```
import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score
from sklearn.model_selection import cross_val_score
from sklearn.neighbors import KNeighborsClassifier
from sklearn.preprocessing import StandardScaler

# Load the wine dataset from a CSV file
red_train = pd.read_csv('red_wine_train.csv', index_col=0)
red_test = pd.read_csv('red_wine_test.csv', index_col=0)

total = pd.concat([red_train, red_test])
total = total.drop(["quality"], axis=1)
scaler = StandardScaler()
scaler.fit(total)


# Split the data into training and test sets
X_train = red_train.drop("quality", axis=1)
X_test = red_train.drop("quality", axis=1)
y_train = red_train["quality"]
y_test = red_train["quality"]

X_train_scaled = scaler.transform(X_train)
X_test_scaled = scaler.transform(X_test)

X_train = pd.DataFrame(X_train_scaled, columns=X_train.columns)
X_test = pd.DataFrame(X_test_scaled, columns=X_test.columns)

X = pd.concat([X_train, X_test])
y = pd.concat([y_test, y_train])

# Train a KNN classifier with k=2 on the training data
knn = KNeighborsClassifier(n_neighbors=5)
knn.fit(X_train, y_train)

# Make predictions on the test data
y_pred = knn.predict(X_test)

# Compute the accuracy of the predictions
prediction_accuracy = accuracy_score(y_test, y_pred)
print("Prediction Accuracy: {:.2f}%".format(prediction_accuracy*100))

# Compute cross-validation scores for different values of k
k_values = [i for i in range(2, 31)]
scores = []


for k in k_values:
    knn = KNeighborsClassifier(n_neighbors=k)
    score = cross_val_score(knn, X_train, y_train, cv=5)
    scores.append(np.mean(score))

# Plot the cross-validation scores vs. k
plt.plot(k_values, scores, marker='o')
plt.xlabel("K values")
plt.ylabel("Accuracy score")

# Train a KNN classifier with the best value of k on the training data
knn = KNeighborsClassifier(n_neighbors=22)
knn.fit(X_train, y_train)

# Make predictions on the test data
y_pred = knn.predict(X_test)

# Compute scores
```

```
accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred, average='weighted', zero_division=1)
recall = recall_score(y_test, y_pred, average='weighted', zero_division=1)
f1 = f1_score(y_test, y_pred, average='weighted', zero_division=1)

# Print
print("Accuracy: {:.2f}%".format(accuracy*100))
print("Precision: {:.2f}%".format(precision*100))
print("Recall: {:.2f}%".format(recall*100))
print("F1 score: {:.2f}%".format(f1*100))

plt.show()
```