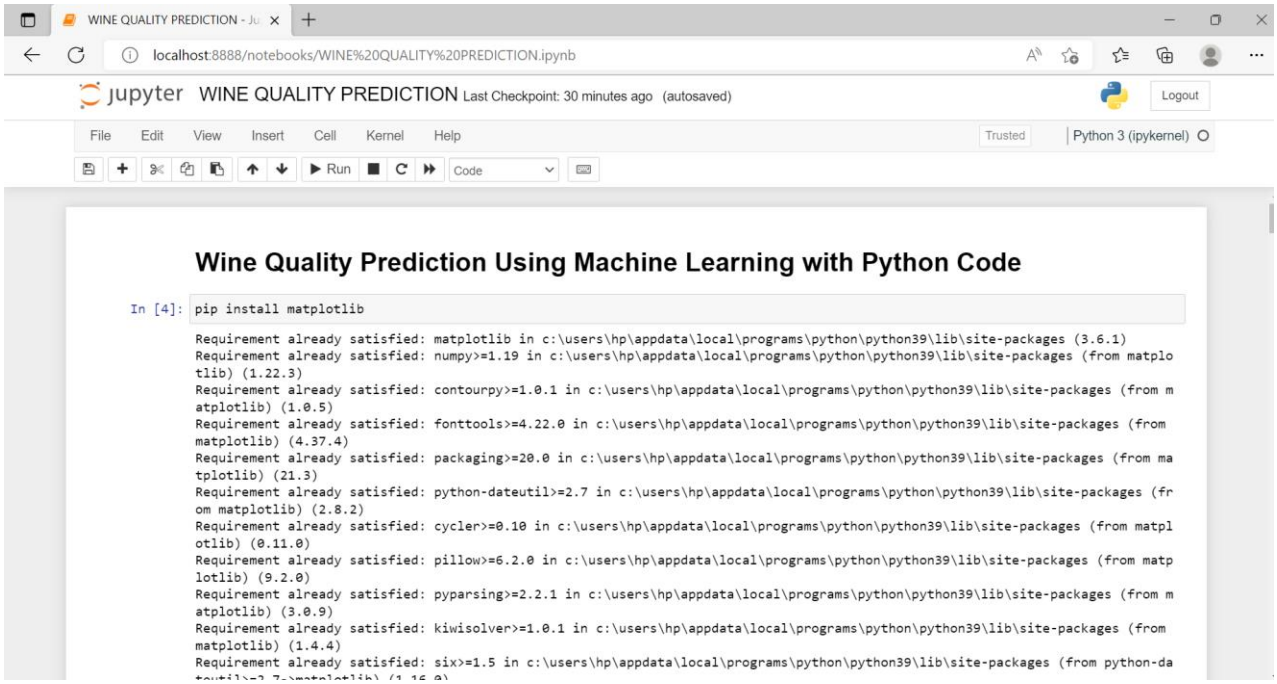
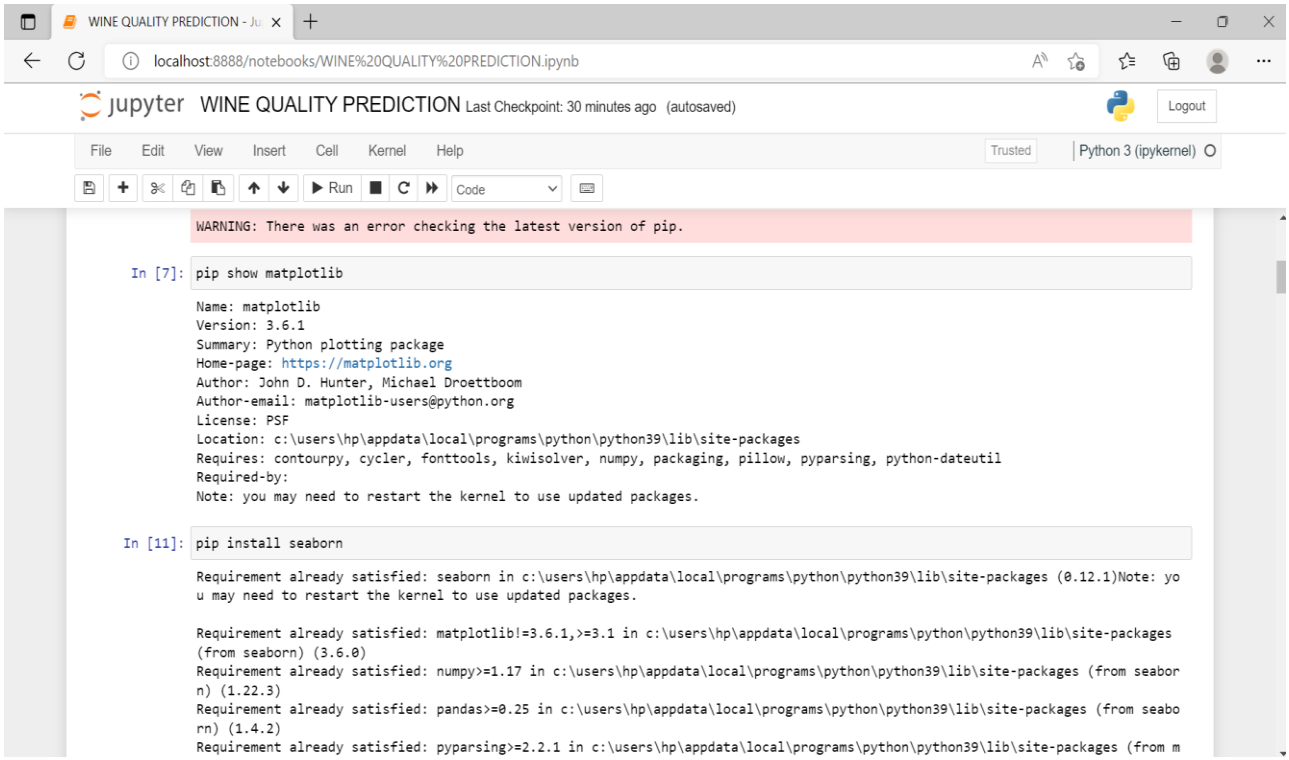
WINE QUALITY PREDICTION USING MACHINE LEARNING WITH PYTHON CODE

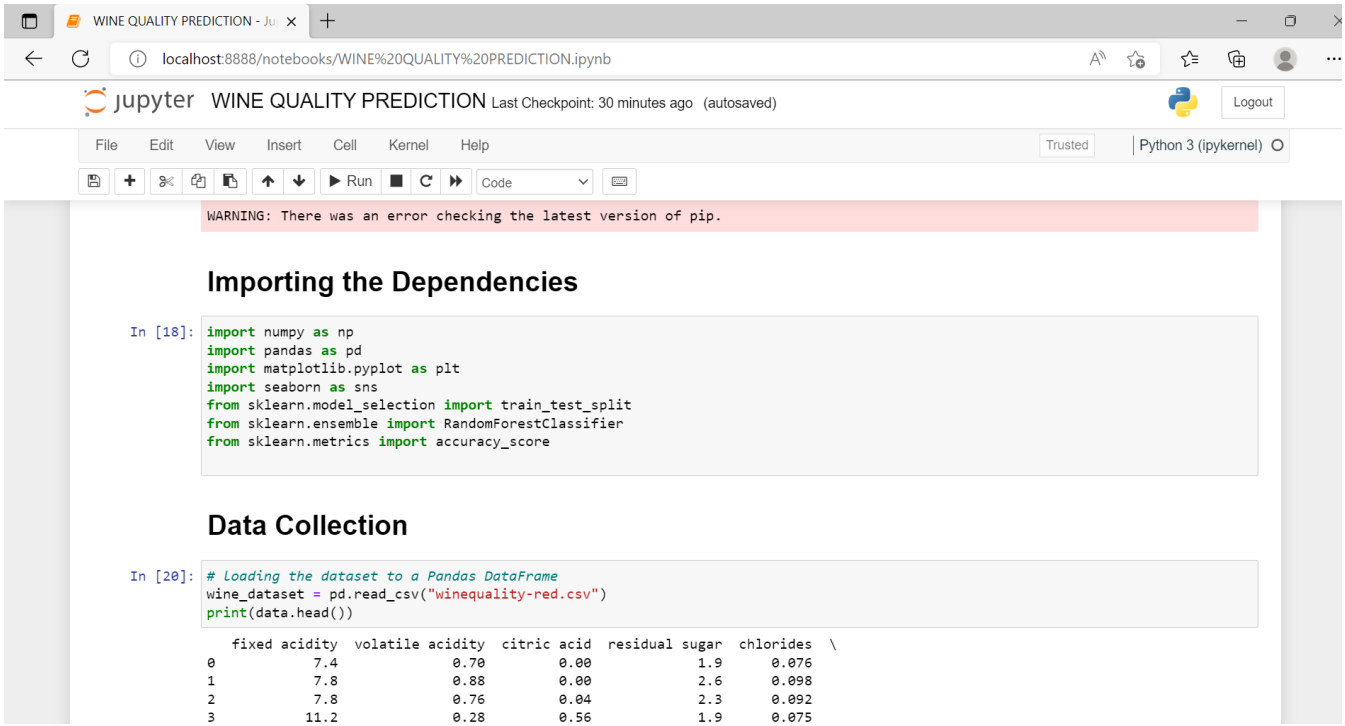
TEAM MEMBERS:

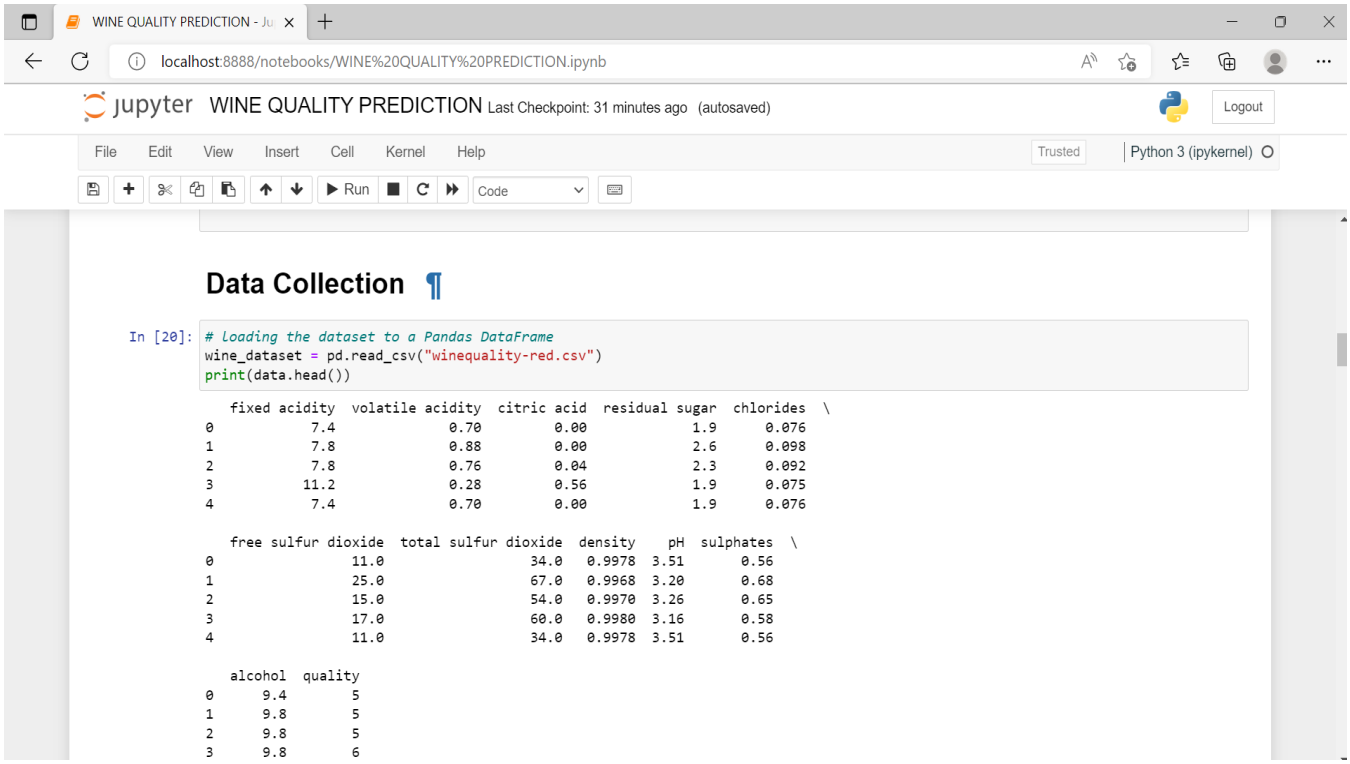
1)PRASAD NAYAK, nayakprasad57@gmail.com, +971557716724

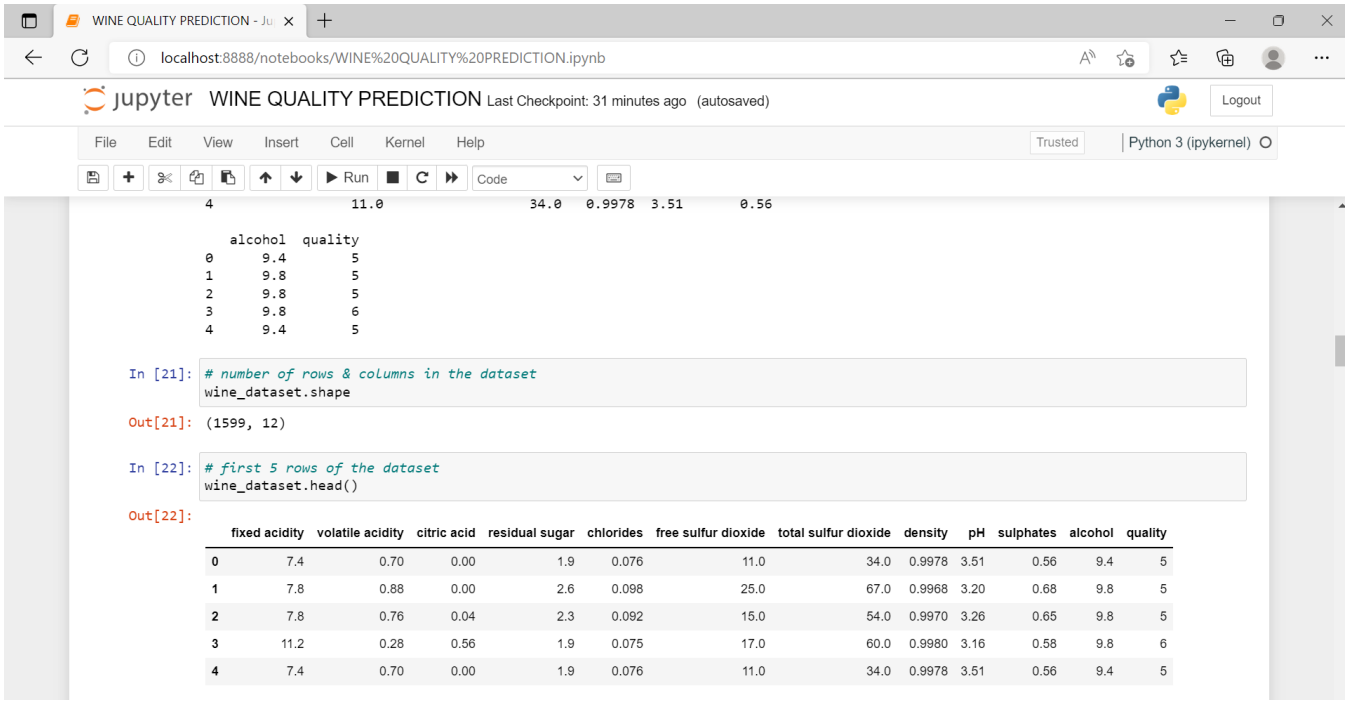
2)GAGAN SB, gagansb37@gmail.com, 8296301709



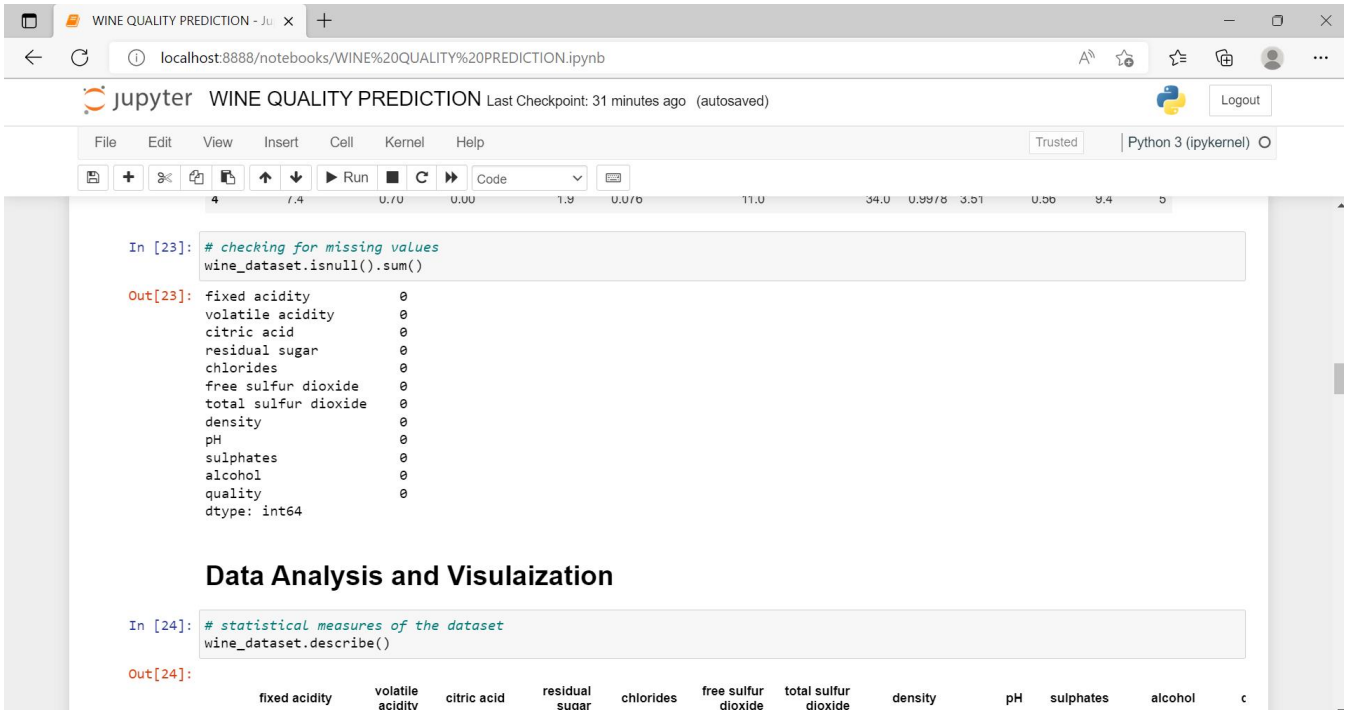



WINE QUALITY PREDICTION - Ju
localhost:8888/notebooks/WINE%20QUALITY%20PREDICTION.ipynb
jupyter WINE QUALITY PREDICTION Last Checkpoint: 30 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Help
Trusted
Python 3 (ipykernel)
Run
Code
WARNING: There was an error checking the latest version of pip.
Importing the Dependencies
In [18]:
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score
Data Collection
In [20]:
# loading the dataset to a Pandas DataFrame
wine_dataset = pd.read_csv("winequality-red.csv")
print(data.head())
fixed acidity volatile acidity citric acid residual sugar chlorides \
0 7.4 0.70 0.00 1.9 0.076
1 7.8 0.88 0.00 2.6 0.098
2 7.8 0.76 0.04 2.3 0.092
3 11.2 0.28 0.56 1.9 0.075


WINE QUALITY PREDICTION - Ju
localhost:8888/notebooks/WINE%20QUALITY%20PREDICTION.ipynb
jupyter WINE QUALITY PREDICTION Last Checkpoint: 31 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Help
Trusted
Python 3 (ipykernel)
Run
Code
Data Collection ¶
In [20]:
# loading the dataset to a Pandas DataFrame
wine_dataset = pd.read_csv("winequality-red.csv")
print(data.head())
fixed acidity volatile acidity citric acid residual sugar chlorides \
0 7.4 0.70 0.00 1.9 0.076
1 7.8 0.88 0.00 2.6 0.098
2 7.8 0.76 0.04 2.3 0.092
3 11.2 0.28 0.56 1.9 0.075
4 7.4 0.70 0.00 1.9 0.076
free sulfur dioxide total sulfur dioxide density pH sulphates \
0 11.0 34.0 0.9978 3.51 0.56
1 25.0 67.0 0.9968 3.20 0.68
2 15.0 54.0 0.9970 3.26 0.65
3 17.0 60.0 0.9980 3.16 0.58
4 11.0 34.0 0.9978 3.51 0.56
alcohol quality
0 9.4 5
1 9.8 5
2 9.8 5
3 9.8 6



	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
0	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5
1	7.8	0.88	0.00	2.6	0.098	25.0	67.0	0.9968	3.20	0.68	9.8	5
2	7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.9970	3.26	0.65	9.8	5
3	11.2	0.28	0.56	1.9	0.075	17.0	60.0	0.9980	3.16	0.58	9.8	6
4	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5



Data Analysis and Visulaization

In [24]:

```
# statistical measures of the dataset
wine_dataset.describe()
```

Out[24]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	c

WINE QUALITY PREDICTION - Ju

localhost:8888/notebooks/WINE%20QUALITY%20PREDICTION.ipynb

jupyter WINE QUALITY PREDICTION Last Checkpoint: 31 minutes ago (autosaved)

Logout

File Edit View Insert Cell Kernel Help Trusted Python 3 (ipykernel)

Run Code

dtype: int64

Data Analysis and Visulaization

In [24]:
```
# statistical measures of the dataset
wine_dataset.describe()
```

Out[24]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	c
count	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.0
mean	8.319637	0.527821	0.270976	2.538806	0.087467	15.874922	46.467792	0.996747	3.311113	0.658149	10.422983	5.6
std	1.741096	0.179060	0.194801	1.409928	0.047065	10.460157	32.895324	0.001887	0.154386	0.169507	1.065668	0.8
min	4.600000	0.120000	0.000000	0.900000	0.012000	1.000000	6.000000	0.990070	2.740000	0.330000	8.400000	3.0
25%	7.100000	0.390000	0.090000	1.900000	0.070000	7.000000	22.000000	0.995600	3.210000	0.550000	9.500000	5.0
50%	7.900000	0.520000	0.260000	2.200000	0.079000	14.000000	38.000000	0.996750	3.310000	0.620000	10.200000	6.0
75%	9.200000	0.640000	0.420000	2.600000	0.090000	21.000000	62.000000	0.997835	3.400000	0.730000	11.100000	6.0
max	15.900000	1.580000	1.000000	15.500000	0.611000	72.000000	289.000000	1.003690	4.010000	2.000000	14.900000	8.0

In [25]:
```
# number of values for each quality
sns.catplot(x='quality', data = wine_dataset, kind = 'count')
```

WINE QUALITY PREDICTION - Ju

localhost:8888/notebooks/WINE%20QUALITY%20PREDICTION.ipynb

jupyter WINE QUALITY PREDICTION Last Checkpoint: 31 minutes ago (autosaved)

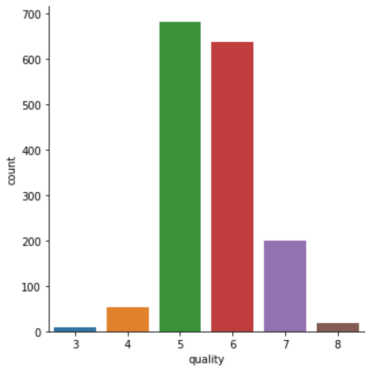
Logout

File Edit View Insert Cell Kernel Help Trusted Python 3 (ipykernel)

Run Code

In [25]:
```
# number of values for each quality
sns.catplot(x='quality', data = wine_dataset, kind = 'count')
```

Out[25]: <seaborn.axisgrid.FacetGrid at 0x290edc20df0>



In [26]:
```
# volatile acidity vs Quality
plot = plt.figure(figsize=(5,5))
```

WINE QUALITY PREDICTION - Ju

localhost:8888/notebooks/WINE%20QUALITY%20PREDICTION.ipynb

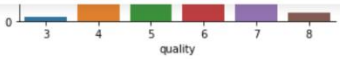
jupyter **WINE QUALITY PREDICTION** Last Checkpoint: 31 minutes ago (autosaved)

Logout

File Edit View Insert Cell Kernel Help

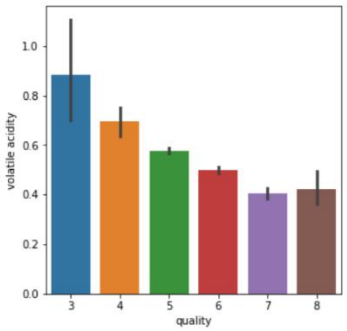
Trusted | Python 3 (ipykernel)

Run Code



In [26]:
```
# volatile acidity vs Quality
plot = plt.figure(figsize=(5,5))
sns.barplot(x='quality', y = 'volatile acidity', data = wine_dataset)
```

Out[26]: <AxesSubplot: xlabel='quality', ylabel='volatile acidity'>



WINE QUALITY PREDICTION - Ju

localhost:8888/notebooks/WINE%20QUALITY%20PREDICTION.ipynb

jupyter **WINE QUALITY PREDICTION** Last Checkpoint: 31 minutes ago (autosaved)

Logout

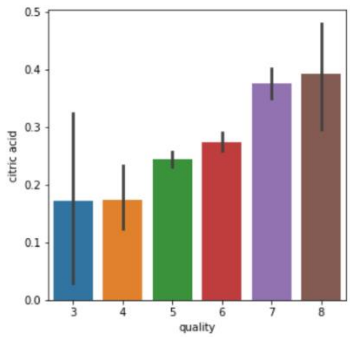
File Edit View Insert Cell Kernel Help

Trusted | Python 3 (ipykernel)

Run Code

In [27]:
```
# citric acid vs Quality
plot = plt.figure(figsize=(5,5))
sns.barplot(x='quality', y = 'citric acid', data = wine_dataset)
```

Out[27]: <AxesSubplot: xlabel='quality', ylabel='citric acid'>



WINE QUALITY PREDICTION - Ju

localhost:8888/notebooks/WINE%20QUALITY%20PREDICTION.ipynb

jupyter WINE QUALITY PREDICTION Last Checkpoint: 31 minutes ago (autosaved)

Logout

File Edit View Insert Cell Kernel Help

Trusted Python 3 (ipykernel)

Run Code

quality

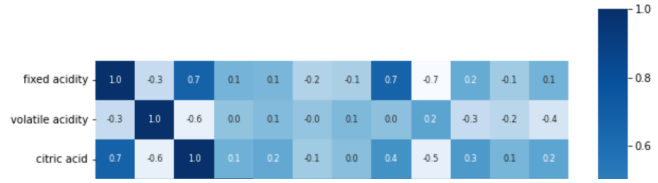
Correlation

- 1)Positive Correlation
- 2)Negative Correlation

In [28]:
```
correlation = wine_dataset.corr()
```

In [29]:
```
# constructing a heatmap to understand the correlation between the columns
plt.figure(figsize=(10,10))
sns.heatmap(correlation, cbar=True, square=True, fmt = '.1f', annot = True, annot_kws={'size':8}, cmap = 'Blues')
```

Out[29]: <AxesSubplot: >



WINE QUALITY PREDICTION - Ju

localhost:8888/notebooks/WINE%20QUALITY%20PREDICTION.ipynb

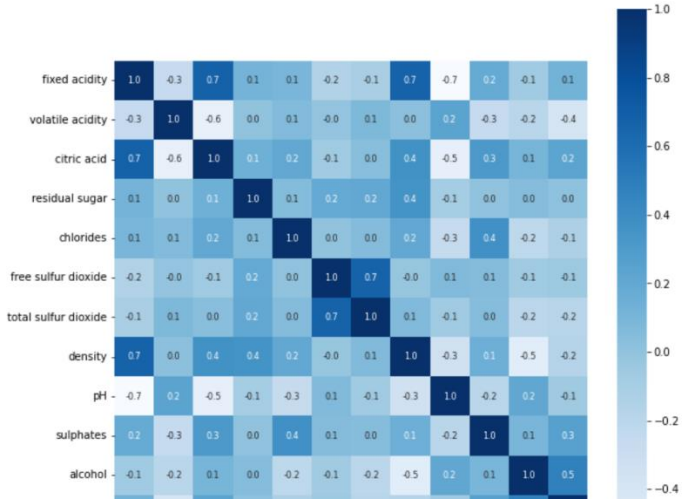
jupyter WINE QUALITY PREDICTION Last Checkpoint: 32 minutes ago (autosaved)

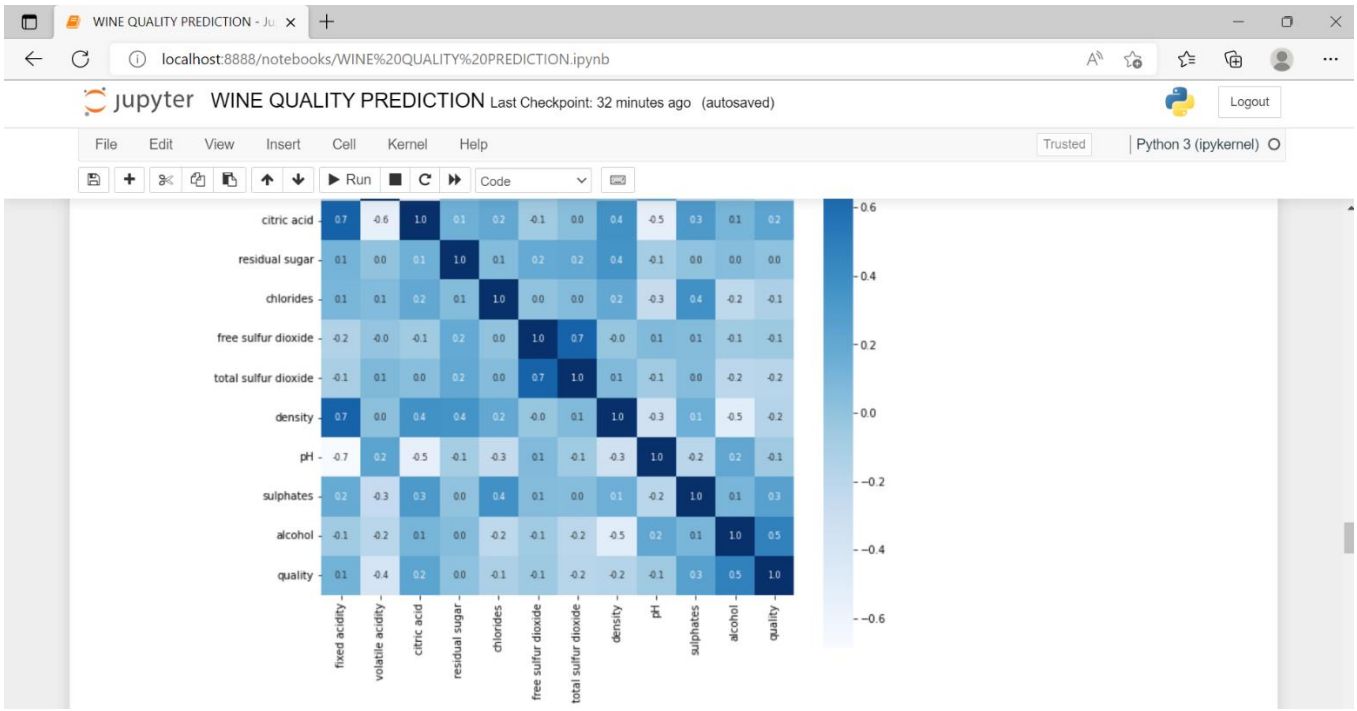
Logout

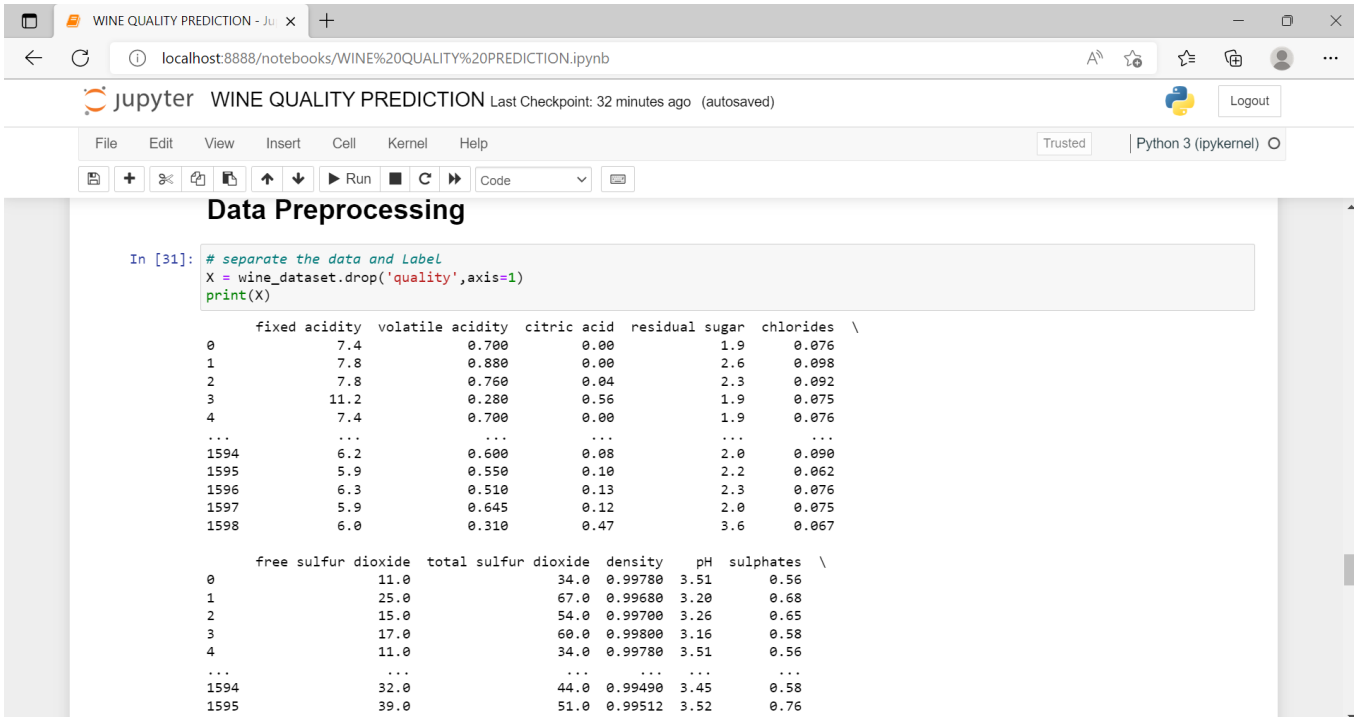
File Edit View Insert Cell Kernel Help

Trusted Python 3 (ipykernel)

Run Code





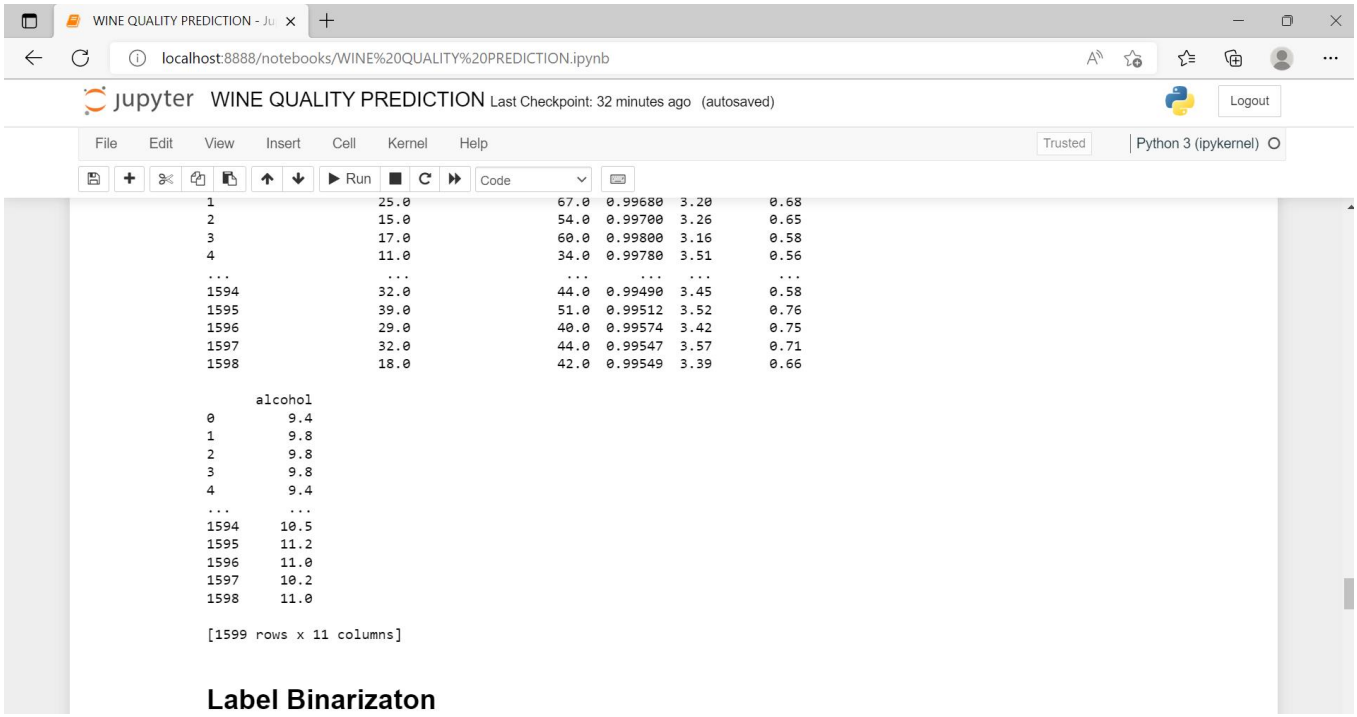


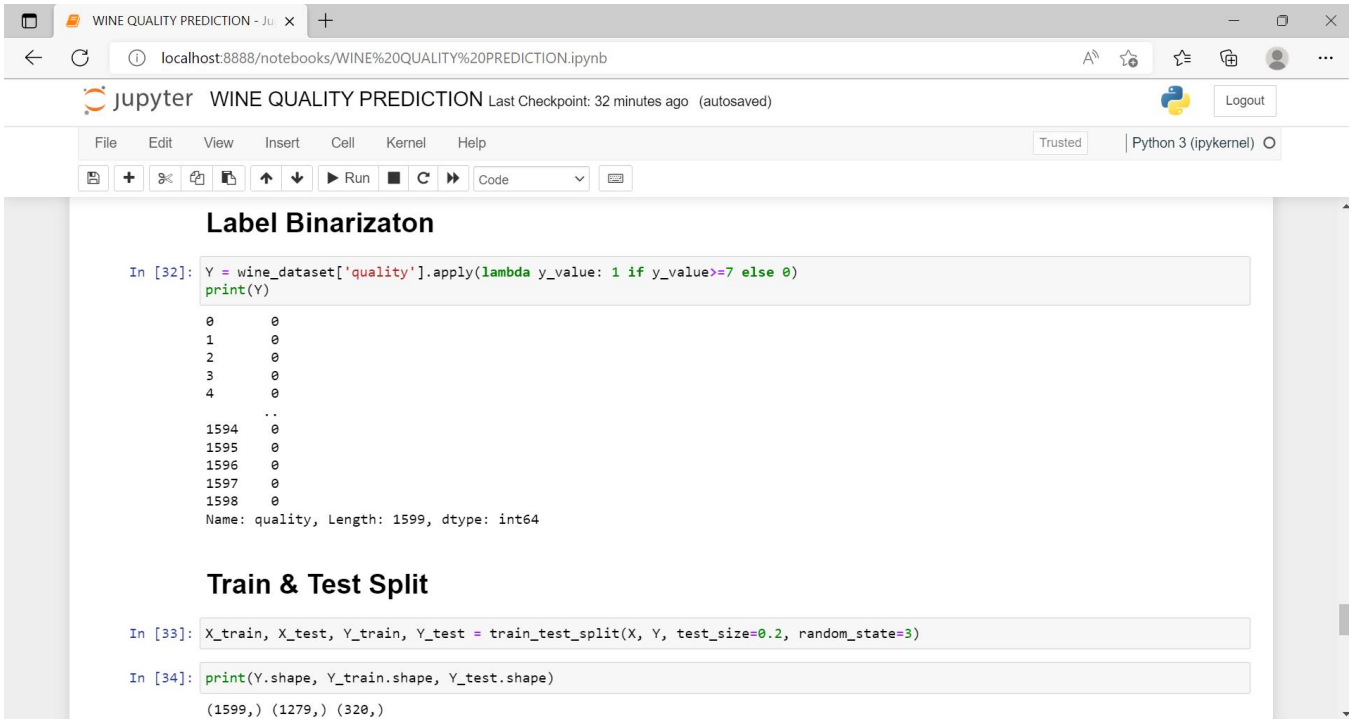
Data Preprocessing

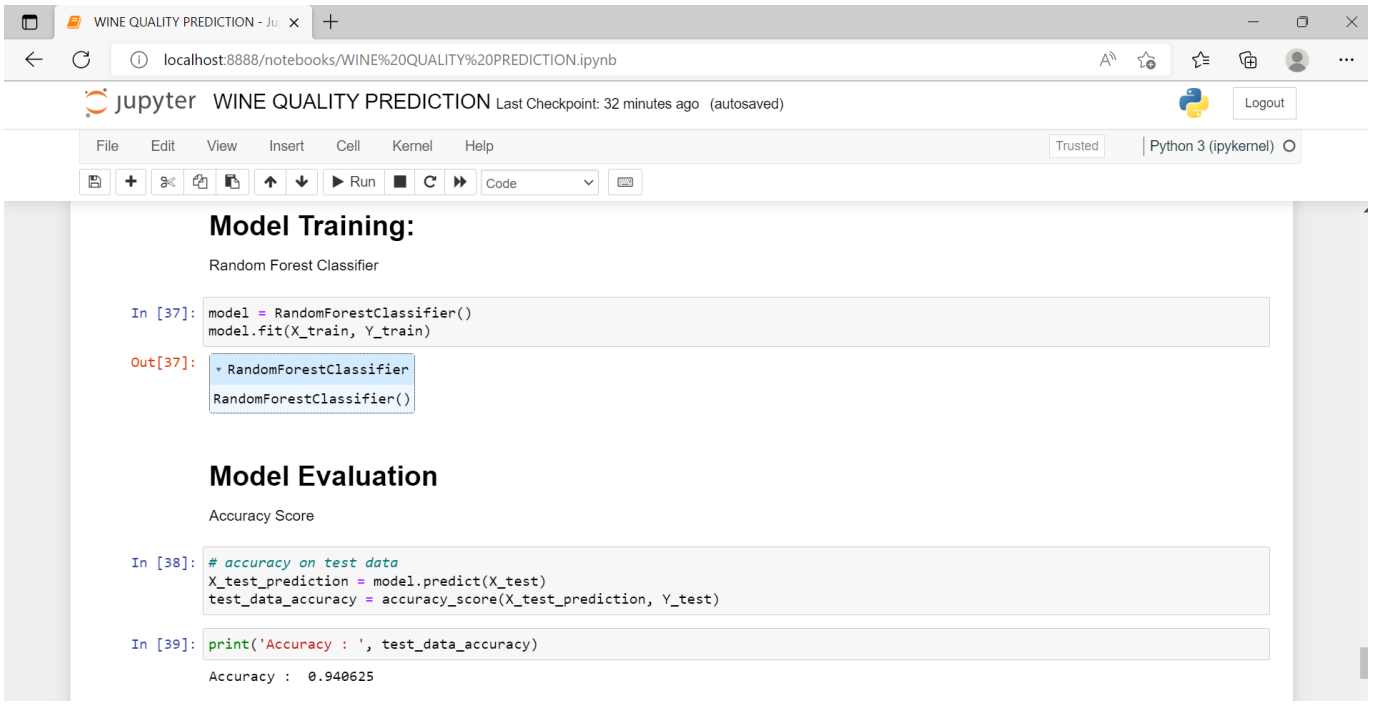
In [31]:
```
# separate the data and Label
X = wine_dataset.drop('quality',axis=1)
print(X)
```

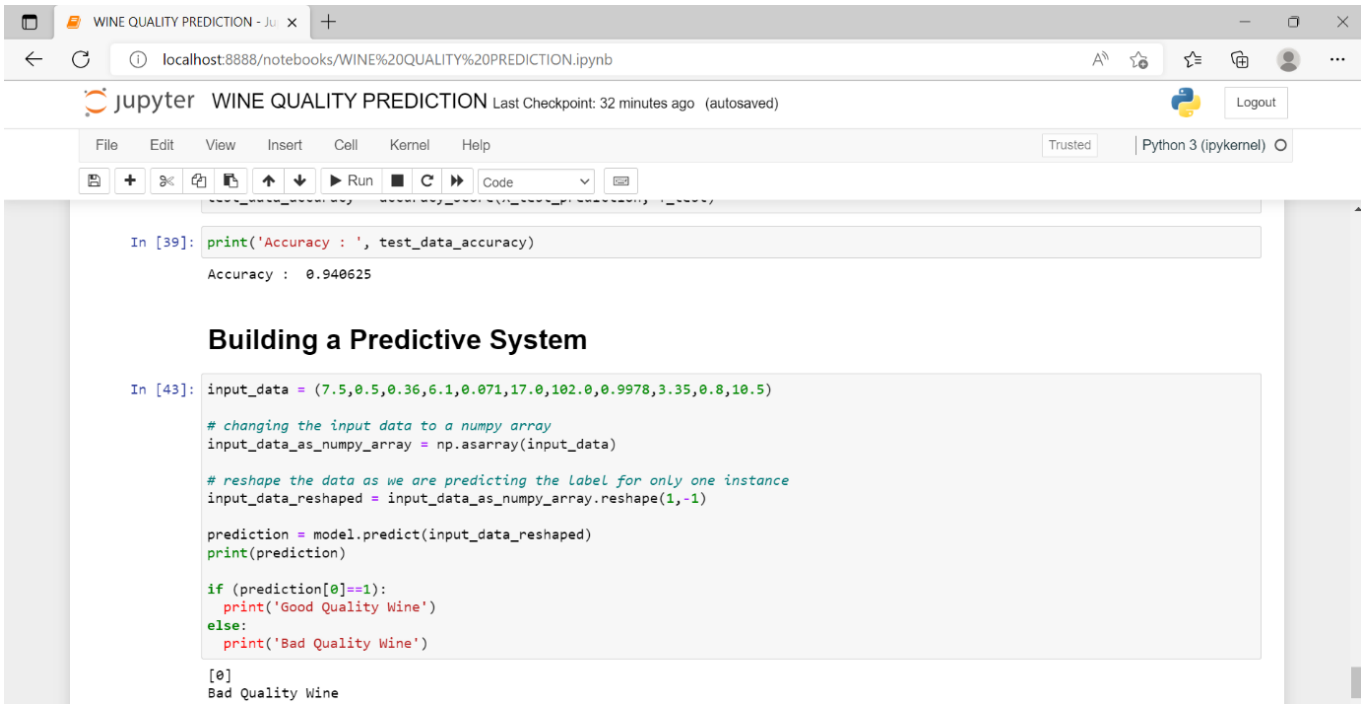
```
      fixed acidity  volatile acidity  citric acid  residual sugar  chlorides  \
0               7.4             0.700         0.00             1.9      0.076
1               7.8             0.880         0.00             2.6      0.098
2               7.8             0.760         0.04             2.3      0.092
3              11.2             0.280         0.56             1.9      0.075
4               7.4             0.700         0.00             1.9      0.076
...             ...               ...          ...             ...        ...
1594            6.2             0.600         0.08             2.0      0.090
1595            5.9             0.550         0.10             2.2      0.062
1596            6.3             0.510         0.13             2.3      0.076
1597            5.9             0.645         0.12             2.0      0.075
1598            6.0             0.310         0.47             3.6      0.067

      free sulfur dioxide  total sulfur dioxide  density    pH  sulphates  \
0                    11.0                  34.0  0.99780  3.51       0.56
1                    25.0                  67.0  0.99680  3.20       0.68
2                    15.0                  54.0  0.99700  3.26       0.65
3                    17.0                  60.0  0.99800  3.16       0.58
4                    11.0                  34.0  0.99780  3.51       0.56
...                   ...                   ...      ...   ...        ...
1594                 32.0                  44.0  0.99490  3.45       0.58
1595                 39.0                  51.0  0.99512  3.52       0.76
```


WINE QUALITY PREDICTION - Ju
localhost:8888/notebooks/WINE%20QUALITY%20PREDICTION.ipynb
jupyter WINE QUALITY PREDICTION Last Checkpoint: 32 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Help
Trusted
Python 3 (ipykernel)
Run
Code
1 25.0 67.0 0.99680 3.20 0.68
2 15.0 54.0 0.99700 3.26 0.65
3 17.0 60.0 0.99800 3.16 0.58
4 11.0 34.0 0.99780 3.51 0.56
... ...
1594 32.0 44.0 0.99490 3.45 0.58
1595 39.0 51.0 0.99512 3.52 0.76
1596 29.0 40.0 0.99574 3.42 0.75
1597 32.0 44.0 0.99547 3.57 0.71
1598 18.0 42.0 0.99549 3.39 0.66
alcohol
0 9.4
1 9.8
2 9.8
3 9.8
4 9.4
... ...
1594 10.5
1595 11.2
1596 11.0
1597 10.2
1598 11.0
[1599 rows x 11 columns]
Label Binarizaton


WINE QUALITY PREDICTION - Ju
localhost:8888/notebooks/WINE%20QUALITY%20PREDICTION.ipynb
jupyter WINE QUALITY PREDICTION Last Checkpoint: 32 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Help
Trusted
Python 3 (ipykernel)
Run
Code
Label Binarizaton
In [32]:
Y = wine_dataset['quality'].apply(lambda y_value: 1 if y_value>=7 else 0)
print(Y)
0 0
1 0
2 0
3 0
4 0
..
1594 0
1595 0
1596 0
1597 0
1598 0
Name: quality, Length: 1599, dtype: int64
Train & Test Split
In [33]:
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, random_state=3)
In [34]:
print(Y.shape, Y_train.shape, Y_test.shape)
(1599,) (1279,) (320,)


WINE QUALITY PREDICTION - Ju
localhost:8888/notebooks/WINE%20QUALITY%20PREDICTION.ipynb
jupyter WINE QUALITY PREDICTION Last Checkpoint: 32 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Help
Trusted
Python 3 (ipykernel)
Run
Code
Model Training:
Random Forest Classifier
In [37]:
model = RandomForestClassifier()
model.fit(X_train, Y_train)
Out[37]:
RandomForestClassifier
RandomForestClassifier()
Model Evaluation
Accuracy Score
In [38]:
# accuracy on test data
X_test_prediction = model.predict(X_test)
test_data_accuracy = accuracy_score(X_test_prediction, Y_test)
In [39]:
print('Accuracy : ', test_data_accuracy)
Accuracy : 0.940625


WINE QUALITY PREDICTION - Ju
localhost:8888/notebooks/WINE%20QUALITY%20PREDICTION.ipynb
jupyter WINE QUALITY PREDICTION Last Checkpoint: 32 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Help
Trusted
Python 3 (ipykernel)
Run
Code
In [39]:
print('Accuracy : ', test_data_accuracy)
Accuracy : 0.940625
Building a Predictive System
In [43]:
input_data = (7.5,0.5,0.36,6.1,0.071,17.0,102.0,0.9978,3.35,0.8,10.5)
# changing the input data to a numpy array
input_data_as_numpy_array = np.asarray(input_data)
# reshape the data as we are predicting the label for only one instance
input_data_reshaped = input_data_as_numpy_array.reshape(1,-1)
prediction = model.predict(input_data_reshaped)
print(prediction)
if (prediction[0]==1):
print('Good Quality Wine')
else:
print('Bad Quality Wine')
[0]
Bad Quality Wine

TO: CHAITRA SUVARNA, chaitras822@gmail.com

@Zephyr technologies and solutions pvt ltd

********** **THANK YOU** **********