000 0.12918828 0.00000000 0.00000000   
## textur that there thick   
## 0.00000000 0.15647508 0.00000000 0.00000000   
## though tight time toast   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## toasti tobacco togeth tomato   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## tone toscana touch tropic   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## turn tuscani two underbrush   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## valley vanilla variet varieti   
## 0.05228140 0.11012388 0.00000000 0.00000000   
## velveti veneto verdot veri   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## vibrant vine vineyard vintag   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## viognier violet warm washington   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## way weight well wet   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## whiff white wild will   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## willamett wine winemak wineri   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## without wonder wood wrap   
## 0.00000000 0.00000000 0.14000828 0.00000000   
## year yellow yet york   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## young zealand zest zesti   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## zinfandel   
## 0.00000000

And let see the first document of DTM for testing set to make sure that the prepared data is correct. As we can see, the terms still remain. Just the points are different.

wine\_test\_set[1,]

## accent acid across add   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## africa aftertast age alcohol   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## almond almost along alongsid   
## 0.00000000 0.00000000 0.16027711 0.00000000   
## alreadi alsac also although   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## ampl anis anoth appeal   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## appl approach apricot argentina   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## aroma aromat around astring   
## 0.05810157 0.00000000 0.00000000 0.00000000   
## attract australia austria back   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## bake balanc barbara barolo   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## barrel beauti berri best   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## better big bit bitter   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## black blackberri blanc blend   
## 0.08717125 0.00000000 0.00000000 0.00000000   
## blossom blue blueberri bodi   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## bold bordeaux bordeauxstyl bottl   
## 0.00000000 0.00000000 0.00000000 0.14176587   
## bouquet boysenberri brambl bright   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## bring brisk brut burgundi   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## butter cab cabernet california   
## 0.00000000 0.00000000 0.00000000 0.05886771   
## can candi caramel carnero   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## carri cassi catalonia cedar   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## cellar central champagn char   
## 0.00000000 0.10262244 0.00000000 0.00000000   
## charact chardonnay cherri chewi   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## chile chocol chunki cinnamon   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## citrus citrusi classic classico   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## clean close clove coast   
## 0.00000000 0.00000000 0.00000000 0.10710576   
## cocoa coffe cola color   
## 0.00000000 0.00000000 0.00000000 0.16347060   
## columbia combin come complex   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## concentr cool core counti   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## cranberri creami creek crisp   
## 0.18309503 0.00000000 0.00000000 0.00000000   
## cru crush ctes currant   
## 0.00000000 0.19399253 0.00000000 0.00000000   
## cut cuve dark deep   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## del delic delici deliv   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## dens depth despit develop   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## doesnt domin dri drink   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## dusti earth earthi easi   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## edg eleg element end   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## enjoy enough espresso estat   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## even excel exot express   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## extra extract fair famili   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## featur feel ferment fill   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## find fine finger finish   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## firm first flavor fleshi   
## 0.00000000 0.00000000 0.03361649 0.00000000   
## floral flower focus follow   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## food foothil forest forward   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## fragrant frame franc french   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## fresh front fruit fruiti   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## full fullbodi generous gentl   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## germani get give glass   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## good grand grape grapefruit   
## 0.00000000 0.00000000 0.15286684 0.00000000   
## graphit great green grenach   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## grill grip gris grown   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## hard heavi herb herbal   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## high highlight hill hint   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## hold honey honeysuckl hot   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## impress includ integr intens   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## interest intrigu invit itali   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## jam jammi juic juici   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## just keep lack lake   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## last layer lead leaf   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## lean least leather leav   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## lemon lend length les   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## licoric lift light like   
## 0.00000000 0.00000000 0.21708525 0.00000000   
## lime linger littl live   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## load loir long lot   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## love lush made make   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## malbec mango mani mark   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## matur meat medium mediumbodi   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## melon mendoza merlot midpal   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## mild miner mint mix   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## mocha moder montalcino month   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## mountain mourvdr mouth mouthfeel   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## much napa napasonoma natur   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## nebbiolo nectarin need new   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## next nice noir north   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## northeastern northern nose note   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## now nuanc oak oaki   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## offer old oliv one   
## 0.10675293 0.20729912 0.00000000 0.00000000   
## open opul orang oregon   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## overal pack pair palat   
## 0.00000000 0.00000000 0.00000000 0.06026340   
## paso peach pear peel   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## pepper pepperi perfect perfum   
## 0.12844115 0.00000000 0.00000000 0.00000000   
## persist petit pie piedmont   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## pineappl pinot play pleasant   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## plenti plum plump polish   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## pomegran portug portugues potenti   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## power present pretti price   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## produc provid provinc prune   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## pure purpl qualiti quit   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## raci raisin ranch raspberri   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## rather readi red refresh   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## region remain reserv reserva   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## reveal rhnestyl rich riesl   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## right rioja ripe riserva   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## river roast robl ros   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## rose round russian rustic   
## 0.18517684 0.00000000 0.00000000 0.00000000   
## sage sangioves santa sardinia   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## sauvignon savori scent seem   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## select sens set sharp   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## show sicili side sierra   
## 0.09596939 0.00000000 0.00000000 0.00000000   
## silki simpl sip sirah   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## skin slight smell smoke   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## smoki smooth soft soften   
## 0.16658973 0.00000000 0.00000000 0.00000000   
## soil solid somewhat sonoma   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## soon sour sourc south   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## southern southwest spain sparkl   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## spice spici start still   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## stone straightforward strawberri streak   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## strong structur style subtl   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## sugar suggest superior suppl   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## support sweet syrah take   
## 0.00000000 0.00000000 0.27086967 0.00000000   
## tangerin tangi tannic tannin   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## tart tast tea tempranillo   
## 0.00000000 0.00000000 0.20864375 0.00000000   
## textur that there thick   
## 0.10817950 0.00000000 0.00000000 0.00000000   
## though tight time toast   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## toasti tobacco togeth tomato   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## tone toscana touch tropic   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## turn tuscani two underbrush   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## valley vanilla variet varieti   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## velveti veneto verdot veri   
## 0.00000000 0.00000000 0.00000000 0.12898784   
## vibrant vine vineyard vintag   
## 0.00000000 0.00000000 0.09436784 0.00000000   
## viognier violet warm washington   
## 0.00000000 0.19008872 0.00000000 0.00000000   
## way weight well wet   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## whiff white wild will   
## 0.00000000 0.11183840 0.00000000 0.00000000   
## willamett wine winemak wineri   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## without wonder wood wrap   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## year yellow yet york   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## young zealand zest zesti   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## zinfandel   
## 0.00000000

## 5.2 Compare kernels

Now, the data is ready. I train that dataset with three different kernels as mentioned. And here are the results:

train\_nb\_model

## Naive Bayes   
##   
## 19356 samples  
## 505 predictor  
## 2 classes: 'excellent', 'good'   
##   
## No pre-processing  
## Resampling: Bootstrapped (25 reps)   
## Summary of sample sizes: 19356, 19356, 19356, 19356, 19356, 19356, ...   
## Resampling results across tuning parameters:  
##   
## usekernel Accuracy Kappa   
## FALSE 0.7513089 0.5033206  
## TRUE 0.5757021 0.1881251  
##   
## Tuning parameter 'laplace' was held constant at a value of 0  
##   
## Tuning parameter 'adjust' was held constant at a value of 1  
## Accuracy was used to select the optimal model using the largest value.  
## The final values used for the model were laplace = 0, usekernel =  
## FALSE and adjust = 1.

confusionMatrix(conf\_nb\_train)

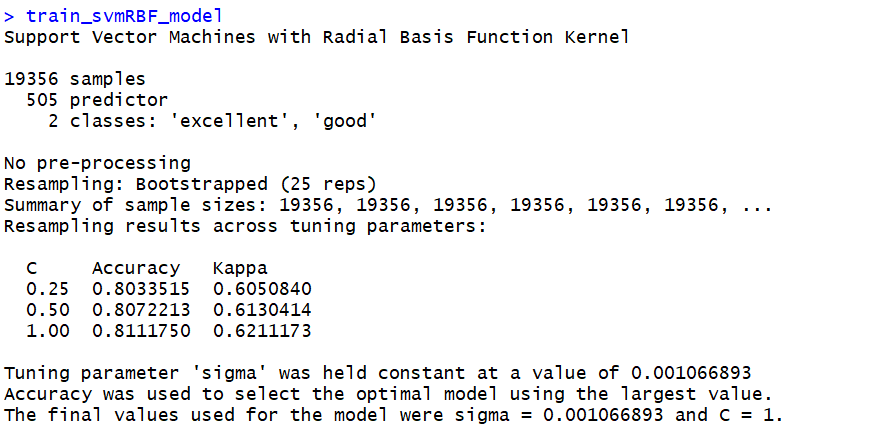
## Confusion Matrix and Statistics  
##   
## Actual class  
## Predicted class excellent good  
## excellent 1525 487  
## good 738 2089  
##   
## Accuracy : 0.7468   
## 95% CI : (0.7343, 0.7591)  
## No Information Rate : 0.5323   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.4881   
## Mcnemar's Test P-Value : 9.141e-13   
##   
## Sensitivity : 0.6739   
## Specificity : 0.8109   
## Pos Pred Value : 0.7580   
## Neg Pred Value : 0.7389   
## Prevalence : 0.4677   
## Detection Rate : 0.3151   
## Detection Prevalence : 0.4158   
## Balanced Accuracy : 0.7424   
##   
## 'Positive' Class : excellent   
##

train\_svmLinear\_model

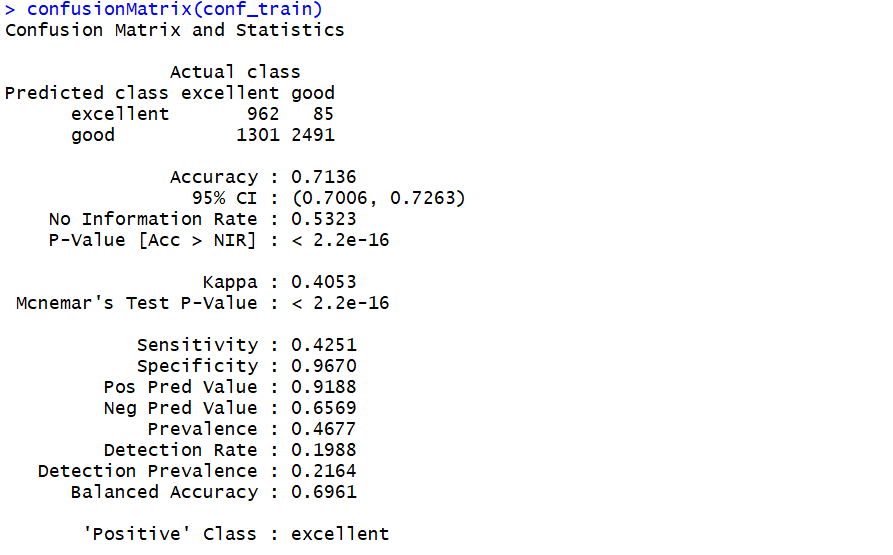
## L2 Regularized Support Vector Machine (dual) with Linear Kernel   
##   
## 19356 samples  
## 505 predictor  
## 2 classes: 'excellent', 'good'   
##   
## No pre-processing  
## Resampling: Bootstrapped (25 reps)   
## Summary of sample sizes: 19356, 19356, 19356, 19356, 19356, 19356, ...   
## Resampling results across tuning parameters:  
##   
## cost Loss Accuracy Kappa   
## 0.25 L1 0.7843732 0.5673696  
## 0.25 L2 0.7861278 0.5709545  
## 0.50 L1 0.7854490 0.5696725  
## 0.50 L2 0.7854986 0.5697548  
## 1.00 L1 0.7853391 0.5694828  
## 1.00 L2 0.7847845 0.5683523  
##   
## Accuracy was used to select the optimal model using the largest value.  
## The final values used for the model were cost = 0.25 and Loss = L2.

confusionMatrix(conf\_svmLinear\_train)

## Confusion Matrix and Statistics  
##   
## Actual class  
## Predicted class excellent good  
## excellent 1676 449  
## good 587 2127  
##   
## Accuracy : 0.7859   
## 95% CI : (0.7741, 0.7974)  
## No Information Rate : 0.5323   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.5684   
## Mcnemar's Test P-Value : 2.078e-05   
##   
## Sensitivity : 0.7406   
## Specificity : 0.8257   
## Pos Pred Value : 0.7887   
## Neg Pred Value : 0.7837   
## Prevalence : 0.4677   
## Detection Rate : 0.3464   
## Detection Prevalence : 0.4391   
## Balanced Accuracy : 0.7832   
##   
## 'Positive' Class : excellent   
##



SVMRBF kenel



As we can see in the result. The accuracy when training data for NB, SVM Linear and SVM RBF is 75.1%, 78.6%, and 81.1%. As a result, I expected that the accuracy when testing with test data will be similar. But they are 74.68% for NB, 78.59% for SVM Linear and 71.36%. SVM Linear is the kernel which has the highest value of accuracy for testing. Noticed that No information rate (NIR) 0.5323 mean that a class takes 53.23% (“good” class), which mean the data is balanced. We can judge that the models are actually worked.

But, accuracy is just one side of the story. Let see about the classification: We have two classes “good” and “excellent”. If the class is predicted exactly, it’s perfect. Obviously, excellent wine is better than good wine. So, if a good wine is predicted as excellent wine, it’s hard to accept (similar to false negative-FN). In contrast, If you pretend to buy good wine but have excellent wine. You were just lucky and nothing happened. (similar to false positive-FP)

SVM RBF kernel show the lowest number of false negative. But, I also see that the predicted result is biased to the “good” class: The number of good wine is predicted is triple as the number of excellent wine. It’s quite hard for understand. On the other hand, SVM Linear is better than NB in all indicators.

After all, I consider the accuracy, the number of each class in confusion matrix and the training time for choosing the best kernel. In my opinion, **SVM Linear is the best kernel**. (SVM RBF is interesting but it’s hard when I try to improve the model with that high training time, consider the scope of this project)

## 5.3 Improve the model

I’ll try to improve the SVM linear model with the following methods:

* Improve with n-gram: Firstly, I try to train the model with 2-gram, (the preprocess still remain) here is the first line of DTM:

wine\_train\_set[1,]

## acid\_us age\_drink alsac\_alsac   
## 0.0000000 0.0000000 0.0000000   
## appl\_pear aroma\_flavor aroma\_lead   
## 0.0000000 0.0000000 0.0000000   
## bake\_spice barolo\_nebbiolo berri\_aroma   
## 0.0000000 0.0000000 0.0000000   
## berri\_flavor berri\_fruit black\_cherri   
## 0.0000000 0.0000000 0.0000000   
## black\_currant black\_fruit black\_pepper   
## 0.0000000 0.0000000 0.0000000   
## black\_plum blackberri\_cherri blend\_cabernet   
## 0.0000000 0.0000000 0.0000000   
## bordeaux\_bordeaux bordeauxstyl\_red bright\_acid   
## 0.0000000 0.0000000 0.0000000   
## cabernet\_franc cabernet\_sauvignon california\_california   
## 0.0000000 0.0000000 0.0000000   
## california\_napa california\_paso california\_russian   
## 0.0000000 0.0000000 0.0000000   
## california\_santa california\_sonoma central\_coast   
## 0.0000000 0.0000000 0.0000000   
## central\_valley champagn\_blend champagn\_champagn   
## 0.0000000 0.0000000 0.0000000   
## cherri\_flavor cherri\_fruit cherri\_raspberri   
## 0.0000000 0.0000000 0.0000000   
## coast\_chardonnay coast\_pinot coast\_sonoma   
## 0.0000000 0.0000000 0.0000000   
## columbia\_valley counti\_central counti\_sonoma   
## 0.0000000 0.0000000 0.0000000   
## crisp\_acid ctes\_de dark\_chocol   
## 0.0000000 0.0000000 0.0000000   
## dark\_fruit di\_montalcino dri\_herb   
## 0.0000000 0.0000000 0.0000000   
## drink\_franc drink\_itali drink\_now   
## 0.0000000 0.0000000 0.0000000   
## drink\_portug estat\_california estat\_grown   
## 0.0000000 0.0000000 0.0000000   
## finger\_lake finish\_drink finish\_itali   
## 0.0000000 0.0000000 0.0000000   
## finish\_us firm\_tannin flavor\_blackberri   
## 0.0000000 0.0000000 0.0000000   
## flavor\_finish flavor\_us franc\_bordeaux   
## 0.1711132 0.0000000 0.0000000   
## french\_oak fresh\_acid fruit\_flavor   
## 0.0000000 0.0000000 0.0000000   
## full\_bodi green\_appl itali\_tuscani   
## 0.0000000 0.0000000 0.0000000   
## lake\_finger lead\_nose linger\_finish   
## 0.0000000 0.0000000 0.0000000   
## loir\_valley long\_finish medium\_bodi   
## 0.0000000 0.0000000 0.0000000   
## mendoza\_provinc montalcino\_sangioves napa\_cabernet   
## 0.0000000 0.0000000 0.0000000   
## napa\_valley new\_york new\_zealand   
## 0.0000000 0.0000000 0.0000000   
## northeastern\_itali northern\_spain nose\_palat   
## 0.0000000 0.0000000 0.0000000   
## now\_franc now\_us old\_vine   
## 0.0000000 0.0000000 0.0000000   
## open\_aroma oregon\_willamett palat\_deliv   
## 0.0000000 0.0000000 0.0000000   
## palat\_offer palat\_show paso\_robl   
## 0.0000000 0.0000000 0.0000000   
## petit\_sirah petit\_verdot piedmont\_barolo   
## 0.0000000 0.0000000 0.0000000   
## pinot\_gris pinot\_noir portugues\_red   
## 0.0000000 0.0000000 0.0000000   
## provinc\_mendoza raspberri\_cherri readi\_drink   
## 0.0000000 0.0000000 0.0000000   
## red\_berri red\_blend red\_cherri   
## 0.0000000 0.0000000 0.0000000   
## red\_currant red\_fruit reserv\_california   
## 0.0000000 0.0000000 0.0000000   
## rhnestyl\_red ripe\_fruit river\_valley   
## 0.0000000 0.0000000 0.0000000   
## robl\_central russian\_river santa\_barbara   
## 0.0000000 0.0000000 0.0000000   
## sauvignon\_blanc sicili\_sardinia sierra\_foothil   
## 0.0000000 0.0000000 0.0000000   
## sonoma\_chardonnay sonoma\_coast sonoma\_counti   
## 0.0000000 0.0000000 0.0000000   
## sonoma\_pinot south\_africa south\_australia   
## 0.0000000 0.0000000 0.0000000   
## southwest\_franc spain\_rioja sparkl\_blend   
## 0.0000000 0.0000000 0.0000000   
## spice\_flavor stone\_fruit tannin\_drink   
## 0.0000000 0.0000000 0.0000000   
## tropic\_fruit us\_california us\_estat   
## 0.0000000 0.0000000 0.0000000   
## us\_oregon us\_reserv us\_washington   
## 0.0000000 0.0000000 0.0000000   
## valley\_cabernet valley\_central valley\_chardonnay   
## 0.0000000 0.0000000 0.1922822   
## valley\_napa valley\_pinot valley\_red   
## 0.0000000 0.0000000 0.0000000   
## valley\_sonoma valley\_syrah valley\_wa   
## 0.0000000 0.0000000 0.0000000   
## valley\_willamett vineyard\_california vineyard\_washington   
## 0.0000000 0.0000000 0.0000000   
## wa\_columbia washington\_columbia white\_blend   
## 0.0000000 0.0000000 0.0000000   
## white\_peach white\_pepper willamett\_valley   
## 0.0000000 0.0000000 0.0000000   
## wood\_age year\_us york\_finger   
## 0.0000000 0.0000000 0.0000000

Here are the results:

train\_svmLinear\_model

## L2 Regularized Support Vector Machine (dual) with Linear Kernel   
##   
## 19356 samples  
## 156 predictor  
## 2 classes: 'excellent', 'good'   
##   
## No pre-processing  
## Resampling: Bootstrapped (25 reps)   
## Summary of sample sizes: 19356, 19356, 19356, 19356, 19356, 19356, ...   
## Resampling results across tuning parameters:  
##   
## cost Loss Accuracy Kappa   
## 0.25 L1 0.6436272 0.2823927  
## 0.25 L2 0.6494504 0.2968135  
## 0.50 L1 0.6455226 0.2896008  
## 0.50 L2 0.6498153 0.2976152  
## 1.00 L1 0.6466283 0.2928574  
## 1.00 L2 0.6499836 0.2979705  
##   
## Accuracy was used to select the optimal model using the largest value.  
## The final values used for the model were cost = 1 and Loss = L2.

confusionMatrix(conf\_svmLinear\_train)

## Confusion Matrix and Statistics  
##   
## Actual class  
## Predicted class excellent good  
## excellent 1200 668  
## good 1063 1908  
##   
## Accuracy : 0.6423   
## 95% CI : (0.6286, 0.6558)  
## No Information Rate : 0.5323   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.2739   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.5303   
## Specificity : 0.7407   
## Pos Pred Value : 0.6424   
## Neg Pred Value : 0.6422   
## Prevalence : 0.4677   
## Detection Rate : 0.2480   
## Detection Prevalence : 0.3860   
## Balanced Accuracy : 0.6355   
##   
## 'Positive' Class : excellent   
##

The result is worse than the previous in all indicators, which means that 2-gram is not useful. Based on my observation. After preprocess and steaming, the descriptions are quite discrete. I mean the words are likely not connected to each other. It leads to the 2-gram model does not work.

I continue with both bag of words and 2-gram (which means 1-2gram). Here is the first line of DTM:

wine\_train\_set\_ngram[1,]

## accent acid acid\_us   
## 0.00000000 0.00000000 0.00000000   
## across add africa   
## 0.00000000 0.00000000 0.00000000   
## aftertast age age\_drink   
## 0.00000000 0.00000000 0.00000000   
## alcohol almond almost   
## 0.00000000 0.00000000 0.07871341   
## along alongsid alreadi   
## 0.00000000 0.00000000 0.00000000   
## alsac alsac\_alsac also   
## 0.00000000 0.00000000 0.00000000   
## although ampl anis   
## 0.00000000 0.00000000 0.00000000   
## anoth appeal appl   
## 0.00000000 0.00000000 0.00000000   
## appl\_pear approach apricot   
## 0.00000000 0.00000000 0.00000000   
## argentina aroma aroma\_flavor   
## 0.00000000 0.00000000 0.00000000   
## aroma\_lead aromat around   
## 0.00000000 0.00000000 0.00000000   
## astring attract australia   
## 0.00000000 0.00000000 0.00000000   
## austria back bake   
## 0.00000000 0.00000000 0.00000000   
## bake\_spice balanc barbara   
## 0.00000000 0.00000000 0.00000000   
## barolo barolo\_nebbiolo barrel   
## 0.00000000 0.00000000 0.07770259   
## beauti berri berri\_aroma   
## 0.00000000 0.00000000 0.00000000   
## berri\_flavor berri\_fruit best   
## 0.00000000 0.00000000 0.00000000   
## better big bit   
## 0.00000000 0.00000000 0.00000000   
## bitter black black\_cherri   
## 0.00000000 0.00000000 0.00000000   
## black\_currant black\_fruit black\_pepper   
## 0.00000000 0.00000000 0.00000000   
## black\_plum blackberri blackberri\_cherri   
## 0.00000000 0.00000000 0.00000000   
## blanc blend blend\_cabernet   
## 0.00000000 0.00000000 0.00000000   
## blossom blue blueberri   
## 0.00000000 0.00000000 0.00000000   
## bodi bold bordeaux   
## 0.00000000 0.00000000 0.00000000   
## bordeaux\_bordeaux bordeauxstyl bordeauxstyl\_red   
## 0.00000000 0.00000000 0.00000000   
## bottl bouquet boysenberri   
## 0.00000000 0.00000000 0.00000000   
## brambl bright bright\_acid   
## 0.00000000 0.00000000 0.00000000   
## bring brisk brut   
## 0.00000000 0.00000000 0.00000000   
## burgundi butter cab   
## 0.00000000 0.00000000 0.00000000   
## cabernet cabernet\_franc cabernet\_sauvignon   
## 0.00000000 0.00000000 0.00000000   
## california california\_california california\_napa   
## 0.00000000 0.00000000 0.00000000   
## california\_paso california\_russian california\_santa   
## 0.00000000 0.00000000 0.00000000   
## california\_sonoma can candi   
## 0.00000000 0.00000000 0.00000000   
## caramel carnero carri   
## 0.00000000 0.00000000 0.00000000   
## cassi catalonia cedar   
## 0.00000000 0.00000000 0.00000000   
## cellar central central\_coast   
## 0.00000000 0.00000000 0.00000000   
## central\_valley champagn champagn\_blend   
## 0.00000000 0.00000000 0.00000000   
## champagn\_champagn char charact   
## 0.00000000 0.00000000 0.00000000   
## chardonnay cherri cherri\_flavor   
## 0.10125128 0.00000000 0.00000000   
## cherri\_fruit cherri\_raspberri chewi   
## 0.00000000 0.00000000 0.00000000   
## chile chocol chunki   
## 0.07468663 0.00000000 0.00000000   
## cinnamon citrus citrusi   
## 0.00000000 0.00000000 0.00000000   
## classic classico clean   
## 0.00000000 0.00000000 0.00000000   
## close clove coast   
## 0.00000000 0.07721394 0.00000000   
## coast\_chardonnay coast\_pinot coast\_sonoma   
## 0.00000000 0.00000000 0.00000000   
## cocoa coffe cola   
## 0.00000000 0.00000000 0.00000000   
## color columbia columbia\_valley   
## 0.00000000 0.00000000 0.00000000   
## combin come complex   
## 0.00000000 0.00000000 0.00000000   
## concentr cool core   
## 0.00000000 0.00000000 0.00000000   
## counti counti\_central counti\_sonoma   
## 0.00000000 0.00000000 0.00000000   
## cranberri creami creek   
## 0.00000000 0.00000000 0.00000000   
## crisp crisp\_acid cru   
## 0.00000000 0.00000000 0.00000000   
## crush ctes ctes\_de   
## 0.00000000 0.00000000 0.00000000   
## currant cut cuve   
## 0.00000000 0.00000000 0.00000000   
## dark dark\_chocol dark\_fruit   
## 0.00000000 0.00000000 0.00000000   
## deep del delic   
## 0.00000000 0.00000000 0.00000000   
## delici deliv dens   
## 0.00000000 0.00000000 0.00000000   
## depth despit develop   
## 0.00000000 0.00000000 0.00000000   
## di\_montalcino doesnt domin   
## 0.00000000 0.00000000 0.00000000   
## dri dri\_herb drink   
## 0.00000000 0.00000000 0.00000000   
## drink\_franc drink\_itali drink\_now   
## 0.00000000 0.00000000 0.00000000   
## drink\_portug dusti earth   
## 0.00000000 0.00000000 0.00000000   
## earthi easi edg   
## 0.00000000 0.00000000 0.00000000   
## eleg element end   
## 0.00000000 0.00000000 0.00000000   
## enjoy enough espresso   
## 0.00000000 0.00000000 0.00000000   
## estat estat\_california estat\_grown   
## 0.00000000 0.00000000 0.00000000   
## even excel exot   
## 0.00000000 0.00000000 0.00000000   
## express extra extract   
## 0.00000000 0.00000000 0.00000000   
## fair famili featur   
## 0.00000000 0.00000000 0.00000000   
## feel ferment fill   
## 0.00000000 0.00000000 0.00000000   
## find fine finger   
## 0.00000000 0.00000000 0.00000000   
## finger\_lake finish finish\_drink   
## 0.00000000 0.02533285 0.00000000   
## finish\_itali finish\_us firm   
## 0.00000000 0.00000000 0.00000000   
## firm\_tannin first flavor   
## 0.00000000 0.00000000 0.01500940   
## flavor\_blackberri flavor\_finish flavor\_us   
## 0.00000000 0.08424033 0.00000000   
## fleshi floral flower   
## 0.00000000 0.00000000 0.00000000   
## focus follow food   
## 0.00000000 0.00000000 0.00000000   
## foothil forest forward   
## 0.00000000 0.00000000 0.00000000   
## fragrant frame franc   
## 0.00000000 0.00000000 0.00000000   
## franc\_bordeaux french french\_oak   
## 0.00000000 0.00000000 0.00000000   
## fresh fresh\_acid front   
## 0.00000000 0.00000000 0.00000000   
## fruit fruit\_flavor fruiti   
## 0.02084889 0.00000000 0.00000000   
## full full\_bodi fullbodi   
## 0.00000000 0.00000000 0.00000000   
## generous gentl germani   
## 0.00000000 0.00000000 0.00000000   
## get give glass   
## 0.09370379 0.00000000 0.00000000   
## good grand grape   
## 0.00000000 0.00000000 0.00000000   
## grapefruit graphit great   
## 0.00000000 0.00000000 0.00000000   
## green green\_appl grenach   
## 0.00000000 0.00000000 0.00000000   
## grill grip gris   
## 0.00000000 0.00000000 0.00000000   
## grown hard heavi   
## 0.00000000 0.00000000 0.00000000   
## herb herbal high   
## 0.00000000 0.00000000 0.00000000   
## highlight hill hint   
## 0.00000000 0.00000000 0.00000000   
## hold honey honeysuckl   
## 0.00000000 0.00000000 0.00000000   
## hot impress includ   
## 0.00000000 0.00000000 0.00000000   
## integr intens interest   
## 0.00000000 0.00000000 0.00000000   
## intrigu invit itali   
## 0.00000000 0.00000000 0.00000000   
## itali\_tuscani jam jammi   
## 0.00000000 0.00000000 0.00000000   
## juic juici just   
## 0.00000000 0.00000000 0.00000000   
## keep lack lake   
## 0.00000000 0.00000000 0.00000000   
## lake\_finger last layer   
## 0.00000000 0.00000000 0.00000000   
## lead lead\_nose leaf   
## 0.00000000 0.00000000 0.00000000   
## lean least leather   
## 0.00000000 0.00000000 0.00000000   
## leav lemon lend   
## 0.00000000 0.00000000 0.00000000   
## length les licoric   
## 0.00000000 0.00000000 0.00000000   
## lift light like   
## 0.00000000 0.00000000 0.00000000   
## lime linger linger\_finish   
## 0.00000000 0.00000000 0.00000000   
## littl live load   
## 0.00000000 0.00000000 0.00000000   
## loir loir\_valley long   
## 0.00000000 0.00000000 0.00000000   
## long\_finish lot love   
## 0.00000000 0.00000000 0.00000000   
## lush made make   
## 0.00000000 0.00000000 0.00000000   
## malbec mango mani   
## 0.00000000 0.00000000 0.00000000   
## mark matur meat   
## 0.00000000 0.00000000 0.00000000   
## medium medium\_bodi mediumbodi   
## 0.00000000 0.00000000 0.00000000   
## melon mendoza mendoza\_provinc   
## 0.00000000 0.00000000 0.00000000   
## merlot midpal mild   
## 0.00000000 0.00000000 0.00000000   
## miner mint mix   
## 0.00000000 0.00000000 0.00000000   
## mocha moder montalcino   
## 0.00000000 0.00000000 0.00000000   
## montalcino\_sangioves month mountain   
## 0.00000000 0.00000000 0.00000000   
## mourvdr mouth mouthfeel   
## 0.00000000 0.00000000 0.00000000   
## much napa napa\_cabernet   
## 0.00000000 0.00000000 0.00000000   
## napa\_valley napasonoma natur   
## 0.00000000 0.00000000 0.00000000   
## nebbiolo nectarin need   
## 0.00000000 0.00000000 0.00000000   
## new new\_york new\_zealand   
## 0.00000000 0.00000000 0.00000000   
## next nice noir   
## 0.00000000 0.00000000 0.00000000   
## north northeastern northeastern\_itali   
## 0.00000000 0.00000000 0.00000000   
## northern northern\_spain nose   
## 0.00000000 0.00000000 0.00000000   
## nose\_palat note now   
## 0.00000000 0.00000000 0.00000000   
## now\_franc now\_us nuanc   
## 0.00000000 0.00000000 0.00000000   
## oak oaki offer   
## 0.04498787 0.00000000 0.00000000   
## old old\_vine oliv   
## 0.00000000 0.00000000 0.00000000   
## one open open\_aroma   
## 0.00000000 0.00000000 0.00000000   
## opul orang oregon   
## 0.00000000 0.00000000 0.00000000   
## oregon\_willamett overal pack   
## 0.00000000 0.00000000 0.00000000   
## pair palat palat\_deliv   
## 0.00000000 0.02711122 0.00000000   
## palat\_offer palat\_show paso   
## 0.00000000 0.00000000 0.00000000   
## paso\_robl peach pear   
## 0.00000000 0.00000000 0.00000000   
## peel pepper pepperi   
## 0.00000000 0.00000000 0.00000000   
## perfect perfum persist   
## 0.00000000 0.00000000 0.00000000   
## petit petit\_sirah petit\_verdot   
## 0.00000000 0.00000000 0.00000000   
## pie piedmont piedmont\_barolo   
## 0.00000000 0.00000000 0.00000000   
## pineappl pinot pinot\_gris   
## 0.00000000 0.00000000 0.00000000   
## pinot\_noir play pleasant   
## 0.00000000 0.00000000 0.00000000   
## plenti plum plump   
## 0.00000000 0.00000000 0.09771923   
## polish pomegran portug   
## 0.00000000 0.00000000 0.00000000   
## portugues portugues\_red potenti   
## 0.00000000 0.00000000 0.00000000   
## power present pretti   
## 0.00000000 0.00000000 0.00000000   
## price produc provid   
## 0.00000000 0.00000000 0.00000000   
## provinc provinc\_mendoza prune   
## 0.00000000 0.00000000 0.00000000   
## pure purpl qualiti   
## 0.00000000 0.00000000 0.00000000   
## quit raci raisin   
## 0.00000000 0.00000000 0.00000000   
## ranch raspberri raspberri\_cherri   
## 0.00000000 0.00000000 0.00000000   
## rather readi readi\_drink   
## 0.00000000 0.00000000 0.00000000   
## red red\_berri red\_blend   
## 0.00000000 0.00000000 0.00000000   
## red\_cherri red\_currant red\_fruit   
## 0.00000000 0.00000000 0.00000000   
## refresh region remain   
## 0.00000000 0.00000000 0.00000000   
## reserv reserv\_california reserva   
## 0.00000000 0.00000000 0.00000000   
## reveal rhnestyl rhnestyl\_red   
## 0.00000000 0.00000000 0.00000000   
## rich riesl right   
## 0.00000000 0.00000000 0.00000000   
## rioja ripe ripe\_fruit   
## 0.00000000 0.00000000 0.00000000   
## riserva river river\_valley   
## 0.00000000 0.00000000 0.00000000   
## roast robl robl\_central   
## 0.00000000 0.00000000 0.00000000   
## ros rose round   
## 0.00000000 0.00000000 0.00000000   
## russian russian\_river rustic   
## 0.00000000 0.00000000 0.00000000   
## sage sangioves santa   
## 0.00000000 0.00000000 0.00000000   
## santa\_barbara sardinia sauvignon   
## 0.00000000 0.00000000 0.00000000   
## sauvignon\_blanc savori scent   
## 0.00000000 0.00000000 0.00000000   
## seem select sens   
## 0.00000000 0.00000000 0.00000000   
## set sharp show   
## 0.00000000 0.00000000 0.00000000   
## sicili sicili\_sardinia side   
## 0.00000000 0.00000000 0.00000000   
## sierra sierra\_foothil silki   
## 0.00000000 0.00000000 0.00000000   
## simpl sip sirah   
## 0.00000000 0.00000000 0.00000000   
## skin slight smell   
## 0.00000000 0.00000000 0.08964408   
## smoke smoki smooth   
## 0.00000000 0.00000000 0.00000000   
## soft soften soil   
## 0.00000000 0.00000000 0.00000000   
## solid somewhat sonoma   
## 0.00000000 0.00000000 0.00000000   
## sonoma\_chardonnay sonoma\_coast sonoma\_counti   
## 0.00000000 0.00000000 0.00000000   
## sonoma\_pinot soon sour   
## 0.00000000 0.00000000 0.00000000   
## sourc south south\_africa   
## 0.00000000 0.00000000 0.00000000   
## south\_australia southern southwest   
## 0.00000000 0.00000000 0.00000000   
## southwest\_franc spain spain\_rioja   
## 0.00000000 0.00000000 0.00000000   
## sparkl sparkl\_blend spice   
## 0.00000000 0.00000000 0.07907191   
## spice\_flavor spici start   
## 0.00000000 0.00000000 0.00000000   
## still stone stone\_fruit   
## 0.00000000 0.00000000 0.00000000   
## straightforward strawberri streak   
## 0.00000000 0.00000000 0.00000000   
## strong structur style   
## 0.00000000 0.00000000 0.00000000   
## subtl sugar suggest   
## 0.00000000 0.00000000 0.00000000   
## superior suppl support   
## 0.00000000 0.00000000 0.00000000   
## sweet syrah take   
## 0.00000000 0.00000000 0.00000000   
## tangerin tangi tannic   
## 0.00000000 0.00000000 0.00000000   
## tannin tannin\_drink tart   
## 0.00000000 0.00000000 0.00000000   
## tast tea tempranillo   
## 0.06558790 0.00000000 0.00000000   
## textur that there   
## 0.00000000 0.07944119 0.00000000   
## thick though tight   
## 0.00000000 0.00000000 0.00000000   
## time toast toasti   
## 0.00000000 0.00000000 0.00000000   
## tobacco togeth tomato   
## 0.00000000 0.00000000 0.00000000   
## tone toscana touch   
## 0.00000000 0.00000000 0.00000000   
## tropic tropic\_fruit turn   
## 0.00000000 0.00000000 0.00000000   
## tuscani two underbrush   
## 0.00000000 0.00000000 0.00000000   
## us\_california us\_estat us\_oregon   
## 0.00000000 0.00000000 0.00000000   
## us\_reserv us\_washington valley   
## 0.00000000 0.00000000 0.02654286   
## valley\_cabernet valley\_central valley\_chardonnay   
## 0.00000000 0.00000000 0.09466201   
## valley\_napa valley\_pinot valley\_red   
## 0.00000000 0.00000000 0.00000000   
## valley\_sonoma valley\_syrah valley\_wa   
## 0.00000000 0.00000000 0.00000000   
## valley\_willamett vanilla variet   
## 0.00000000 0.05590905 0.00000000   
## varieti velveti veneto   
## 0.00000000 0.00000000 0.00000000   
## verdot veri vibrant   
## 0.00000000 0.00000000 0.00000000   
## vine vineyard vineyard\_california   
## 0.00000000 0.00000000 0.00000000   
## vineyard\_washington vintag viognier   
## 0.00000000 0.00000000 0.00000000   
## violet wa\_columbia warm   
## 0.00000000 0.00000000 0.00000000   
## washington washington\_columbia way   
## 0.00000000 0.00000000 0.00000000   
## weight well wet   
## 0.00000000 0.00000000 0.00000000   
## whiff white white\_blend   
## 0.00000000 0.00000000 0.00000000   
## white\_peach white\_pepper wild   
## 0.00000000 0.00000000 0.00000000   
## will willamett willamett\_valley   
## 0.00000000 0.00000000 0.00000000   
## wine winemak wineri   
## 0.00000000 0.00000000 0.00000000   
## without wonder wood   
## 0.00000000 0.00000000 0.07108113   
## wood\_age wrap year   
## 0.00000000 0.00000000 0.00000000   
## year\_us yellow yet   
## 0.00000000 0.00000000 0.00000000   
## york york\_finger young   
## 0.00000000 0.00000000 0.00000000   
## zealand zest zesti   
## 0.00000000 0.00000000 0.00000000   
## zinfandel   
## 0.00000000

And here are the results:

train\_svmLinear\_model

## L2 Regularized Support Vector Machine (dual) with Linear Kernel   
##   
## 19356 samples  
## 661 predictor  
## 2 classes: 'excellent', 'good'   
##   
## No pre-processing  
## Resampling: Bootstrapped (25 reps)   
## Summary of sample sizes: 19356, 19356, 19356, 19356, 19356, 19356, ...   
## Resampling results across tuning parameters:  
##   
## cost Loss Accuracy Kappa   
## 0.25 L1 0.7804824 0.5592447  
## 0.25 L2 0.7903403 0.5793207  
## 0.50 L1 0.7862217 0.5711220  
## 0.50 L2 0.7911250 0.5810424  
## 1.00 L1 0.7893797 0.5776108  
## 1.00 L2 0.7902474 0.5793807  
##   
## Accuracy was used to select the optimal model using the largest value.  
## The final values used for the model were cost = 0.5 and Loss = L2.

confusionMatrix(conf\_svmLinear\_train)

## Confusion Matrix and Statistics  
##   
## Actual class  
## Predicted class excellent good  
## excellent 1603 371  
## good 660 2205  
##   
## Accuracy : 0.7869   
## 95% CI : (0.7751, 0.7984)  
## No Information Rate : 0.5323   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.5687   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.7084   
## Specificity : 0.8560   
## Pos Pred Value : 0.8121   
## Neg Pred Value : 0.7696   
## Prevalence : 0.4677   
## Detection Rate : 0.3313   
## Detection Prevalence : 0.4079   
## Balanced Accuracy : 0.7822   
##   
## 'Positive' Class : excellent   
##

The accuracy is just slightly increased. The good news is the false negative is really better (371 compare to 448) of course the false positive shows an increase.

To explain, I see that there are some important pairs that impove the model. Considering those results and the dimension of DTM is large (which lead to high training time), while there are too many features with 0 point. I conclude that 1-2 gram is a little bit better.

The second way to improve is adjusted the weight for DTM (with bag of words). As I observed in the data. The description usually talk about ingredients: lemon, cherry or region: California… So, the first idea is double the point for nouns and see how it works. On the other hand, Adjectives used to describe the flavor of wine could act an important role. So, the second idea is double the point for adjectives only. To do these, I implement POS tagger and see which word is nouns, adj, verb…

Here is the noun list that I extracted:

nouns

## [1] "accent" "acid" "age" "alcohol"   
## [5] "almond" "alreadi" "alsac" "appeal"   
## [9] "approach" "apricot" "argentina" "aroma"   
## [13] "austria" "bake" "balanc" "barbara"   
## [17] "barolo" "barrel" "beauti" "berri"   
## [21] "bit" "blanc" "blend" "bodi"   
## [25] "bordeaux" "bordeauxstyl" "bottl" "bouquet"   
## [29] "boysenberri" "brambl" "brut" "butter"   
## [33] "cab" "cabernet" "california" "caramel"   
## [37] "carnero" "carri" "cassi" "catalonia"   
## [41] "cedar" "cellar" "champagn" "char"   
## [45] "charact" "chardonnay" "cherri" "chewi"   
## [49] "chile" "chocol" "chunki" "cinnamon"   
## [53] "citrus" "classico" "clove" "coast"   
## [57] "cocoa" "coffe" "cola" "color"   
## [61] "columbia" "combin" "concentr" "core"   
## [65] "creek" "crisp" "cru" "crush"   
## [69] "cut" "delic" "depth" "despit"   
## [73] "domin" "dri" "drink" "earth"   
## [77] "earthi" "eleg" "element" "end"   
## [81] "espresso" "estat" "express" "ferment"   
## [85] "fill" "finger" "firm" "flavor"   
## [89] "flower" "focus" "food" "foothil"   
## [93] "forest" "frame" "franc" "french"   
## [97] "front" "fruit" "fruiti" "gentl"   
## [101] "glass" "grape" "grapefruit" "grenach"   
## [105] "grill" "grip" "herb" "high"   
## [109] "highlight" "hill" "hint" "honey"   
## [113] "integr" "interest" "invit" "jam"   
## [117] "jammi" "lack" "lake" "layer"   
## [121] "lead" "leaf" "leather" "leav"   
## [125] "lemon" "length" "lift" "light"   
## [129] "lime" "linger" "load" "mango"   
## [133] "mark" "matur" "meat" "medium"   
## [137] "melon" "mendoza" "merlot" "midpal"   
## [141] "miner" "mint" "mix" "mocha"   
## [145] "moder" "month" "mountain" "mourvdr"   
## [149] "mouth" "mouthfeel" "napa" "napasonoma"   
## [153] "natur" "nebbiolo" "nectarin" "noir"   
## [157] "north" "nose" "note" "oak"   
## [161] "oaki" "one" "opul" "oregon"   
## [165] "pack" "pair" "palat" "paso"   
## [169] "peach" "pear" "peel" "pepper"   
## [173] "pepperi" "perfum" "pie" "piedmont"   
## [177] "pineappl" "play" "plum" "plump"   
## [181] "polish" "pomegran" "portug" "power"   
## [185] "present" "price" "produc" "provinc"   
## [189] "prune" "purpl" "qualiti" "ranch"   
## [193] "raspberri" "region" "reserva" "rhnestyl"   
## [197] "rioja" "riserva" "river" "roast"   
## [201] "sage" "sardinia" "sauvignon" "scent"   
## [205] "show" "side" "sierra" "silki"   
## [209] "sip" "skin" "smell" "smoke"   
## [213] "soil" "sonoma" "sourc" "southwest"   
## [217] "spain" "sparkl" "spice" "stone"   
## [221] "streak" "structur" "style" "sugar"   
## [225] "support" "syrah" "tangerin" "tannin"   
## [229] "tart" "tast" "tea" "textur"   
## [233] "time" "toast" "toasti" "tobacco"   
## [237] "tone" "touch" "tropic" "turn"   
## [241] "valley" "vanilla" "varieti" "velveti"   
## [245] "veneto" "verdot" "veri" "vine"   
## [249] "vineyard" "vintag" "viognier" "violet"   
## [253] "way" "weight" "will" "wine"   
## [257] "winemak" "wineri" "wonder" "wood"   
## [261] "wrap" "year" "yellow" "zealand"   
## [265] "zest" "zesti" "zinfandel"

Let’s take a look at the Dtm of the first document to make sure that the points for nouns is doubled. As we can see below, the points for the word “california” is 0.139. It’s just 0.064 before.

wine\_train\_set[1,]

## accent acid across add   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## africa aftertast age alcohol   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## almond almost along alongsid   
## 0.00000000 0.15504157 0.00000000 0.00000000   
## alreadi alsac also although   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## ampl anis anoth appeal   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## appl approach apricot argentina   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## aroma aromat around astring   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## attract australia austria back   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## bake balanc barbara barolo   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## barrel beauti berri best   
## 0.30610113 0.00000000 0.00000000 0.00000000   
## better big bit bitter   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## black blackberri blanc blend   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## blossom blue blueberri bodi   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## bold bordeaux bordeauxstyl bottl   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## bouquet boysenberri brambl bright   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## bring brisk brut burgundi   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## butter cab cabernet california   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## can candi caramel carnero   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## carri cassi catalonia cedar   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## cellar central champagn char   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## charact chardonnay cherri chewi   
## 0.00000000 0.39886867 0.00000000 0.00000000   
## chile chocol chunki cinnamon   
## 0.29422005 0.00000000 0.00000000 0.00000000   
## citrus citrusi classic classico   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## clean close clove coast   
## 0.00000000 0.00000000 0.30417613 0.00000000   
## cocoa coffe cola color   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## columbia combin come complex   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## concentr cool core counti   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## cranberri creami creek crisp   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## cru crush ctes currant   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## cut cuve dark deep   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## del delic delici deliv   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## dens depth despit develop   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## doesnt domin dri drink   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## dusti earth earthi easi   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## edg eleg element end   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## enjoy enough espresso estat   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## even excel exot express   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## extra extract fair famili   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## featur feel ferment fill   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## find fine finger finish   
## 0.00000000 0.00000000 0.00000000 0.04989804   
## firm first flavor fleshi   
## 0.00000000 0.00000000 0.05912793 0.00000000   
## floral flower focus follow   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## food foothil forest forward   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## fragrant frame franc french   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## fresh front fruit fruiti   
## 0.00000000 0.00000000 0.08213201 0.00000000   
## full fullbodi generous gentl   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## germani get give glass   
## 0.00000000 0.18456806 0.00000000 0.00000000   
## good grand grape grapefruit   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## graphit great green grenach   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## grill grip gris grown   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## hard heavi herb herbal   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## high highlight hill hint   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## hold honey honeysuckl hot   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## impress includ integr intens   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## interest intrigu invit itali   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## jam jammi juic juici   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## just keep lack lake   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## last layer lead leaf   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## lean least leather leav   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## lemon lend length les   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## licoric lift light like   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## lime linger littl live   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## load loir long lot   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## love lush made make   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## malbec mango mani mark   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## matur meat medium mediumbodi   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## melon mendoza merlot midpal   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## mild miner mint mix   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## mocha moder montalcino month   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## mountain mourvdr mouth mouthfeel   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## much napa napasonoma natur   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## nebbiolo nectarin need new   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## next nice noir north   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## northeastern northern nose note   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## now nuanc oak oaki   
## 0.00000000 0.00000000 0.17722495 0.00000000   
## offer old oliv one   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## open opul orang oregon   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## overal pack pair palat   
## 0.00000000 0.00000000 0.00000000 0.10680177   
## paso peach pear peel   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## pepper pepperi perfect perfum   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## persist petit pie piedmont   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## pineappl pinot play pleasant   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## plenti plum plump polish   
## 0.00000000 0.00000000 0.38495454 0.00000000   
## pomegran portug portugues potenti   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## power present pretti price   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## produc provid provinc prune   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## pure purpl qualiti quit   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## raci raisin ranch raspberri   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## rather readi red refresh   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## region remain reserv reserva   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## reveal rhnestyl rich riesl   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## right rioja ripe riserva   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## river roast robl ros   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## rose round russian rustic   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## sage sangioves santa sardinia   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## sauvignon savori scent seem   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## select sens set sharp   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## show sicili side sierra   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## silki simpl sip sirah   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## skin slight smell smoke   
## 0.00000000 0.00000000 0.35314335 0.00000000   
## smoki smooth soft soften   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## soil solid somewhat sonoma   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## soon sour sourc south   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## southern southwest spain sparkl   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## spice spici start still   
## 0.31149539 0.00000000 0.00000000 0.00000000   
## stone straightforward strawberri streak   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## strong structur style subtl   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## sugar suggest superior suppl   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## support sweet syrah take   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## tangerin tangi tannic tannin   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## tart tast tea tempranillo   
## 0.00000000 0.25837656 0.00000000 0.00000000   
## textur that there thick   
## 0.00000000 0.15647508 0.00000000 0.00000000   
## though tight time toast   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## toasti tobacco togeth tomato   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## tone toscana touch tropic   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## turn tuscani two underbrush   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## valley vanilla variet varieti   
## 0.10456280 0.22024776 0.00000000 0.00000000   
## velveti veneto verdot veri   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## vibrant vine vineyard vintag   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## viognier violet warm washington   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## way weight well wet   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## whiff white wild will   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## willamett wine winemak wineri   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## without wonder wood wrap   
## 0.00000000 0.00000000 0.28001657 0.00000000   
## year yellow yet york   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## young zealand zest zesti   
## 0.00000000 0.00000000 0.00000000 0.00000000   
## zinfandel   
## 0.00000000

Here are the results:

train\_svmLinear\_model

## L2 Regularized Support Vector Machine (dual) with Linear Kernel   
##   
## 19356 samples  
## 505 predictor  
## 2 classes: 'excellent', 'good'   
##   
## No pre-processing  
## Resampling: Bootstrapped (25 reps)   
## Summary of sample sizes: 19356, 19356, 19356, 19356, 19356, 19356, ...   
## Resampling results across tuning parameters:  
##   
## cost Loss Accuracy Kappa   
## 0.25 L1 0.7835696 0.5658319  
## 0.25 L2 0.7845502 0.5679076  
## 0.50 L1 0.7843196 0.5674399  
## 0.50 L2 0.7840762 0.5670011  
## 1.00 L1 0.7842031 0.5672562  
## 1.00 L2 0.7835764 0.5660250  
##   
## Accuracy was used to select the optimal model using the largest value.  
## The final values used for the model were cost = 0.25 and Loss = L2.

confusionMatrix(conf\_svmLinear\_train)

## Confusion Matrix and Statistics  
##   
## Actual class  
## Predicted class excellent good  
## excellent 1651 433  
## good 612 2143  
##   
## Accuracy : 0.784   
## 95% CI : (0.7722, 0.7956)  
## No Information Rate : 0.5323   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.5642   
## Mcnemar's Test P-Value : 3.664e-08   
##   
## Sensitivity : 0.7296   
## Specificity : 0.8319   
## Pos Pred Value : 0.7922   
## Neg Pred Value : 0.7779   
## Prevalence : 0.4677   
## Detection Rate : 0.3412   
## Detection Prevalence : 0.4307   
## Balanced Accuracy : 0.7807   
##   
## 'Positive' Class : excellent   
##

We can see that The accuracy is decreasing 0.2%. FN is decreasing a few units. I conclude that the model is not better.

Now double the point only for adjectives: Here is the list adjectives that I extracted:

adj

## [1] "aftertast" "ampl" "appl"   
## [4] "australia" "big" "bitter"   
## [7] "black" "blackberri" "blue"   
## [10] "blueberri" "bold" "bright"   
## [13] "brisk" "central" "classic"   
## [16] "clean" "close" "complex"   
## [19] "cool" "creami" "currant"   
## [22] "cuve" "dark" "deep"   
## [25] "doesnt" "edg" "enough"   
## [28] "exot" "extra" "fair"   
## [31] "fine" "floral" "fragrant"   
## [34] "fresh" "full" "generous"   
## [37] "good" "grand" "great"   
## [40] "green" "hard" "heavi"   
## [43] "hot" "juic" "last"   
## [46] "lean" "licoric" "littl"   
## [49] "live" "lush" "malbec"   
## [52] "mild" "new" "next"   
## [55] "nice" "northeastern" "northern"   
## [58] "nuanc" "old" "oliv"   
## [61] "open" "overal" "perfect"   
## [64] "petit" "pleasant" "pure"   
## [67] "readi" "red" "refresh"   
## [70] "rich" "right" "ripe"   
## [73] "robl" "russian" "rustic"   
## [76] "select" "sharp" "sicili"   
## [79] "simpl" "slight" "smooth"   
## [82] "soft" "solid" "sour"   
## [85] "south" "southern" "straightforward"  
## [88] "strawberri" "strong" "superior"   
## [91] "suppl" "sweet" "tannic"   
## [94] "tempranillo" "thick" "tight"   
## [97] "variet" "vibrant" "warm"   
## [100] "wet" "white" "wild"   
## [103] "young"

And here are the results after double the points for adjectives:

train\_svmLinear\_model

## L2 Regularized Support Vector Machine (dual) with Linear Kernel   
##   
## 19356 samples  
## 505 predictor  
## 2 classes: 'excellent', 'good'   
##   
## No pre-processing  
## Resampling: Bootstrapped (25 reps)   
## Summary of sample sizes: 19356, 19356, 19356, 19356, 19356, 19356, ...   
## Resampling results across tuning parameters:  
##   
## cost Loss Accuracy Kappa   
## 0.25 L1 0.7812716 0.5612232  
## 0.25 L2 0.7844977 0.5676142  
## 0.50 L1 0.7827696 0.5642357  
## 0.50 L2 0.7840476 0.5667318  
## 1.00 L1 0.7826981 0.5640836  
## 1.00 L2 0.7839151 0.5664589  
##   
## Accuracy was used to select the optimal model using the largest value.  
## The final values used for the model were cost = 0.25 and Loss = L2.

confusionMatrix(conf\_svmLinear\_train)

## Confusion Matrix and Statistics  
##   
## Actual class  
## Predicted class excellent good  
## excellent 1667 443  
## good 596 2133  
##   
## Accuracy : 0.7853   
## 95% CI : (0.7734, 0.7968)  
## No Information Rate : 0.5323   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.567   
## Mcnemar's Test P-Value : 2.41e-06   
##   
## Sensitivity : 0.7366   
## Specificity : 0.8280   
## Pos Pred Value : 0.7900   
## Neg Pred Value : 0.7816   
## Prevalence : 0.4677   
## Detection Rate : 0.3445   
## Detection Prevalence : 0.4360   
## Balanced Accuracy : 0.7823   
##   
## 'Positive' Class : excellent   
##

It’s clearly that this model is more or less the same with the original one. To explain why double the points is not work. When reviewing the dataset. I see there are three reasons: Firstly, most important nouns is the ingredients, country, region… They can use the same ingredients, plant in the same region. But, the experts may have some unique techniques to make it become excellent wine. So, nouns are not so helpful. I also see that only famous ingredients are mentioned. Maybe the secret one is not published. Secondly, because I steam when processing data. Many original words are steamed to nouns but it is not the nouns in the original paragraph. Which leads to the method is not as useful as expected. Finally, one factor can affect the system is the POS tagger function doesn’t work perfectly, especially in the adjective list. For example, the word “blackberri” should be in the noun list but it appears in the adjective list, and vice versa for the words “yellow”.

**Finally I conclude that the best model for “good” - “excellent” wine classify is SVM linear with 1-2gram.**

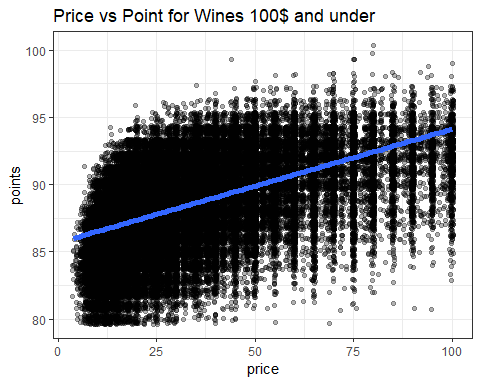
# Second aspect: High value wine

## 6.1 Motivation

In most of the times, the price of the bottle can reflect the quality of the wine. For example, when you want to buy excellent wine, choose the 1000$ bottle. That method should work. But, most of us don’t have the financial condition for buying like this. The point is if we have a limit amount of money, or we just don’t want to pay too much for a bottle, how can we choose? It’s my idea for the second problem: Spend money in the right way.

Just want to confirm that the problem is real and solvable. Let see the scatter plot of price vs points for wines 100$ and under:

ggplot(subset(wine, price <= 100),  
 aes(x = price, y = points)) +  
 geom\_point(alpha = 0.3, position = position\_jitter()) +   
 stat\_smooth(method = "lm", size =2) +  
 labs(title = 'Price vs Point for Wines 100$ and under') +  
 theme\_bw()



As we can see, there is a positive relationship between the points and the price. But, the points for bottles with the same price can be very different. Very cheap wine(<10$) can have up to 92 points while many 25 dollar wines just have less than 85 points.

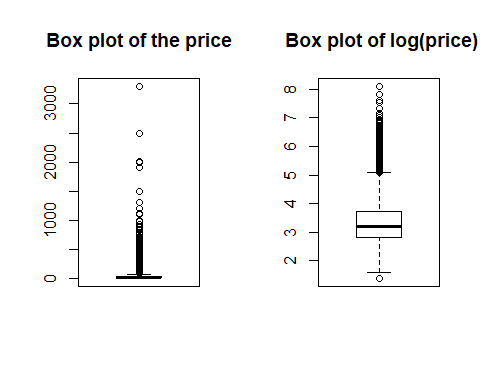
The thing that I see from this graph is: We can have a strategy for choosing excellent wine with a low budget, while there is a high probability of having just a good bottle when choosing randomly even with a quite high price (~50$).

The aim is similar to the first problem: Using NLP and compare ML kernels to solve this problems.

## 6.2 Method and result

Let’s take a look at the price of our data:

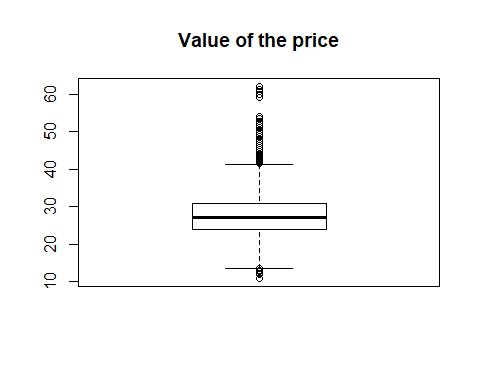
par(mfrow=c(1,2))  
boxplot(wine$price, main = "Box plot of the price")  
boxplot(log(wine$price), main = " Box plot of log(price)")



The price has a wide range. It also has a lot of outliers and high value of SD. otherwise, the log of the price looks better and quite similar to the point.

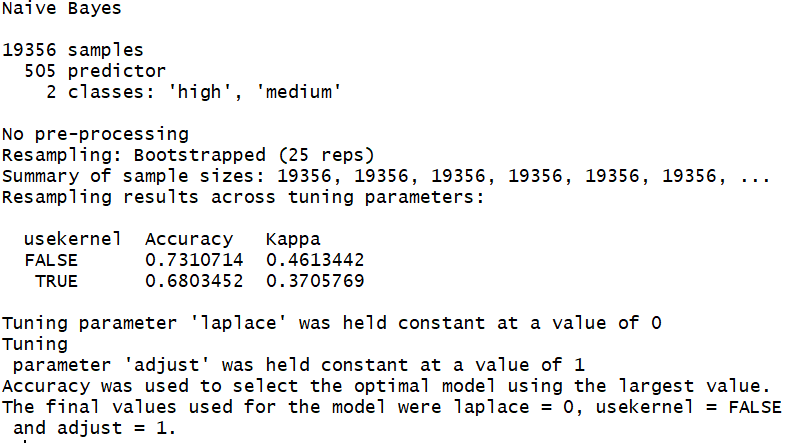
I define the Here is the value:

boxplot(wine$value, main = "Value of the price")

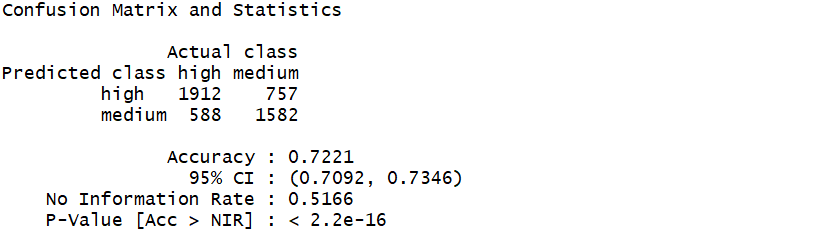


The median number for value is ~27. So the bottle with a higher value than 27 will have a “high” benefit. The rests have “medium” benefit. This way of define class will make all data become interesting. For example. there is a bottle with just 4$ but have 80 points. It becomes the bottle with the highest benefit. In contrast. many bottles with the price higher than 1000 dollar. Of course, they are an excellent wine. But is just have a medium benefit because it’s too expensive.

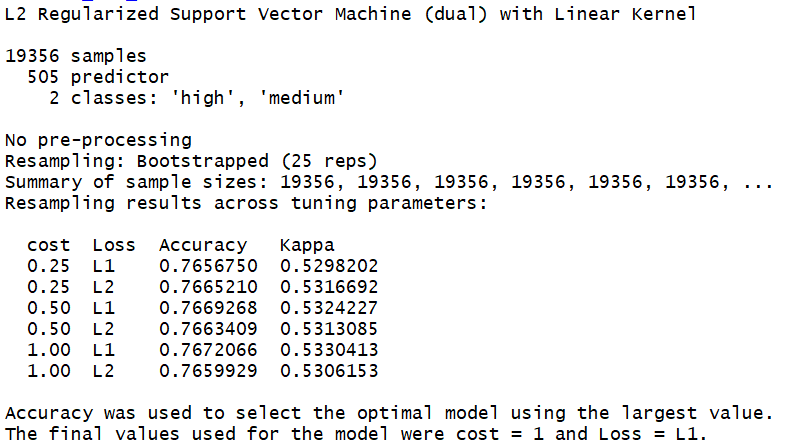
I do the same preprocessing for the text. Then continue to compare models with the same 20% sample of the data. Here are the results.



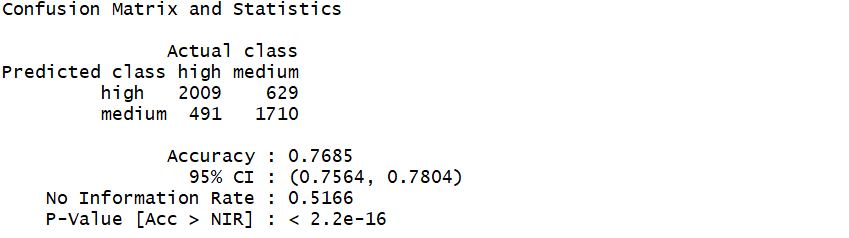
NB benefit



NB benefit predict

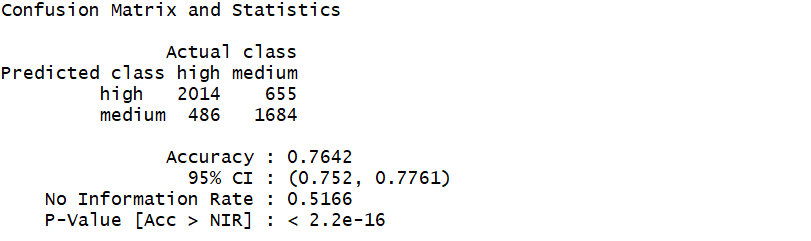


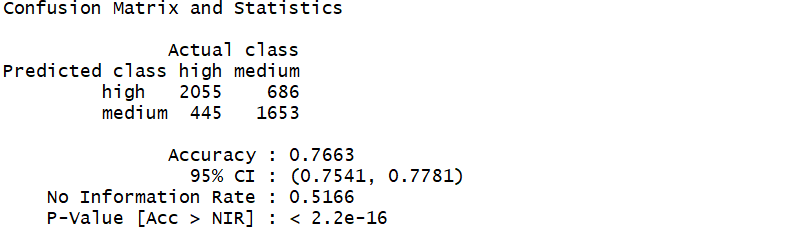
SVM benefit



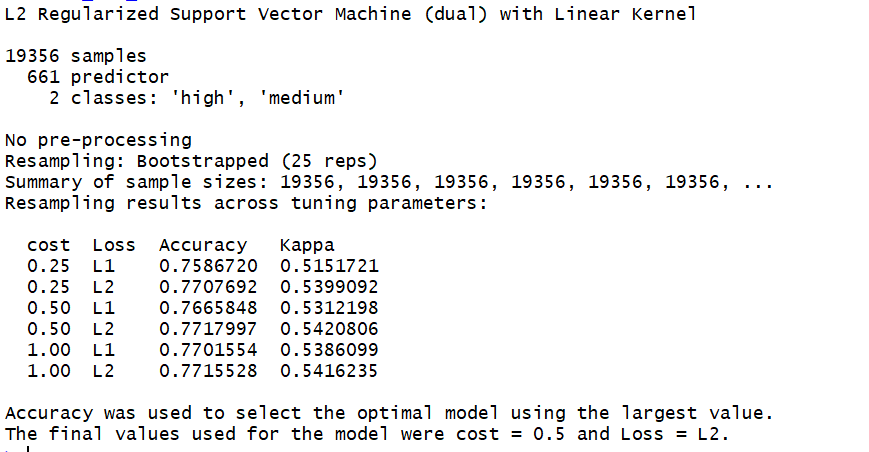
SVMLinear\_benefit

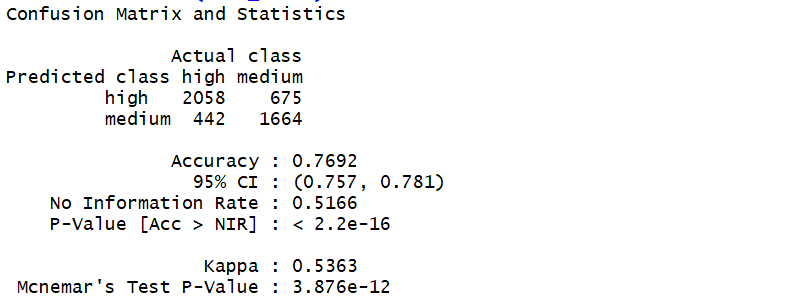
The indicators for comparing models are the same with the previous problem. As we can see, SVM Linear is better than NB in both accuracy and how the bottle is classified. Now I’ll try with POS Tagger

Here are the results when double the point for nouns: 

And here are the results when double the point for adjectives: 

In this case, POS Tagger doesn’t show any clear effect either with double the point for nouns or adjective. All indicator is more or less the same, or even worse. As I said before, the way of classifying here has mix everything.

And here is with 1-2gram model:



The results is more or less the same. In this case, 1-2gram doesn’t improve the model.

**Finally, I conclude that the final model for “high” - “medium” wine value classify is SVM linear with tf-idf, bag of words.**

# Further discussion

## 7.1 Suggestions for choosing wine

Because both problems is solve uing SVMLinear model, which use vector based on tf-idf points. As a result, top points of each class should be the most influences of the class. In this section, I’ll extract the top influences of each class and draw by word cloud.

Here are the results from the SVM Linear for “good” - “excellent” wine model: The first cloud is for excellent wine and the second one is for good wine. Noticed that the words is steamed.

wordcloud(rownames(excellent), excellent[,1], max.words=100, colors=brewer.pal(8, "Dark2"), scale=c(4,.5))



wordcloud(rownames(good), good[,1], max.words=100, colors=brewer.pal(8, "Dark2"), scale=c(4,.5))



Suggestions for choosing wine: For excellent wine, we should pay attention to the word “vineyard”, which is a plantation of grape-bearing vines, grown mainly for winemaking[1]. “concentr” (concentrate) “red\_blen”, “blanc”, “itali” (Italy), “fresh” are some words those you should look when searching for an excellent bottle. “red” and “oak” will appear in both classes. “sauvignon”, “finish”, “fruit” and “franc” (France) are typical words for “good” class.

Here are the top influencer words from the SVM Linear for “high” - “medium” wine model. The first cloud is for high benefit wine and the second one is for medium benefit wine.

wordcloud(rownames(high), high[,1], max.words=100, colors=brewer.pal(8, "Dark2"), scale=c(4,.5))



wordcloud(rownames(medium), medium[,1], max.words=100, colors=brewer.pal(8, "Dark2"), scale=c(4,.5))



Suggesstions for choosing wine: Suggestions for choosing wine: “franc” (France) and “itali” (Italy) likely have more high-value bottle than medium. “spain” (Spain), “carbenet”, “appl” (apply), “pear”, “cherri” are words for high benefit. “valley”, “black”, “champagn” are considered when looking for a medium benefit wine.

## 7.2 Future work:

Here are some points that can improve from the project:

* Check again the SVM RBF kernel with a bigger data set and more powerful computer. In my experiments, SVM RBF is better than SVM Linear in most the cases.
* Improve the POS tagger function.
* Check again the result for 1-2gram with a bigger data set. I try with full data but mycomputer is crashed when train the model because DTM for full data is very heavy (we have a lot of terms)
* Take the advice from wine specialist to improve the model. In my opinion, we should fully understand the dataset before thinking about machine learning. In this case. Because my knowledge of wine is limited. I often don’t clearly understand about a bottle after reading the description. It leads to the ideal for improving the model are limited.
* Split red wine and white wine. These types of wine are quite different in both ingredient and technique. Red wine takes 2/3 of data set while ¼ is white wine (1/12 are others). But you must have some knowledge about wine when split the dataset because the type of wine is not provided, we only have variety like “white blend”, “pinot noir”, “portuguese red”, “riesling”..
* Improve the stop word list. Knowledge about wine is required.

# Conclusion

The issues raised at the beginning are solved with positive results. We figure out that the description and other information printed in a bottle can bring us some idea about how good the bottle is or how high level of benefit it has. We also have some idea about how can we choose a satisfied bottle when reading the descriptions. In the NLP and ML aspect. SVM linear with 1-2gram is the best model for predicting the quality with 78.69% accuracy. For predicting the benefit of a bottle, SVM linear is an acceptable model, with accuracy 77.09%.

# References

1. https://en.wikipedia.org/wiki/Vineyard
2. <https://fderyckel.github.io/2016-12-07-Texts_Classification_in_R/>
3. <https://www.kaggle.com/carkar/classifying-wine-type-by-review>
4. <https://stackoverflow.com/questions/28764056/could-not-find-function-tagpos>
5. <https://machinelearningcoban.com/2017/04/09/smv/>
6. https://machinelearningcoban.com/2017/04/22/kernelsmv/

# Appendix

Here is the code used in this project. POS tagger function is modify from [4]. For some models that require a long time for training, I run it independently and attach the pictures of results.

library(dplyr)  
library(tm)  
library(stringr)  
library(NLP)  
library(openNLP)  
library(caret)  
library(wordcloud)  
library(RColorBrewer)  
knitr::opts\_chunk$set(echo = TRUE, warning=FALSE, message=FALSE, include=FALSE)  
wine <- read.csv("C:/Users/Duong Minh Duc/Documents/GitHub/Text-Mining-Project/wine.csv")  
head(wine)  
boxplot(wine$points, main="Boxplot of points")  
stopwords <- stopwords("english")  
stopwords <- stopwords[!stopwords=="very"]  
stopwords <- c("the", "and", "wine", stopwords)  
  
stopwords  
# Use for check the not available and duplicate data, but not necessary  
wine <- na.omit(wine)  
#wine[duplicated(wine),]  
  
#Assign class  
wine$quality <- wine$points > 88  
wine$quality[wine$quality == TRUE] <- "excellent"  
wine$quality[wine$quality == FALSE] <- "good"  
wine$quality <- as.factor(wine$quality)  
wine$value <- wine$points/log(wine$price)  
wine$benefit <- wine$value > 27  
wine$benefit[wine$benefit == TRUE] <- "high"  
wine$benefit[wine$benefit == FALSE] <- "medium"  
wine$benefit <- as.factor(wine$benefit)  
wine$description <- paste(wine$description, wine$country, wine$designation, wine$province, wine$region\_1, wine$region\_2, wine$variety)  
wine$description <- as.character(wine$description)  
  
#Language convert  
wine$description <- gsub("weissburgunder", "chardonnay", wine$description)  
wine$description <- gsub("spatburgunder", "pinot noir", wine$description)  
wine$description <- gsub("grauburgunder", "pinot gris", wine$description)  
  
#Replace the Spanish garnacha with the french grenache  
wine$description <- gsub("garnacha", "grenache", wine$description)  
  
#Replace the Italian pinot nero with the french pinot noir  
wine$description <- gsub("pinot nero", "pinot noir", wine$description)  
  
#Replace the Portugues alvarinho with the spanish albarino  
wine$description <- gsub("alvarinho", "albarino", wine$description)  
  
#clean non ASCII  
wine$description <- iconv(wine$description, from = "UTF-8", to = "ASCII", sub = "")  
  
#Spit train test  
n = dim(wine)[1]  
set.seed(12345)  
  
id2 = sample(1:n, floor(n\*0.2))  
wine\_sample <- wine[id2,]  
n2 = length(id2)  
id\_test = sample(1:n2, floor(n2\*0.8))  
train = wine\_sample[id\_test,]  
test = wine\_sample[-id\_test,]  
  
##clean function  
clean <- function(text\_vector)  
 {  
 wine\_corpus = VCorpus(VectorSource(text\_vector))  
 wine\_corpus = tm\_map(wine\_corpus, removePunctuation)  
 wine\_corpus = tm\_map(wine\_corpus, content\_transformer(tolower))  
 wine\_corpus = tm\_map(wine\_corpus, removeNumbers)  
 wine\_corpus = tm\_map(wine\_corpus, removeWords, stopwords )  
 #wine\_corpus = tm\_map(wine\_corpus, stripWhitespace)  
 wine\_corpus <- tm\_map(wine\_corpus, stemDocument)  
   
   
 return(wine\_corpus)  
 }  
  
##create the train set  
wine\_train\_set <- clean(train$description)  
  
train$description[1]  
wine\_train\_set[[1]]$content  
  
train\_dtm\_tfidf <- DocumentTermMatrix(wine\_train\_set, control = list(weighting = weightTfIdf))  
train\_dtm\_tfidf <- removeSparseTerms(train\_dtm\_tfidf, 0.99)  
  
  
#create the test set  
wine\_test\_set <- clean(test$description)  
wine\_test\_set <- DocumentTermMatrix(wine\_test\_set, control = list(dictionary = Terms(train\_dtm\_tfidf) ,weighting = weightTfIdf))  
  
#create matrix for training  
wine\_train\_set <<- as.matrix(train\_dtm\_tfidf)  
wine\_test\_set <- as.matrix(wine\_test\_set)  
wine\_test\_set <- wine\_test\_set[,Terms(train\_dtm\_tfidf)]  
  
#create the test result  
wine\_testing\_result <- test$quality  
  
wine\_train\_set[1,]  
wine\_test\_set[1,]  
#train model  
train\_nb\_model <- train(x= wine\_train\_set, y=train$quality , method = 'naive\_bayes')  
  
model\_nb\_result <- predict(train\_nb\_model, newdata = wine\_test\_set)  
conf\_nb\_train <- table(model\_nb\_result, wine\_testing\_result)  
names(dimnames(conf\_nb\_train)) <- c("Predicted class", "Actual class")  
  
train\_nb\_model  
confusionMatrix(conf\_nb\_train)  
#Here is the SVM Linear kenel  
train\_svmLinear\_model <- train(x= wine\_train\_set, y=train$quality , method = 'svmLinear3')  
  
model\_svmLinear\_result <- predict(train\_svmLinear\_model, newdata = wine\_test\_set)  
conf\_svmLinear\_train <- table(model\_svmLinear\_result, wine\_testing\_result)  
names(dimnames(conf\_svmLinear\_train)) <- c("Predicted class", "Actual class")  
  
train\_svmLinear\_model  
confusionMatrix(conf\_svmLinear\_train)  
# #Here is the SVM RBF kenel  
# # Because the trainning time is long. I'll not run this code and only attach images of the previous run.  
# train\_svmRBF\_model <- train(x= wine\_train\_set, y=train$quality , method = 'svmRadial')  
# train\_svmRBF\_model  
# model\_svmRBF\_result <- predict(train\_svmRBF\_model, newdata = wine\_test\_set)  
#   
# conf\_svmRBF\_train <- table(model\_svmRBF\_result, wine\_testing\_result)  
# names(dimnames(conf\_svmRBF\_train)) <- c("Predicted class", "Actual class")  
# confusionMatrix(conf\_svmRBF\_train)  
NLP\_tokenizer <- function(x) {  
 unlist(lapply(ngrams(words(x), 2:2), paste, collapse = "\_"), use.names = FALSE)  
}  
  
wine\_train\_set <- clean(train$description)  
  
train\_dtm\_tfidf <- DocumentTermMatrix(wine\_train\_set, control = list(weighting = weightTfIdf, tokenize=NLP\_tokenizer))  
train\_dtm\_tfidf <- removeSparseTerms(train\_dtm\_tfidf, 0.99)  
  
#create the test set  
wine\_test\_set <- clean(test$description)  
wine\_test\_set <- DocumentTermMatrix(wine\_test\_set, control = list(dictionary = Terms(train\_dtm\_tfidf) ,weighting = weightTfIdf, tokenize=NLP\_tokenizer))  
  
  
#create matrix for training  
wine\_train\_set <<- as.matrix(train\_dtm\_tfidf)  
wine\_test\_set <- as.matrix(wine\_test\_set)  
wine\_test\_set <- wine\_test\_set[,Terms(train\_dtm\_tfidf)]  
#create the test result  
wine\_testing\_result <- test$quality  
  
  
wine\_train\_set[1,]  
train\_svmLinear\_model <- train(x= wine\_train\_set, y=train$quality , method = 'svmLinear3')  
  
model\_svmLinear\_result <- predict(train\_svmLinear\_model, newdata = wine\_test\_set)  
conf\_svmLinear\_train <- table(model\_svmLinear\_result, wine\_testing\_result)  
names(dimnames(conf\_svmLinear\_train)) <- c("Predicted class", "Actual class")  
train\_svmLinear\_model  
confusionMatrix(conf\_svmLinear\_train)  
NLP\_tokenizer <- function(x) {  
 unlist(lapply(ngrams(words(x), 1:2), paste, collapse = "\_"), use.names = FALSE)  
}  
  
wine\_train\_set <- clean(train$description)  
  
train\_dtm\_tfidf <- DocumentTermMatrix(wine\_train\_set, control = list(weighting = weightTfIdf, tokenize=NLP\_tokenizer))  
train\_dtm\_tfidf <- removeSparseTerms(train\_dtm\_tfidf, 0.99)  
  
  
#create the test set  
wine\_test\_set <- clean(test$description)  
wine\_test\_set <- DocumentTermMatrix(wine\_test\_set, control = list(dictionary = Terms(train\_dtm\_tfidf) ,weighting = weightTfIdf, tokenize=NLP\_tokenizer))  
  
#create matrix for training  
wine\_train\_set\_ngram <<- as.matrix(train\_dtm\_tfidf)  
wine\_test\_set <- as.matrix(wine\_test\_set)  
wine\_test\_set\_ngram <- wine\_test\_set[,Terms(train\_dtm\_tfidf)]  
#create the test result  
wine\_testing\_result\_ngram <- test$quality  
  
wine\_train\_set\_ngram[1,]  
train\_svmLinear\_model <- train(x= wine\_train\_set\_ngram, y=train$quality , method = 'svmLinear3')  
  
model\_svmLinear\_result <- predict(train\_svmLinear\_model, newdata = wine\_test\_set\_ngram)  
  
conf\_svmLinear\_train <- table(model\_svmLinear\_result, wine\_testing\_result\_ngram)  
names(dimnames(conf\_svmLinear\_train)) <- c("Predicted class", "Actual class")  
  
train\_svmLinear\_model  
confusionMatrix(conf\_svmLinear\_train)  
#tagPos  
tagPOS <- function(x, ...) {  
 s <- as.String(x)  
 word\_token\_annotator <- Maxent\_Word\_Token\_Annotator()  
 a2 <- Annotation(1L, "sentence", 1L, nchar(s))  
 a2 <- NLP::annotate(s, word\_token\_annotator, a2)  
 a3 <- NLP::annotate(s, Maxent\_POS\_Tag\_Annotator(), a2)  
 a3w <- a3[a3$type == "word"]  
 POStags <- unlist(lapply(a3w$features, `[[`, "POS"))  
 POStagged <- paste(sprintf("%s/%s", s[a3w], POStags), collapse = " ")  
 list(POStagged = POStagged, POStags = POStags)  
}  
wine\_train\_set <- clean(train$description)  
  
train\_dtm\_tfidf <- DocumentTermMatrix(wine\_train\_set, control = list(weighting = weightTfIdf))  
train\_dtm\_tfidf <- removeSparseTerms(train\_dtm\_tfidf, 0.99)  
  
#create the test set  
wine\_test\_set <- clean(test$description)  
wine\_test\_set <- DocumentTermMatrix(wine\_test\_set, control = list(dictionary = Terms(train\_dtm\_tfidf) ,weighting = weightTfIdf))  
  
  
#create matrix for training  
wine\_train\_set <<- as.matrix(train\_dtm\_tfidf)  
wine\_test\_set <- as.matrix(wine\_test\_set)  
wine\_test\_set <- wine\_test\_set[,Terms(train\_dtm\_tfidf)]  
#create the test result  
wine\_testing\_result <- test$quality  
  
  
#extract nouns and adj  
tag <- tagPOS(Terms(train\_dtm\_tfidf))  
tag <- tag$POStags  
noun\_id <- which( tag=="NN")  
nouns <- colnames(wine\_train\_set)[noun\_id]  
adj\_id <- which( tag=="JJ")  
adj <- colnames(wine\_train\_set)[adj\_id]  
  
nouns  
  
column\_id <- c()  
  
#multify for noun  
for (i in 1:dim(wine\_train\_set)[2]) {  
 check <- colnames(wine\_train\_set)[i] %in% nouns  
 if(check)  
 {  
 column\_id <- c(column\_id, i)  
 }  
}  
  
wine\_train\_set[,column\_id] <- wine\_train\_set[,column\_id]\*2  
wine\_test\_set[,column\_id] <- wine\_test\_set[,column\_id]\*2  
  
wine\_train\_set[1,]  
train\_svmLinear\_model <- train(x= wine\_train\_set, y=train$quality , method = 'svmLinear3')  
  
model\_svmLinear\_result <- predict(train\_svmLinear\_model, newdata = wine\_test\_set)  
conf\_svmLinear\_train <- table(model\_svmLinear\_result, wine\_testing\_result)  
names(dimnames(conf\_svmLinear\_train)) <- c("Predicted class", "Actual class")  
  
train\_svmLinear\_model  
confusionMatrix(conf\_svmLinear\_train)  
adj  
#remove double point for nouns  
wine\_train\_set[,column\_id] <- wine\_train\_set[,column\_id]/2  
wine\_test\_set[,column\_id] <- wine\_test\_set[,column\_id]/2  
  
#multify for adj  
column\_id <- c()  
for (i in 1:dim(wine\_train\_set)[2]) {  
 check <- colnames(wine\_train\_set)[i] %in% adj  
 if(check)  
 {  
 column\_id <- c(column\_id, i)  
 }  
}  
  
wine\_train\_set[,column\_id] <- wine\_train\_set[,column\_id]\*2  
wine\_test\_set[,column\_id] <- wine\_test\_set[,column\_id]\*2  
  
train\_svmLinear\_model <- train(x= wine\_train\_set, y=train$quality , method = 'svmLinear3')  
  
model\_svmLinear\_result <- predict(train\_svmLinear\_model, newdata = wine\_test\_set)  
conf\_svmLinear\_train <- table(model\_svmLinear\_result, wine\_testing\_result)  
names(dimnames(conf\_svmLinear\_train)) <- c("Predicted class", "Actual class")  
  
train\_svmLinear\_model  
confusionMatrix(conf\_svmLinear\_train)  
ggplot(subset(wine, price <= 100),  
 aes(x = price, y = points)) +  
 geom\_point(alpha = 0.3, position = position\_jitter()) +   
 stat\_smooth(method = "lm", size =2) +  
 labs(title = 'Price vs Point for Wines 100$ and under') +  
 theme\_bw()  
par(mfrow=c(1,2))  
boxplot(wine$price, main = "Box plot of the price")  
boxplot(log(wine$price), main = " Box plot of log(price)")  
boxplot(wine$value, main = "Value of the price")  
good<- which(train$quality=="good")  
good <- wine\_train\_set\_ngram[good,]  
good = data.frame(sort(colSums(good), decreasing=TRUE))  
excellent<- which(train$quality=="excellent")  
excellent <- wine\_train\_set\_ngram[excellent,]  
excellent = data.frame(sort(colSums(excellent), decreasing=TRUE))  
  
wordcloud(rownames(excellent), excellent[,1], max.words=100, colors=brewer.pal(8, "Dark2"), scale=c(4,.5))  
  
wordcloud(rownames(good), good[,1], max.words=100, colors=brewer.pal(8, "Dark2"), scale=c(4,.5))  
#remove double points for adj  
wine\_train\_set[,column\_id] <- wine\_train\_set[,column\_id]/2  
wine\_test\_set[,column\_id] <- wine\_test\_set[,column\_id]/2  
  
high<- which(train$benefit=="high")  
high <- wine\_train\_set[high,]  
medium<- which(train$benefit=="medium")  
medium <- wine\_train\_set[medium,]  
  
high = data.frame(sort(colSums(high), decreasing=TRUE))  
  
medium = data.frame(sort(colSums(medium), decreasing=TRUE))  
  
wordcloud(rownames(high), high[,1], max.words=100, colors=brewer.pal(8, "Dark2"), scale=c(4,.5))  
  
wordcloud(rownames(medium), medium[,1], max.words=100, colors=brewer.pal(8, "Dark2"), scale=c(4,.5))