In [1]:
```
import pandas as pd
import numpy as np
```

In [2]:
```
data = pd.read_csv('wine_data_processed.csv')

xin = data.drop('quality', axis='columns')
yout = data['quality']
```

In [3]:
```
# 75-25 split
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler

xin_train, xin_test, yout_train, yout_test = train_test_split(xin, yout, test_size=0.25, random_state=42)


# Data balancing
from imblearn.over_sampling import RandomOverSampler
ros = RandomOverSampler(random_state=42)
xin_train, yout_train = ros.fit_resample(xin_train, yout_train)
print(yout_train.value_counts())
```

```
quality
6    2102
5    2102
7    2102
Name: count, dtype: int64
```

In [4]:
```
# function for forward feature selection using KNN

print("The below code will help in identifying the most relevant features with a forward search using the k-Nearest Neig

from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score

def forward_feature_selection_using_knn(x_train, x_test, y_train, y_test):
    # store the selected features
    selected_features = []
    max_accuracy = 0

    while len(selected_features) < x_train.shape[1]:
        best_feature = None
        for feature in x_train.columns:
```

```
            if feature not in selected_features:
                selected_features.append(feature)
                knn = KNeighborsClassifier(n_neighbors=5)
                # Train the KNN using only the selected features
                knn.fit(x_train[selected_features], y_train)
                # Predict on the test set
                y_pred = knn.predict(x_test[selected_features])
                # Calculate accuracy
                accuracy = accuracy_score(y_test, y_pred)
                # Check if adding this feature improves the accuracy
                print(f"Selected Feature: {selected_features}")
                print(f"Accuracy: {accuracy}")
                if accuracy > max_accuracy:
                    max_accuracy = accuracy
                    best_feature = feature
                selected_features.remove(feature)
        # if the feature improves accuracy then add it to the set of selected features
        if best_feature is not None:
            selected_features.append(best_feature)
        else:
            break;
    return selected_features

print(xin_train.shape)

# Select the combination where we can achieve a higher accuracy with comparitively low num of features.
selected_features = forward_feature_selection_using_knn(xin_train, xin_test, yout_train, yout_test)
print("Final selected features:", selected_features)
```

```
The below code will help in identifying the most relevant features with a forward search using the k-Nearest Neighbors
algorithm.
(6306, 12)
Selected Feature: ['fixed acidity']
Accuracy: 0.4035667107001321
Selected Feature: ['volatile acidity']
Accuracy: 0.42932628797886396
Selected Feature: ['citric acid']
Accuracy: 0.39299867899603697
Selected Feature: ['residual sugar']
Accuracy: 0.45310435931307796
Selected Feature: ['chlorides']
Accuracy: 0.40885072655217963
Selected Feature: ['free sulfur dioxide']
Accuracy: 0.37318361955085866
Selected Feature: ['total sulfur dioxide']
Accuracy: 0.42602377807133424
Selected Feature: ['density']
Accuracy: 0.45904887714663145
Selected Feature: ['pH']
Accuracy: 0.38110964332893
Selected Feature: ['sulphates']
Accuracy: 0.40752972258916775
Selected Feature: ['alcohol']
Accuracy: 0.45772787318361957
Selected Feature: ['wine_type']
Accuracy: 0.3414795244385733
Selected Feature: ['density', 'fixed acidity']
Accuracy: 0.4795244385733157
Selected Feature: ['density', 'volatile acidity']
Accuracy: 0.4775429326287979
Selected Feature: ['density', 'citric acid']
Accuracy: 0.4848084544253633
Selected Feature: ['density', 'residual sugar']
Accuracy: 0.535006605019815
Selected Feature: ['density', 'chlorides']
Accuracy: 0.4808454425363276
Selected Feature: ['density', 'free sulfur dioxide']
Accuracy: 0.49273447820343463
Selected Feature: ['density', 'total sulfur dioxide']
Accuracy: 0.47424042272126815
Selected Feature: ['density', 'pH']
Accuracy: 0.47886393659180976
Selected Feature: ['density', 'sulphates']
Accuracy: 0.47886393659180976
```

```
Selected Feature: ['density', 'alcohol']
Accuracy: 0.5343461030383091
Selected Feature: ['density', 'wine_type']
Accuracy: 0.4630118890356671
Selected Feature: ['density', 'residual sugar', 'fixed acidity']
Accuracy: 0.46367239101717306
Selected Feature: ['density', 'residual sugar', 'volatile acidity']
Accuracy: 0.5026420079260238
Selected Feature: ['density', 'residual sugar', 'citric acid']
Accuracy: 0.4715984147952444
Selected Feature: ['density', 'residual sugar', 'chlorides']
Accuracy: 0.5085865257595773
Selected Feature: ['density', 'residual sugar', 'free sulfur dioxide']
Accuracy: 0.46036988110964333
Selected Feature: ['density', 'residual sugar', 'total sulfur dioxide']
Accuracy: 0.4418758256274769
Selected Feature: ['density', 'residual sugar', 'pH']
Accuracy: 0.452443857331572
Selected Feature: ['density', 'residual sugar', 'sulphates']
Accuracy: 0.47424042272126815
Selected Feature: ['density', 'residual sugar', 'alcohol']
Accuracy: 0.5568031704095112
Selected Feature: ['density', 'residual sugar', 'wine_type']
Accuracy: 0.5363276089828269
Selected Feature: ['density', 'residual sugar', 'alcohol', 'fixed acidity']
Accuracy: 0.5422721268163805
Selected Feature: ['density', 'residual sugar', 'alcohol', 'volatile acidity']
Accuracy: 0.583223249669749
Selected Feature: ['density', 'residual sugar', 'alcohol', 'citric acid']
Accuracy: 0.5528401585204755
Selected Feature: ['density', 'residual sugar', 'alcohol', 'chlorides']
Accuracy: 0.5541611624834875
Selected Feature: ['density', 'residual sugar', 'alcohol', 'free sulfur dioxide']
Accuracy: 0.5085865257595773
Selected Feature: ['density', 'residual sugar', 'alcohol', 'total sulfur dioxide']
Accuracy: 0.5257595772787318
Selected Feature: ['density', 'residual sugar', 'alcohol', 'pH']
Accuracy: 0.5422721268163805
Selected Feature: ['density', 'residual sugar', 'alcohol', 'sulphates']
Accuracy: 0.5528401585204755
Selected Feature: ['density', 'residual sugar', 'alcohol', 'wine_type']
Accuracy: 0.5700132100396301
Selected Feature: ['density', 'residual sugar', 'alcohol', 'volatile acidity', 'fixed acidity']
Accuracy: 0.5541611624834875
Selected Feature: ['density', 'residual sugar', 'alcohol', 'volatile acidity', 'citric acid']
```

```
Accuracy: 0.5852047556142669
Selected Feature: ['density', 'residual sugar', 'alcohol', 'volatile acidity', 'chlorides']
Accuracy: 0.5845442536327609
Selected Feature: ['density', 'residual sugar', 'alcohol', 'volatile acidity', 'free sulfur dioxide']
Accuracy: 0.5085865257595773
Selected Feature: ['density', 'residual sugar', 'alcohol', 'volatile acidity', 'total sulfur dioxide']
Accuracy: 0.5237780713342141
Selected Feature: ['density', 'residual sugar', 'alcohol', 'volatile acidity', 'pH']
Accuracy: 0.560105680317041
Selected Feature: ['density', 'residual sugar', 'alcohol', 'volatile acidity', 'sulphates']
Accuracy: 0.5785997357992074
Selected Feature: ['density', 'residual sugar', 'alcohol', 'volatile acidity', 'wine_type']
Accuracy: 0.5865257595772787
Selected Feature: ['density', 'residual sugar', 'alcohol', 'volatile acidity', 'wine_type', 'fixed acidity']
Accuracy: 0.5548216644649934
Selected Feature: ['density', 'residual sugar', 'alcohol', 'volatile acidity', 'wine_type', 'citric acid']
Accuracy: 0.5766182298546896
Selected Feature: ['density', 'residual sugar', 'alcohol', 'volatile acidity', 'wine_type', 'chlorides']
Accuracy: 0.582562747688243
Selected Feature: ['density', 'residual sugar', 'alcohol', 'volatile acidity', 'wine_type', 'free sulfur dioxide']
Accuracy: 0.5072655217965654
Selected Feature: ['density', 'residual sugar', 'alcohol', 'volatile acidity', 'wine_type', 'total sulfur dioxide']
Accuracy: 0.5250990752972259
Selected Feature: ['density', 'residual sugar', 'alcohol', 'volatile acidity', 'wine_type', 'pH']
Accuracy: 0.5700132100396301
Selected Feature: ['density', 'residual sugar', 'alcohol', 'volatile acidity', 'wine_type', 'sulphates']
Accuracy: 0.5772787318361955
Final selected features: ['density', 'residual sugar', 'alcohol', 'volatile acidity', 'wine_type']
```

From the results above, it is clear that the model performs best when using 5 features, namely, 'density', 'residual sugar', 'alcohol', 'volatile acidity', 'wine_type'.

Further, we will train the model using the training datset of these 5 features and then predict the labels for the test data. We will compute the training as well as the testing time and also computer the accuracy score and confusion metrics.

In [5]:
```
# training, prediction and record training time, accuracy and confusion matrix
import time
from sklearn.metrics import ConfusionMatrixDisplay

xin_selected_features = xin[selected_features]
print(xin_selected_features.head())
x_train, x_test, y_train, y_test = train_test_split(xin_selected_features, yout,test_size = 0.25, random_state=42)
knn = KNeighborsClassifier(n_neighbors=5)
start = time.time()
```

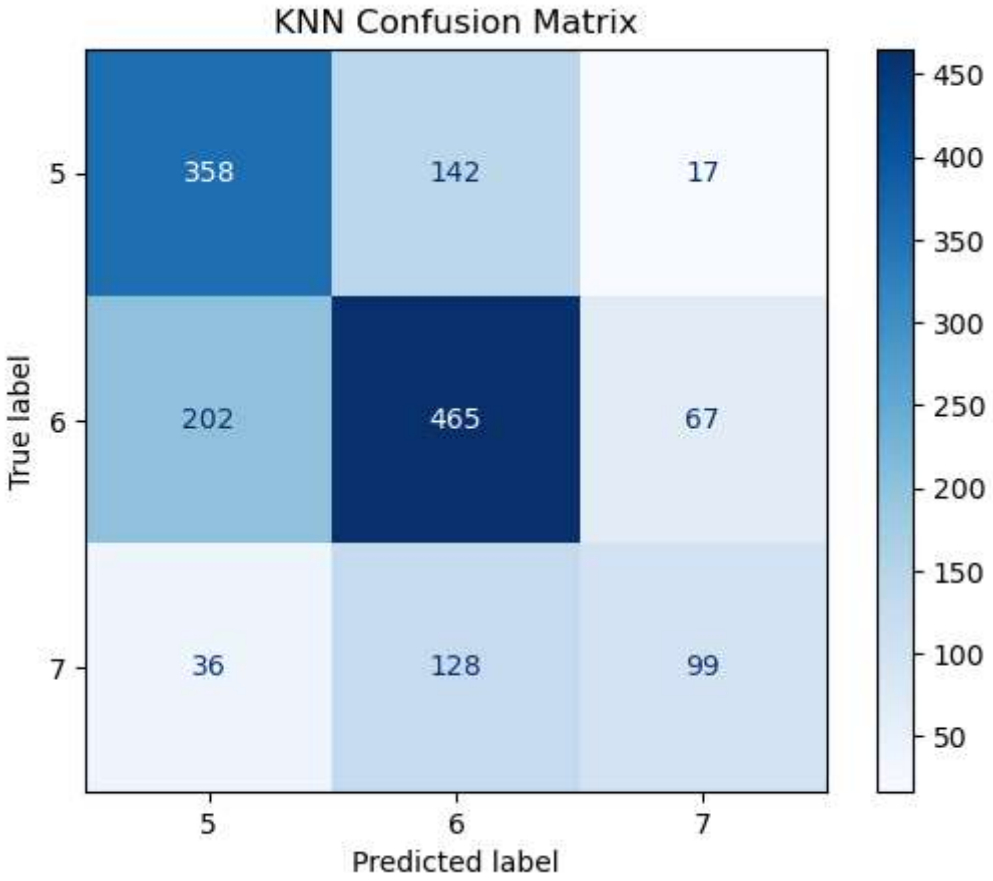
```
knn.fit(x_train, y_train)
stop = time.time()
print("Training Time: ", stop - start)

start = time.time()
y_pred = knn.predict(x_test)
stop = time.time()
print("Testing Time: ", stop - start)

print('Accuracy of KNN on test set: ', accuracy_score(y_test,y_pred))

import matplotlib.pyplot as plt
plt.figure(figsize=(8, 6))
ConfusionMatrixDisplay.from_predictions(y_test, y_pred, display_labels=[5, 6, 7], cmap='Blues')
plt.title('KNN Confusion Matrix')
plt.show()
```

```
   density  residual sugar  alcohol  volatile acidity  wine_type
0   0.9978             1.9      9.4              0.70          1
1   0.9968             2.6      9.8              0.88          1
2   0.9970             2.3      9.8              0.76          1
3   0.9980             1.9      9.8              0.28          1
4   0.9978             1.9      9.4              0.70          1
Training Time:  0.0029544830322265625
Testing Time:  0.033229827880859375
Accuracy of KNN on test set:  0.6089828269484808
<Figure size 800x600 with 0 Axes>
```



In [6]:
```
# forward feature selection using decision tree classifier

print("The below code will help in identifying the most relevant features with a forward search using the Decision Tree

from sklearn.tree import DecisionTreeClassifier

def forward_feature_selection_using_dtc(x_train, x_test, y_train, y_test):
    # store the selected features
    selected_features = []
    max_accuracy = 0.0

    while len(selected_features) < x_train.shape[1]:
        best_feature = None
        for feature in x_train.columns:
            if feature not in selected_features:
                selected_features.append(feature)
```

```
                dtc = DecisionTreeClassifier()
                # Train
                dtc.fit(x_train[selected_features], y_train)
                # Predict on the test set
                y_pred = dtc.predict(x_test[selected_features])
                # Calculate accuracy
                accuracy = accuracy_score(y_test, y_pred)
                # Check if adding this feature improves the accuracy
                print(f"Selected Feature: {selected_features}")
                print(f"Accuracy: {accuracy}")
                if accuracy > max_accuracy:
                    max_accuracy = accuracy
                    best_feature = feature
                selected_features.remove(feature)
        # if the feature improves accuracy then add it to the set of selected features
        if best_feature is not None:
            selected_features.append(best_feature)
        else:
            break;
    return selected_features

# Select the combination where we can achieve a higher accuracy with comparitively low num of features.
selected_features = forward_feature_selection_using_knn(xin_train, xin_test, yout_train, yout_test)
print("Final selected features:", selected_features)
```

```
The below code will help in identifying the most relevant features with a forward search using the Decision Tree algori
thm.
Selected Feature: ['fixed acidity']
Accuracy: 0.4035667107001321
Selected Feature: ['volatile acidity']
Accuracy: 0.42932628797886396
Selected Feature: ['citric acid']
Accuracy: 0.39299867899603697
Selected Feature: ['residual sugar']
Accuracy: 0.45310435931307796
Selected Feature: ['chlorides']
Accuracy: 0.40885072655217963
Selected Feature: ['free sulfur dioxide']
Accuracy: 0.37318361955085866
Selected Feature: ['total sulfur dioxide']
Accuracy: 0.42602377807133424
Selected Feature: ['density']
Accuracy: 0.45904887714663145
Selected Feature: ['pH']
Accuracy: 0.38110964332893
Selected Feature: ['sulphates']
Accuracy: 0.40752972258916775
Selected Feature: ['alcohol']
Accuracy: 0.45772787318361957
Selected Feature: ['wine_type']
Accuracy: 0.3414795244385733
Selected Feature: ['density', 'fixed acidity']
Accuracy: 0.4795244385733157
Selected Feature: ['density', 'volatile acidity']
Accuracy: 0.4775429326287979
Selected Feature: ['density', 'citric acid']
Accuracy: 0.4848084544253633
Selected Feature: ['density', 'residual sugar']
Accuracy: 0.535006605019815
Selected Feature: ['density', 'chlorides']
Accuracy: 0.4808454425363276
Selected Feature: ['density', 'free sulfur dioxide']
Accuracy: 0.49273447820343463
Selected Feature: ['density', 'total sulfur dioxide']
Accuracy: 0.47424042272126815
Selected Feature: ['density', 'pH']
Accuracy: 0.47886393659180976
Selected Feature: ['density', 'sulphates']
Accuracy: 0.47886393659180976
Selected Feature: ['density', 'alcohol']
```

```
Accuracy: 0.5343461030383091
Selected Feature: ['density', 'wine_type']
Accuracy: 0.4630118890356671
Selected Feature: ['density', 'residual sugar', 'fixed acidity']
Accuracy: 0.46367239101717306
Selected Feature: ['density', 'residual sugar', 'volatile acidity']
Accuracy: 0.5026420079260238
Selected Feature: ['density', 'residual sugar', 'citric acid']
Accuracy: 0.4715984147952444
Selected Feature: ['density', 'residual sugar', 'chlorides']
Accuracy: 0.5085865257595773
Selected Feature: ['density', 'residual sugar', 'free sulfur dioxide']
Accuracy: 0.46036988110964333
Selected Feature: ['density', 'residual sugar', 'total sulfur dioxide']
Accuracy: 0.4418758256274769
Selected Feature: ['density', 'residual sugar', 'pH']
Accuracy: 0.452443857331572
Selected Feature: ['density', 'residual sugar', 'sulphates']
Accuracy: 0.47424042272126815
Selected Feature: ['density', 'residual sugar', 'alcohol']
Accuracy: 0.5568031704095112
Selected Feature: ['density', 'residual sugar', 'wine_type']
Accuracy: 0.5363276089828269
Selected Feature: ['density', 'residual sugar', 'alcohol', 'fixed acidity']
Accuracy: 0.5422721268163805
Selected Feature: ['density', 'residual sugar', 'alcohol', 'volatile acidity']
Accuracy: 0.583223249669749
Selected Feature: ['density', 'residual sugar', 'alcohol', 'citric acid']
Accuracy: 0.5528401585204755
Selected Feature: ['density', 'residual sugar', 'alcohol', 'chlorides']
Accuracy: 0.5541611624834875
Selected Feature: ['density', 'residual sugar', 'alcohol', 'free sulfur dioxide']
Accuracy: 0.5085865257595773
Selected Feature: ['density', 'residual sugar', 'alcohol', 'total sulfur dioxide']
Accuracy: 0.5257595772787318
Selected Feature: ['density', 'residual sugar', 'alcohol', 'pH']
Accuracy: 0.5422721268163805
Selected Feature: ['density', 'residual sugar', 'alcohol', 'sulphates']
Accuracy: 0.5528401585204755
Selected Feature: ['density', 'residual sugar', 'alcohol', 'wine_type']
Accuracy: 0.5700132100396301
Selected Feature: ['density', 'residual sugar', 'alcohol', 'volatile acidity', 'fixed acidity']
Accuracy: 0.5541611624834875
Selected Feature: ['density', 'residual sugar', 'alcohol', 'volatile acidity', 'citric acid']
Accuracy: 0.5852047556142669
```

```
Selected Feature: ['density', 'residual sugar', 'alcohol', 'volatile acidity', 'chlorides']
Accuracy: 0.5845442536327609
Selected Feature: ['density', 'residual sugar', 'alcohol', 'volatile acidity', 'free sulfur dioxide']
Accuracy: 0.5085865257595773
Selected Feature: ['density', 'residual sugar', 'alcohol', 'volatile acidity', 'total sulfur dioxide']
Accuracy: 0.5237780713342141
Selected Feature: ['density', 'residual sugar', 'alcohol', 'volatile acidity', 'pH']
Accuracy: 0.560105680317041
Selected Feature: ['density', 'residual sugar', 'alcohol', 'volatile acidity', 'sulphates']
Accuracy: 0.5785997357992074
Selected Feature: ['density', 'residual sugar', 'alcohol', 'volatile acidity', 'wine_type']
Accuracy: 0.5865257595772787
Selected Feature: ['density', 'residual sugar', 'alcohol', 'volatile acidity', 'wine_type', 'fixed acidity']
Accuracy: 0.5548216644649934
Selected Feature: ['density', 'residual sugar', 'alcohol', 'volatile acidity', 'wine_type', 'citric acid']
Accuracy: 0.5766182298546896
Selected Feature: ['density', 'residual sugar', 'alcohol', 'volatile acidity', 'wine_type', 'chlorides']
Accuracy: 0.582562747688243
Selected Feature: ['density', 'residual sugar', 'alcohol', 'volatile acidity', 'wine_type', 'free sulfur dioxide']
Accuracy: 0.5072655217965654
Selected Feature: ['density', 'residual sugar', 'alcohol', 'volatile acidity', 'wine_type', 'total sulfur dioxide']
Accuracy: 0.5250990752972259
Selected Feature: ['density', 'residual sugar', 'alcohol', 'volatile acidity', 'wine_type', 'pH']
Accuracy: 0.5700132100396301
Selected Feature: ['density', 'residual sugar', 'alcohol', 'volatile acidity', 'wine_type', 'sulphates']
Accuracy: 0.5772787318361955
Final selected features: ['density', 'residual sugar', 'alcohol', 'volatile acidity', 'wine_type']
```

From the results above, it is clear that the model performs best when using 5 features, namely, 'density', 'residual sugar', 'alcohol', 'volatile acidity', 'wine_type'.

Further, we will train the model using the training datset of these 5 features and then predict the labels for the test data. We will compute the training as well as the testing time and also computer the accuracy score and confusion metrics.

In [14]:
```
# training, prediction and record training time, accuracy and confusion matrix
import time
from sklearn.metrics import ConfusionMatrixDisplay

xin_selected_features = xin[selected_features]
print(xin_selected_features.head())
x_train, x_test, y_train, y_test = train_test_split(xin_selected_features, yout,test_size = 0.25, random_state=42)

start = time.time()
dtc = DecisionTreeClassifier()
```

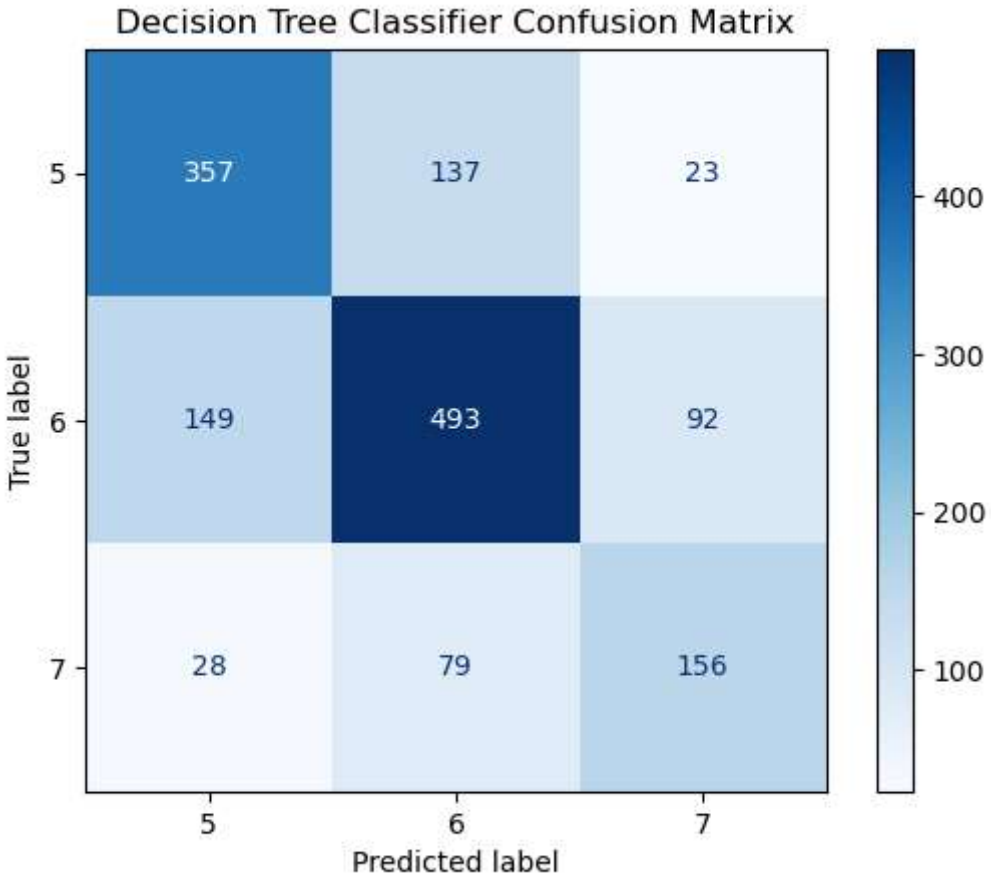
```
dtc.fit(x_train, y_train)
stop = time.time()
print("Training Time: ", stop - start)

start = time.time()
y_pred = dtc.predict(x_test)
stop = time.time()
print("Testing Time: ", stop - start)

print('Accuracy of Decision Tree Classifier on test set: ', accuracy_score(y_test,y_pred))

import matplotlib.pyplot as plt
plt.figure(figsize=(8, 6))
ConfusionMatrixDisplay.from_predictions(y_test, y_pred, display_labels=[5, 6, 7], cmap='Blues')
plt.title('Decision Tree Classifier Confusion Matrix')
plt.show()
```

```
   density  residual sugar  alcohol  volatile acidity  wine_type
0   0.9978             1.9      9.4              0.70          1
1   0.9968             2.6      9.8              0.88          1
2   0.9970             2.3      9.8              0.76          1
3   0.9980             1.9      9.8              0.28          1
4   0.9978             1.9      9.4              0.70          1
Training Time:  0.015247821807861328
Testing Time:  0.0
Accuracy of Decision Tree Classifier on test set:  0.6644649933949802
<Figure size 800x600 with 0 Axes>
```


Decision Tree Classifier Confusion Matrix
357
137
23
149
493
92
28
79
156
5
6
7
True label
5
6
7
Predicted label
400
300
200
100

In [ ]:

In [ ]: