A case of Champagne

Machine Learning project for the Capstone module
of the HarvardX/EdX Professional Certificate in Data Science

Thomas Marx

12/06/2020

Contents

1 Introduction

This report presents the second of the two projects required for the PH125.9x Capstone module of the EdX/HarvardX Professional Certificate in Data Science.

As a French person, I feel that Champagne is quite an important part of the national image of France. However, how do you rate a good Champagne? Will an average Champagne rate better than an average non-Champagne sparkling wine? More generally, can you model the rating of a sparkling wine based on a number of factors, possibly diffenciating real Champagnes from other wines?

This project consists in building a sparkling wines database, then constructing two machine learning models, each of which assessed through three algorithms.

The first model aims, after an exploratory data analysis step (which will exhibit some specific relations between a wine rating and other variables), at proposing different algorithms to classify the quality of a sparkling wine among three categories, based on a few explanatory variables.

The second one will attempt to model the lexical field used to described spakling wines, and to infer from this whether a given wine is a Champagne or not.

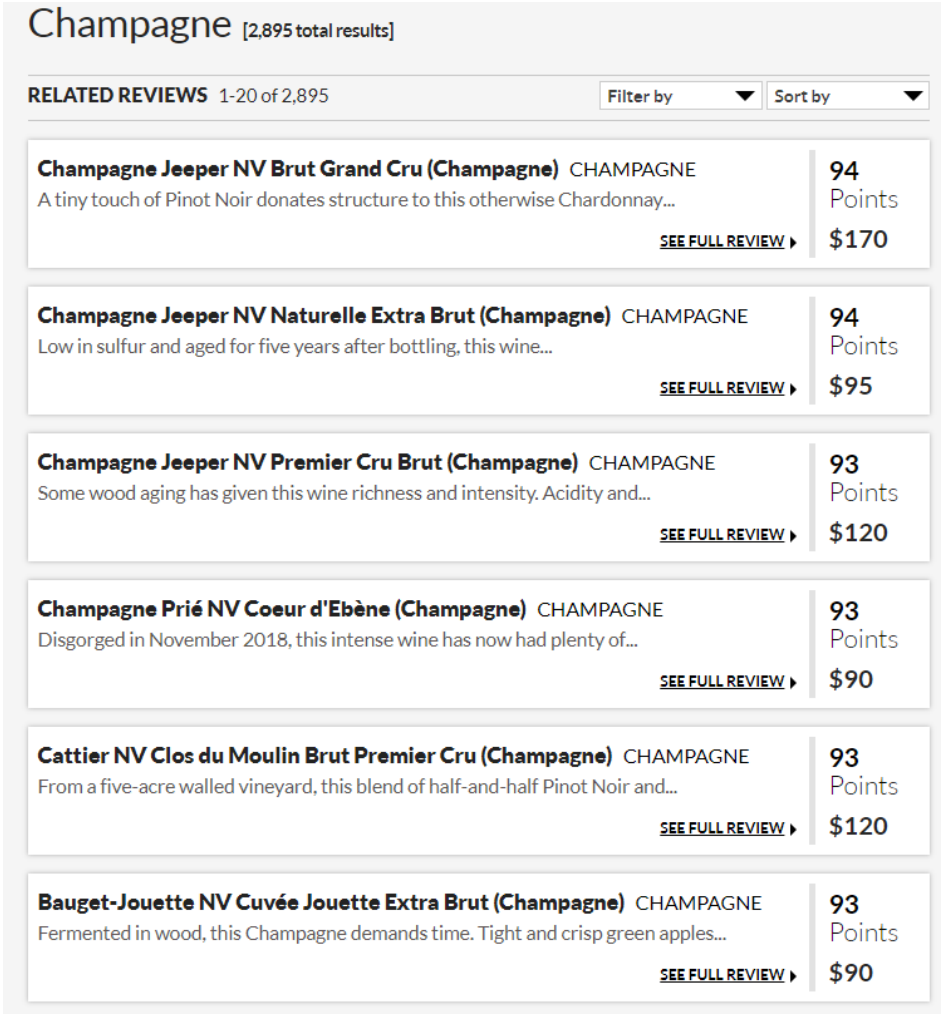
2 Dataset construction

While both Kaggle and the UCI Machine learning repositiry have large, clean datasets on wine, they did not encompass specifically sparkling wines. We thus turned to Wine Enthusiast Magazine, and used their search engine to identify the URL format of sparkling wine reviews.

We then used the web scraping functions of the *rvest* package to :

1. load each consecutive page of listings summary (each page containing 20 boxes, from which the URL to the individual review, the professional rating, the price and the occasional reviewer badge were collected);
2. access each individual rating page, and obtain from there the review text, the appellation, the specific desgination of the bottle, the name of the domain, the alcoholic content of the wine and the bottle size. We did not retrieve the "user average rating" since it appeared to be scarecely populated, nor the "variety" of grapes (always a blend; analysis of the review text was more helpful in that matter), nor the review date (as it was coded inconsistenly in the webpage depending on the wine).

From a navigator user point of view, the first step corresponds to the analysis of pages under the following format:



The second one corresponds to the analysis of individual wine pages under the following format:

The relevant fields were identified in terms of CSS code through the use of the SelectorGadget plugin for Google Chrome.

This information was then put in a data frame, and the variables converted to numeric when necessary, while treating the outliers due to input errors and NA values (badge and vintage). Additional factors were computed, extracting (through the *stringr* package) review length information from the review text, and removing the author first & last names. While it should normally have been an interesting modelling factor, a first analysis suggested a strong relationship between the reviewer name and the type of wine, in particular with a bijection between Champagnes and Roger Voss. Hence, keeping the reviewer name in the review text would have given away the wine type too easily in our models and/or lead to colinearity issues.

We also extracted the specific appellation and country of production from the appellation field (expression occurring respectively before the first comma and after the last one).

We initially intended to construct a factor regarding the optimal drink time (based on the review text analysis), but were confronted to three issues:

1. Most reviewers did not provide such recommandation, which was for a large part limited to Roger Voss, the site Champagne expert (which makes sense: non-Champagne sparkling wines are not really expected to benefit from longer bottle aging);
2. When it was provided, the optimal drink date was taking different formats (a specific year, “now”, “a few years”), making it hard to encode;
3. Extracting a four-digit figure starting with “20” expecting to grab the optimal drink date was not feasible, as it could also capture disgorgement time (a factor we initially intended to incorporate in our models, but finally left aside due to its scarce mentions).

3 Exploratory data analysis

3.1 Dataset review

To start assessing how wines characteristics affect the rating, we start with a display of the header of the dataset, orderered by ascending points:

Table 1: Table continues below

wineId	domain	designation
3729	Viejo Isaias	Don Isaias Brut Nature Cuvée Especial
4041	Naveran	Perles d'Or
4042	ViÃ±edos Balmoral	EdonÃ© Gran CuvÃ©e Extra Brut

Table 2: Table continues below

review	badge	alcohol	bottle_size	wine_year
A soft orchard-fruit aroma is muted and earthy rather than fresh. This brut nature is flat in feel, with a sweet, waxy white-fruit flavor that culminates on a wheaty finish	None	11.7	750	None
Sorry folks, but this smells like vegetables in the garbage can, meaning earthy and rotten. A foamy clumsy mouthfeel is unhelpful, while this tastes of pickled citrus fruits and cole slaw. All in all, this is barely palatable	None	12.5	750	None
Highly volatile and pickled aromas are unpleasant and uninviting. This is flat on the palate. Sour flavors of brine and green herbs are the final nails in the coffin of this barely acceptable Cava	None	12	750	None

points	price	appellation	country	rev_length
80	16	Mendoza	Argentina	172
80	25	Cava	Spain	225
80	35	Spain	Spain	196

then by descending points:

Table 4: Table continues below

wineId	domain	designation
1538	Louis Roederer	Cristal VinothÃ¨que Brut
1539	Billecart-Salmon	Le Clos Saint-Hilaire Brut
1551	Louis Roederer	Cristal VinothÃ¨que Brut

Table 5: Table continues below

review	badge	alcohol	bottle_size
A re-release from one of the legendary vintages of the last 30 years, this Champagne still shows some of the intense acidity that marked that year. It has gained maturity, but it's an ageless, magnificent wine. Although it is ready to drink, it will hold well through 2029	None	12	750
From just over two acres of old vines in the producer's home village of Mareuil-sur-Aÿ, this pure Pinot Noir Champagne is magnificent. A toasty flavor is balanced by concentrated white fruits, with a touch of tannin adding texture. It's an unforgettable wine. Enjoy in the nearterm	Editors' Choice	12.5	750
A re-release from one of the legendary vintages of the last 30 years, this Champagne still shows some of the intense acidity that marked that year. It has gained maturity, but it's an ageless, magnificent wine. Although it is ready to drink, it will hold well through 2029	None	12	750

wine_year	points	price	appellation	country	rev_length
None	100	1100	Champagne	France	272
None	100	500	Champagne	France	281
2002	100	1100	Champagne	France	272

and then some basic statistics, to assess the range of values and types of variables (after cleaning):

Table 7: Table continues below

wineId	domain	designation	review
Min. : 1	Length:19641	Length:19641	Length:19641
1st Qu.: 4911	Class :character	Class :character	Class :character
Median : 9821	Mode :character	Mode :character	Mode :character
Mean : 9821	NA	NA	NA
3rd Qu.:14731	NA	NA	NA
Max. :19641	NA	NA	NA

Table 8: Table continues below

badge	alcohol	bottle_size	wine_year
Length:19641	Min. : 4.50	Min. :187	Length:19641
Class :character	1st Qu.:11.50	1st Qu.:750	Class :character
Mode :character	Median :12.00	Median :750	Mode :character
NA	Mean :11.87	Mean :748	NA
NA	3rd Qu.:12.50	3rd Qu.:750	NA
NA	Max. :21.50	Max. :750	NA

Table 9: Table continues below

points	price	appellation	country
Min. : 80.0	Min. : 5.00	Length:19641	Length:19641
1st Qu.: 87.0	1st Qu.: 19.00	Class :character	Class :character
Median : 88.0	Median : 30.00	Mode :character	Mode :character
Mean : 88.6	Mean : 42.31	NA	NA
3rd Qu.: 91.0	3rd Qu.: 49.00	NA	NA
Max. :100.0	Max. :2400.00	NA	NA

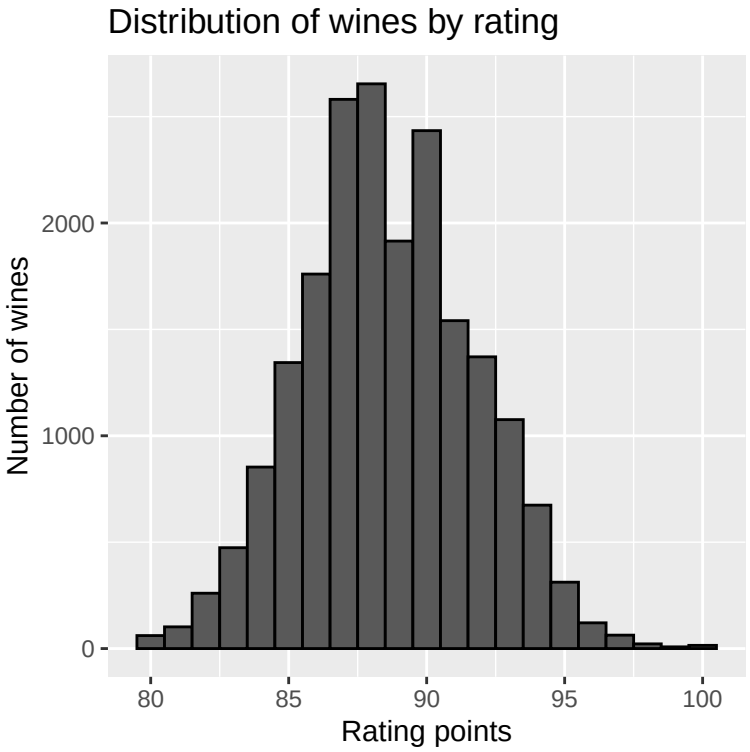
rev_length
Min. : 1.0
1st Qu.:196.0
Median :231.0
Mean :238.7
3rd Qu.:272.0
Max. :623.0

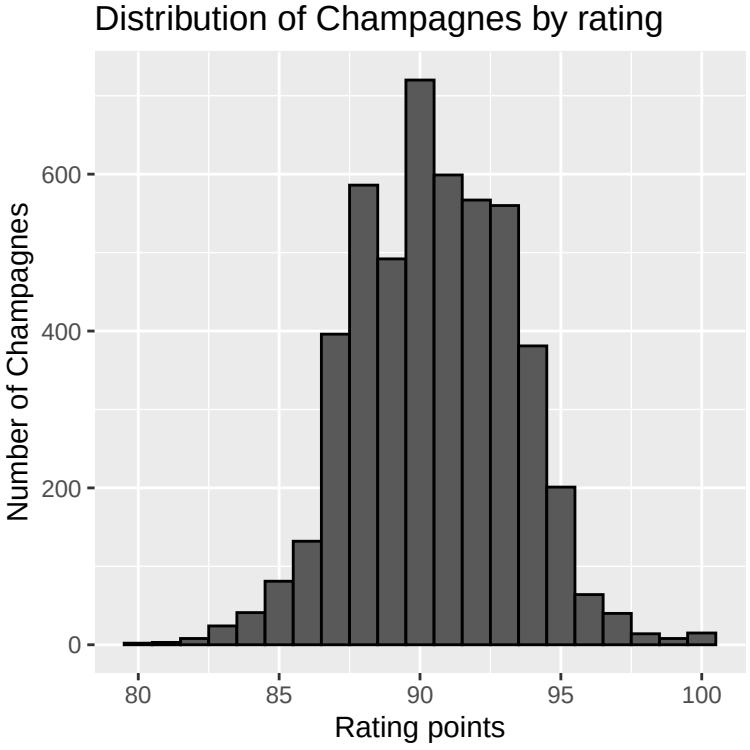
The main take-aways from this review are:

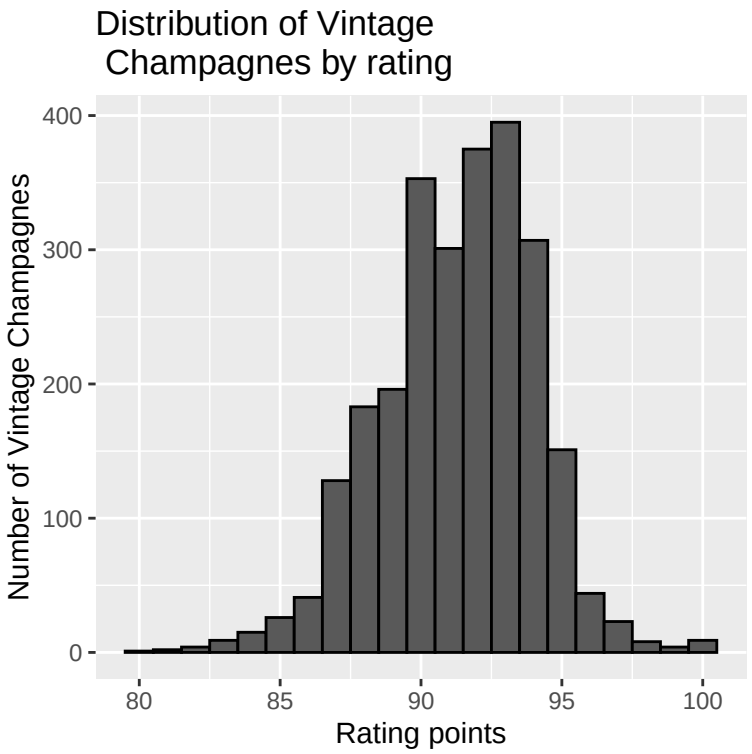
1. At time of scraping, there were 19641 sparkling wines reviewed on Wine Spectator, from 2133 domains;
2. They represent 407 appellations, with Champagne covering 4934 wines;
3. Alcohol grades are mostly in the traditional 12 to 12.5° range, but there are some outliers;
4. While mid-size bottles were reviewed, no large format (magnum, jeroboam or larger, generally considered as offering better evolution perspectives) were reviewed;
5. Review ratings range from 80 to 100 points;
6. The prices range from 5 to 2400 USD, with a median of 30 USD;
7. Review length varies wildly, from a single character (a single space, to ensure the field is populated) to 623 (though the kurtosis should be low, considering the 1st and 3rd quarter levels).

3.2 Graphical analysis

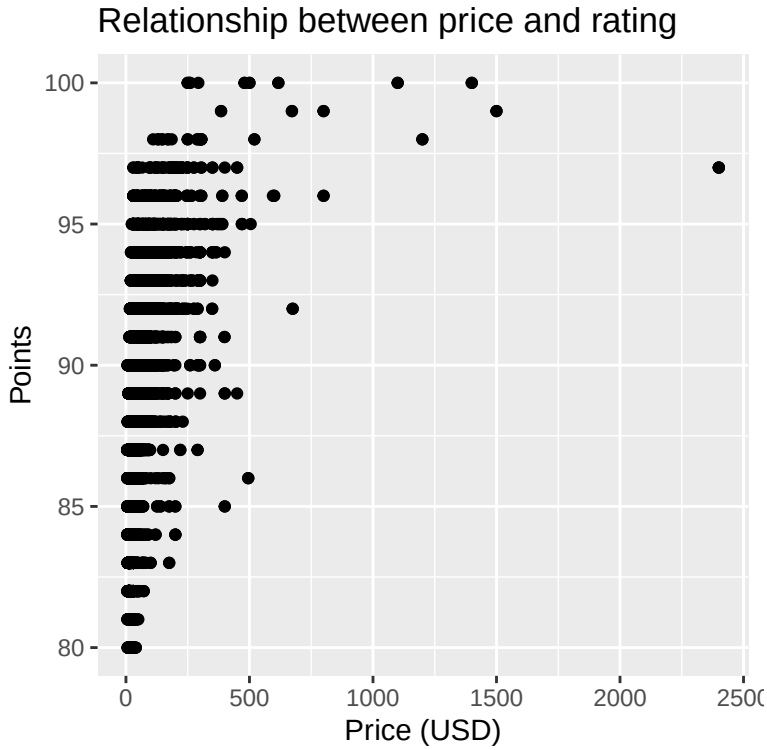
Let's first check how ratings are distributed:

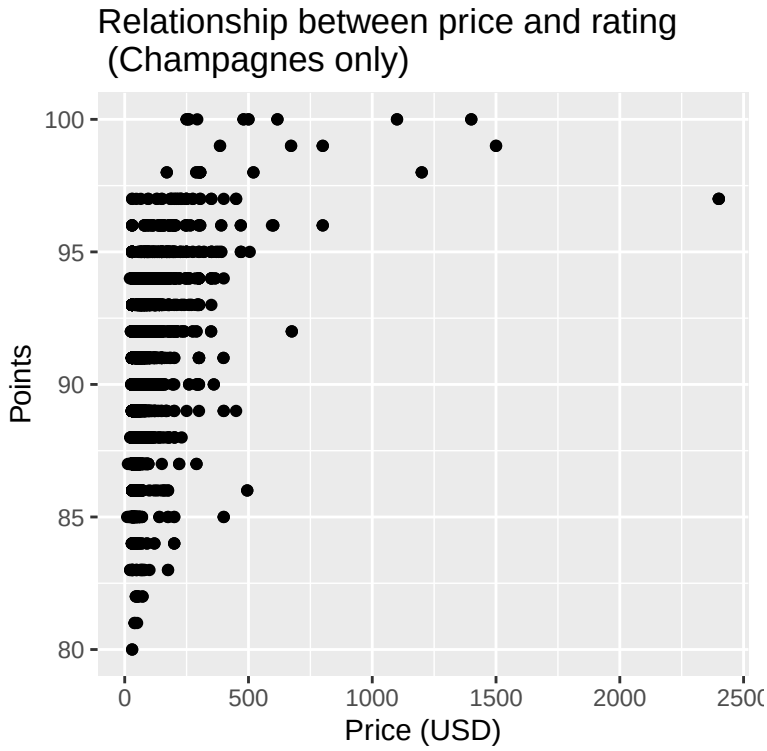


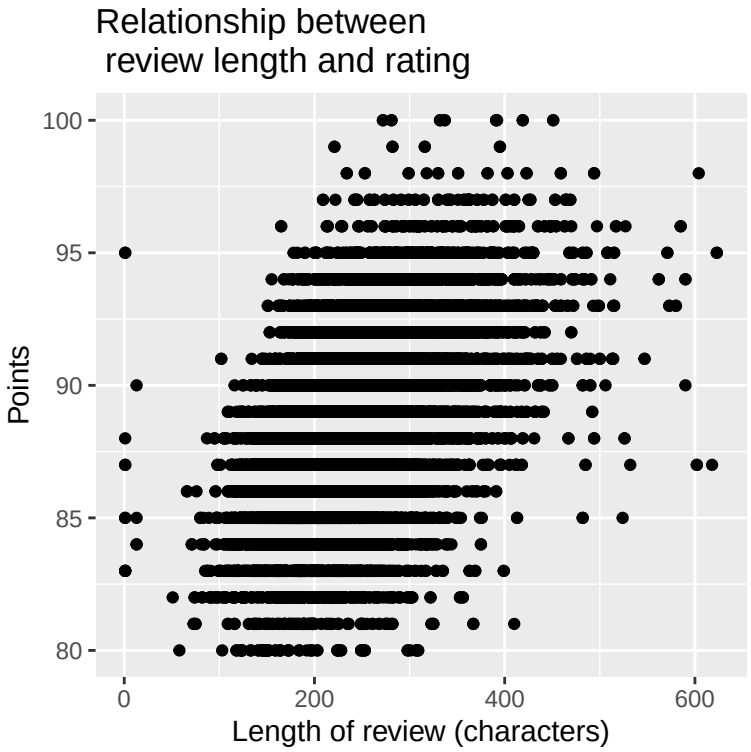


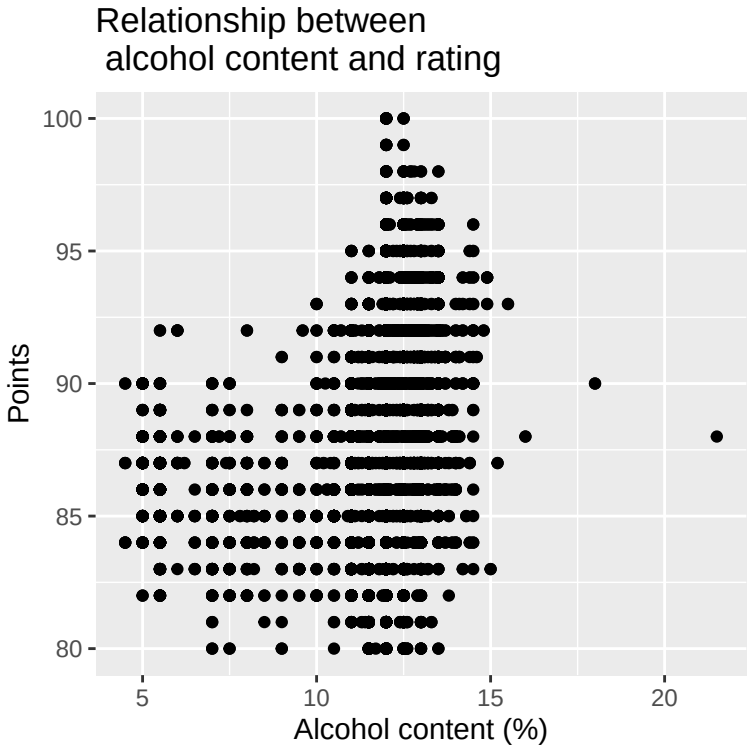

Distribution of Vintage
Champagnes by rating
Number of Vintage Champagnes
400
300
200
100
0
80
85
90
95
100
Rating points

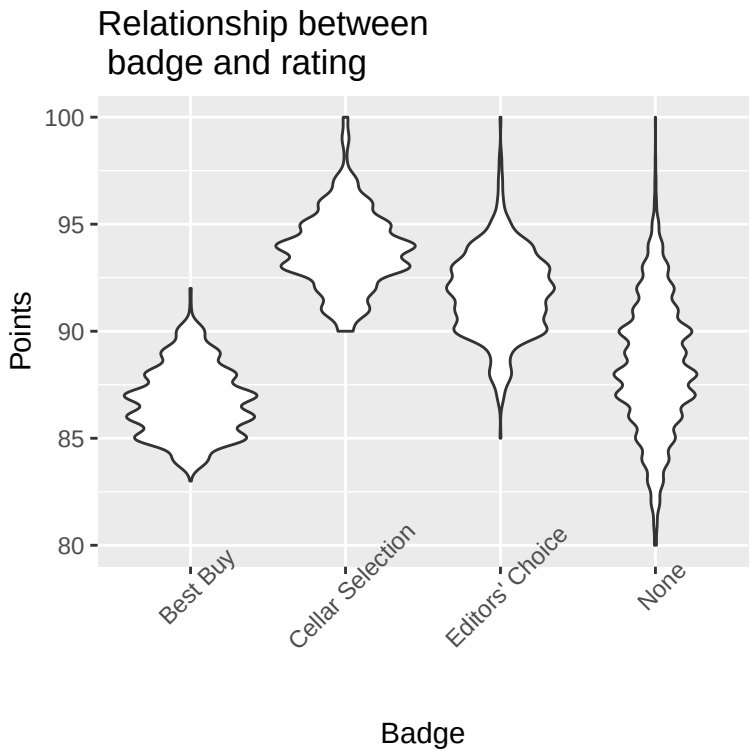
We now plot relationships between some variables and ratings:



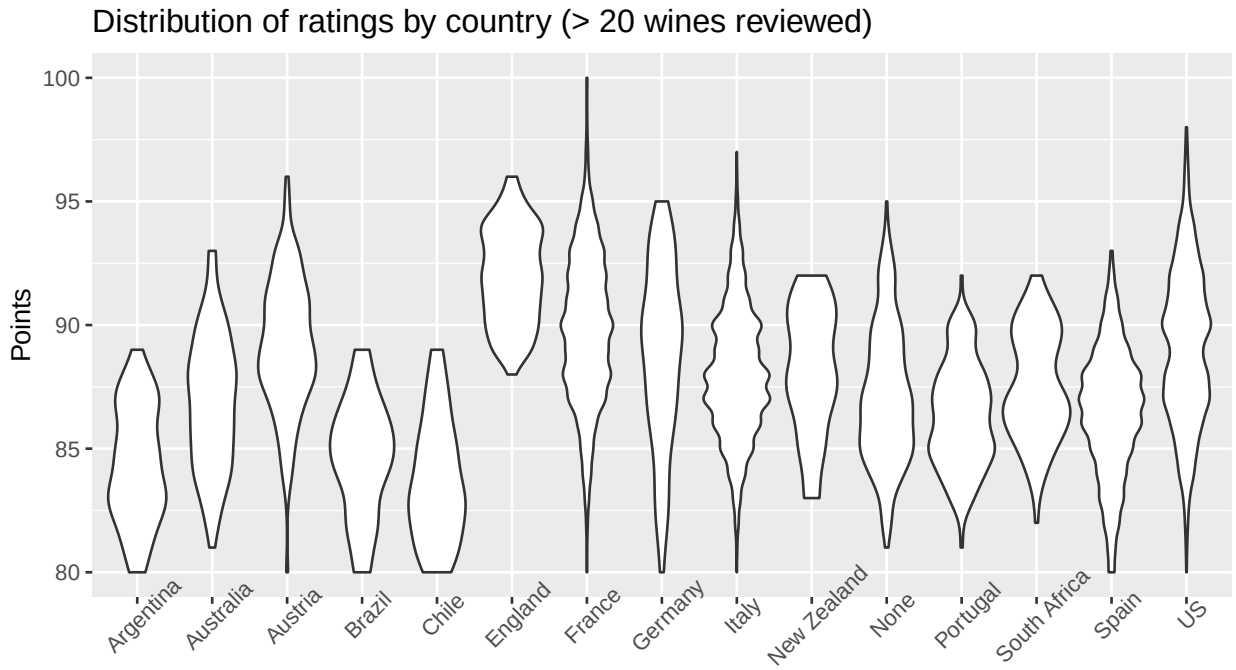


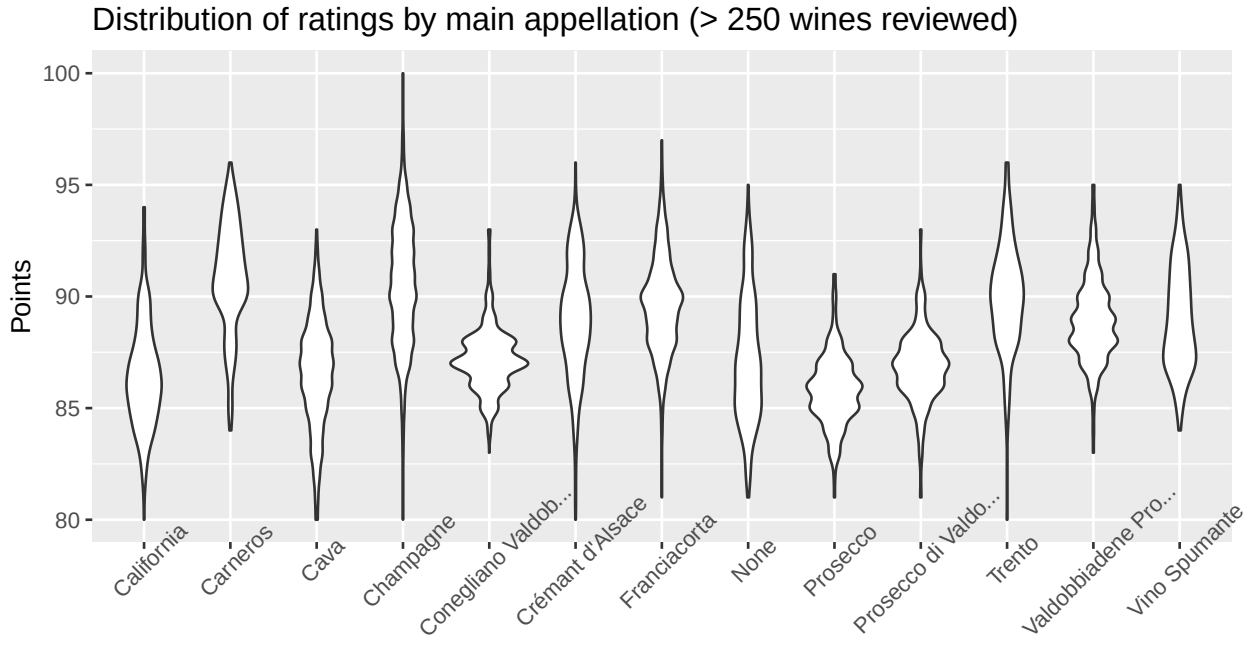

Relationship between
review length and rating
Points
100
95
90
85
80
0
200
400
600
Length of review (characters)


Relationship between
alcohol content and rating
Points
100
95
90
85
80
5
10
15
20
Alcohol content (%)

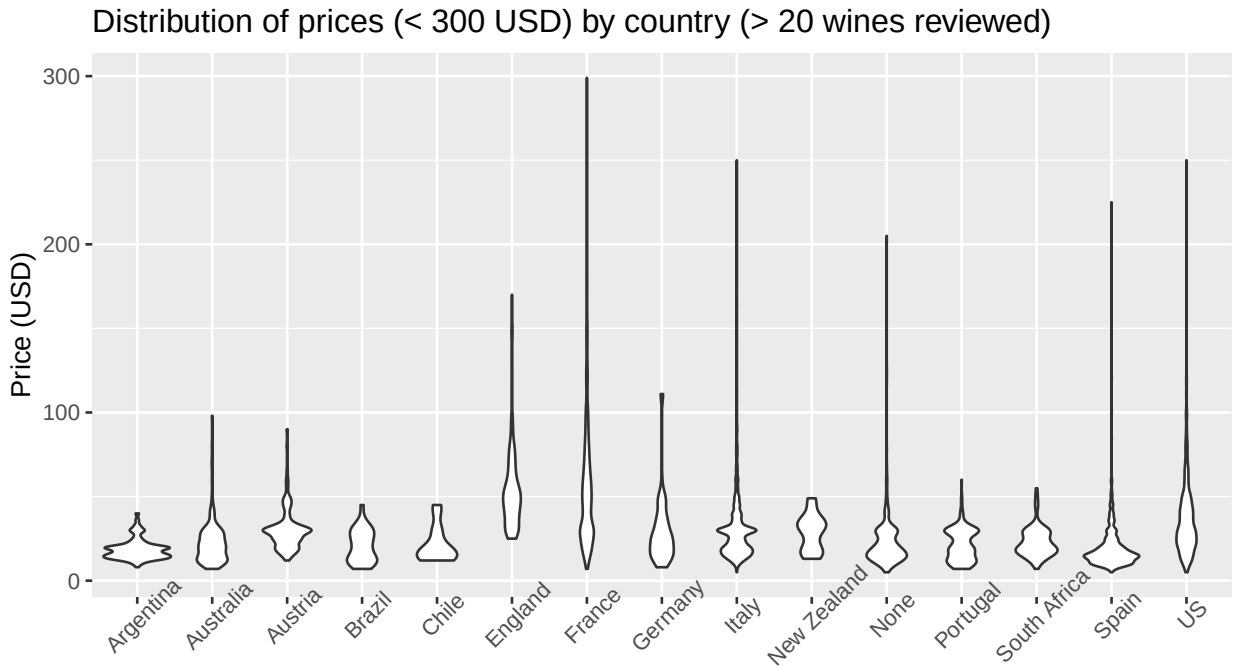

Relationship between
badge and rating
Points
100
95
90
85
80
Best Buy
Cellar Selection
Editors' Choice
None
Badge

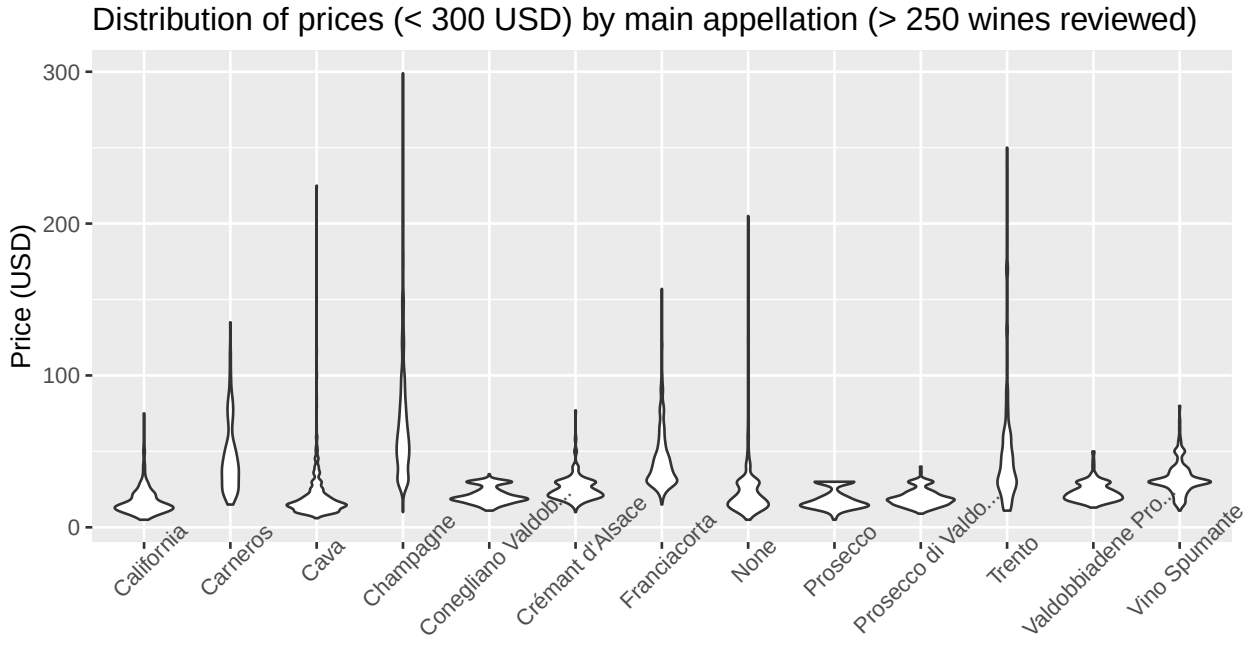
We now plot relationships between origin variables and ratings:





Finally, we plot relationships between origin variables and prices:





From these different graphics, we can infer the following relations:

1. The distrbution of ratings appears to be roughly bell-shaped, aside from an under-representation of the 89 rating (probably for psychological reasons, 90 being the threshold for entry among very good wines);
2. Actual Champagnes tend to be rated higher (mode at 90 instead of 88 for the whole population), with a distribution skewed to the right (as well as a fatter right tail, compared to a fat left tail for the whole population);
3. This phenomenon is even more exacerbated for Vintage Champagnes, with a mode at 93 and an even stronger right skew;
4. The amplitude of price range tends to grow with rating levels; however, as we don't know if tastings were blind, it is difficult to assess whether ratings were influenced by prices. In addition, the minimum price of a sparkling wine for a given rating tends to rise with the rating;
5. The universe of Champagnes exhibits similar behaviours in terms of price / rating relation, which suggests that reviews were done fairly, on their own merit between all sparkling wines;
6. Better-rated wines tend to have longer reviews, which could be the expression of a higher enthusiasm, or more details regarding subtle notes of more complex wines;
7. There does not appear to be a clear relationship between alcohol contents and ratings, though we observe that low-alcohol wines (less thant 10 °) tend to be capped in their ratings;
8. While no-badge wines can be found across all rating levels, each of the three explicit badges seems to *de facto* cover a specific rating range: 85 to 90 points for 'Best Buy', 90 to 100 points for 'Cellar Selection' and 90 to 95 points for 'Editor's Choice';
9. The distribution of ratings by country exhibits a specific French behavior (only country covering the whole rating range, and in particular only country to register perfect scores), which the ratings by appellation graphic shows is largely attributable to Champagnes (only wines to register perfect scores) ;
10. Finally, the distribution of prices by country does not repeat this French exception, as US or English wines appear to have a similar distribution (for wines below 300 USD), including with a higher entry-price for UK.

3.3 Take-aways from the EDA for the modelling

This analysis gave us some hints as to interesting models that could be applied to this dataset:

1. A classification model among wine qualities could be applied bassed on a few explanatory variables, selected among those having showed the strongest correlation with ratings (country, price and review length);
2. A regression model could be applied on the review text corpus, so as to verify if the sole review text is sufficient to distinguish between a Champagne and a non-Champagne wine.

4 Modelling approach

4.1 First model : classification of wines among three quality levels

The first of the two meachine-learning models will be a classification-based one. The objective of our classification algorithms (we will compare three of them) is to predict to which of three quality categories a wine from the test set belongs, based on the explanatory variables. We will judge the three algorithms based on their accuracy.

We start by reencoding wine ratings intro three quality categories:

- 'Low' for ratings from 80 to 87 points;
- 'Average' for ratings from 88 to 90 points;
- 'High' for ratings from 91 to 100 points.

These three categories have roughly similar populations:

Quality	Population
Average	7003
High	5203
Low	7435

After setting a seed, we construct training and test sets along a 90/10 line. We also define a resampling parameter (*trControl*) as 5-fold cross-validation (*i.e.* the training set is divided into five subsamples of equal size, with four of them used for training the model and the last one used for validation; this process is repeated five times, with each subsample being used once and only once as validation).

Before training each algorithm, we will set a seed to ensure reproducible results.

4.1.1 First algorithm: Random Forest

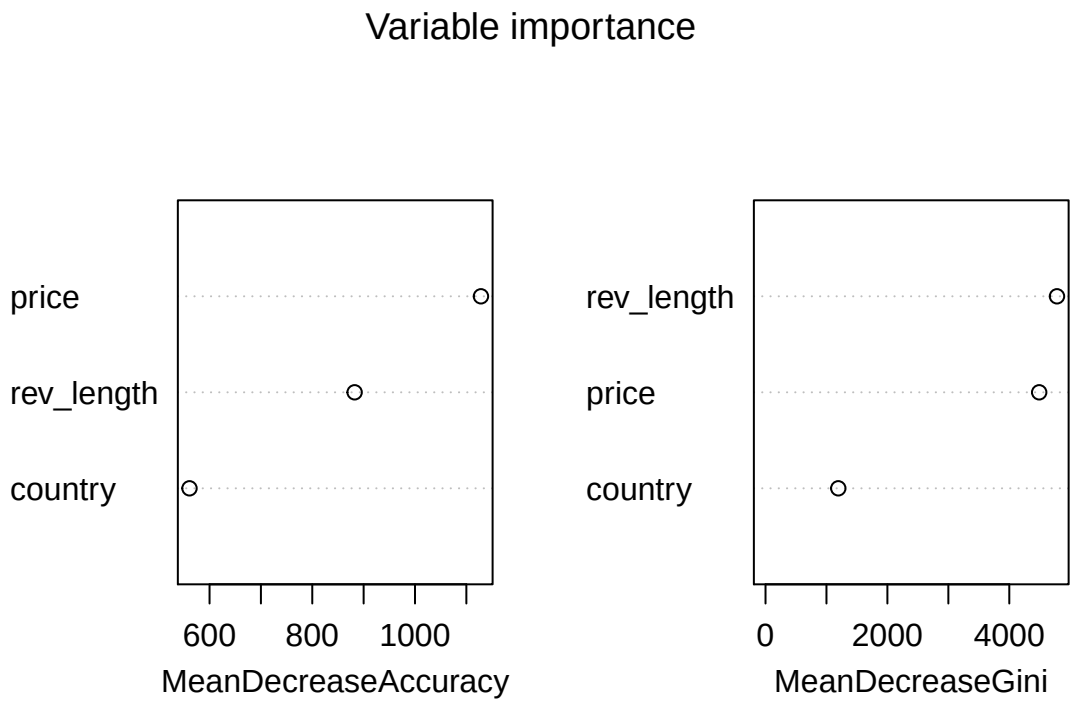
We start with a standard algorithm for this kind of classification exercise, a Random Forest with 1000 trees (as suggested by the litterature) and a node size at 1 (better for classification);

We tried the different possible values of the *mtry* parameter (*i.e.* 1, 2 and 3 in our case), and a value of 3 gives the best accuracy.

The model is thus expressed as such:

```
rf_model <- randomForest(quality ~ price + rev_length + country,
                         data = wines_class_train,
                         importance = TRUE,
                         ntree = 1000,
                         nodesize = 1,
                         mtry = 3,
                         trControl = trainControl)
```

which leads to the following variable importance graphic:



We then use the *predict* function, then build the confusion matrix, from which we extract the accuracy:

```
rf_pred <- predict(rf_model, wines_class_test)
confusionMatrix(rf_pred, wines_class_test$quality)
```

We then start to build a table comparing the accuracies of our three algorithms:

	Method	Accuracy
Accuracy	Random Forest (1000 trees, node size = 1, mtry = 3)	0.6852

4.1.2 Second algorithm: Classification and Regression Trees (CART), with bagging

A similar approach to random forests is CART, which we try to enhance through bagging, so as to reduce variance and minimize overfitting. To that end, we use the *train* function of the *caret* package with a *treebag* model, along with the aforementioned 5-fold cross-validation training control.

This model is expressed as such:

```
tree_model <- train(quality ~ price + rev_length + country,
                    data = wines_class_train,
                    method = "treebag",
                    trControl = trainControl)
```

We then use once again the *predict* function and build the confusion matrix:

```
tree_pred <- predict(tree_model, wines_class_test)
confusionMatrix(tree_pred, wines_class_test$quality)
```

And we add this accuracy to our comparison table:

Method	Accuracy
Random Forest (1000 trees, node size = 1, mtry = 3)	0.6852
Bagged CART (no parameters)	0.7937

4.1.3 Third algorithm: K-Nearest Neighbors

The third classification algorithm used is K-nearest neighbors, which we will train with a grid for the k parameter (number of neighbors) ranging from 1 to 10.

```
knn_model <- train(quality ~ price + rev_length + country,
                   data = wines_class_train,
                   method = "knn",
                   trControl = trainControl,
                   tuneGrid = expand.grid(k = 1:10))
```

A comment in the model object suggests a parameter of k = 1 was used as optimal.

We proceed one last time with the *predict* and *confusionMatrix* functions:

```
knn_pred <- predict(knn_model, wines_class_test)
confusionMatrix(knn_pred, wines_class_test$quality)
```

And we get the final line of our classification algorithm comparison table:

Method	Accuracy
Random Forest (1000 trees, node size = 1, mtry = 3)	0.6852
Bagged CART (no parameters)	0.7937
K nearest neighbors (grid for k: 1 to 10, value of 1 selected)	0.8013

4.2 Second model: identifying if a sparkling wine is a Champagne

As we observed in the EDA part, there is a relatively strong correlation between the rating of a wine and the length of its review (which has justified including *rev_length* as one of the three explanatory variables in the first model). However, can we do more, and use text mining to analyze the corpus of the text review, and guess from there if a wine is a Champagne, based solely on the text of its review?

To that effect, we will use in particular two packages:

- *text2vec* for text vectorization;
- *glmnet* for fast lasso regularization on generalized linear models regressions on these vectors.

From our webscraped and cleaned dataset, we extract only the review text and appellation, then set a binary variable (1 if the appellation is Champagne, 0 otherwise), and drop the appellation column. Our new dataset is thus of the following form:

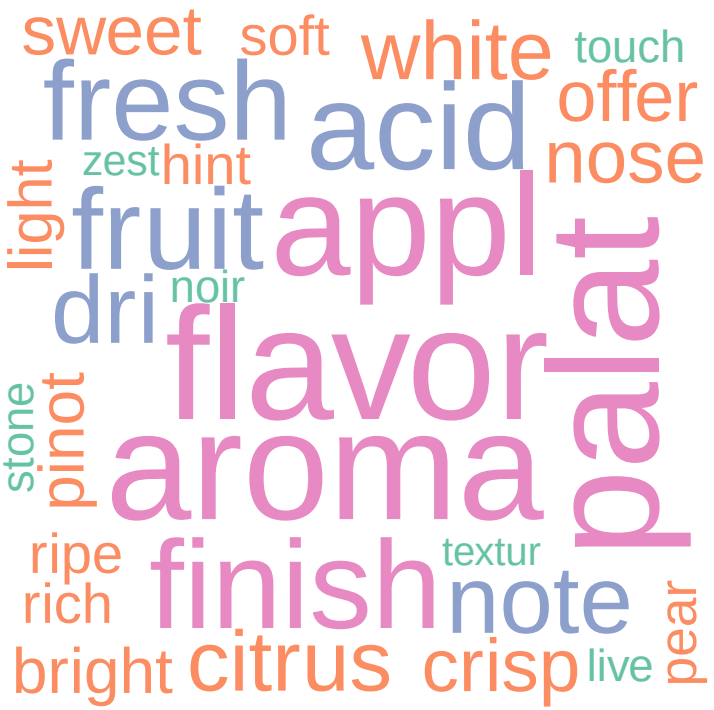
review	isChamp
This inaugural release is stunning and incredibly impressiveâ€"a wine well worth stocking up on for the price and quality. A beautiful jasmine aroma leads to marzipan, peach and cherry flavors as lively acidity envelopes the palate. It's a blend of 80% Chardonnay and 20% Pinot Noir	0
A blend of 75% Pinot Noir and 25% Chardonnay, this dry brut is complex, mineral-driven and enduringly fresh. Peach, pear and apple flavors meet a briny salt note that buzzes on the palate	0
One of the only domestic sparkling Picpoul Blancs, this bottling starts with aromas of orange rind, wet clay and the slightest hint of petrol. It's laser sharp, crisp and racy on the palate, offering flavors of Asian pear, orange blossom and a hint of jasmine	0
Strongly aromatic on the nose, this sparkling Riesling shows melon, crisp lime and kiwi on the nose, as well as a hint of sourdough. The mousse fires up rapidly once sipped, delivering rounded kiwi and sharper yuzu flavors	0

We then do a partition of the dataset between Champagnes and non-Champagnes, and use the *Corpus* and *tm_map* functions of the *tm* package to produce a standardized (lower case, without spaces, punctuation or numbers) corpus of each subset. We remove the words "champagne", "wine" and "drink" along with stop-words (common, short function words with little informational content).

We can then draw word clouds of the most common words for each category.

- For Champagnes:

- For non-Champagnes:



We can observe that while the vocabulary is roughly similar between both categories, term frequencies do vary. For example, under the set seed, "acid" is one of the key term of the corpus describing Champagnes, while it is relatively secondary for non-Champagnes. The opposite occurs for "apple".

It shoud thus be possible to use these differences in vocabulary to construct vocabulary vectors, and conduct regression on these variables to assess whether a given wine review refers to a Champagne or not.

To that effect, we first partition (after setting a seed) the reduced dataset (text of review and binary variable for Champagnes) into training and test subsets, then use the *itoken* function of the *text2vec* package to tokenize the text of the review of the train subset. We then use the *create_vocabulary* (removing the obvious "champagne" as stopword) and *vocab_vectorizer* functions of this package to obtain the vectors of the training set, which will be used as explanatory variables in the regression (after a ultimate transformation in Document Term Matrix format).

```
token_train <- itoken(wines_t2v_train$review,
                      preprocessor = tolower,
                      tokenizer = word_tokenizer)

vectors <- create_vocabulary(token_train, stopwords = "champagne") %>%
  vocab_vectorizer(.)

dtm_train <- create_dtm(token_train, vectors)
```

We tokenize as well the test set, which we transform in DTM using the training set-based vectors.

```
token_test <- itoken(wines_t2v_test$review,
                     preprocessor = tolower,
                     tokenizer = word_tokenizer)

dtm_test <- create_dtm(token_test, vectors)
```

We will now test three variations of a similar 5-fold GLM algorithm, on three versions of the DTMs:

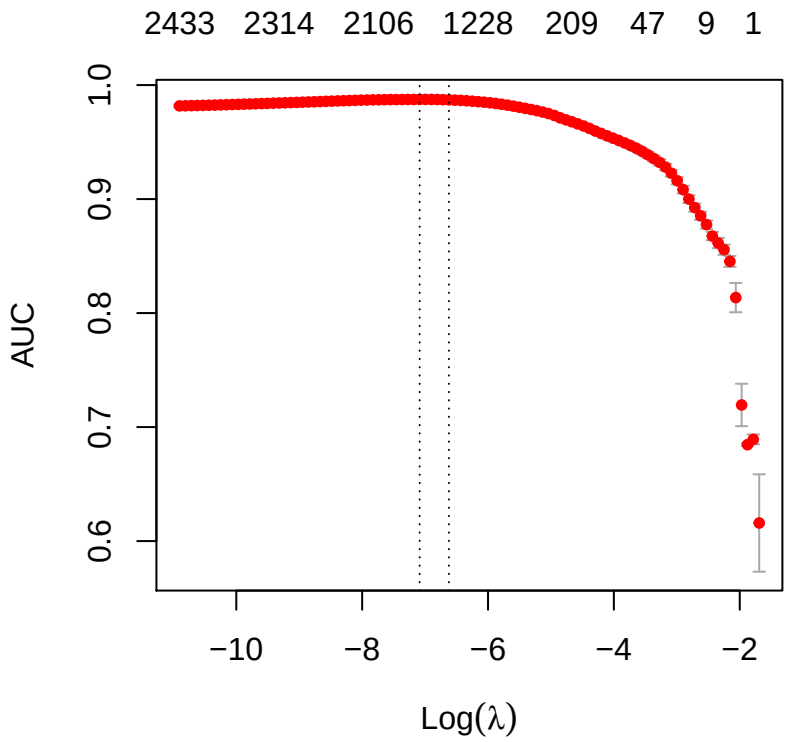
1. With the base DTMs;
2. With pruned, N-grams vectors;
3. With a TF-IDF transformation of the DTMs.

4.2.1 First algorithm: base Document Term Matrices

The *cv.glmnet* function of the *glmnet* package does cross-validation on the sequence of models provided by the *glmnet* function. The underlying modelling will be a logistic regression (with binomial distribution), with a lasso penalty ($\alpha = 1$). As for the cross validation, we will use a five-fold one and a criterion of Area under curve. The model will thus be expressed as:

```
glmnet_train <- cv.glmnet(x = dtm_train,
                          y = wines_t2v_train$isChamp,
                          family = "binomial",
                          alpha = 1,
                          type.measure = "auc",
                          nfolds = 5)
```

Run on the base DTMs, this model gives the following AUCs as a function of $\log(\lambda)$:



for a maximum AUC value of 0.9874.

We can then use this model (with maximum AUC) to run our prediction (with a *response* type given the GLM modelling). The resulting prediction will be transformed in a binary variable based on a threshold of 0.5.

```
glmnet_pred <- predict(glmnet_train, dtm_test, type = 'response')[,1]

glmnet_res <- data.frame(cbind(test = wines_t2v_test$isChamp,
                               pred = ifelse(glmnet_pred>0.5,1,0)))
```

After running the *confusionMatrix* function on these results (transformed as factors), we start building our table comparing accuracies of the three algorithms (which in fact are just one algorithm, run over three sligthly diffrent versions of the DTMs).

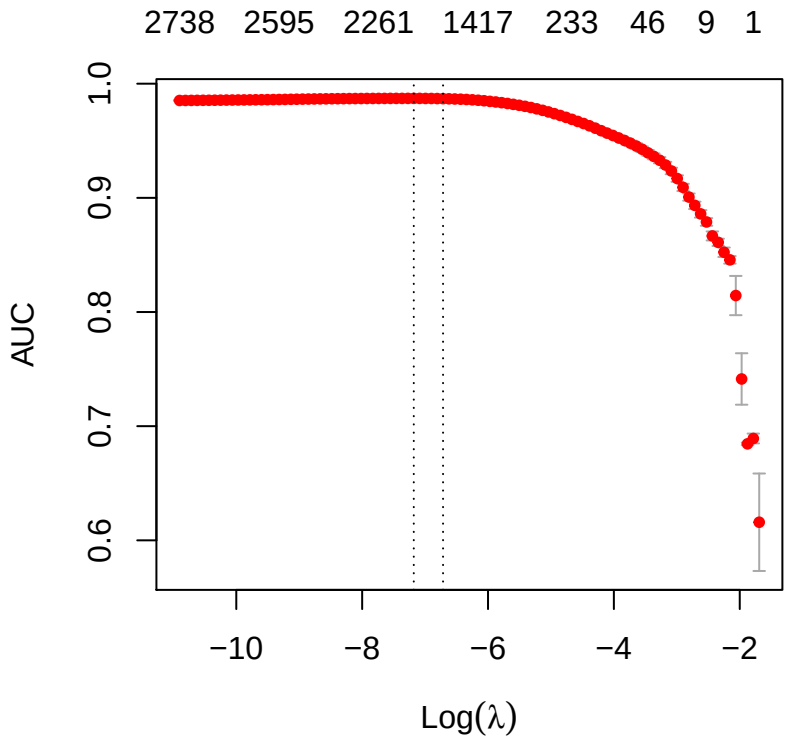
	Method	Accuracy
Accuracy	Standard GLM with 5-fold cross-validation	0.9644

4.2.2 Second algorithm: N-grams with pruning

This algorithm will use the same modelling approach as before, but on a filtered input vocabulary. We first create the vocabulary based on 1-gram and 2-grams, then impose some restrictions on the minimum number of occurrences of each term over all documents, as well as on the maximum proportion of documents which contain this term. The corresponding code looks like this:

```
pruned_vectors <- prune_vocabulary(create_vocabulary(token_train,
                                                     stopwords = "champagne",
                                                     ngram = c(1L, 2L)),
                                   term_count_min = 10,
                                   doc_proportion_max = 0.5) %>%
  vocab_vectorizer(.)
```

We then use these new vectors to create pruned DTMs, both for the training and test datasets, and we run the same model as before on the pruned DTMs. We get the following set of AUCs:



After running the prediction and confusion matrix functions as before, we can add a new accuracy to our table:

Method	Accuracy
Standard GLM with 5-fold cross-validation	0.9644
GLM with 5-fold cross-validation, N-grams with pruning	0.9695

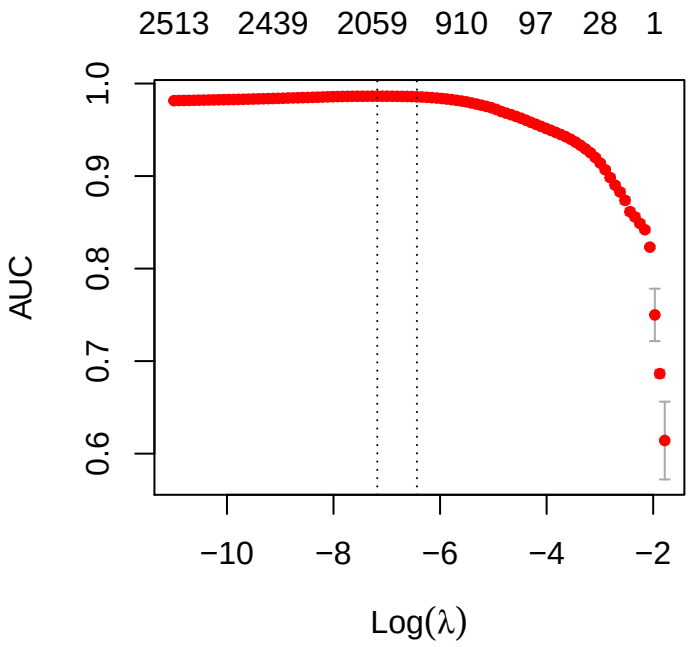
4.2.3 Third algorithm: TF-IDF transformation

Our third algorithm will use another transformation of the base vocabulary, by increasing the weight of terms specific to a few of the documents and decreasing that of terms encountered in a vast majority of documents. That transformation is known as Term Frequency - Inverse Document Frequency (TF-IDF).

We first need to create a neww *TfIdf* model object, then use the *fit_transform* function to apply the TF-IDF transformation to our training DTM. We proceed slightly differently for the testing set, as we should not apply a fitting step on this set, but only the transforming one.

```
tfidf <- TfIdf$new()
dtm_tfidf_train <- dtm_train %>%
  fit_transform(., tfidf)
dtm_tfidf_test <- create_dtm(token_test, vectors) %>%
  transform(tfidf)
```

We once again run our GLM modelling on this transformed set, which gives the following set of AUCs:



After running the prediction and confusion matrix functions, we get our last accuracy measure:

Method	Accuracy
Standard GLM with 5-fold cross-validation	0.9644
GLM with 5-fold cross-validation, N-grams with pruning	0.9695
GLM with 5-fold cross-validation and TF-IDF transformation	0.9664

5 Modelling results and interpretation

5.1 Classification model

Method	Accuracy
Random Forest (1000 trees, node size = 1, mtry = 3)	0.6852
Bagged CART (no parameters)	0.7937
K nearest neighbors (grid for k: 1 to 10, value of 1 selected)	0.8013

The Random Forest appears relatively weaker than the two other classification algorithms, even though its accuracy is still about twice that of a random choice.

We initially coded the country field based on a list of countries (with more than 20 wines reviewed) extracted from the website interface (and not from the last string of the appellation field), leading to only 12 possible values for *country* instead of 32, and the Random Forest accuracy under that construction was similar to that of the two other algorithms, at about 0.79.

Overall, the accuracy reached under the last two models appears decent considering the simplicity of the model, with only three explanatory variables, for a three-bucket classification.

5.2 Regression model

Method	Accuracy
Standard GLM with 5-fold cross-validation	0.9644
GLM with 5-fold cross-validation, N-grams with pruning	0.9695
GLM with 5-fold cross-validation and TF-IDF transformation	0.9664

These three algorithms exhibit very similar accuracies, and at a high average of 0.964.

Transformations of the DTMs so as to refine the vectors hence do not bring noticeable improvement in the already high accuracy of the base vocabulary vectors. This tends to confirm that the base vocabularies of Champagnes and non-Champagnes reviews are rather different from scratch, and enable a reliable identification of the underlying wine. This could be for example due to the different kind of grapes used in both categories of wine, or to references to different kinds of fruit flavors.

6 General conclusion and perspectives

This project has enabled us to practice a wide array of teachings covered in the course of the first eight modules of the HarvardX/EdX Professional Certificate in Data Science, from graphic representations to webscraping to machine learning techniques. Regarding this latter, it has also given us the opportunity to experiment with different approaches of machine learning, with both a classification model and a regression one, and comparisons betweens different algorithms for each of these models.

There is obviously still room for improvement in both models:

1. Regarding the classification model, the accuracy, while decent, is not extremely high either. We have tried adding additional explanatory variables, but the gain in accuracy was marginal (and even negative for *badge*), while computational times were significantly increased at the algorithm training level. Were it not for the bijection between some wine reviewers and appellations (most notably Roger Voss for Champagnes), thus leading to some colinearities with the appellations, we feel adding a "reviewer name" explanatory variable in the model could have lifted the accuracy of these three algorithms;
2. Regarding the regression model, the black box nature of the *cv.glmnet* function makes it difficult to fullt assess the subtleties of the resulting modelling, and hence to evaluate whether some tweakings coul have been made. However, considering the number of vectors used in the modelling, this function appears extremely time-efficient.

Moreover, we regret not having access to a more frequently populated "user average rating" field, which could have led to some additional fascinating modelling (is the user rating influenced by the professional review rating? What about the badge or the price? Is the right skew observed in professional ratings for Champagnes, and even more so vintage ones, also observed among site users?).

7 Technical annex

Computing environment:

CPU	OS	R_Version
Intel64 Family 6 Model 142 Stepping 10, GenuineIntel	Windows_NT	R version 4.0.0 (2020-04-24)

Construction of the provided *wines.RDS* database: webscraping performed between June 11th at 2.15 pm and June 12th at 1.30 am.