Report on Capstone/WineData-project 02/2022

alex-stocker

01.02.2022

Contents

Executive Summary

This report summaries the activities within my second Capstone project. Therefore, I chose to work on a publicly available dataset on wine quality. This dataset includes several measured input variables such as citric acid, residual sugar or alcohol as well as one output variable, (wine) quality, rated by professional wine testers. My overall goal was to apply machine learning to predict wine quality in a best possible way based on the input variables. I will go into more detail on data exploration, visualization and model design in the following subsections of my report.

Introduction and motivation

As I am in general highly interested in product quality, I was looking for a dataset on product quality parameters and corresponding product quality ratings. In particular, I was interested in (subjective) human product quality ratings.

While searching for a publicly available dataset on product quality using Google, I came across two datasets on wine quality (https://archive.ics.uci.edu/ml/datasets/wine+quality) in the UCI Machine Learning Repository operated by the University of California, Irvine (https://uci.edu/).

After carefully reading the description of the wine datasets on the corresponding dataset website, I quickly decided to work on wine quality data. While there are two datasets available in the repository related to red and white variants of the Portuguese "Vinho Verde" wine, I used the white wine dataset, only. As taught in the ML course, I created a training set and a test set and applied all analyses on the training set to avoid biases.

As a first step, I conducted an exploration of the wine dataset (the training dataset) to gain more insight into the data set structure, the variables and the distribution of the variables. I therefore created a series of different plots (including histograms, boxplots, and correlation plots) that were very helpful to me to gain a better understanding.

In a second step, I created a series of models to predict wine quality based on all input variables, starting from a very simple model and ranging to more complex machine learning models. I computed both Accuracy and RMSE as the main performance metrics of comparing the developed machine learning models against each other. While accuracy is a usable metrics for categorical values (and classification problems), RMSE is a proper metrics for continuous values (and regression problems). So, what is the nature of my problem? In my example, the experts rate the wine quality with integers from 1 to 10. Hence, it may be treated as a categorical problem. However, I did not convert the expert rating into a factor before developing the models but left the values as numeric. I can therefore describe the variable as both, since a rating of 5.5 (as predicted by the model) should also be possible, for example. Hence, I compute both, Accuracy and RMSE (like in the MovieLens movie rating example, which can be treated similarly).

Finally, I summarized the results of all predictions in a table and selected the model with the highest prediction accuracy and the lowest RMSE as the winner.

The following subsections provide a summary of all my activities related to the application of machine learning to predict wine quality based on a series of wine measurements.

Method/analysis

Data exploration

The white wine dataset includes **11 input variables that are based on objective chemical tests**, (1) fixed acidity, (2) volatile acidity, (3) citric acid, (4) residual sugar, (5) chlorides, (6) free sulfur dioxide, (7) total sulfur dioxide, (8) density, (9) pH, (10) sulphates, and (11) alcohol, and **one output variable based on subjective human ratings**, (wine) quality, a score between 1 and 10 based on the median of at least 3 evaluations given by human wine experts.

While my major goal is to predict the wine quality as rated by the human experts when wine testing from measured parameters, I want to get better insights into the dataset, first. As taught in the ML courses, I split my dataset into training and test data and perform all analyses on the training data. My **training set** has **3919 observations (80%)** and my **test set** has **979 observations (20%)**.

The table below provides an overview on the training dataset, showing the 11 input variables and one output variable, (wine) quality.

All variables in the dataset are numeric, including the quality variable.

Table 1: Observations in the wine dataset

fixed_acidity	volatile_acidity	citric_acid	residual_sugar	chlorides	free_so2	total_so2	density	pH	sulphates	alcohol	quality
7.0	0.27	0.36	20.7	0.045	45	170	1.0010	3.00	0.45	8.8	6
6.3	0.30	0.34	1.6	0.049	14	132	0.9940	3.30	0.49	9.5	6
8.1	0.28	0.40	6.9	0.050	30	97	0.9951	3.26	0.44	10.1	6
7.2	0.23	0.32	8.5	0.058	47	186	0.9956	3.19	0.40	9.9	6
6.2	0.32	0.16	7.0	0.045	30	136	0.9949	3.18	0.47	9.6	6
6.3	0.30	0.34	1.6	0.049	14	132	0.9940	3.30	0.49	9.5	6

```
## tibble [3,919 x 12] (S3: tbl_df/tbl/data.frame)
##  $ fixed_acidity   : num [1:3919] 7 6.3 8.1 7.2 6.2 6.3 8.1 8.1 8.6 7.9 ...
##  $ volatile_acidity: num [1:3919] 0.27 0.3 0.28 0.23 0.32 0.3 0.22 0.27 0.23 0.18 ...
##  $ citric_acid     : num [1:3919] 0.36 0.34 0.4 0.32 0.16 0.34 0.43 0.41 0.4 0.37 ...
##  $ residual_sugar  : num [1:3919] 20.7 1.6 6.9 8.5 7 1.6 1.5 1.45 4.2 1.2 ...
##  $ chlorides       : num [1:3919] 0.045 0.049 0.05 0.058 0.045 0.049 0.044 0.033 0.035 0.04 ...
##  $ free_so2        : num [1:3919] 45 14 30 47 30 14 28 11 17 16 ...
##  $ total_so2       : num [1:3919] 170 132 97 186 136 132 129 63 109 75 ...
##  $ density         : num [1:3919] 1.001 0.994 0.995 0.996 0.995 ...
##  $ pH              : num [1:3919] 3 3.3 3.26 3.19 3.18 3.3 3.22 2.99 3.14 3.18 ...
##  $ sulphates       : num [1:3919] 0.45 0.49 0.44 0.4 0.47 0.49 0.45 0.56 0.53 0.63 ...
##  $ alcohol         : num [1:3919] 8.8 9.5 10.1 9.9 9.6 9.5 11 12 9.7 10.8 ...
##  $ quality         : num [1:3919] 6 6 6 6 6 6 6 5 5 5 ...
```

The following output provides a summary of the training data set and shows basic statistical parameters such as minimum, average, and maximum values for all variables.

```
##  fixed_acidity    volatile_acidity  citric_acid     residual_sugar
##  Min.   : 3.800   Min.   :0.0800   Min.   :0.0000   Min.   : 0.600
##  1st Qu.: 6.300   1st Qu.:0.2100   1st Qu.:0.2700   1st Qu.: 1.700
##  Median : 6.800   Median :0.2600   Median :0.3200   Median : 5.100
##  Mean   : 6.855   Mean   :0.2779   Mean   :0.3339   Mean   : 6.388
##  3rd Qu.: 7.300   3rd Qu.:0.3200   3rd Qu.:0.3800   3rd Qu.: 9.900
##  Max.   :11.800   Max.   :1.1000   Max.   :1.6600   Max.   :65.800
##    chlorides          free_so2        total_so2        density
##  Min.   :0.00900   Min.   :  2.00   Min.   :  9.0   Min.   :0.9872
##  1st Qu.:0.03600   1st Qu.: 23.00   1st Qu.:109.0   1st Qu.:0.9917
##  Median :0.04300   Median : 34.00   Median :134.0   Median :0.9938
##  Mean   :0.04569   Mean   : 35.21   Mean   :138.3   Mean   :0.9940
##  3rd Qu.:0.05000   3rd Qu.: 46.00   3rd Qu.:167.0   3rd Qu.:0.9961
##  Max.   :0.27100   Max.   :146.50   Max.   :366.5   Max.   :1.0390
##        pH          sulphates         alcohol         quality
##  Min.   :2.720   Min.   :0.2200   Min.   : 8.00   Min.   :3.000
##  1st Qu.:3.085   1st Qu.:0.4100   1st Qu.: 9.50   1st Qu.:5.000
##  Median :3.180   Median :0.4800   Median :10.40   Median :6.000
##  Mean   :3.189   Mean   :0.4916   Mean   :10.52   Mean   :5.875
##  3rd Qu.:3.280   3rd Qu.:0.5500   3rd Qu.:11.40   3rd Qu.:6.000
##  Max.   :3.820   Max.   :1.0800   Max.   :14.20   Max.   :9.000
```

Finally, I checked the dataset for missing values (NAs), but fortunately did not find any.

```
##    fixed_acidity volatile_acidity      citric_acid   residual_sugar
##                0                0                0                0
```

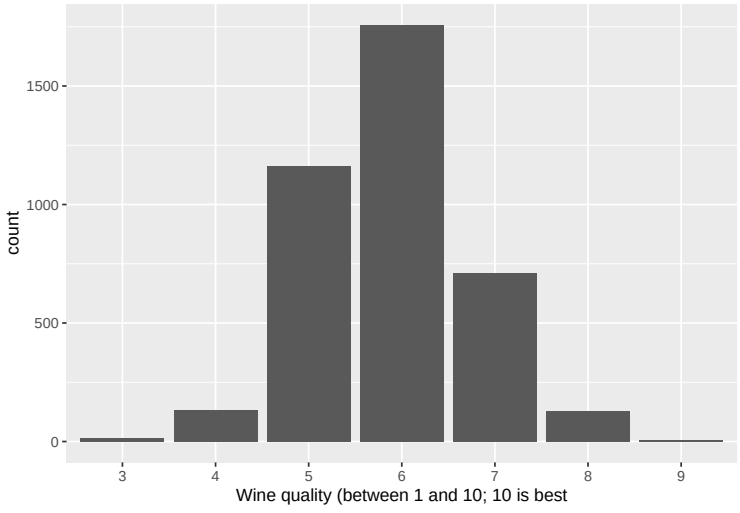
Table 2: Quality of wines

Var1	Freq
3	16
4	134
5	1162
6	1759
7	713
8	130
9	5

```
##        chlorides          free_so2         total_so2           density
##                0                 0                 0                 0
##               pH         sulphates           alcohol           quality
##                0                 0                 0                 0
```

Data visualisation

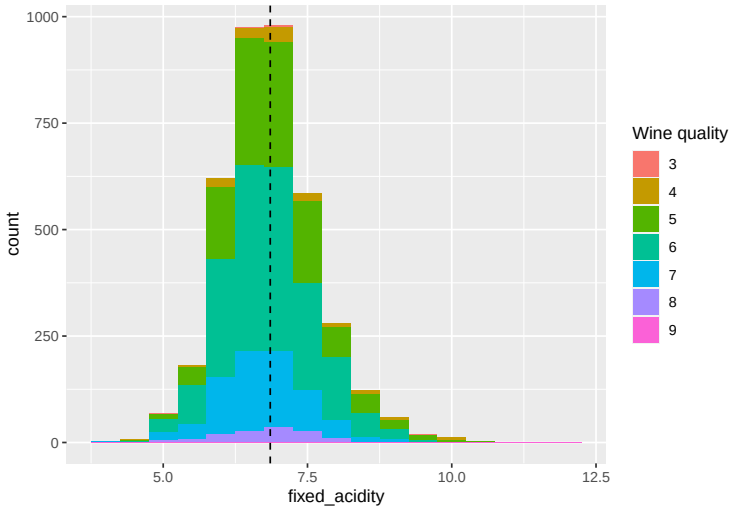
In the next step, I applied several plots to get better insights into the dataset, including histograms. In a first step, I created a plot of the quality variable. The plot shows a rather centered distribution of the wine quality variable around the median value 6. Most experts rated the wines more in the midfield.

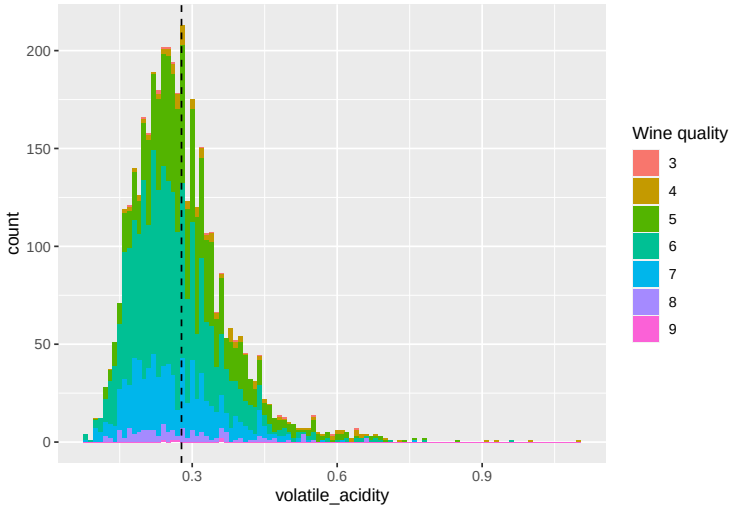


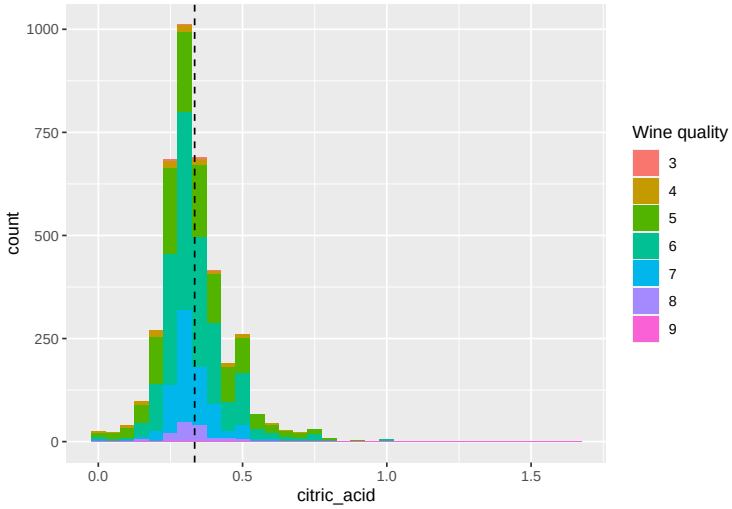
The majority of experts rated the wines they have tested with 5, 6 and 7. The table below shows that there are almost no really bad wines and also comparatively few really good ones. In the training data, no one rated a wine 1, 2 or 10.

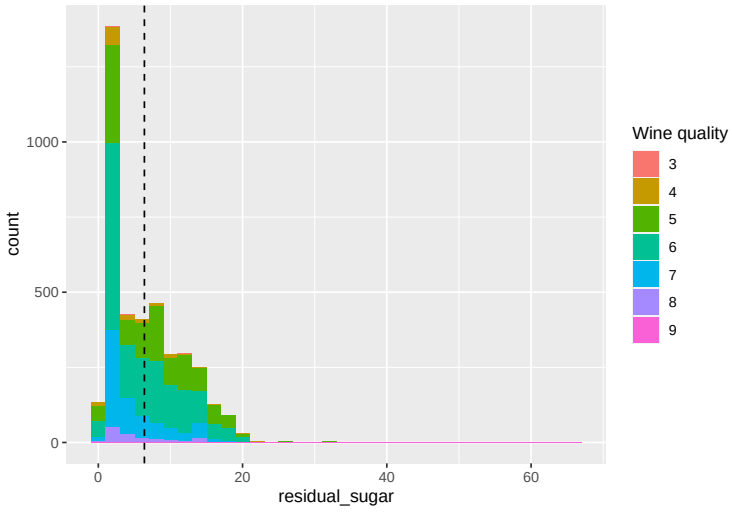
To gain additional insights into the data set and the distribution of the input variables, I plotted all 11 input variables as histograms. I also plotted the median as a vertical line in each histogram and filled the histograms with wine quality information. Therefore, I converted the numerical quality variable into a factor. To avoid repetition of code, I created a function for these 11 plots.

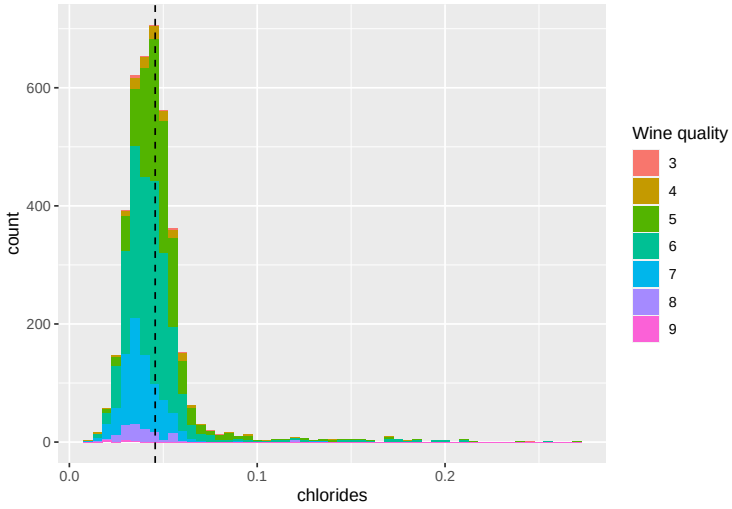
The subsequent plots show the distributions of all input variables as histograms.

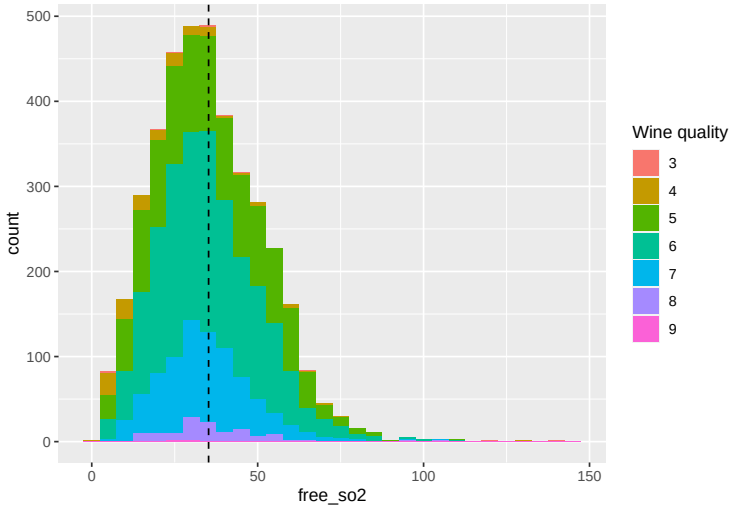

1000
750
500
250
0
count
5.0
7.5
10.0
12.5
fixed_acidity
Wine quality
3
4
5
6
7
8
9

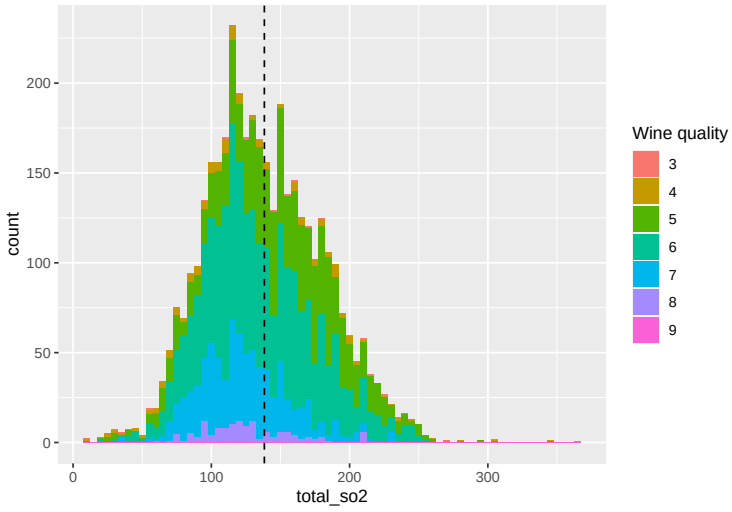

200
150
100
50
0
count
0.3
0.6
0.9
volatile_acidity
Wine quality
3
4
5
6
7
8
9

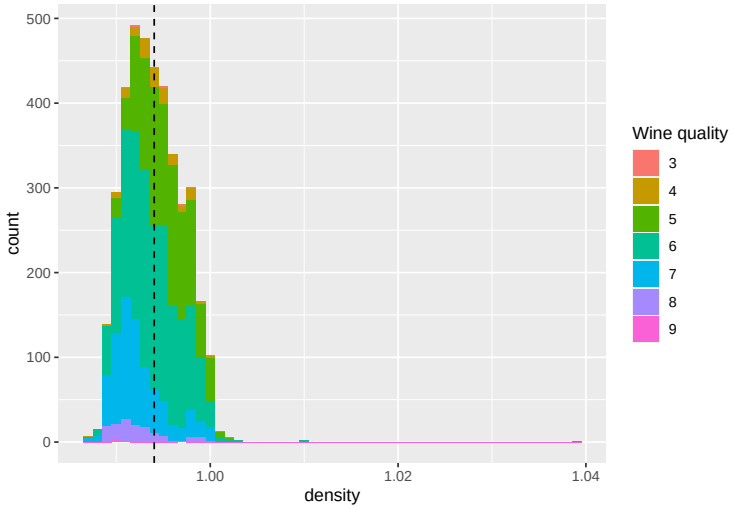

1000
750
500
250
0
count
0.0
0.5
1.0
1.5
citric_acid
Wine quality
3
4
5
6
7
8
9

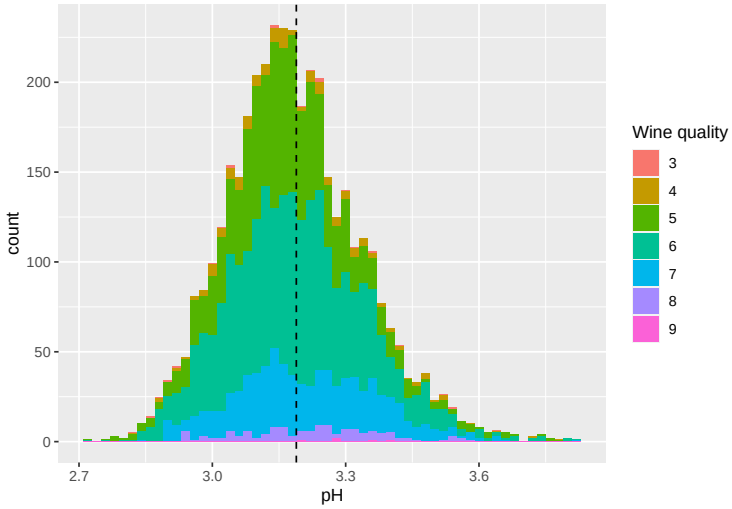

count
1000
500
0
0
20
40
60
residual_sugar
Wine quality
3
4
5
6
7
8
9

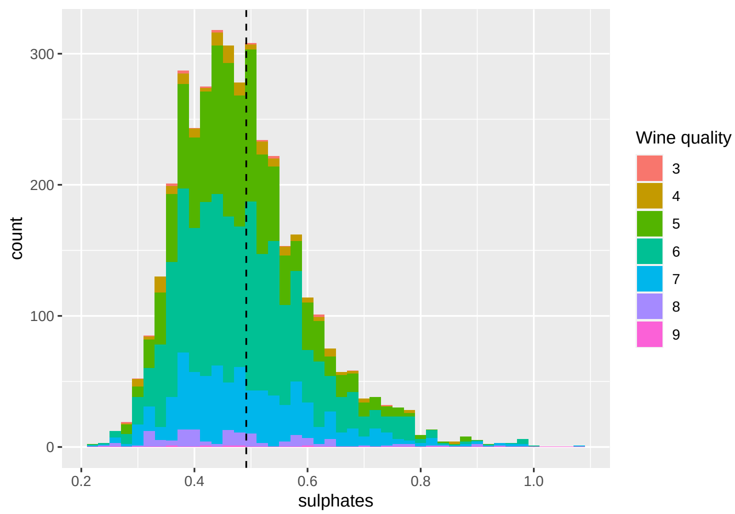

count
600
400
200
0
0.0
0.1
0.2
chlorides
Wine quality
3
4
5
6
7
8
9

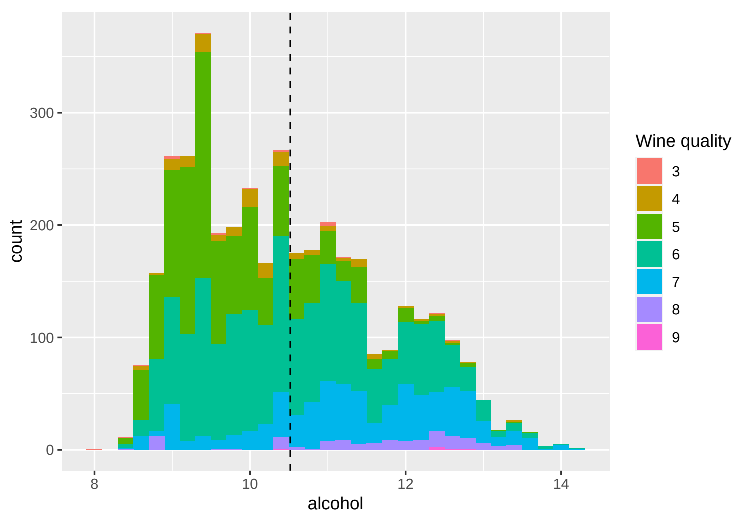

count
500
400
300
200
100
0
0
50
100
150
free_so2
Wine quality
3
4
5
6
7
8
9


count
200
150
100
50
0
0
100
200
300
total_so2
Wine quality
3
4
5
6
7
8
9


count
500
400
300
200
100
0
1.00
1.02
1.04
density
Wine quality
3
4
5
6
7
8
9


count
200
150
100
50
0
2.7
3.0
3.3
3.6
pH
Wine quality
3
4
5
6
7
8
9



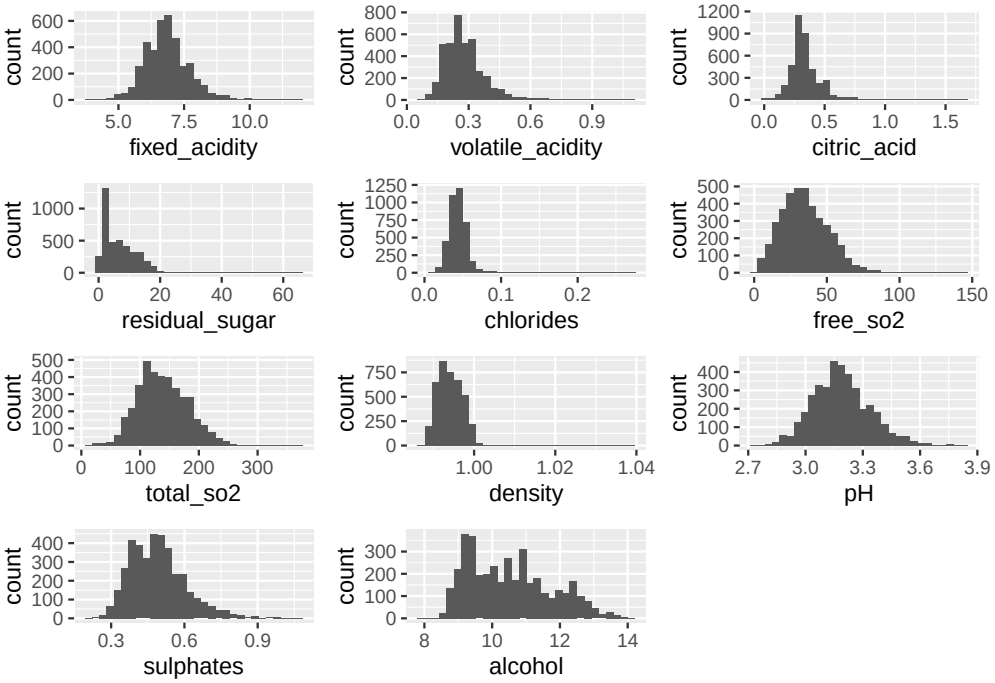


I also created a **faceted plot** with all input variables for a quick overview.

```
## ‘stat_bin()‘ using ‘bins = 30‘. Pick better value with ‘binwidth‘.
## ‘stat_bin()‘ using ‘bins = 30‘. Pick better value with ‘binwidth‘.
## ‘stat_bin()‘ using ‘bins = 30‘. Pick better value with ‘binwidth‘.
## ‘stat_bin()‘ using ‘bins = 30‘. Pick better value with ‘binwidth‘.
## ‘stat_bin()‘ using ‘bins = 30‘. Pick better value with ‘binwidth‘.
## ‘stat_bin()‘ using ‘bins = 30‘. Pick better value with ‘binwidth‘.
## ‘stat_bin()‘ using ‘bins = 30‘. Pick better value with ‘binwidth‘.
## ‘stat_bin()‘ using ‘bins = 30‘. Pick better value with ‘binwidth‘.
## ‘stat_bin()‘ using ‘bins = 30‘. Pick better value with ‘binwidth‘.
## ‘stat_bin()‘ using ‘bins = 30‘. Pick better value with ‘binwidth‘.
## ‘stat_bin()‘ using ‘bins = 30‘. Pick better value with ‘binwidth‘.
```

Table 3: Correlation Matrix

	fixed_acidity	volatile_acidity	citric_acid	residual_sugar	chlorides	free_so2	total_so2	density	pH	sulphates	alcohol	quality
fixed_acidity	1.0000000	-0.0356187	0.2836957	0.0879878	0.0248405	-0.0488531	0.0801337	0.2708552	-0.4239917	-0.0206143	-0.1374676	-0.1168843
volatile_acidity	-0.0356187	1.0000000	-0.1404526	0.0649850	0.0496389	-0.1018808	0.0822483	0.0190761	-0.0192061	-0.0353493	0.0924021	-0.1890719
citric_acid	0.2836957	-0.1404526	1.0000000	0.0882394	0.1088160	0.0975080	0.1167675	0.1466961	-0.1580438	0.0621568	-0.0782724	-0.0110849
residual_sugar	0.0879878	0.0649850	0.0882394	1.0000000	0.0824840	0.3007373	0.3940930	0.8388519	-0.1888509	-0.0325911	-0.4458216	-0.1026977
chlorides	0.0248405	0.0496389	0.1088160	0.0824840	1.0000000	0.1086663	0.2047143	0.2586204	-0.0772008	0.0090631	-0.3676168	-0.2083639
free_so2	-0.0488531	-0.1018808	0.0975080	0.3007373	0.1086663	1.0000000	0.6141917	0.2952679	-0.0069809	0.0563580	-0.2603097	0.0116184
total_so2	0.0801337	0.0822483	0.1167675	0.3940930	0.2047143	0.6141917	1.0000000	0.5220846	0.0029535	0.1287744	-0.4531759	-0.1808521
density	0.2708552	0.0190761	0.1466961	0.8388519	0.2586204	0.2952679	0.5220846	1.0000000	-0.0869164	0.0706324	-0.7738367	-0.3099692
pH	-0.4239917	-0.0192061	-0.1580438	-0.1888509	-0.0772008	-0.0069809	0.0029535	-0.0869164	1.0000000	0.1623329	0.1234847	0.0992475
sulphates	-0.0206143	-0.0353493	0.0621568	-0.0325911	0.0090631	0.0563580	0.1287744	0.0706324	0.1623329	1.0000000	-0.0093088	0.0447373
alcohol	-0.1374676	0.0924021	-0.0782724	-0.4458216	-0.3676168	-0.2603097	-0.4531759	-0.7738367	0.1234847	-0.0093088	1.0000000	0.4350495
quality	-0.1168843	-0.1890719	-0.0110849	-0.1026977	-0.2083639	0.0116184	-0.1808521	-0.3099692	0.0992475	0.0447373	0.4350495	1.0000000

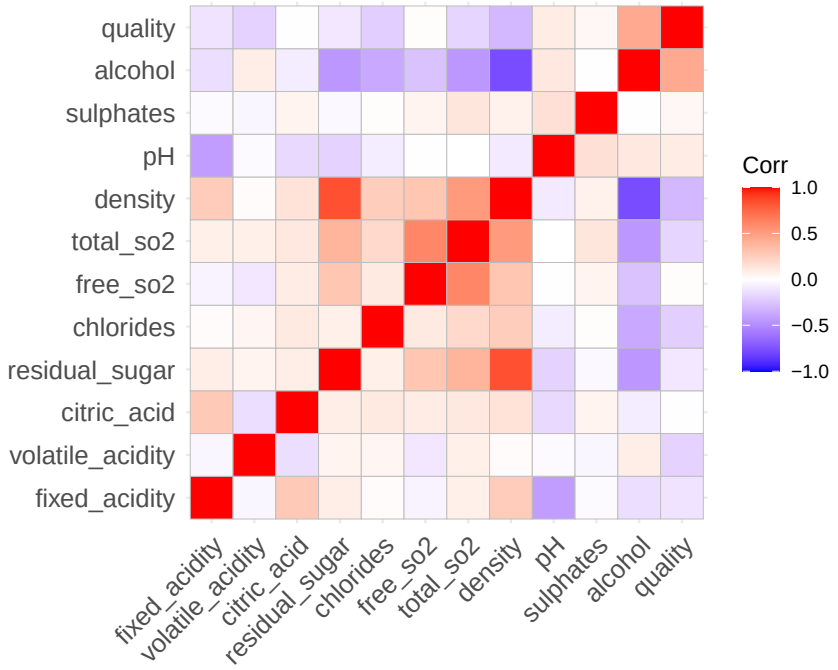


Correlation analysis

In a further step of my data exploration, I wanted to examine the **(linear) correlations** between all variables. Hence, I calculated the correlation matrix and created a visualization with all correlations in the form of a heat map.

The correlation matrix shows the correlations between all variables. Especially interesting for me is the correlation between the objective input variables and the subjective output variable, wine quality.

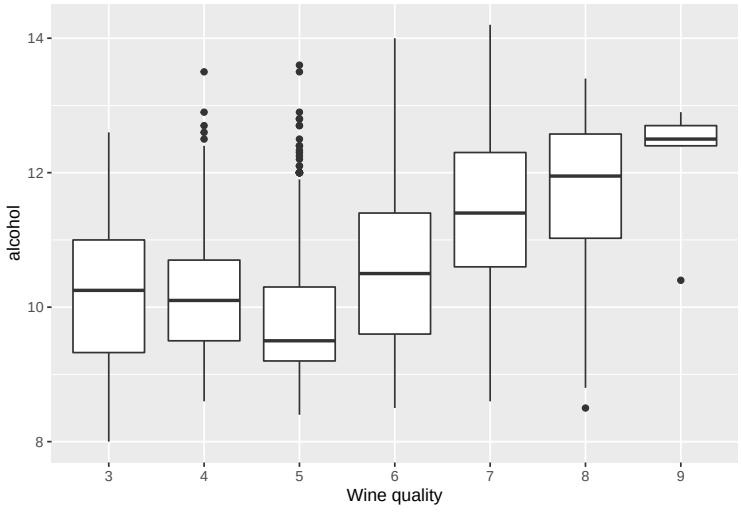
To get a better (visual) overview on the correlations, I decided to plot a correlation **heatmap**. Red color indicates positive correlations, while blue color indicates negative correlations. As previously said, I am especially interested, which wine property (as measured by the chemical tests) has what influence on (perceived) wine quality.



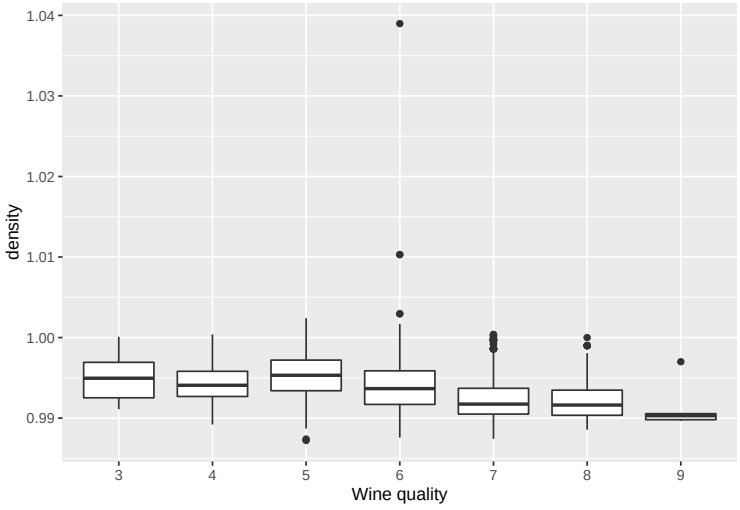
In a further step, I have again clearly presented the correlations between all input variables and the wine quality. The result below suggests that **alcohol has the highest positive (linear) correlation with wine quality** and **density has the highest negative (linear) correlation with wine quality**.

```
##      fixed_acidity volatile_acidity citric_acid residual_sugar chlorides
## [1,]   -0.08678133       -0.1888178  0.01292482    -0.09163214 -0.311248
##        free_so2  total_so2    density       pH  sulphates   alcohol
## [1,] 0.01663614 -0.2086096 -0.3536447 0.104738 0.02808121 0.4413672
```

The subsequent box plot shows the positive correlation between alcohol and wine quality. The higher the alcohol level in the wine, the better the (perceived) wine quality.



The subsequent box plot shows the negative correlation between density and wine quality. The higher the density level in the wine, the lower the (perceived) wine quality.



Data Modeling

In the second part of my report, I focus on designing the models for predicting the wine quality from the input data. Thereby I apply a series of approaches and evaluate their performance. I always present the confusion matrix, the accuracy of the model, and the RMSE of the model.

Median model

I start with a very simple model, the median model. Thereby I set all values to predict to the median of the wine ratings, 6. The table below shows the correlation matrix of this model.

```
##            Reference
## Prediction   3   4   5   6   7   8   9
##          3   0   0   0   0   0   0   0
##          4   0   0   0   0   0   0   0
##          5   0   0   0   0   0   0   0
##          6   4  29 295 439 167  45   0
##          7   0   0   0   0   0   0   0
##          8   0   0   0   0   0   0   0
##          9   0   0   0   0   0   0   0
```

The accuracy of this simple median model is 0.4484168. It may be important to mention that this score is also computed as "no information rate" in the summaries of the other models that will follow in the next subsections. The RMSE of the simple model is 0.900573.

Linear model

In the next step, I computed a linear regression model. The confusion matrix below shows that the Lm model confuses many predictions with neighbouring values. For example, the model was able to correctly predict a rating of "6" 324 times (i.e. 324 wines were predicted to have a quality of 6), but confused 6 with 5 156 times (i.e. 156 wines were predicted to have a quality of 6 but were actually only rated 5) and 6 with 7 120 times (i.e. 120 wines were predicted to have a quality of 6 but were actually rated 7). I will give an explanation for this phenomenon in my discussion section.

```
##            Reference
## Prediction   3   4   5   6   7   8   9
##          3   0   0   0   0   0   0   0
##          4   0   2   3   0   0   0   0
##          5   0  17 131  72   8   0   0
##          6   3  10 156 324 120  26   0
##          7   1   0   5  43  39  19   0
##          8   0   0   0   0   0   0   0
##          9   0   0   0   0   0   0   0
```

The accuracy of this linear regression model is 0.5066394, which is not really good. However, the prediction of the model is better than when using the simple median model. The RMSE of the lm model is 0.7609342.

Generalized (linear) model

In the next step, I computed a generalized linear model. The prediction of this model is of course identical to that of the linear regression model (as I used glm to fit a linear regression model), but I wanted to execute the code, as we learned it in the course.

```
##            Reference
## Prediction   3   4   5   6   7   8   9
##          3   0   0   0   0   0   0   0
##          4   0   2   3   0   0   0   0
##          5   0  17 131  72   8   0   0
##          6   3  10 156 324 120  26   0
##          7   1   0   5  43  39  19   0
##          8   0   0   0   0   0   0   0
##          9   0   0   0   0   0   0   0
```

The accuracy of the generalized linear model is 0.5066394. The RMSE of the glm model is 0.7609342.

Knn model

In the next step, I computed a k-nearest neighbor model. The confusion matrix shows a similar phenomenon to the previously used models. The Knn model is not very effective at predicting the exact rating as made by the human experts, e.g. “6” (and in many cases it predicts the immediate neighbour values, e.g. “5” or “7”).

```
##            Reference
## Prediction   3   4   5   6   7   8   9
##          3   0   0   0   0   0   0   0
##          4   0   0   1   0   0   0   0
##          5   3  12 113  67   8   2   0
##          6   1  17 171 335 125  28   0
##          7   0   0  10  37  34  15   0
##          8   0   0   0   0   0   0   0
##          9   0   0   0   0   0   0   0
```

The accuracy of the k-nearest neighbor model is 0.4923391. The RMSE of the knn model is 0.8098325.

Decision tree model

In the next step, I computed a decision tree model. Also, this model is not very effective in predicting the exact human quality ratings.

```
##            Reference
## Prediction   3   4   5   6   7   8   9
##          3   0   0   0   0   0   0   0
##          4   0   0   0   0   0   0   0
##          5   1  22 176 120  10   2   0
##          6   3   7 112 244  85  18   0
##          7   0   0   7  75  72  25   0
##          8   0   0   0   0   0   0   0
##          9   0   0   0   0   0   0   0
```

The accuracy of the decision tree model is 0.5025536. The RMSE of the decision tree model is 0.7560709.

Random forest

In the next step, I computed a Random forest model. The Random forest model performs much better, than the models computed previously, as also the much higher accuracy suggests.

```
##            Reference
## Prediction   3   4   5   6   7   8   9
##          3   0   0   0   0   0   0   0
##          4   0   1   0   0   0   0   0
##          5   2  23 210  49   0   0   0
##          6   2   5  84 368  59  16   0
##          7   0   0   1  22 108  25   0
##          8   0   0   0   0   0   4   0
##          9   0   0   0   0   0   0   0
```

The accuracy of the Random forest model is 0.7058223. The RMSE of the Random forest model is 0.5772042.

SVM model

In the next and final modeling step, I computed a Support Vector Machine model. Also, this model is not too effective in predicting the exact human quality ratings, but performs better than most of the previous models.

```
##            Reference
## Prediction   3   4   5   6   7   8   9
##          3   0   0   0   0   0   0   0
##          4   0   3   3   0   0   0   0
##          5   1  19 183  90   2   0   0
##          6   3   7 107 322 103  21   0
##          7   0   0   2  27  62  24   0
##          8   0   0   0   0   0   0   0
##          9   0   0   0   0   0   0   0
```

The accuracy of the Support Vector Machine model is 0.5822268. The RMSE of the Svm model is 0.680142.

Table 4: Accuracy of models

Method	Accuracy
Random Forest model	0.7058223
SVM model	0.5822268
Lm model	0.5066394
Glm model	0.5066394
Decision Tree model	0.5025536
KNN model	0.4923391
Median model	0.4484168

Table 5: RMSE of models

Method	RMSE
Random Forest model	0.5772042
SVM model	0.6801420
Decision Tree model	0.7560709
Lm model	0.7609342
Glm model	0.7609342
KNN model	0.8098325
Median model	0.9005730

Results

Finally, I merge my results and create a data frame with the results by ranking for all methods, starting with the best method. All models perform better than the simple mean model. However, the best performing model in terms of "accuracy" and RMSE is the Random Forrest model, followed by the Support Vector Machine model.

Key results of my project:

The best performing model in terms of **Accuracy** is the **Random Forest model** with an accuracy of **0.7058223**.

The best performing model in terms of **RMSE** is the **Random Forest model** with an RMSE of **0.5772042**.

Discussion and conclusion

This report presents my work in the Capstone WineData project related to the development of machine learning models to predict wine quality from several input variables such as density or alcohol in a best possible way. In a first step, I performed some exploratory analyses of the dataset and created a series of plots to get more insights into the distributions of the variables and the correlations of the variables with wine quality. In a second step, starting from a very simple model, I developed a series of models and assessed the accuracy of each. In a third step, I finally identified **Random Forest model** as the best model that was able to achieve an **Accuracy** of **0.7058223** and an **RMSE** of 0.5772042.

The report, and especially the confusion matrices, shows the **inaccuracy of the machine learning models to effectively predict the subjective human ratings** of the wines. Although Random Forest offers the best prediction accuracy, the results are still far from perfect. The presented work clearly shows that human ratings can not be so exact that they can be really efficiently predicted using objective parameters. The confusion matrices show that the algorithms are very efficient in predicting a range (including neighbour ratings) but fail in predicting the exact rating. One limitation of the results is that the wine dataset is not balanced but biased towards the middle.

In future work, the wine dataset could be split into a new category scheme with fewer categories (e.g. bad, medium, excellent) and based on this, models could be developed that allow prediction for one of these categories. It can be assumed that this will be done with higher accuracy, but the unbalanced nature of the dataset will probably be only slightly affected.