Name:- Shresthi Singh

Reg. No:- 21BCE2682

Assignment 4

Project Title: Grapes to Greatness: Machine Learning in Wine Quality Prediction

Task:

• Load the Dataset

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split

from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import classification_report

d = pd.read_csv('C:\\Users\\wwwad\\Downloads\\archive\\winequality-red.csv')
df=pd.DataFrame(d)
print(d)
```

```
      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
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  quality
0         9.4        5
1         9.8        5
```

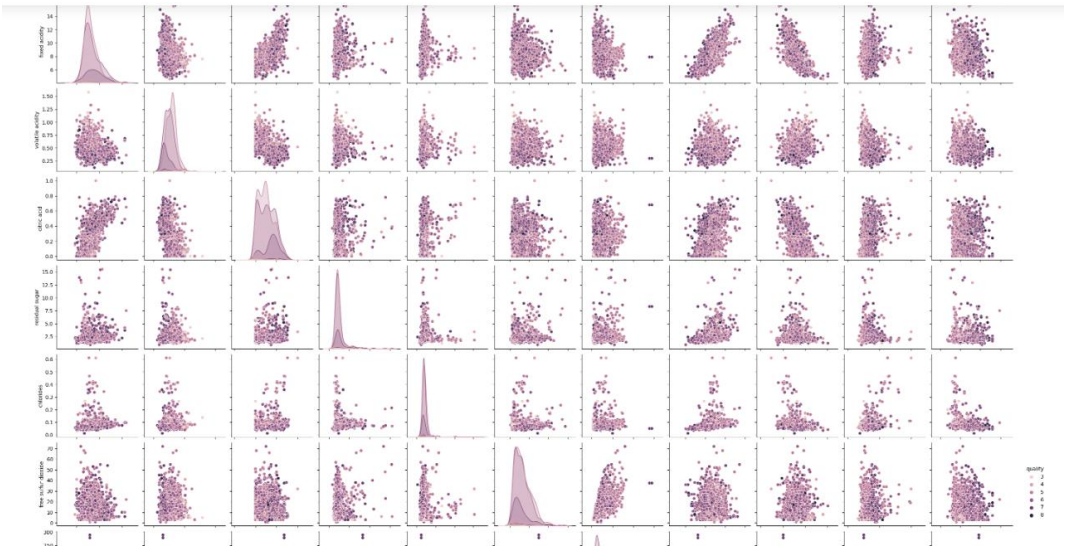
```
      alcohol  quality
0         9.4        5
1         9.8        5
2         9.8        5
3         9.8        6
4         9.4        5
...       ...      ...
1594     10.5        5
1595     11.2        6
1596     11.0        6
1597     10.2        5
1598     11.0        6

[1599 rows x 12 columns]
```

• Data preprocessing including visualization

```
print(d.info())
sns.pairplot(d, hue="quality")
plt.show()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1599 entries, 0 to 1598
Data columns (total 12 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   fixed acidity         1599 non-null   float64
 1   volatile acidity      1599 non-null   float64
 2   citric acid           1599 non-null   float64
 3   residual sugar        1599 non-null   float64
 4   chlorides             1599 non-null   float64
 5   free sulfur dioxide   1599 non-null   float64
 6   total sulfur dioxide  1599 non-null   float64
 7   density               1599 non-null   float64
 8   pH                    1599 non-null   float64
 9   sulphates             1599 non-null   float64
 10  alcohol               1599 non-null   float64
 11  quality               1599 non-null   int64  
dtypes: float64(11), int64(1)
memory usage: 150.0 KB
None
```



• Machine Learning Model building

```
X = d.drop("quality", axis=1)
y = d["quality"]

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=45)
clf = DecisionTreeClassifier(random_state=45)
clf.fit(X_train, y_train)
```

```
▾ DecisionTreeClassifier
DecisionTreeClassifier(random_state=45)
```

• Evaluate the model

```
y_pred = clf.predict(X_test)
report = classification_report(y_test, y_pred)
print("Classification Report:\n", report)
```

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

           3       0.00      0.00      0.00         4
           4       0.15      0.20      0.17        10
           5       0.69      0.75      0.72       132
           6       0.72      0.64      0.68       136
           7       0.57      0.59      0.58        34
           8       0.12      0.25      0.17         4

    accuracy                           0.65       320
   macro avg       0.38      0.40      0.39       320
weighted avg       0.66      0.65      0.65       320
```

- Test with random observation

```
random_sample = X.sample(n=5, random_state=42)
predictions = clf.predict(random_sample)
print("Predicted wine quality for random samples:")
for i, pred in enumerate(predictions):
    print(f"Sample {i + 1}: Predicted Quality = {pred}")
```

```
Predicted wine quality for random samples:
Sample 1: Predicted Quality = 6
Sample 2: Predicted Quality = 4
Sample 3: Predicted Quality = 6
Sample 4: Predicted Quality = 4
Sample 5: Predicted Quality = 6
```