SWE584 – Term Project

Wine Quality Prediction

Final Report

Zeynep Deniz Yılmaz

13.06.2022

GitHub: https://github.com/ZeynepDYilmaz/wine-quality-prediction

Problem

- I want to try to calculate the quality of red wines based on their physical and chemical properties like Ph value, density, acidity, etc.
- Wine tasting can be very subjective and not reliable. By analysing the material properties of wine, I aim to find an overlap between the objective properties of the wines and the subjective quality score.

Dataset

To train my model, I will use a dataset that contains psychochemical values of the "Vinho Verde" wine from northern Portugal, as well as the quality score for each entry that is based on sensory data.

1600 entries for red wine
4899 entries for white wine

Source:
https://archive.ics.uci.edu/ml/datasets/wine+quality
https://www.kaggle.com/datasets/uciml/red-wine-quality-cortez-et-al-2009

Columns:

- fixed acidity
- volatile acidity
- citric acid
- residual sugar
- chlorides
- free sulfur dioxide
- total sulfur dioxide
- density
- pH
- sulphates
- alcohol
- quality (between 0 and 10)

Method

- 1st stage:
  - I will train and test my dataset with different sets using k-fold, then compare the accuracy and the standard deviation of the methods below:
    - Decision trees
    - Random Forest
    - Gaussian Naïve Bayes
- 2nd stage:
  - I will try to optimize the parameters of the method chosen in the first stage using a sample parameters grid and scikit's RandomizedSearchCV.

Stage 1 - Decision trees

- **One off**: the results that where the predicted quality was one point above or below the actual score.
- **Two off**: the results that where the predicted quality was two points above or below the actual score.
- **Fail**: the results that where the predicted quality was more than two points above or below the actual score.

random state	accuracy	one off	two off	fail
1	0.646875	30.3125	4.0625	0.9375
27	0.6	35	4.0625	0.9375
53	0.64375	27.5	7.5	0.625
79	0.634375	29.0625	5.9375	1.5625
105	0.628125	31.5625	5.3125	0.3125
131	0.5875	33.4375	5.9375	1.875
157	0.61875	33.75	3.4375	0.9375
183	0.596875	33.75	4.6875	1.875
209	0.625	32.1875	4.375	0.9375
235	0.61875	31.875	5.625	0.625

Stage 1 - Random forest

random state	accuracy	one off	two off	fail
1	0.7	27.1875	2.8125	0
27	0.71875	25.3125	2.8125	0
53	0.69375	26.5625	4.0625	0
79	0.65625	29.6875	4.375	0.3125
105	0.725	24.375	2.8125	0.3125
131	0.690625	28.125	2.5	0.3125
157	0.675	29.375	3.125	0
183	0.675	28.75	3.125	0.625
209	0.728125	23.4375	3.75	0
235	0.66875	30.625	2.5	0

Stage 1 - Gaussian Naïve Bayes

random state	accuracy	one off	two off	fail
1	0.73125	27.1875	2.8125	0
27	0.7	25.3125	2.8125	0
53	0.709375	26.5625	4.0625	0
79	0.646875	29.6875	4.375	0.3125
105	0.70625	24.375	2.8125	0.3125
131	0.70625	28.125	2.5	0.3125
157	0.6625	29.375	3.125	0
183	0.66875	28.75	3.125	0.625
209	0.725	23.4375	3.75	0
235	0.69375	30.625	2.5	0

Stage 1 - Summary

- Random forest and naïve bayes have a similar accuracy score but slightly different standard deviations. Since the random forest method is slightly more consistent, I will be continuing to stage 2 with **random fores**t method.

method	average accuracy	standard deviation
decision tree	0.62	0.019939
random forest	0.693125	0.01973
naive bayes	0.555	0.025396

Stage 2 – Optimizing parameters

The list of parameters below is used as the sample parameters list for optimization using scikit’s RandomizedSearchCV.

Sample parameters:

n_estimators: [200, 400, 600, 800, 1000, 1200, 1400, 1600, 1800, 2000]

max_features: ['sqrt', 'log2’]

max_depth: [10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, None]

min_samples_split: [2, 3, 5, 7, 10]

min_samples_leaf: [1, 2, 3]

criterion: ['gini', 'entropy', 'log_loss’]

Stage 2 – Results

	values
n_estimators	2000
min_samples_split	2
min_samples_leaf	2
max_features	sqrt
max_depth	50
criterion	entropy

Comparison: initial vs. optimized

- I ran the Random Forest Classifier the same way I implemented the k-fold method in stage 1. Surprisingly, the classifier with the “optimized” parameters did not have a better accuracy score or a standard derivation. This means that a finer tuning is needed for this dataset.

	average accuracy	**standard deviation**
initial	0.697188	0.019734
optimized	0.69375	0.024026

Extra – Random Forest feature importance

	importance
alcohol	0.135423
sulphates	0.118852
total_sulfur_dioxide	0.102659
volatile_acidity	0.102655
density	0.092146
chlorides	0.079657
fixed_acidity	0.0769
pH	0.076007
citric_acid	0.075479
free_sulfur_dioxide	0.070391
residual_sugar	0.069833