Wine quality

Introduction

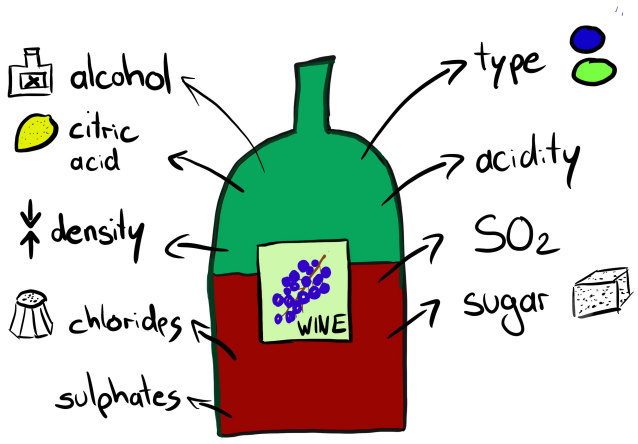
Load the iml package

Before we start, we need to load some libraries:

- `ranger` is a package that supplies the random forest algorithm for classification and regression. We use it to train a model for our data.
- `iml` is a package that implements several model-agnostic interpretation methods that explain the the model behavior and its predictions.
- `ggplot` , `grid.Extra` , and `DataExplorer` will be used for plotting and a quick exploratory data analysis.

```
library('ranger')
library('iml')
library('ggplot2')
library('gridExtra')
library('DataExplorer')
```

Wine Data



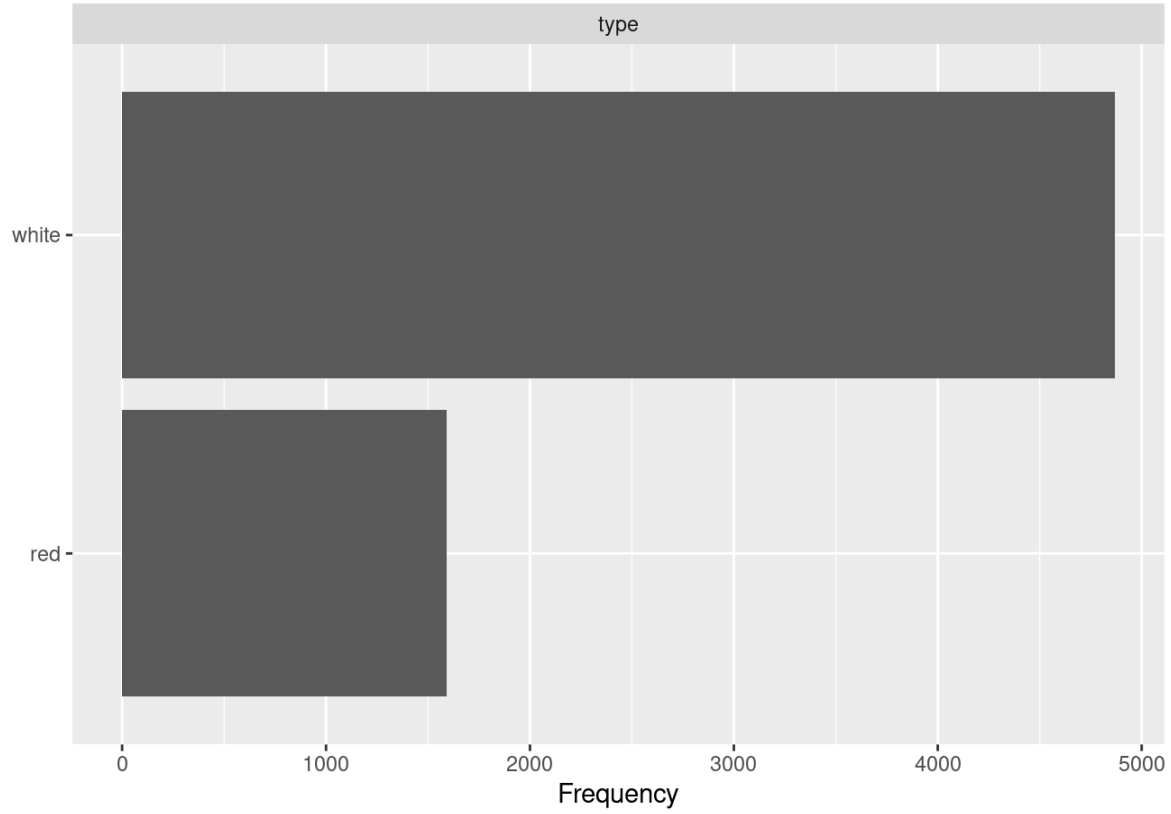
We can import the wine data set from the ‘data’ folder and apply some pre-processing steps.

```
wine_complete = read.csv(file.path("../data", "wine.csv"))
# Removing 36 wines with missing values
wine_complete = na.omit(wine_complete)
# convert wine type from data type character to type factor (create levels/categories for modellin
g purposes)
wine_complete$type = as.factor(wine_complete$type)
```

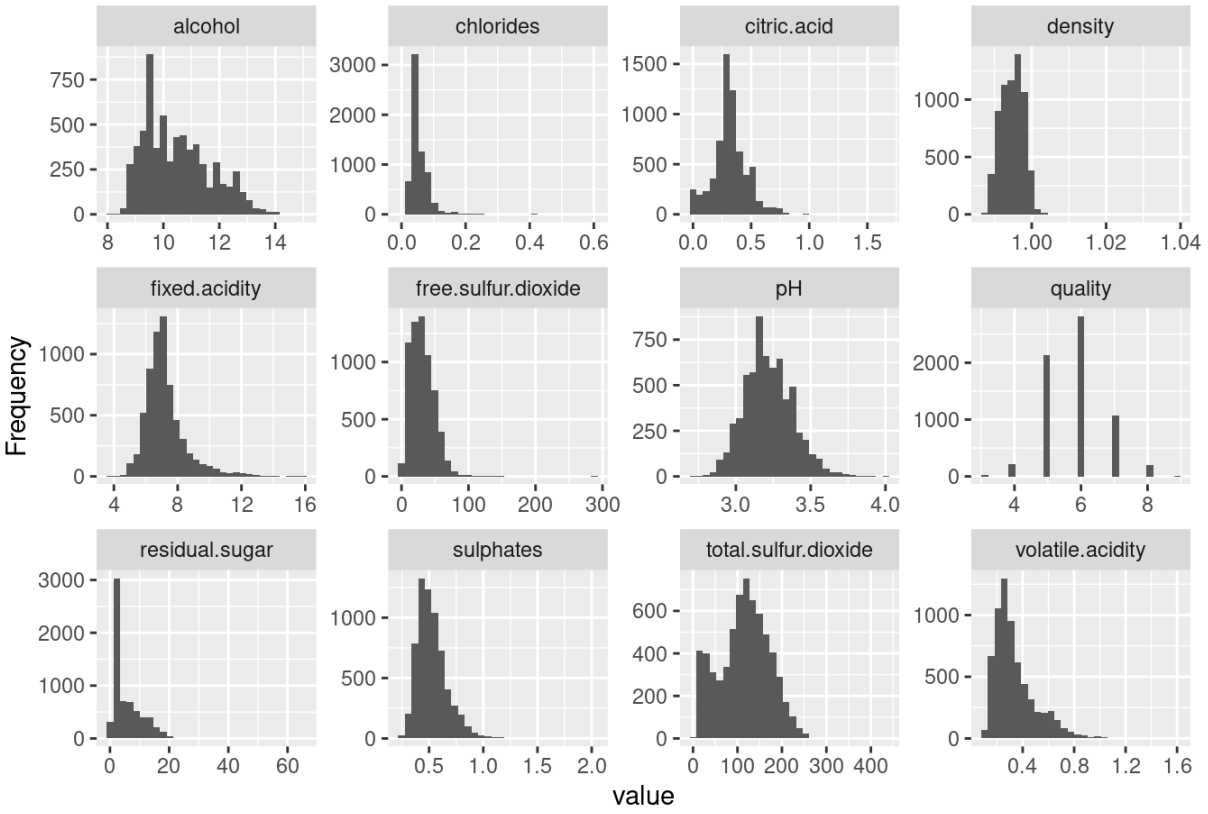
The data set:

- 6500 red and white Portuguese “Vinho Verde” wines (the ratio between white and red is approximately 3:1)
- Features: Physicochemical properties
- Quality assessed by blind tasting, from 0 (very bad) to 10 (excellent)

```
plot_bar(wine_complete)
```

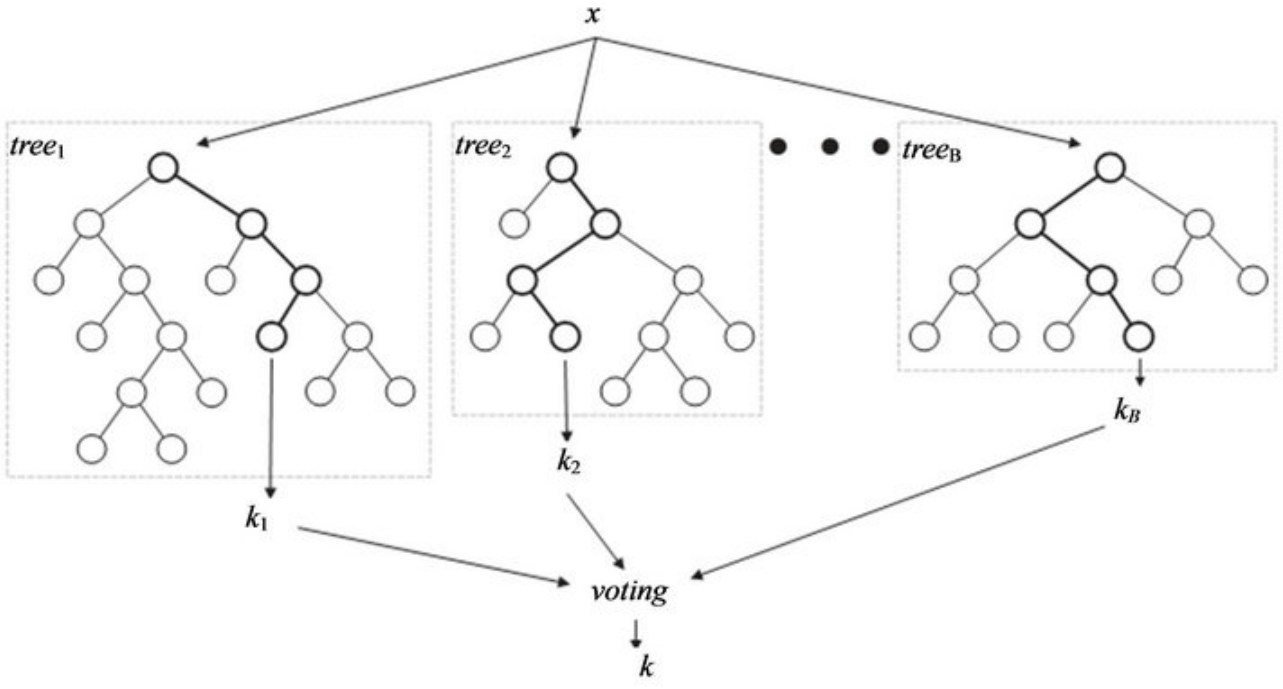



```
plot_histogram(wine_complete)
```



Model

Finally, we apply machine learning to predict the quality of wine using the random forest algorithm.



Some interpretation methods make more sense if they are applied on test data (i.e., data that was not used to fit a model). Hence, we split up the data into a training set on which the random forest is trained and a test set which we will use to analyze our model using several interpretation methods.

We store the trained model in an object `rfmod` and use 1000 observations for the test set and IML analysis.

```
# sample 1000 observations randomly as test set and use all other observations for training the mo
del
set.seed(1)
test_ind = sample(1:nrow(wine_complete), size = 1000, replace = FALSE)
train_ind = setdiff(1:nrow(wine_complete), test_ind)
wine_train = wine_complete[train_ind,]
wine = wine_complete[test_ind,]

# fit the random forest to be analyzed
rfmod = ranger(quality ~ ., data = wine_train)
```

Exercises

You can copy and paste the previously shown code to import the data and model. Note that for all following exercises, we will use the test data `wine` that contains 1000 observations.

Exercise 0: The Predictor

Create a `Predictor` object using the `iml` package.

▶ **Solution**

Exercise 1 (a): PD and ALE plots

Create PDP and ALE plots for at least 6 features of your dataset you are most interested in how they affect your target variable. Can you see any differences between PDP and ALE plots? Can you explain those differences?

Exercise 1 (b): ICE plots

Create ICE plots for the same features as above and interpret the difference to the PDPs. Are your chosen features interacting with any other features?

Exercise 2 (a): Permutation Feature Importance

Make yourself familiar with the `FeatureImp` method in the **iml** package and calculate the permutation feature importance (PFI) for all features in your dataset. Calculate the PFI on your training and on your test dataset. Do you observe any differences? Interpret your results.

Exercise 2 (b): LOCO

Write a function which calculates the LOCO feauture importance on the test dataset. How does it differ to the PFI results? Explain your observation. Hint: Refit your model on the training data without the feature of interest. Predict with the refitted model on the test dataset and calculate the loss difference between your original model and your refitted model. Repeat for all features.

Exercise 2 (c): Comparison with feature effects

Compare your feature importance values to your feature effect results of the first exercise sheet. Are they conclusive?

Exercise 3 (a): LIME

Choose 2-3 observations from your dataset which you would like to explain. How do the different features influence the predictions of the regarded observations? Use LIME for your analysis and interpret your results. Hint: Use `LocalModel` from the **iml** package.

Exercise 3 (b): Counterfactual Explanations

Your task is to choose an observation in your dataset of which you want to change the label (e.g. credit accepted vs. credit declined in a classification task or change a wine with a rating of 5 to a predicted wine rating of 6 in a regression task). Try to find such a counterfactual by changing as few features as possible (e.g., 1-2 features) and with rather small changes.

Bonus: Try the **counterfactuals** package (https://github.com/dandls/counterfactuals (https://github.com/dandls/counterfactuals)), e.g. with the method `MOCRegr` or `MOCClassif`, and compare your results.

Exercise 4 (a): Shapely Values - local analysis

Choose one or two observations of your dataset which might be worth to explain via Shapley values. Calculate and visualize the Shapley values for these observations and interpret your results.

Exercise 4 (b): Shapely Values - global analysis

Sample 100 observations from your dataset at hand and calculate the Shapley values for these observations

- Calculate the Shapley feature importance for your sample. How does it differ from the PFI?
- Create a Shapley dependency plot for the 6 features you have chosen in the feature effects exercise and compare your plots with the PDPs.