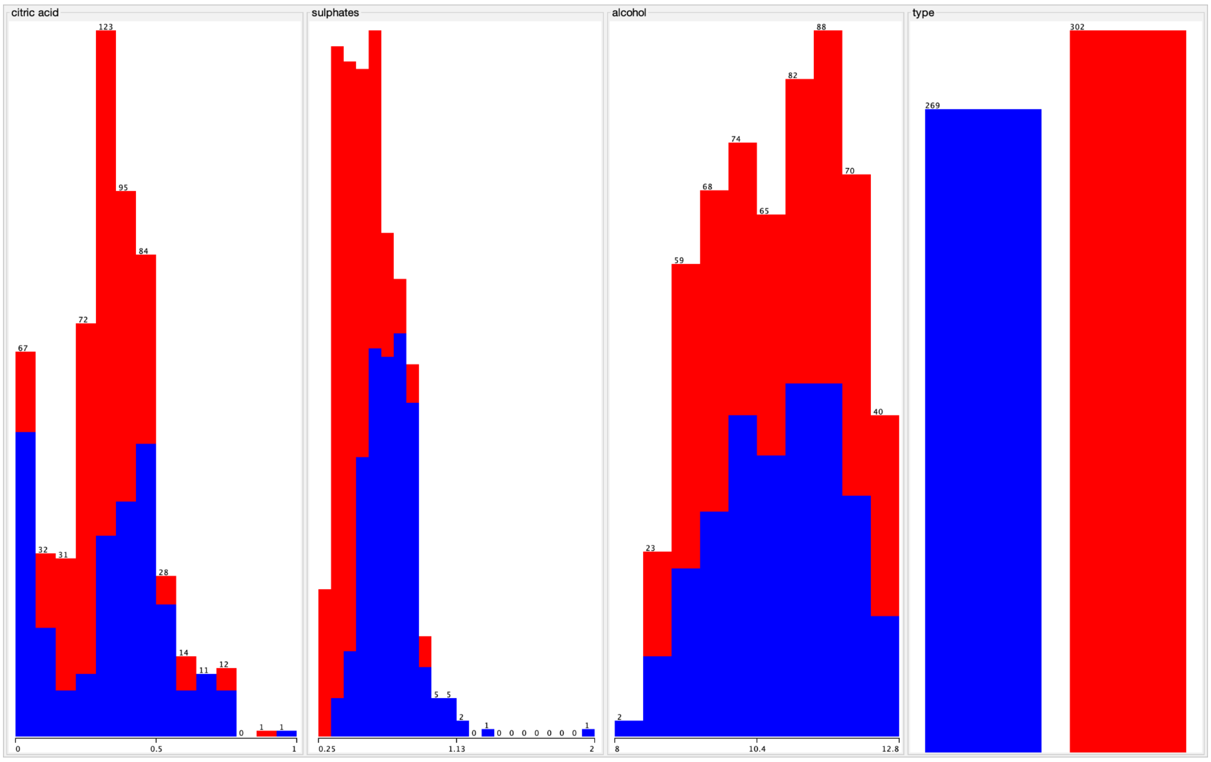
IS_733 Assignment 2

WEKA Task

1.

a) Conditional Distribribution for red-wine.csv is:



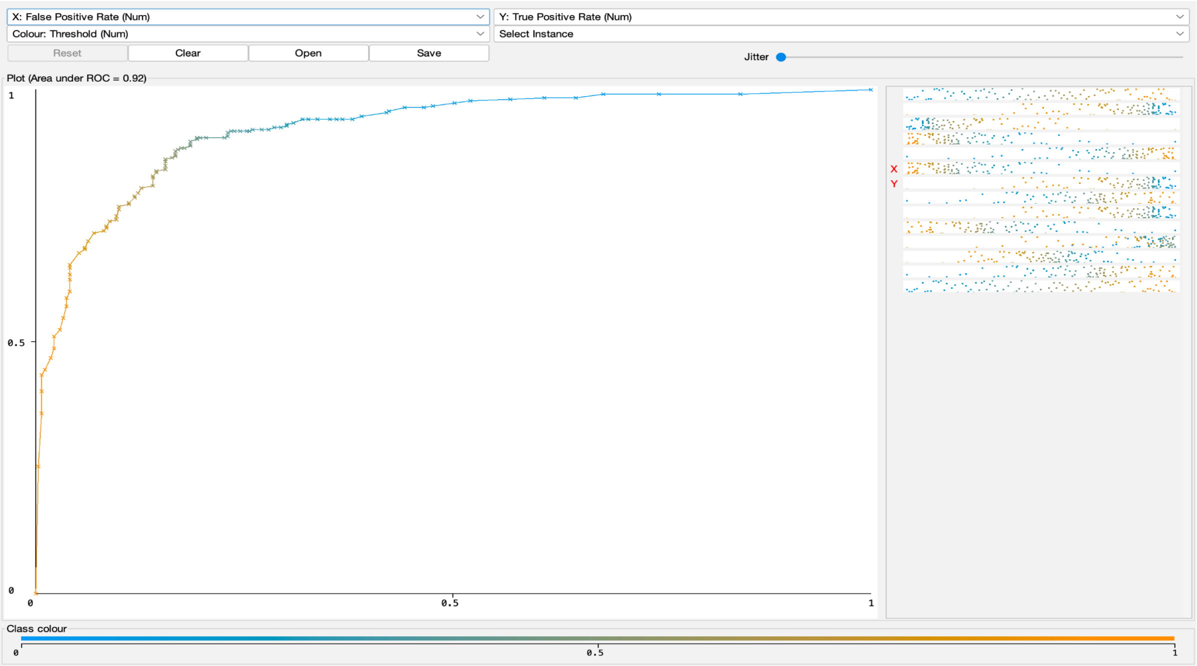
b) I think alcohol is more predective of the wine quality after comparing the plots of Sulphates and Alcohol because wine quality of alcohol plot is more consistent with its highs and lows when compared to Sulphates. Moreover, alcohol in general contributes alot to the wine's body, aroma and flavor. Higher alcohol content can give the wine a fuller body and a richer flavor, but it can also make the wine more alcoholic and overpowering.

c) After using the Logistic Regression Model, I have realised that it is not consistent with my speculation in (b) i.e., logistic regression model shows that sulphates can predict the wine quality better than the alcohol.

2. The performance metrics report of 10-fold cross validation using red-wine.csv as the training set for the following methods:

Model	ZeroR	OneR	LR	NB	DT	SVM	RF
AUC	N/A	N/A	0.873	0.89	0.877	0.781	0.92
Accuracy	52.8897%	78.4588%	79.3345%	82.4869%	84.2382%	78.2837%	85.289%

3.

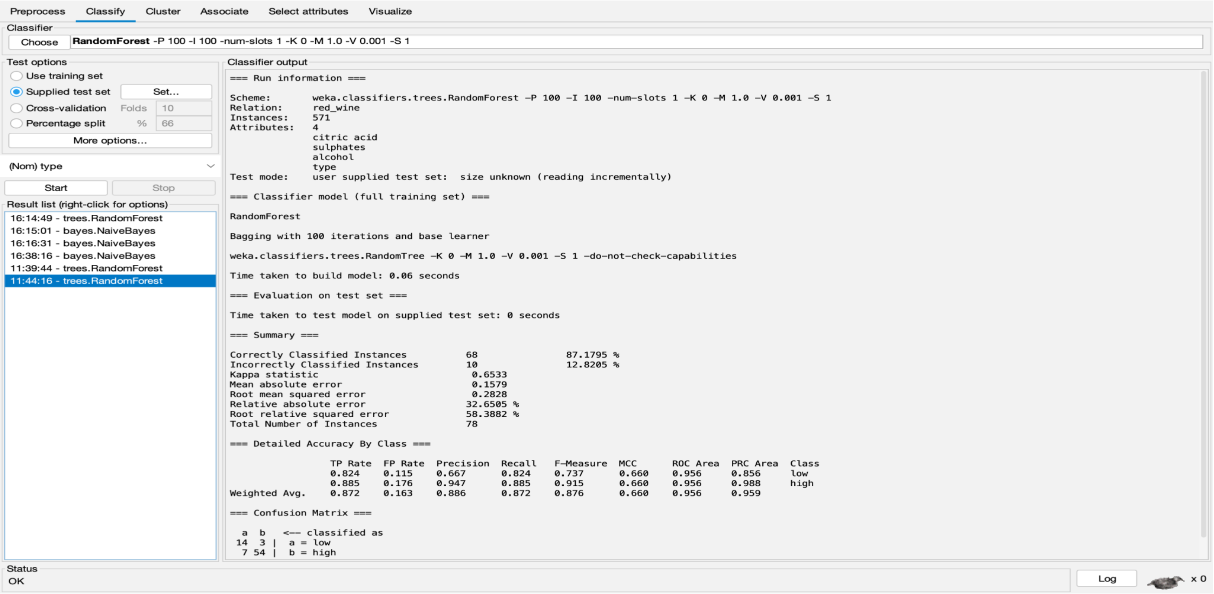

X: False Positive Rate (Num)
Y: True Positive Rate (Num)
Colour: Threshold (Num)
Select Instance

According to the performance metrics report, the best performing model in terms of AUC score would be Random Forest. The AUC for Random Forest is 0.920 which is closer to 1 i.e., it has the very good ability to distinguish the positive and negative instances in the dataset.

4. After running the model on white_wine.csv with the best performing model (Random Forest), the AUC score and accuracy values are as follows:

AUC score: 0.956
Accuracy: 87.1795%

Our model also performed really well on white-wine.csv dataset with AUC score of 0.956. The model’s high AUC value suggests that it is highly effective in distinguishing in between positive and negative instances in the dataset, indicating strong predictive power.



RandomForest -P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1

Test options
○ Use training set
● Supplied test set   Set...
○ Cross-validation   Folds 10
○ Percentage split   % 66
More options...

(Nom) type

Result list (right-click for options)

```
16:14:49 - trees.RandomForest
16:15:01 - bayes.NaiveBayes
16:16:31 - bayes.NaiveBayes
16:38:16 - bayes.NaiveBayes
11:39:44 - trees.RandomForest
11:44:16 - trees.RandomForest
```

Classifier output

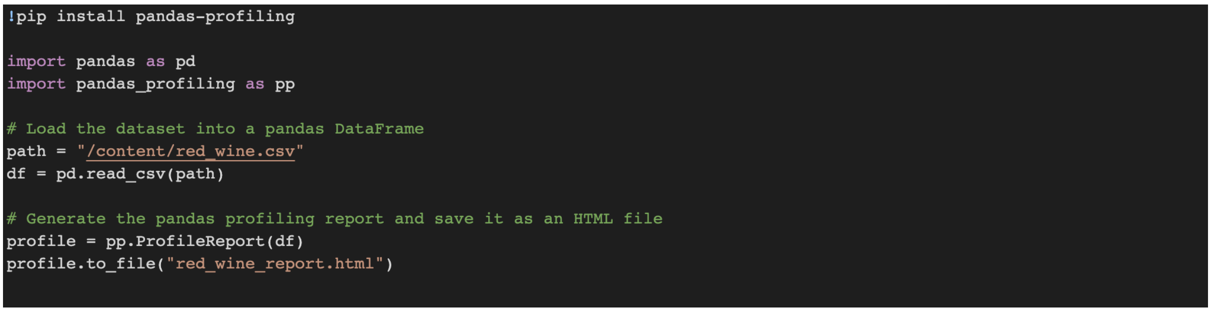
```
=== Run information ===

Scheme:       weka.classifiers.trees.RandomForest -P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1
Relation:     red_wine
Instances:    571
Attributes:   4
              citric acid
              sulphates
              alcohol
              type
Test mode:    user supplied test set:  size unknown (reading incrementally)

=== Classifier model (full training set) ===

RandomForest

Bagging with 100 iterations and base learner

weka.classifiers.trees.RandomTree -K 0 -M 1.0 -V 0.001 -S 1 -do-not-check-capabilities

Time taken to build model: 0.06 seconds

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0 seconds

=== Summary ===

Correctly Classified Instances          68               87.1795 %
Incorrectly Classified Instances        10               12.8205 %
Kappa statistic                          0.6533
Mean absolute error                      0.1579
Root mean squared error                  0.2828
Relative absolute error                 32.6505 %
Root relative squared error             58.3882 %
Total Number of Instances               78

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.824    0.115    0.667      0.824    0.737      0.660    0.956     0.856     low
                 0.885    0.176    0.947      0.885    0.915      0.660    0.956     0.988     high
Weighted Avg.    0.872    0.163    0.886      0.872    0.876      0.660    0.956     0.959

=== Confusion Matrix ===

  a  b   <-- classified as
 14  3 |  a = low
  7 54 |  b = high
```

Status
OK

Python Task

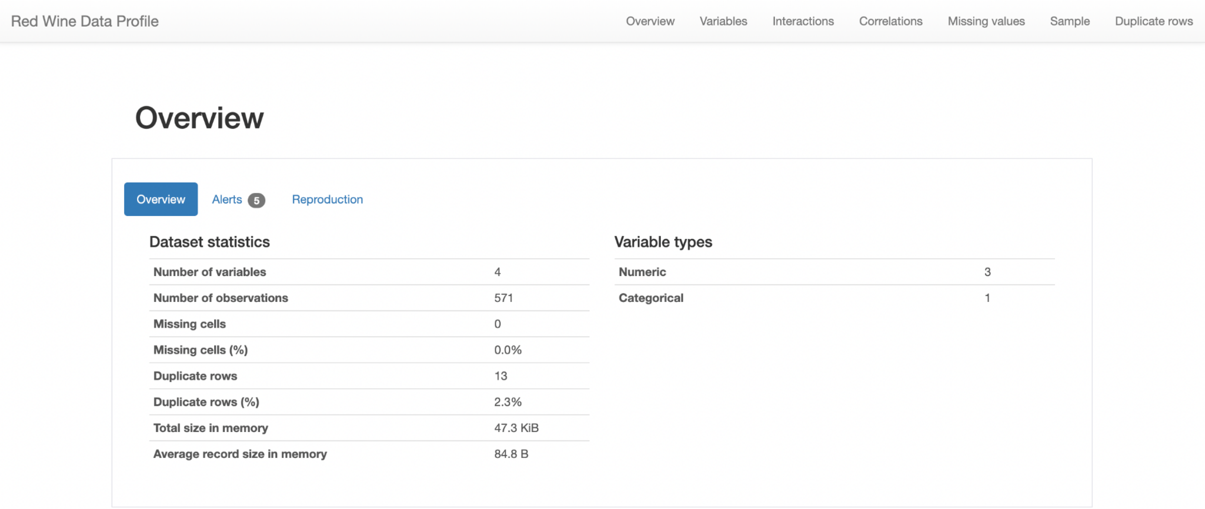
Here is the GitHub link for the Python Notebook:
https://github.com/sujit-3699/Sujit-Kandala_IS_733-Assignment-2.git

1. I used Google Colab to solve this question using Python. Below is the code for reading the red_wine.csv and creating a HTML report:



```
!pip install pandas-profiling

import pandas as pd
import pandas_profiling as pp

# Load the dataset into a pandas DataFrame
path = "/content/red_wine.csv"
df = pd.read_csv(path)

# Generate the pandas profiling report and save it as an HTML file
profile = pp.ProfileReport(df)
profile.to_file("red_wine_report.html")
```

Screenshots of HTML Report:



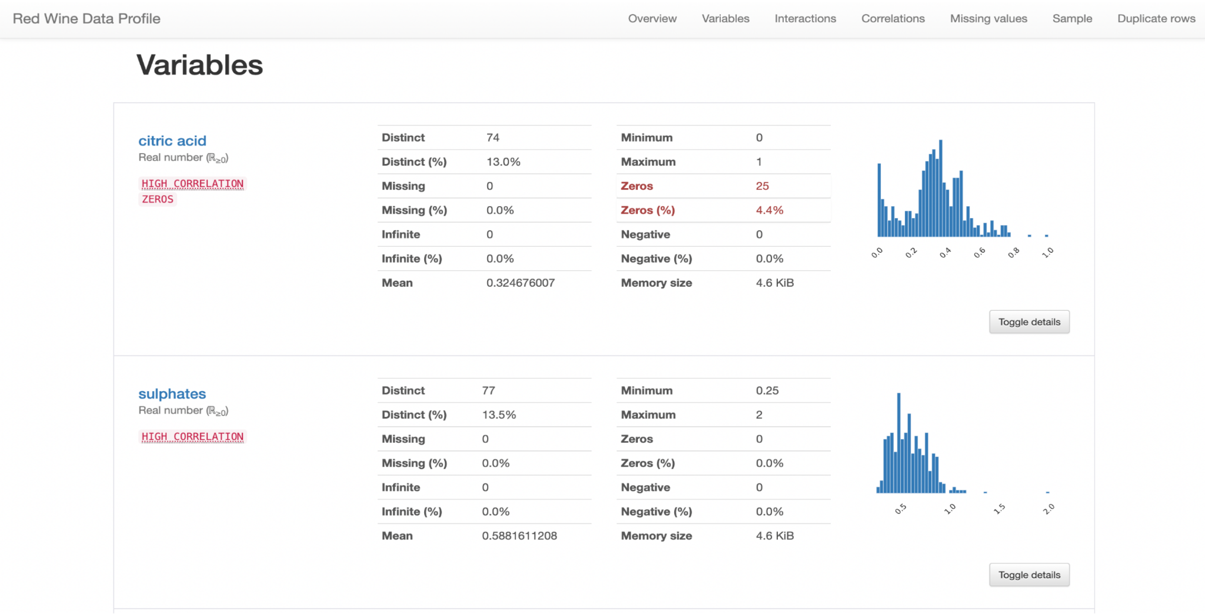
Overview

Dataset statistics

Number of variables	4
Number of observations	571
Missing cells	0
Missing cells (%)	0.0%
Duplicate rows	13
Duplicate rows (%)	2.3%
Total size in memory	47.3 KiB
Average record size in memory	84.8 B

Variable types

Numeric	3
Categorical	1



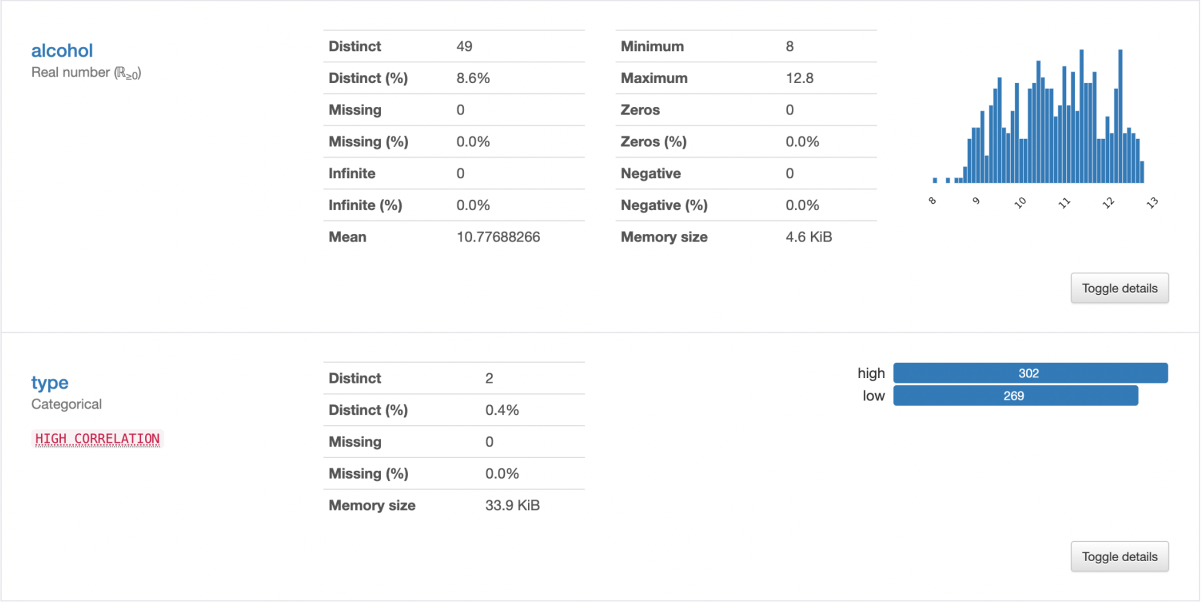
Variables

citric acid
Real number ($\mathbb{R}_{\geq 0}$)
HIGH CORRELATION
ZEROS

Distinct	74	Minimum	0
Distinct (%)	13.0%	Maximum	1
Missing	0	Zeros	25
Missing (%)	0.0%	Zeros (%)	4.4%
Infinite	0	Negative	0
Infinite (%)	0.0%	Negative (%)	0.0%
Mean	0.324676007	Memory size	4.6 KiB

sulphates
Real number ($\mathbb{R}_{\geq 0}$)
HIGH CORRELATION

Distinct	77	Minimum	0.25
Distinct (%)	13.5%	Maximum	2
Missing	0	Zeros	0
Missing (%)	0.0%	Zeros (%)	0.0%
Infinite	0	Negative	0
Infinite (%)	0.0%	Negative (%)	0.0%
Mean	0.5881611208	Memory size	4.6 KiB



alcohol
Real number ($\mathbb{R}_{\geq 0}$)

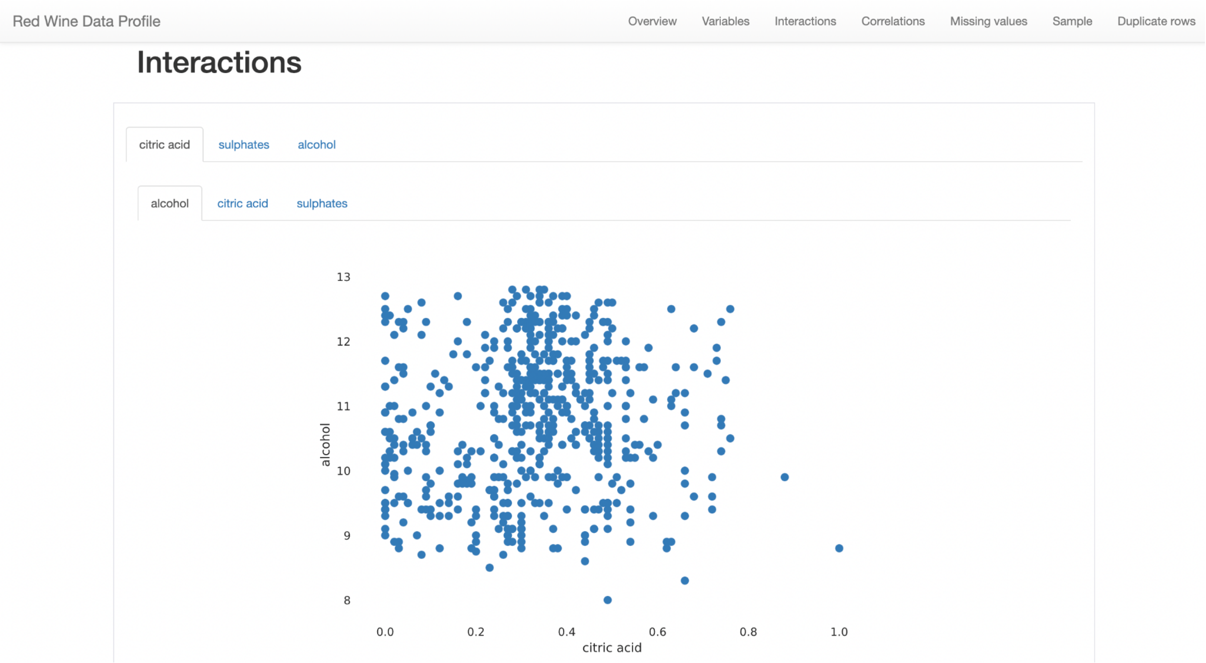
Distinct	49
Distinct (%)	8.6%
Missing	0
Missing (%)	0.0%
Infinite	0
Infinite (%)	0.0%
Mean	10.77688266

Minimum	8
Maximum	12.8
Zeros	0
Zeros (%)	0.0%
Negative	0
Negative (%)	0.0%
Memory size	4.6 KiB

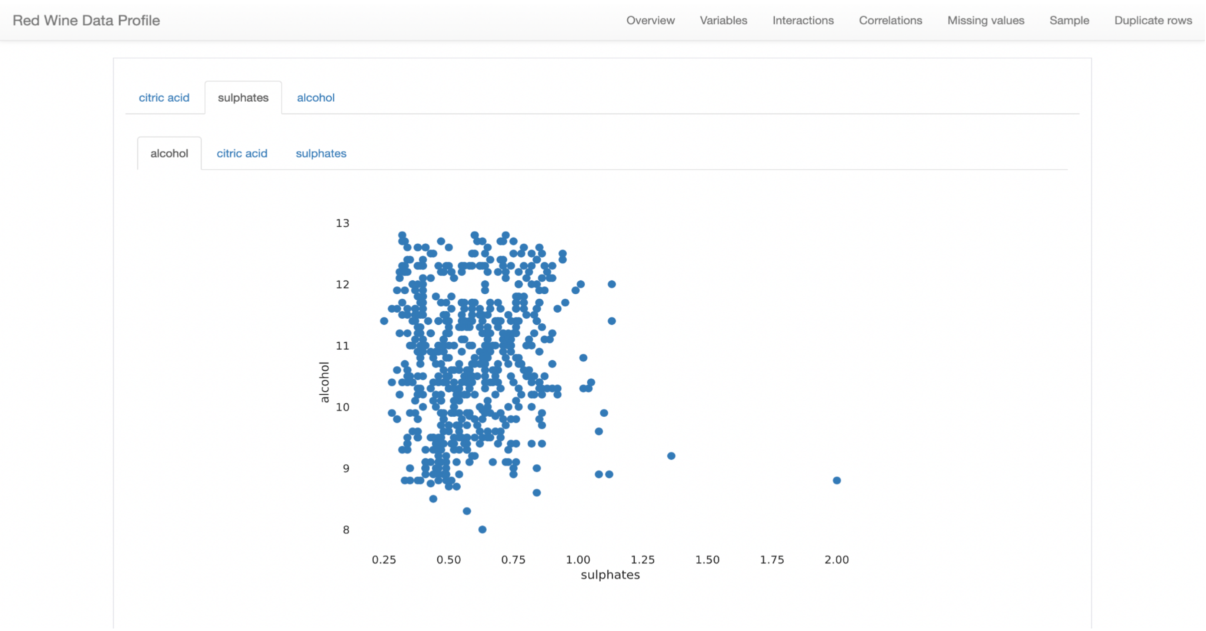
type
Categorical

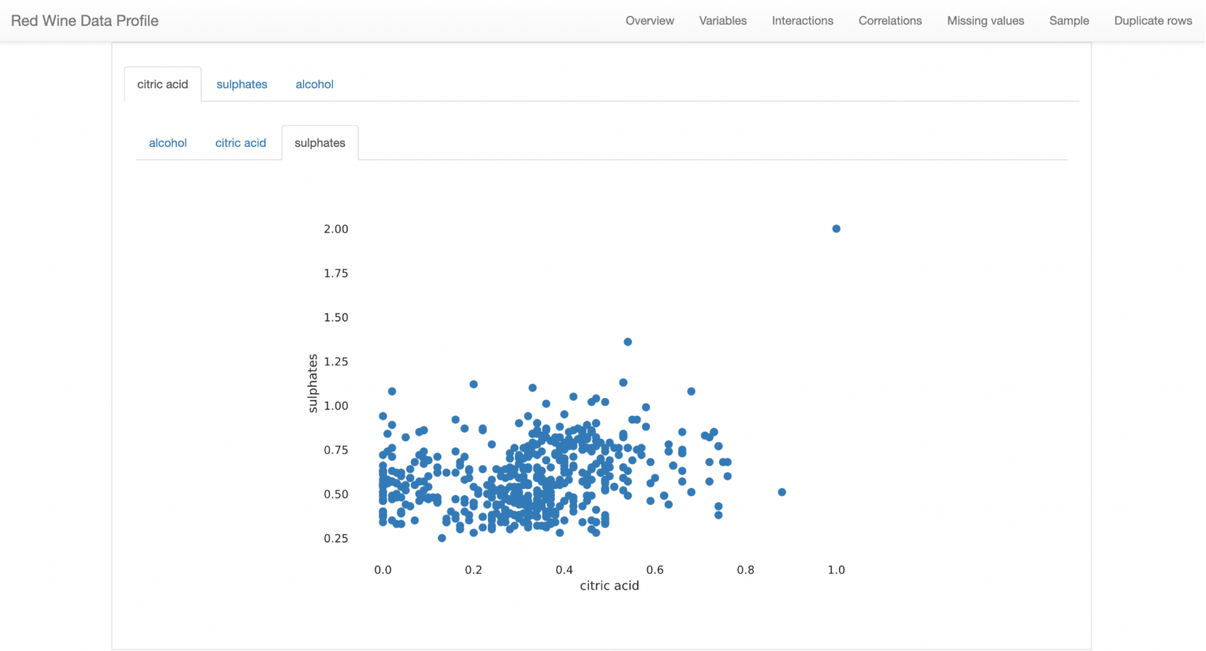
HIGH CORRELATION

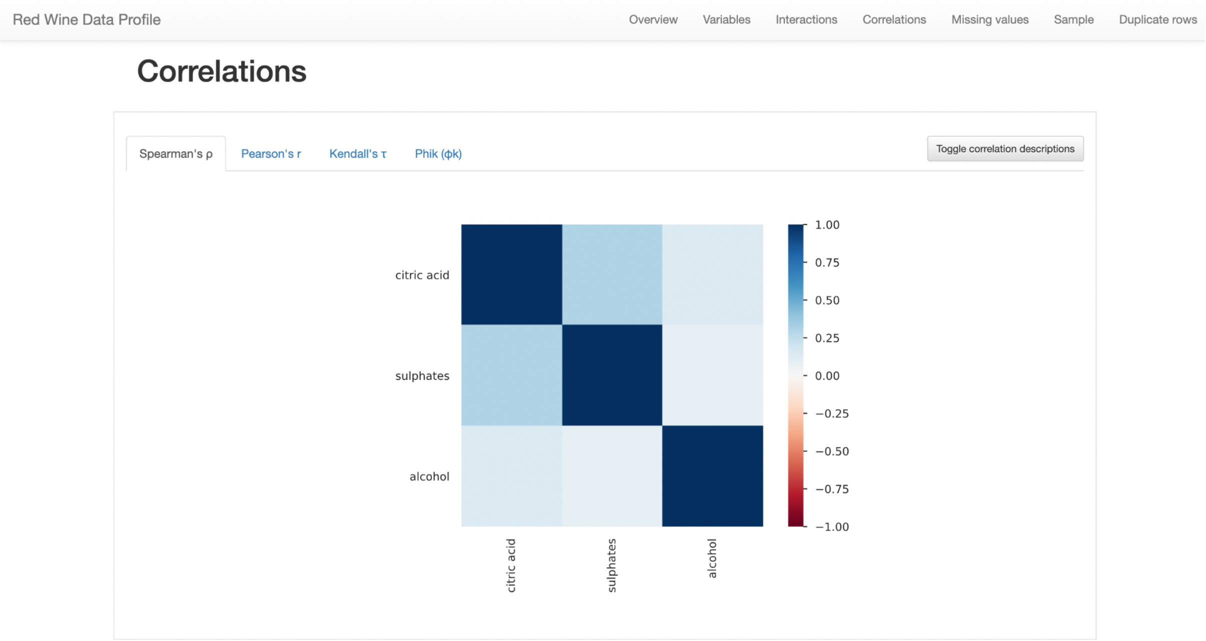
Distinct	2
Distinct (%)	0.4%
Missing	0
Missing (%)	0.0%
Memory size	33.9 KiB



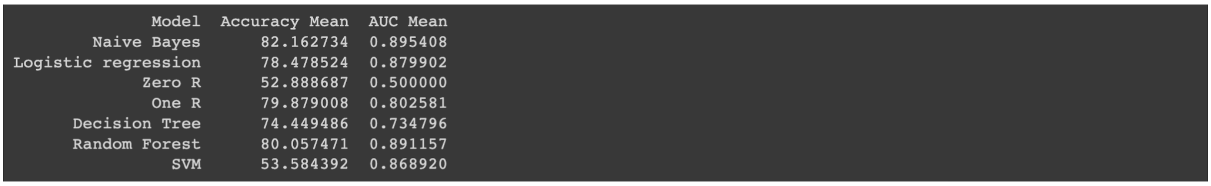
Interactions





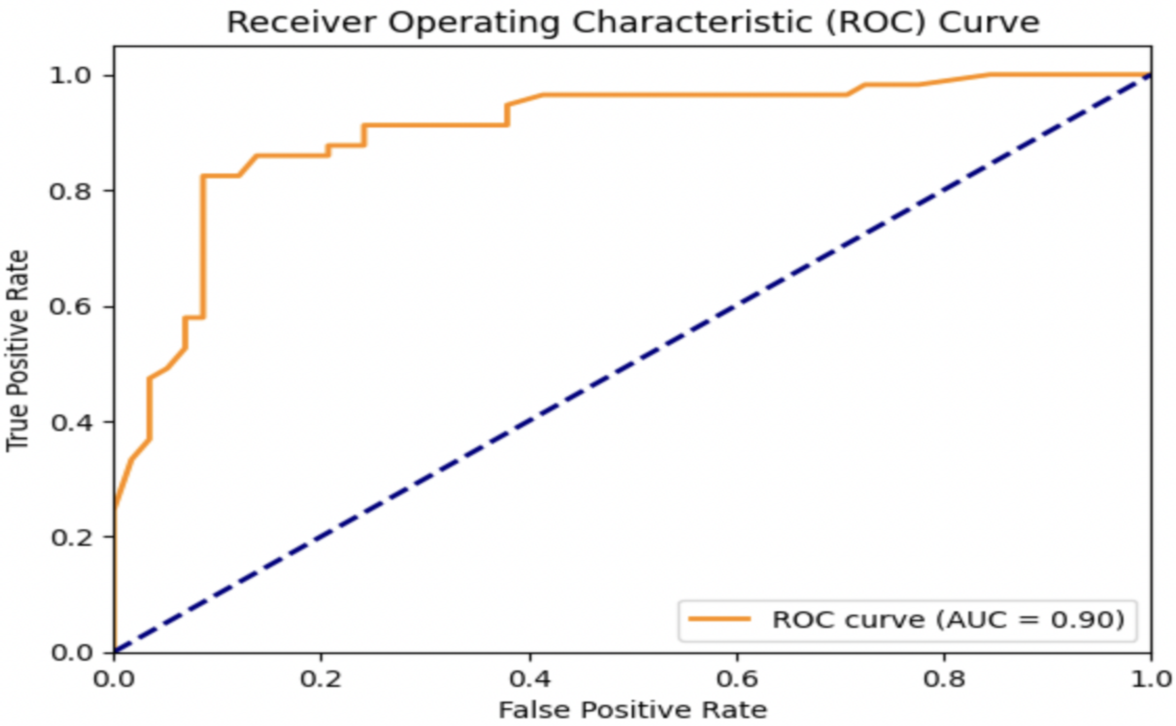


2. Results of all the models from Python Notebook:

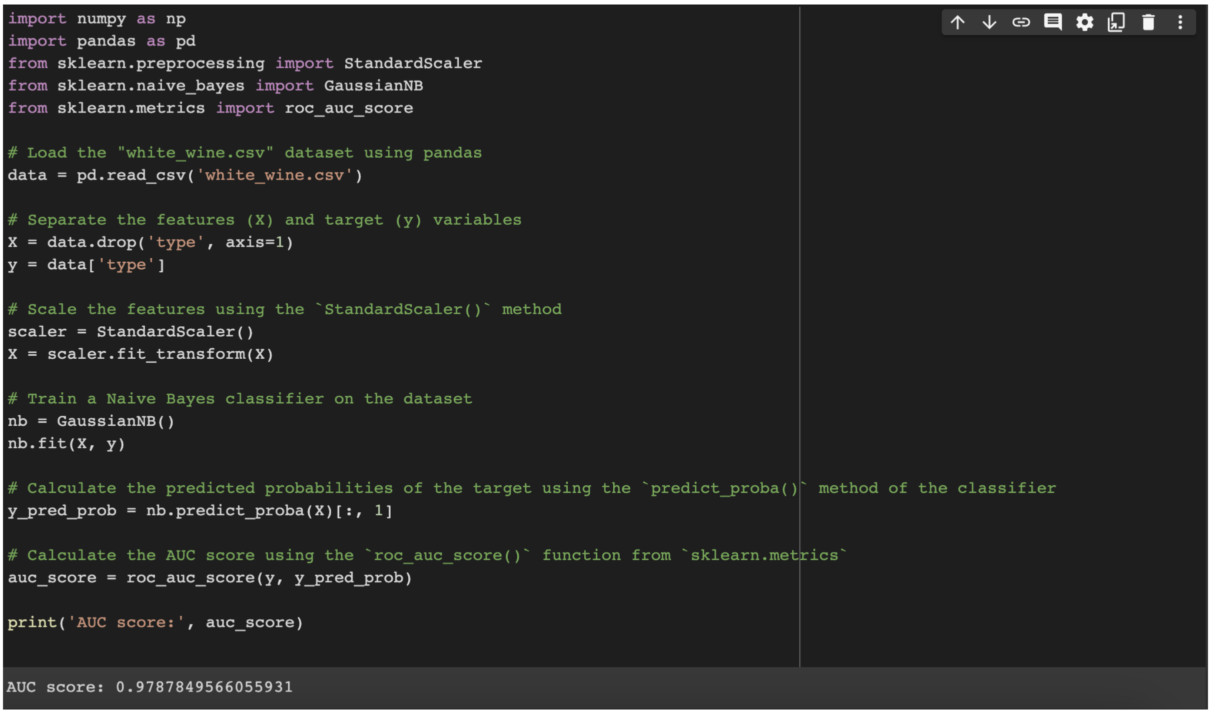


```
              Model  Accuracy Mean  AUC Mean
        Naive Bayes      82.162734  0.895408
Logistic regression      78.478524  0.879902
             Zero R      52.888687  0.500000
              One R      79.879008  0.802581
      Decision Tree      74.449486  0.734796
      Random Forest      80.057471  0.891157
                SVM      53.584392  0.868920
```

3. Screenshot of ROC curve plot for the Random Forest Classifier:



4.



```
import numpy as np
import pandas as pd
from sklearn.preprocessing import StandardScaler
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import roc_auc_score

# Load the "white_wine.csv" dataset using pandas
data = pd.read_csv('white_wine.csv')

# Separate the features (X) and target (y) variables
X = data.drop('type', axis=1)
y = data['type']

# Scale the features using the `StandardScaler()` method
scaler = StandardScaler()
X = scaler.fit_transform(X)

# Train a Naive Bayes classifier on the dataset
nb = GaussianNB()
nb.fit(X, y)

# Calculate the predicted probabilities of the target using the `predict_proba()` method of the classifier
y_pred_prob = nb.predict_proba(X)[:, 1]

# Calculate the AUC score using the `roc_auc_score()` function from `sklearn.metrics`
auc_score = roc_auc_score(y, y_pred_prob)

print('AUC score:', auc_score)
```

```
AUC score: 0.9787849566055931
```

AUC score of the Naive Bayes Classifier on the white_wine.csv is 0.9787 which is closer to 1 (perfect model) which indicates that the classifier can effectively distinguish positive and negative samples and can make highly accurate predictions.

5. If interpretability and gaining insights into the model is a priority for the wine tasting experts, then I would choose a decision-tree based model like Random Forest provided all the models have comparable performances. Because I believe it can provide importance of the features and decision rules that can be used to gain insights into the model's predictions. Moreover, it is easier to understand and can visualize the trees without any serious efforts.