PREDICTION OF WINE

Group 8

GROUP MEMBERS

Nguyễn Tương Quỳnh - BI10-154

Lương Nguyễn Việt Sơn - BI10-156

Phạm Đức Thắng - BI10-159

Nguyễn Đức Thắng - BI10-160

Nguyễn Tự Tùng - BI10-187

01

INTRODUCTION

- Problem Declaration
- Objectives
- Dataset Specification

02

DATA EXPLORATION AND PREPROCESSING

03

MODEL RESULTS IMPROVEMENTS

04

CONCLUSIONS AND FUTURE DIRECTION

01

INTRODUCTION

PROBLEM DECLARATION

Using the characteristics available from the dataset to estimate the quality of a dataset including over a thousand red wine bottles.

OBJECTIVES

RANKING

A trustworthy ranking for everyone

FOR COMPANY

To qualify the best product before putting it into the marketplace

FOR CUSTOMER

To have an objective basis when choosing the reasonable red wine

DATASET SPECIFICATION

This datasets is related to red variants of the Portuguese "Vinho Verde" wine. Due to privacy and logistic issues, only physicochemical (inputs) and sensory (the output) variables are available (e.g. there is no data about grape types, wine brand, wine selling price, etc.).

This dataset is also available from the UCI machine learning repository, https://archive.ics.uci.edu/ml/datasets/wine+quality

In total: 1599 rows and 12 columns

02

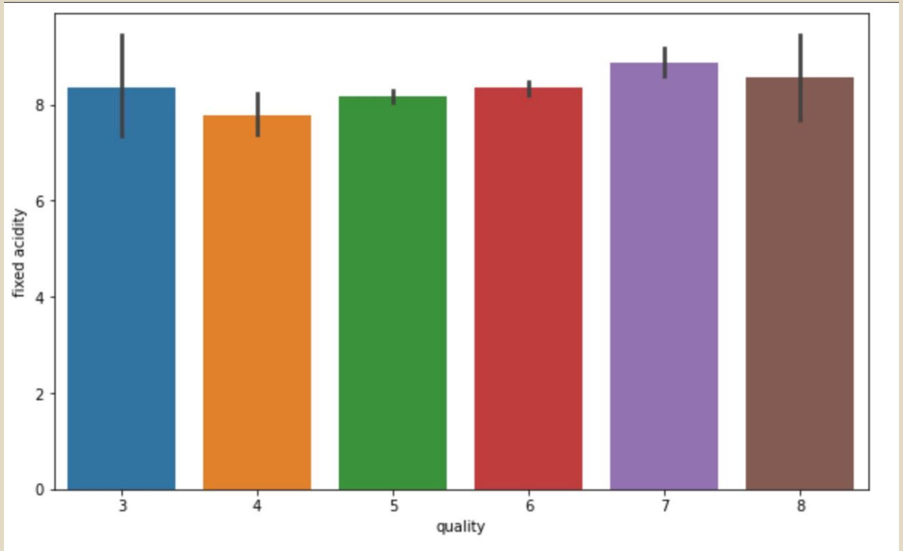
DATA EXPLORATION AND PREPROCESSING

DATA EXPLORATION

DATA EXPLORATION

Fixed Acidity

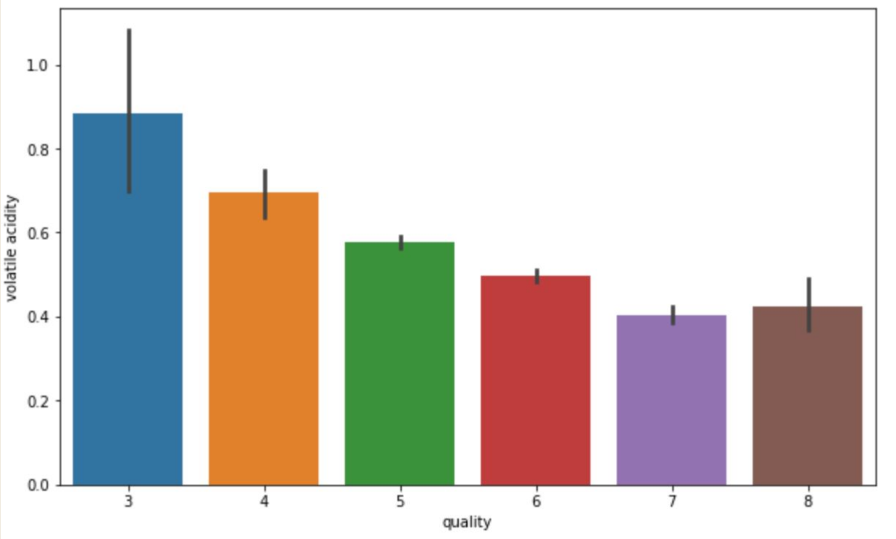
- most acids involved with wine or fixed or nonvolatile (do not evaporate readily)
- fixed acidity does not give any specification to classify the quality.



DATA EXPLORATION

Volatide Acidity

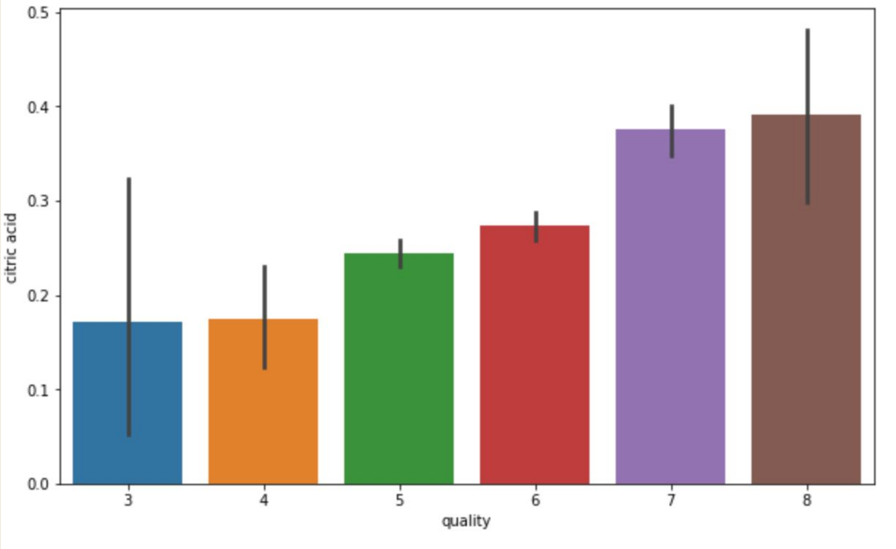
- the amount of acetic acid in wine, which at too high of levels can lead to an unpleasant, vinegar taste
- It is quite a downing trend in the volatile acidity as we go higher the quality



DATA EXPLORATION

Citric Acid

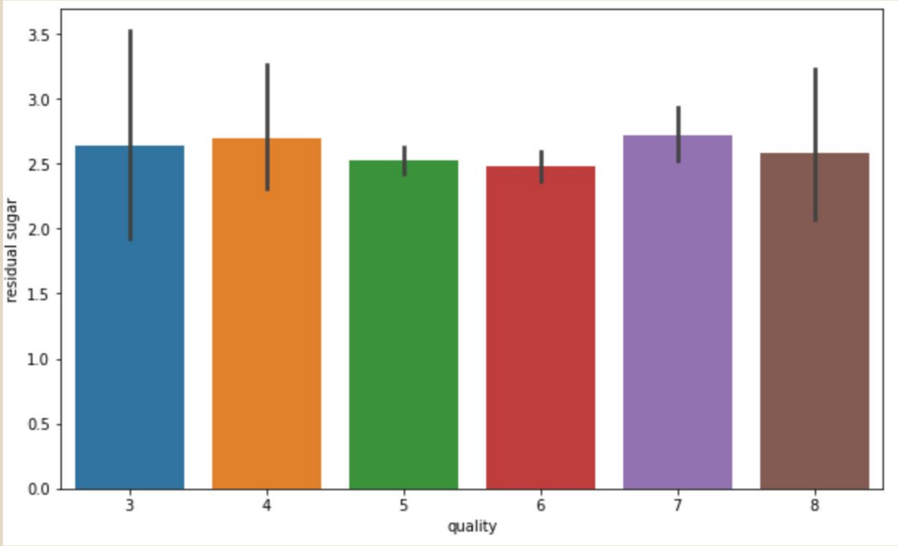
- found in small quantities, citric acid can add 'freshness' and flavor to wines
- Composition of citric acid go higher as we go higher in the quality of the wine



DATA EXPLORATION

Residual Sugar

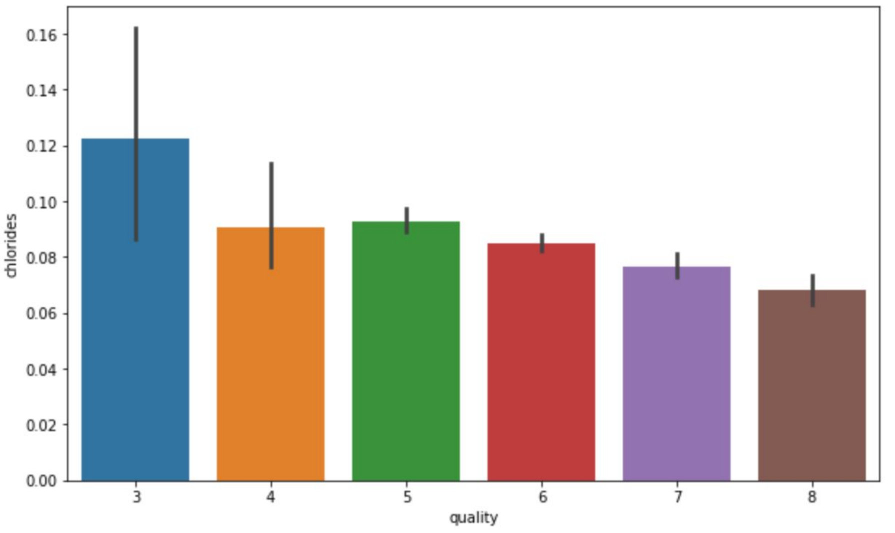
- the amount of sugar remaining after fermentation stops, it's rare to find wines with less than 1 gram/liter and wines with greater than 45 grams/liter are considered sweet
- The residual sugar does not have any effect on the quality



DATA EXPLORATION

Chlorides

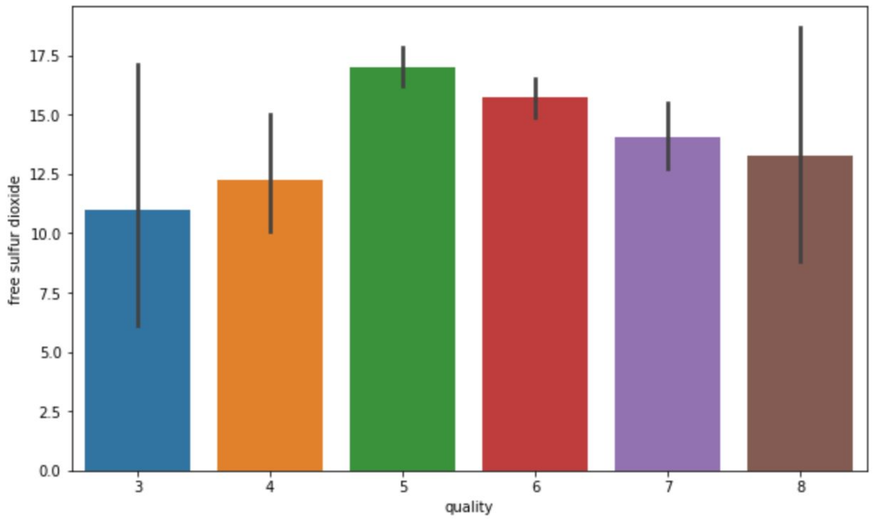
- the amount of salt in the wine
- Composition of chloride also go down as we go higher in the quality of the wine



DATA EXPLORATION

Free Sulfur Dioxide

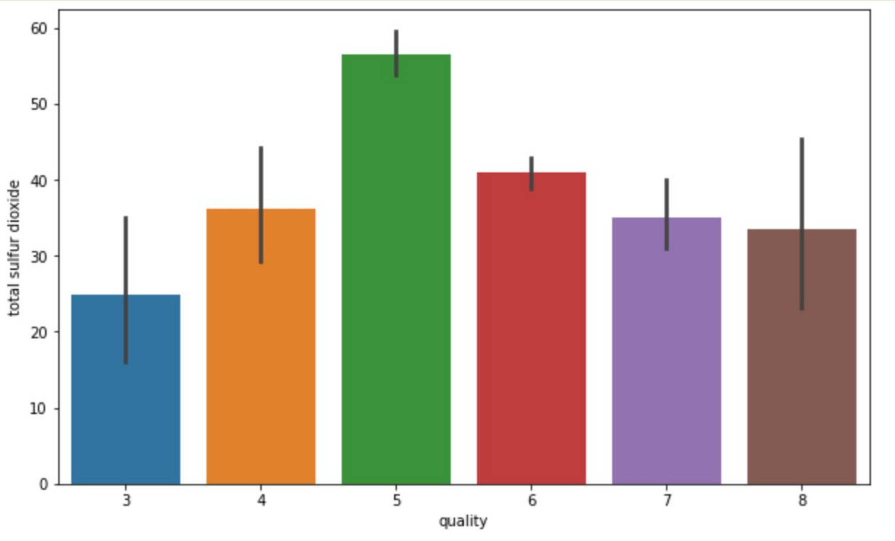
- the free form of SO2 exists in equilibrium between molecular SO2 (as a dissolved gas) and bisulfite ion; it prevents microbial growth and the oxidation of wine
- The free sulfur dioxide does not have much effect on the quality of wine



DATA EXPLORATION

Total Sulfur Dioxide

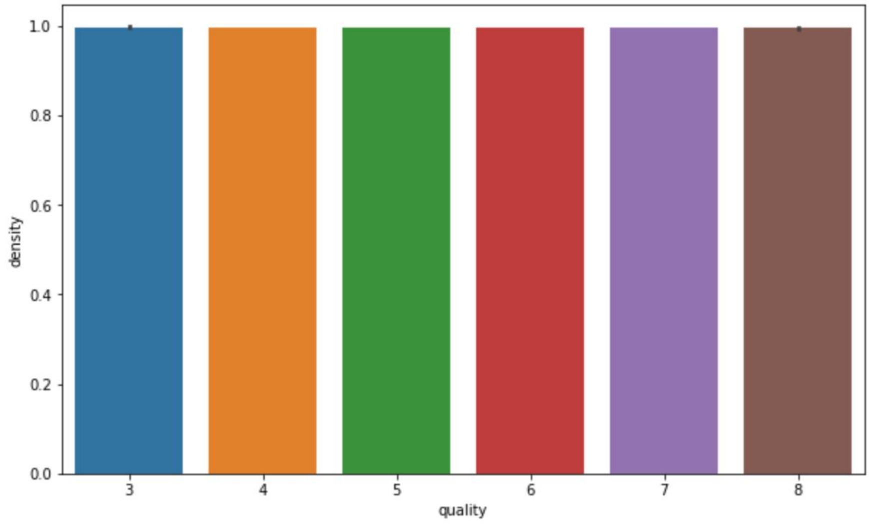
- amount of free and bound forms of S02; in low concentrations, SO2 is mostly undetectable in wine, but at free SO2 concentrations over 50 ppm, SO2 becomes evident in the nose and taste of wine
- The total sulfur dioxide is also not important in creating the quality



DATA EXPLORATION

Density

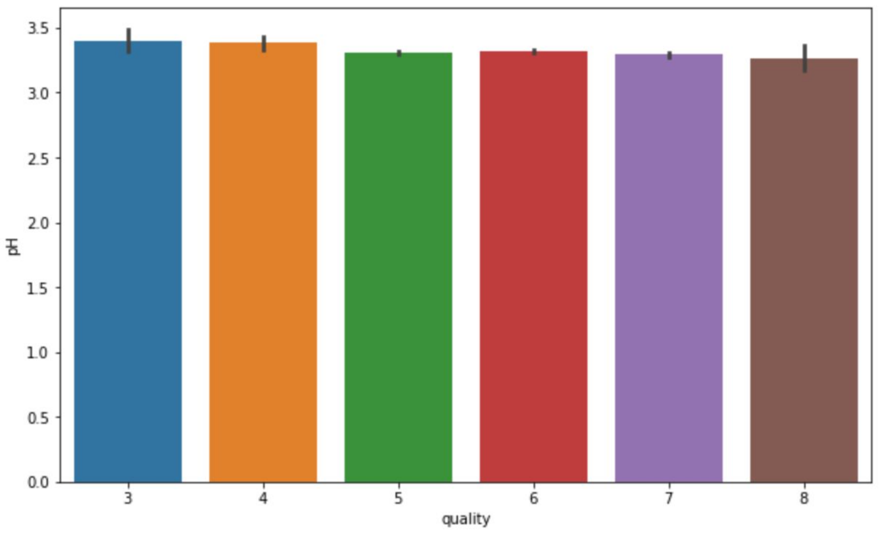
- the density of water is close to that of water depending on the percent alcohol and sugar content
- Density levels are the same in every quality



DATA EXPLORATION

pH

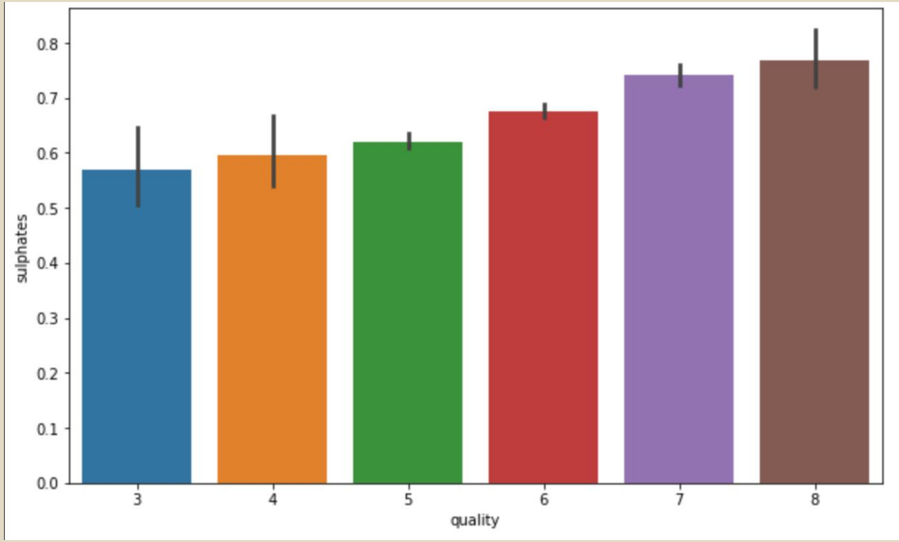
- describes how acidic or basic a wine is on a scale from 0 (very acidic) to 14 (very basic); most wines are between 3-4 on the pH scale
- pH has a slowly slight down trend as the quality goes higher



DATA EXPLORATION

Sulphates

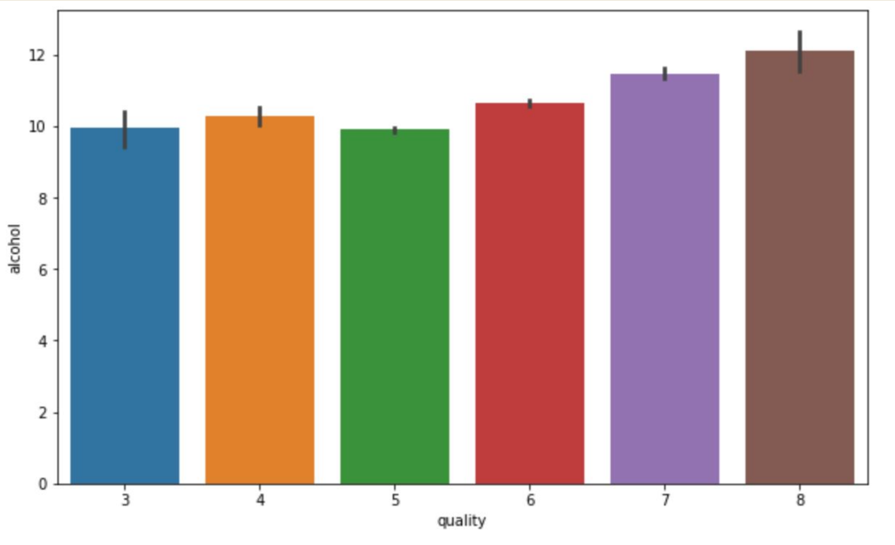
- a wine additive which can contribute to sulfur dioxide gas (S02) levels, which acts as an antimicrobial and antioxidant
- Sulphates level goes higher with the quality of wine



DATA EXPLORATION

Alcohol

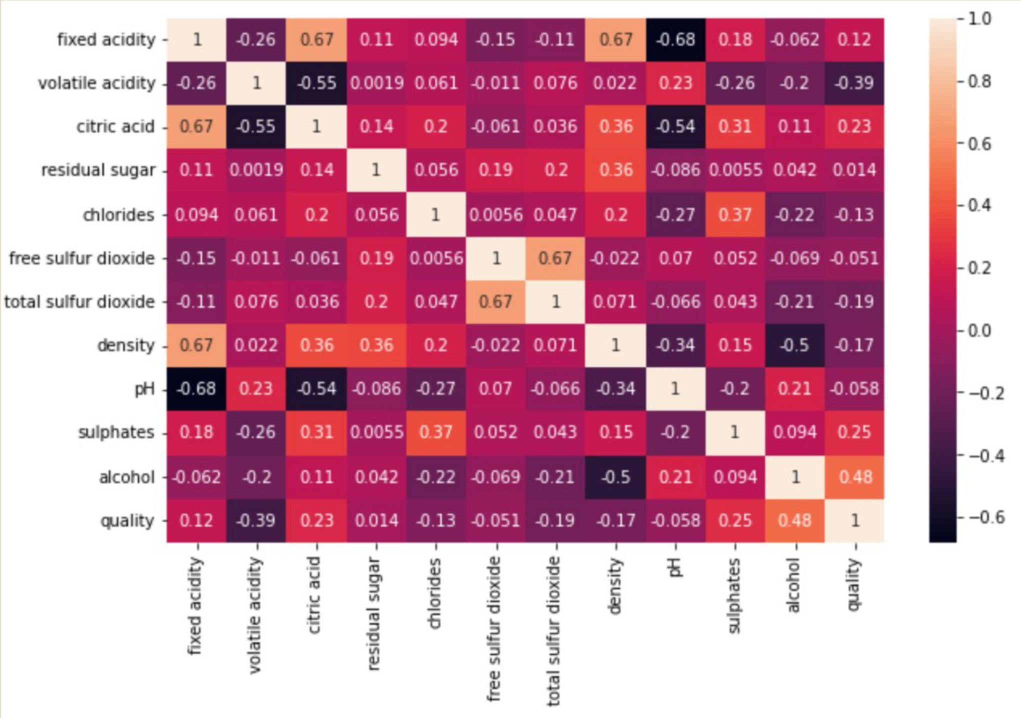
- the percent alcohol content of the wine
- From quality 3-5: the alcohol level does not have impact on quality
- From quality 6-8: alcohol level goes higher as the quality of wine increases



PREPROCESSING

PREPROCESSING

Correlation Matrix

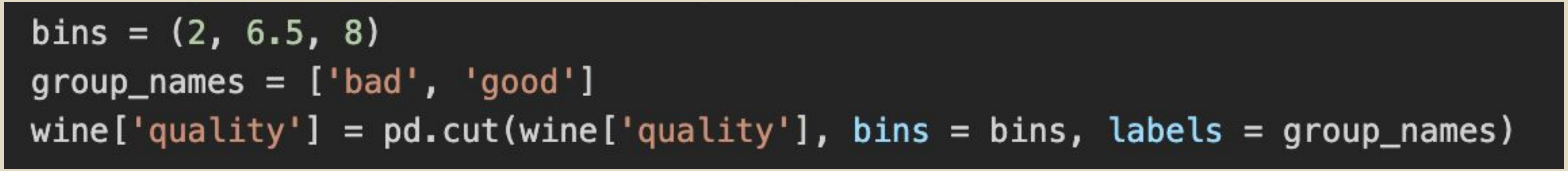


	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
fixed acidity	1	-0.26	0.67	0.11	0.094	-0.15	-0.11	0.67	-0.68	0.18	-0.062	0.12
volatile acidity	-0.26	1	-0.55	0.0019	0.061	-0.011	0.076	0.022	0.23	-0.26	-0.2	-0.39
citric acid	0.67	-0.55	1	0.14	0.2	-0.061	0.036	0.36	-0.54	0.31	0.11	0.23
residual sugar	0.11	0.0019	0.14	1	0.056	0.19	0.2	0.36	-0.086	0.0055	0.042	0.014
chlorides	0.094	0.061	0.2	0.056	1	0.0056	0.047	0.2	-0.27	0.37	-0.22	-0.13
free sulfur dioxide	-0.15	-0.011	-0.061	0.19	0.0056	1	0.67	-0.022	0.07	0.052	-0.069	-0.051
total sulfur dioxide	-0.11	0.076	0.036	0.2	0.047	0.67	1	0.071	-0.066	0.043	-0.21	-0.19
density	0.67	0.022	0.36	0.36	0.2	-0.022	0.071	1	-0.34	0.15	-0.5	-0.17
pH	-0.68	0.23	-0.54	-0.086	-0.27	0.07	-0.066	-0.34	1	-0.2	0.21	-0.058
sulphates	0.18	-0.26	0.31	0.0055	0.37	0.052	0.043	0.15	-0.2	1	0.094	0.25
alcohol	-0.062	-0.2	0.11	0.042	-0.22	-0.069	-0.21	-0.5	0.21	0.094	1	0.48
quality	0.12	-0.39	0.23	0.014	-0.13	-0.051	-0.19	-0.17	-0.058	0.25	0.48	1

PREPROCESSING

Making binary classification for the response variable.

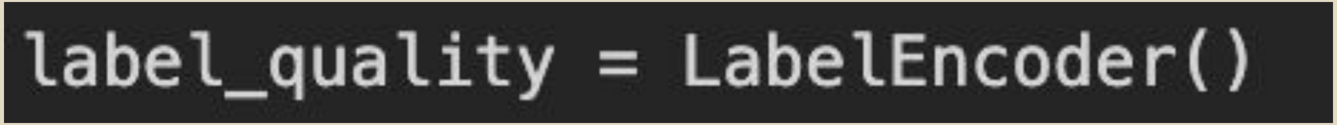
Dividing wine as good and bad by giving the limit for the quality



```
bins = (2, 6.5, 8)
group_names = ['bad', 'good']
wine['quality'] = pd.cut(wine['quality'], bins = bins, labels = group_names)
```

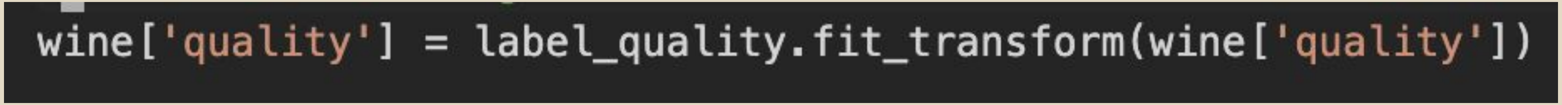
PREPROCESSING

Assign a labels to our quality variable



```
label_quality = LabelEncoder()
```

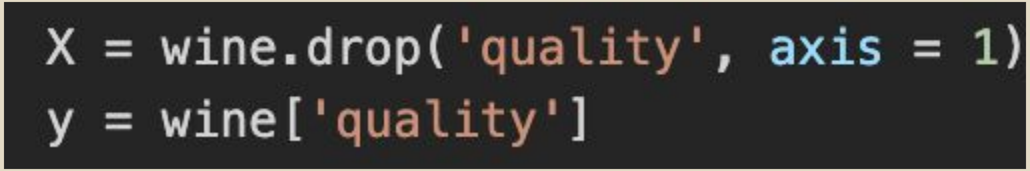
Bad becomes 0 and good becomes 1



```
wine['quality'] = label_quality.fit_transform(wine['quality'])
```

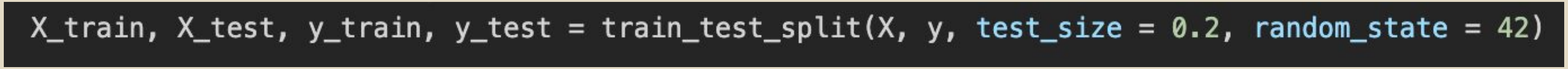
PREPROCESSING

Separate the dataset as response variable and feature variables



```
X = wine.drop('quality', axis = 1)
y = wine['quality']
```

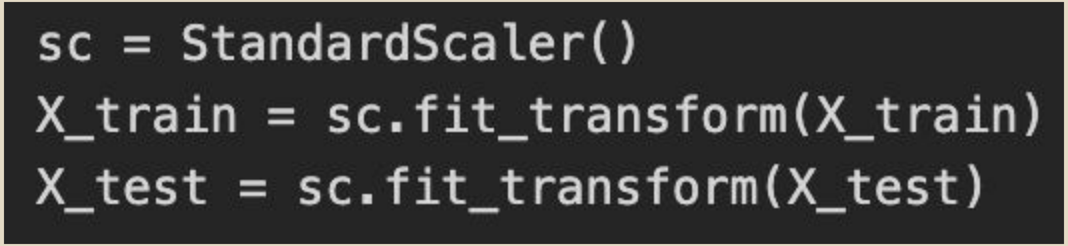
Train and Test splitting of data



```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = 42)
```

PREPROCESSING

Applying Standard scaling to get optimized result



```
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.fit_transform(X_test)
```

03

MODEL AND RESULTS

MODEL

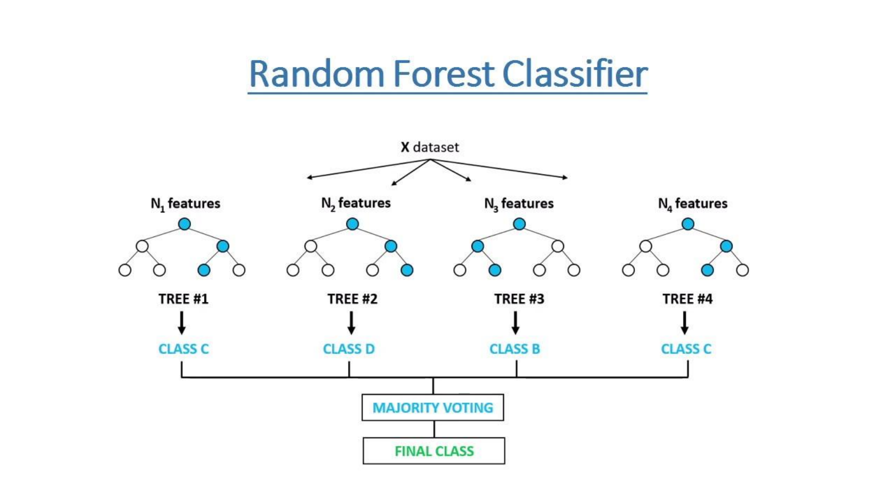
MODEL

3 model

1. Random Forest Classifier
2. Stochastic Gradient Descent Classifier
3. Support Vector Classifier

RANDOM FOREST CLASSIFIER

The term “Random Forest Classifier” refers to the classification algorithm made up of several decision trees. The algorithm uses randomness to build each individual tree to promote uncorrelated forests, which then uses the forest’s predictive powers to make accurate decisions.

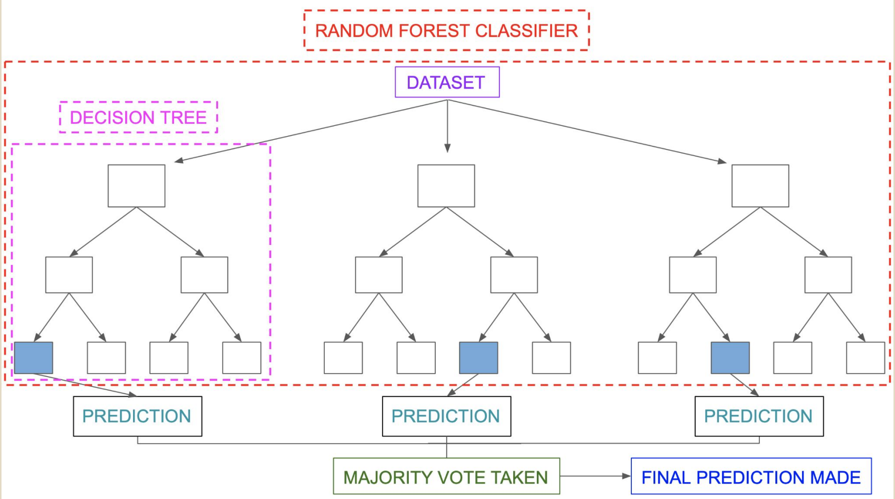


RANDOM FOREST CLASSIFIER

It selects some rows and characteristics from which to draw samples.

It predicts and builds trees depending on the samples.

It combines a number of poorly predicted estimators to produce a powerful prediction and estimation when used together.

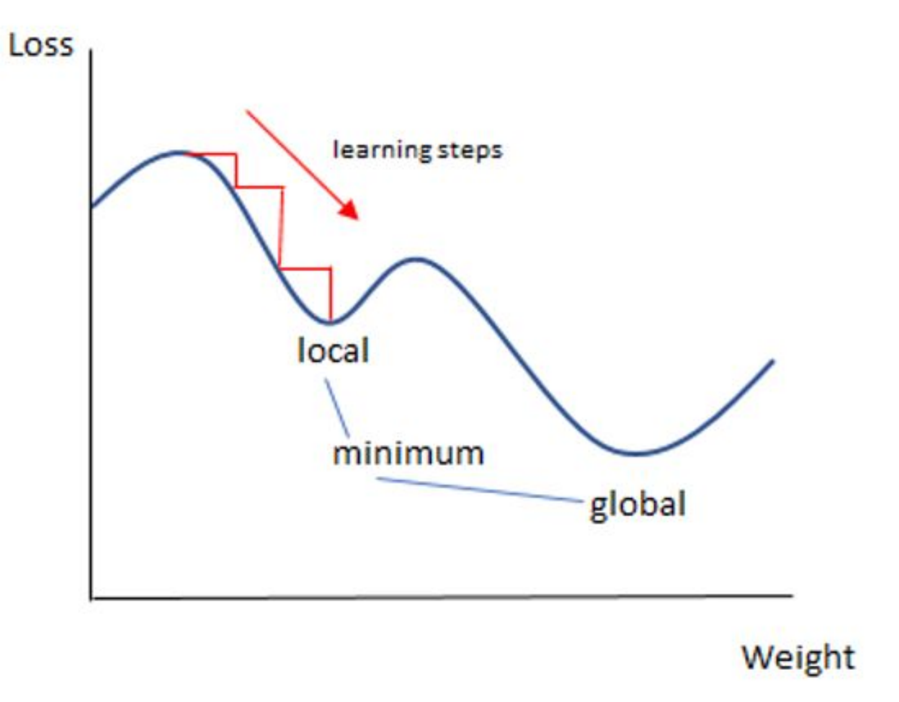


STOCHASTIC GRADIENT DESCENT CLASSIFIER

The iterative approach of stochastic gradient descent (commonly abbreviated SGD) is used to optimize an objective function.

It substitutes the real gradient (derived from the whole data set) with an estimate.

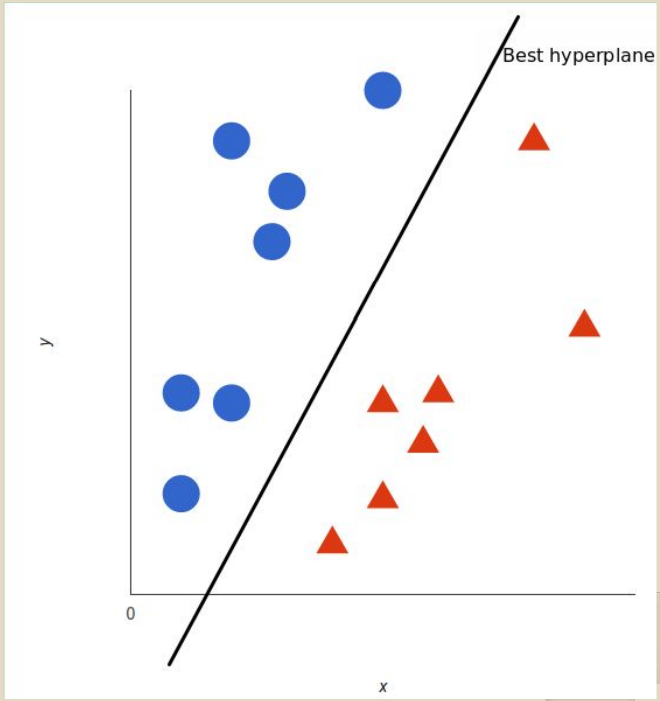
This decreases the computing cost, especially in high-dimensional optimization problems.



SUPPORT VECTOR CLASSIFIER

For two-group classification issues, a support vector machine (SVM) is a supervised machine learning model that employs classification methods.

SVM models can categorize new data after being given sets of labeled training data for each category.



RESULTS

RESULTS

In 3 models

Stochastic Gradient Descent Classifier has the highest RMSE

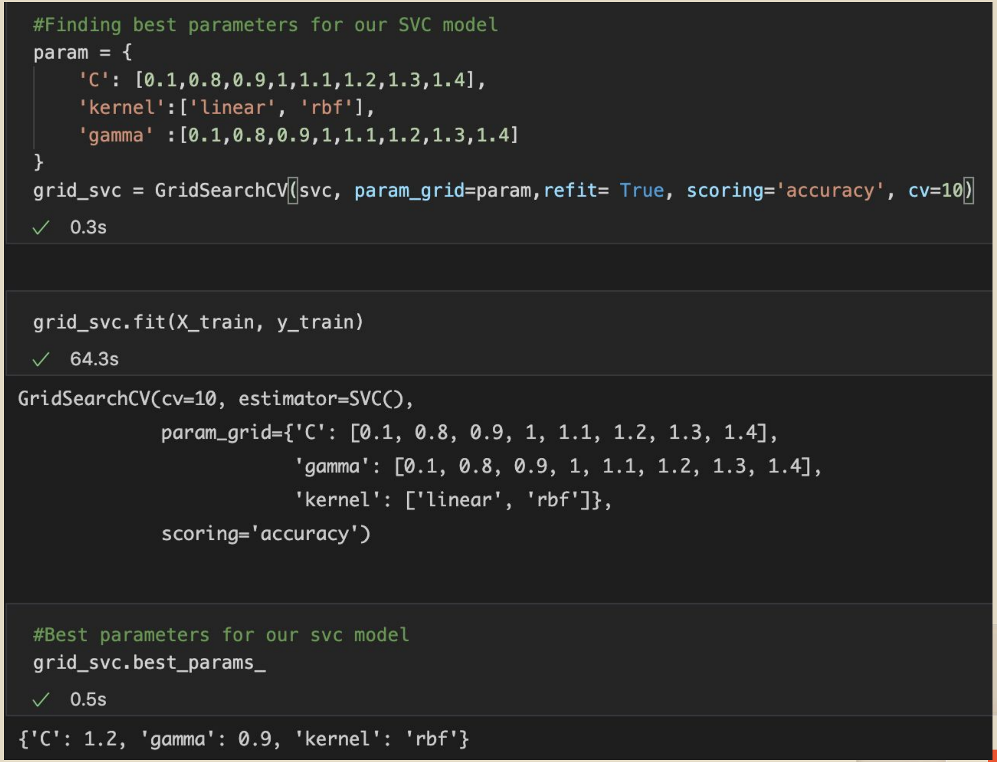
Random Forest Classifier has the lowest RMSE

Algorithm	RMSE
Random Forest Classifier	0.34003676271838607
Stochastic Gradient Descent Classifier	0.3872983346207417
Support Vector Classifier	0.3535533905932738

IMPROVEMENTS

USING GRIDSEARCH CV FOR SVC MODEL

Firstly, we have to find the best parameters for the model



```
#Finding best parameters for our SVC model
param = {
    'C': [0.1,0.8,0.9,1,1.1,1.2,1.3,1.4],
    'kernel':['linear', 'rbf'],
    'gamma' :[0.1,0.8,0.9,1,1.1,1.2,1.3,1.4]
}
grid_svc = GridSearchCV(svc, param_grid=param,refit= True, scoring='accuracy', cv=10)
```

✓ 0.3s

```
grid_svc.fit(X_train, y_train)
```

✓ 64.3s

```
GridSearchCV(cv=10, estimator=SVC(),
             param_grid={'C': [0.1, 0.8, 0.9, 1, 1.1, 1.2, 1.3, 1.4],
                         'gamma': [0.1, 0.8, 0.9, 1, 1.1, 1.2, 1.3, 1.4],
                         'kernel': ['linear', 'rbf']},
             scoring='accuracy')
```

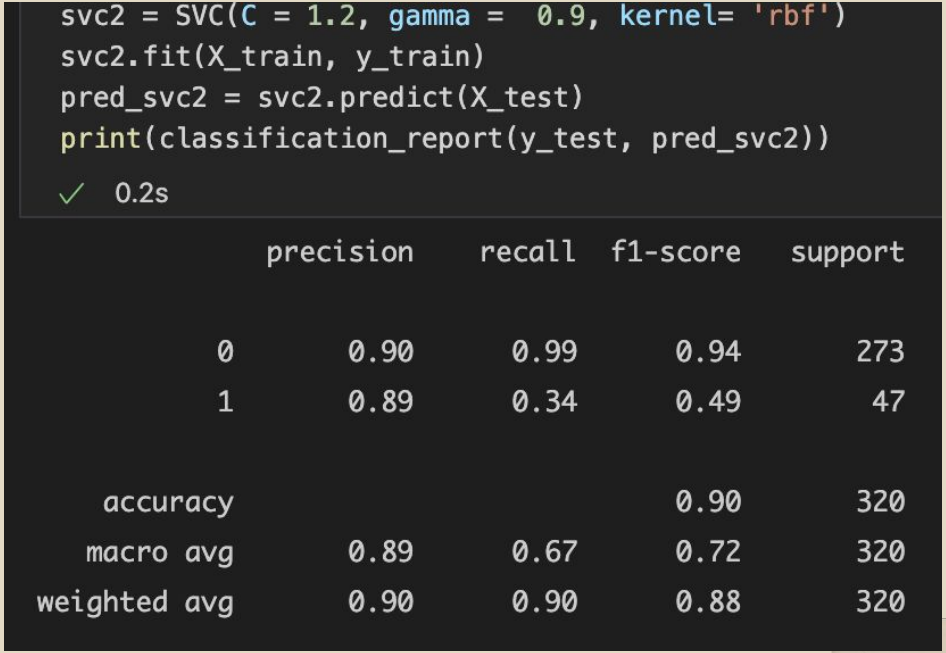
```
#Best parameters for our svc model
grid_svc.best_params_
```

✓ 0.5s

```
{'C': 1.2, 'gamma': 0.9, 'kernel': 'rbf'}
```

USING GRIDSEARCH CV FOR SVC MODEL

SVC improves from 86% to 90% using Grid Search CV



```
svc2 = SVC(C = 1.2, gamma =  0.9, kernel= 'rbf')
svc2.fit(X_train, y_train)
pred_svc2 = svc2.predict(X_test)
print(classification_report(y_test, pred_svc2))
```

✓ 0.2s

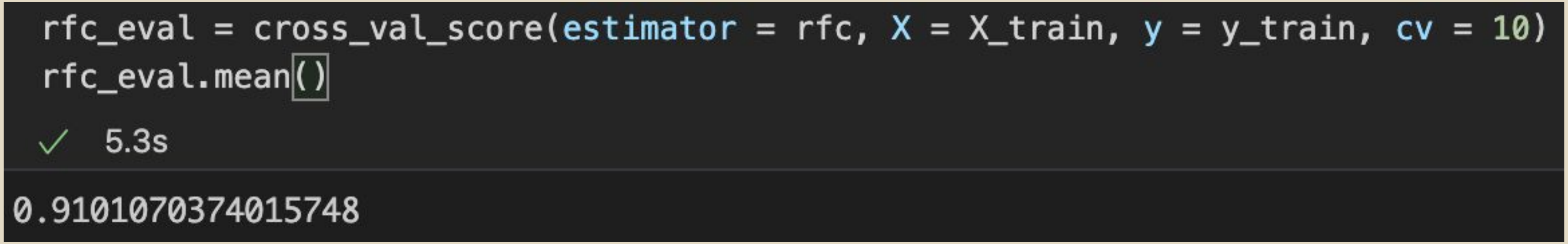
```
              precision    recall  f1-score   support

           0       0.90      0.99      0.94       273
           1       0.89      0.34      0.49        47

    accuracy                           0.90       320
   macro avg       0.89      0.67      0.72       320
weighted avg       0.90      0.90      0.88       320
```

USING CROSS VALIDATION FOR RFC

Random forest accuracy increases from 86% to 91 % using cross validation score



```
rfc_eval = cross_val_score(estimator = rfc, X = X_train, y = y_train, cv = 10)
rfc_eval.mean()

✓  5.3s

0.9101070374015748
```

04

CONCLUSION AND FUTURE DIRECTIONS

CONCLUSIONS FUTURE DIRECTIONS

1 CONCLUSIONS

We try multiple models in order to get as the lowest error as we could

After some improvements, the results has improved significantly

2 FUTURE DIRECTIONS

Try to find the improvements for Stochastic Gradient Descent Classifier

Use more algorithms to find out the best model in this particular case

REFERENCES

- https://cafedev.vn/tu-hoc-ml-dieu-chinh-sieu-tham-so-svm-bang-gridsearchcv-ml/
- https://monkeylearn.com/blog/introduction-to-support-vector-machines-svm/
- https://michael-fuchs-python.netlify.app/2019/11/11/introduction-to-sgd-classifier/#gridsearch
- https://www.kite.com/python/answers/how-to-take-root-mean-square-error-(rmse)-in-python

THANK YOU