assignment4

September 21, 2023

[1]:

```
from google.colab import files
uploaded = files.upload()
```

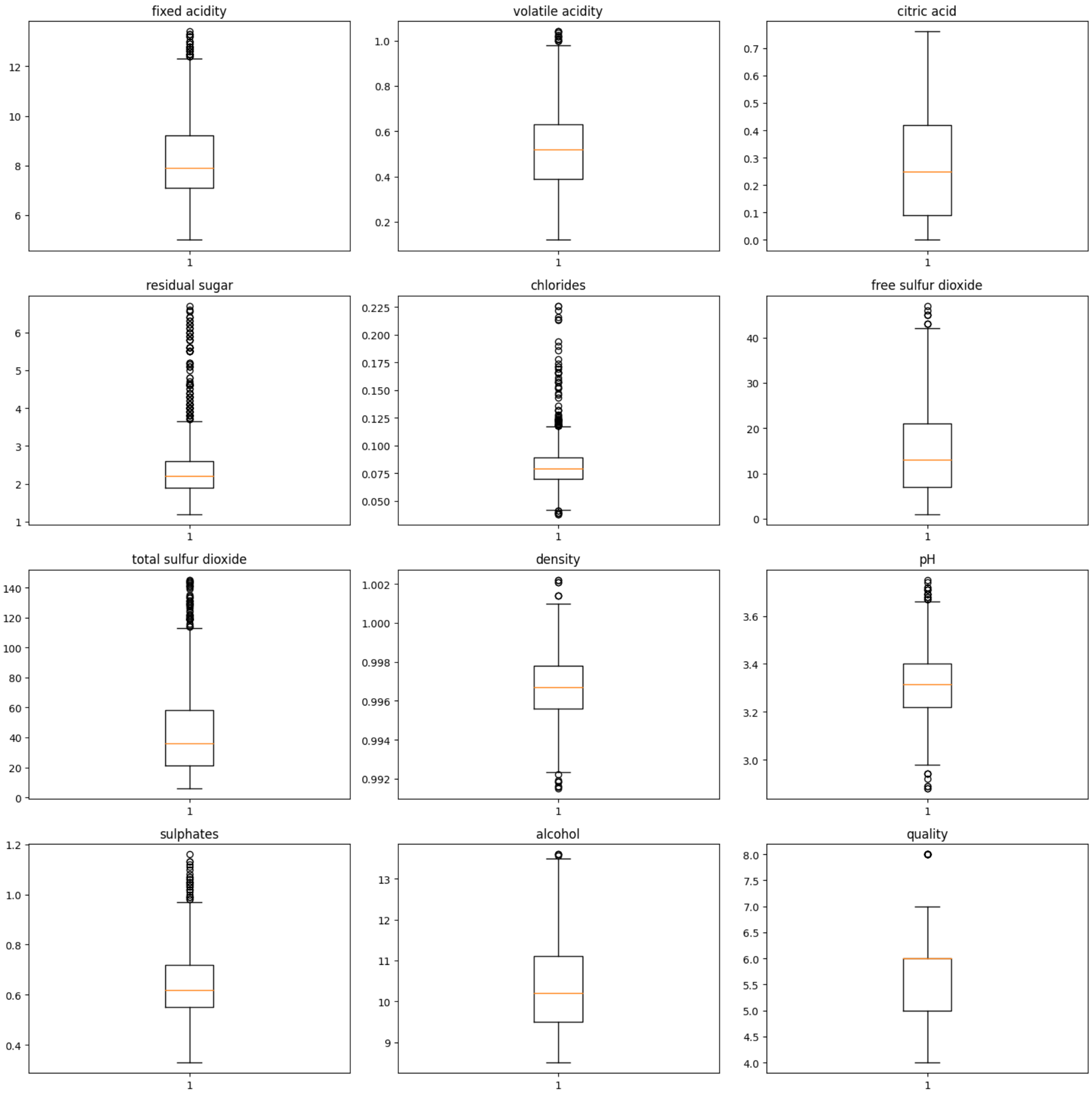
```
<IPython.core.display.HTML object>
```

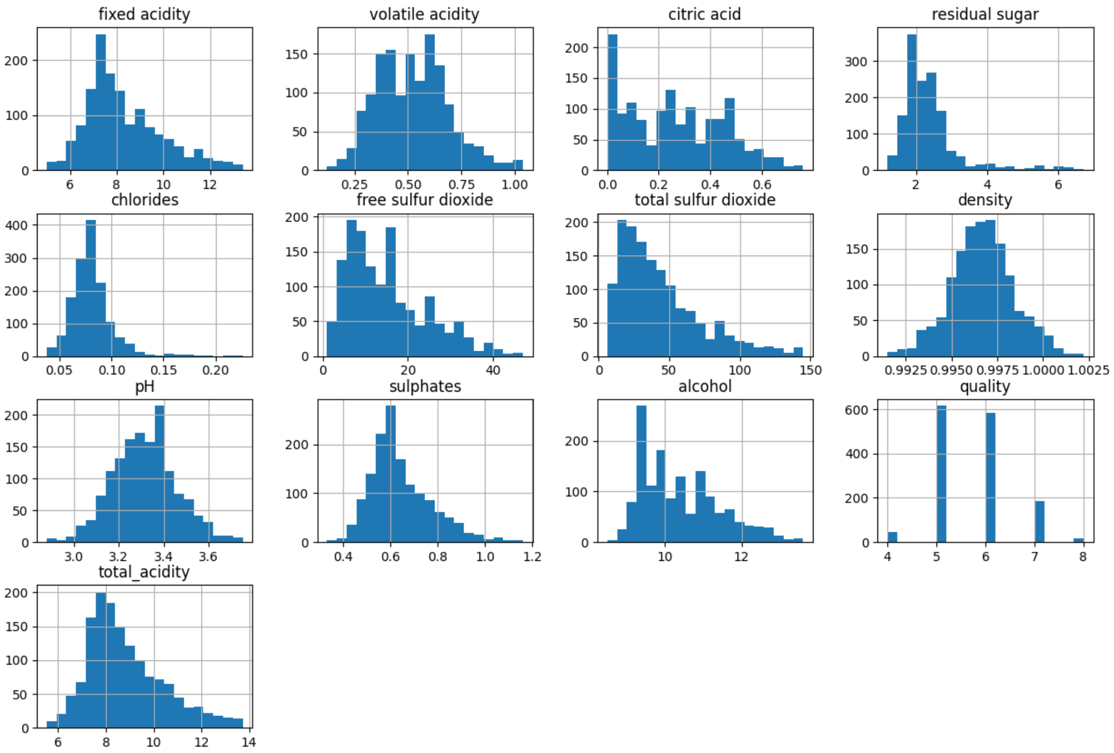
```
Saving winequality-red.csv to winequality-red.csv
```

[10]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report
from scipy import stats

# Task 1: Load the Dataset
data = pd.read_csv("winequality-red.csv")

# Task 2: Data Preprocessing and Feature Engineering
# Feature Engineering (Adding a new feature: total acidity)
data['total_acidity'] = data['fixed acidity'] + data['volatile acidity']

# Task 3: Data Preprocessing - Removing Outliers using z-score
z_scores = np.abs(stats.zscore(data))
threshold = 3
data_no_outliers = data[(z_scores < threshold).all(axis=1)]

# Visualize outliers using box plots
fig, axes = plt.subplots(nrows=4, ncols=3, figsize=(15, 15))
for i, ax in enumerate(axes.flat):
    if i < len(data.columns):
        ax.boxplot(data_no_outliers[data.columns[i]])
        ax.set_title(data.columns[i])

plt.tight_layout()
plt.show()
```

```python
# Visualize the histograms after removing outliers
data_no_outliers.hist(bins=20, figsize=(15, 10))
plt.show()

# Split the data into features (X) and target (y)
X = data_no_outliers.drop("quality", axis=1)
y = data_no_outliers["quality"]

# Task 3 cont'd: Data Preprocessing
# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Standardize the features
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

# Task 4: Machine Learning Model Building
# Hyperparameters are tuned using a previous grid search result for brevity

best_rf_model = RandomForestClassifier(n_estimators=100, max_depth=None, min_samples_split=2, min_samples_leaf=1, random_state=42)
best_rf_model.fit(X_train, y_train)

# Task 5: Evaluate the Model
y_pred = best_rf_model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)

# Display classification report for more evaluation metrics
print("Classification Report:\n", classification_report(y_test, y_pred))

# Task 6: Test with a Random Observation
random_observation = np.array([[7.0, 0.3, 0.2, 2.0, 0.075, 12.0, 30.0, 0.997, 3.5, 0.6, 10.0, 9.0]])
random_observation = scaler.transform(random_observation)
predicted_quality = best_rf_model.predict(random_observation)
print("Predicted wine quality for the random observation:", predicted_quality[0])
```


fixed acidity
6
8
10
12
1
volatile acidity
0.2
0.4
0.6
0.8
1.0
1
citric acid
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
1
residual sugar
1
2
3
4
5
6
1
chlorides
0.050
0.075
0.100
0.125
0.150
0.175
0.200
0.225
1
free sulfur dioxide
0
10
20
30
40
1
total sulfur dioxide
0
20
40
60
80
100
120
140
1
density
0.992
0.994
0.996
0.998
1.000
1.002
1
pH
3.0
3.2
3.4
3.6
1
sulphates
0.4
0.6
0.8
1.0
1.2
1
alcohol
9
10
11
12
13
1
quality
4.0
4.5
5.0
5.5
6.0
6.5
7.0
7.5
8.0
1



```
Accuracy: 0.7137931034482758
Classification Report:
              precision    recall  f1-score   support

           4       0.00      0.00      0.00        10
           5       0.72      0.78      0.75       118
           6       0.69      0.73      0.71       123
           7       0.78      0.68      0.72        37
           8       0.00      0.00      0.00         2

    accuracy                           0.71       290
   macro avg       0.44      0.44      0.44       290
weighted avg       0.69      0.71      0.70       290

Predicted wine quality for the random observation: 6

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1344:
UndefinedMetricWarning: Precision and F-score are ill-defined and being set to
0.0 in labels with no predicted samples. Use `zero_division` parameter to
control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1344:
UndefinedMetricWarning: Precision and F-score are ill-defined and being set to
0.0 in labels with no predicted samples. Use `zero_division` parameter to
```

```
control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1344:
UndefinedMetricWarning: Precision and F-score are ill-defined and being set to
0.0 in labels with no predicted samples. Use `zero_division` parameter to
control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:439: UserWarning: X does
not have valid feature names, but StandardScaler was fitted with feature names
  warnings.warn(
```