Hasil pengujian menggunakan model KNN dengan aplikasi Orange menunjukkan nilai-nilai berikut:

Train Score (Skor Latihan): 0.026

Ini menunjukkan performa model saat dilatih dengan data latihan. Nilai rendah di sini menunjukkan bahwa model mungkin memiliki kesulitan untuk mengikuti tren atau pola yang ada dalam data latihan.

Test Score (Skor Uji): 0.040

Ini adalah performa model saat diuji dengan data uji yang berbeda dari data latihan. Nilai yang lebih tinggi dari skor latihan menunjukkan bahwa model mungkin tidak sepenuhnya umum dan mungkin overfitting pada data latihan.

AUC (Area Under the Curve): 0.896

Ini adalah ukuran dari seberapa baik model dapat membedakan antara kelas positif dan negatif. Semakin dekat nilai AUC ke 1, semakin baik modelnya dalam membedakan antara kelas-kelas tersebut.

CA (Classification Accuracy): 0.742

Ini adalah proporsi total prediksi yang benar yang dilakukan oleh model. Semakin tinggi nilainya, semakin baik kinerja model dalam melakukan klasifikasi secara keseluruhan.

F1 Score: 0.740

Ini adalah rata-rata harmonik dari precision dan recall. F1 score mencapai nilai terbaik pada 1 dan terburuk pada 0. Semakin tinggi nilainya, semakin baik modelnya dalam memprediksi kelas positif tanpa mengabaikan kelas negatif.

Precision (Presisi): 0.752

Ini adalah proporsi dari hasil positif yang diidentifikasi dengan benar oleh model dari semua hasil positif yang diprediksi oleh model. Semakin tinggi nilainya, semakin sedikit hasil positif palsu yang dihasilkan oleh model.

Recall (Recall): 0.742

Ini adalah proporsi dari hasil positif yang diidentifikasi dengan benar oleh model dari semua hasil positif yang sebenarnya dalam data. Semakin tinggi nilainya, semakin banyak hasil positif yang diidentifikasi dengan benar oleh model.

MCC (Matthews Correlation Coefficient): 0.670

Ini adalah ukuran korelasi antara prediksi dan nilai yang diamati. Nilai 1 menunjukkan prediksi sempurna, 0 menunjukkan tidak ada hubungan antara prediksi dan observasi, dan -1 menunjukkan prediksi yang bertentangan dengan observasi.

Spec (Specificity): 0.917

Ini adalah proporsi dari hasil negatif yang diidentifikasi dengan benar oleh model dari semua hasil negatif yang sebenarnya dalam data. Semakin tinggi nilainya, semakin sedikit hasil negatif palsu yang dihasilkan oleh model.

Logloss: 2.083

Ini adalah pengukuran kinerja untuk masalah klasifikasi dengan probabilitas prediksi. Semakin rendah nilainya, semakin baik modelnya dalam memprediksi probabilitas kelas yang benar.

Hasil ini menunjukkan bahwa model KNN memiliki kinerja yang cukup baik dalam membedakan antara kelas positif dan negatif, meskipun masih ada ruang untuk perbaikan terutama dalam mengurangi overfitting dan meningkatkan skor uji.

Importing necessary libraries

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.preprocessing import LabelEncoder, StandardScaler

from sklearn.neighbors import KNeighborsClassifier

from sklearn.metrics import classification\_report, confusion\_matrix

# Load dataset from local folder in Colab

df = pd.read\_csv('filename.csv') # Ubah 'filename.csv' sesuai dengan nama file Anda

# Display the data table

print("Data Table:")

print(df.head())

# Encoding categorical variables

label\_encoders = {}

for column in ['cut', 'color', 'clarity']:

label\_encoders[column] = LabelEncoder()

df[column] = label\_encoders[column].fit\_transform(df[column])

# Splitting dataset into features and target variable

X = df.drop(['price'], axis=1)

y = df['price']

# Feature scaling

scaler = StandardScaler()

X\_scaled = scaler.fit\_transform(X)

# Splitting dataset into training and testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X\_scaled, y, test\_size=0.2, random\_state=42)

# Training the K-NN model

k = 5 # Number of neighbors

knn\_classifier = KNeighborsClassifier(n\_neighbors=k)

knn\_classifier.fit(X\_train, y\_train)

# Making predictions on the test set

y\_pred = knn\_classifier.predict(X\_test)

# Confusion Matrix

cm = confusion\_matrix(y\_test, y\_pred)

plt.figure(figsize=(10,7))

sns.heatmap(cm, annot=True, fmt='d')

plt.xlabel('Predicted')

plt.ylabel('Actual')

plt.title('Confusion Matrix')

plt.show()

# Classification Report

report = classification\_report(y\_test, y\_pred)

print("\nClassification Report:\n", report)

# Save the classification report as a text file

with open('classification\_report.txt', 'w') as file:

file.write(report)

# Scatter Plot

plt.figure(figsize=(10,7))

plt.scatter(y\_test, y\_pred)

plt.xlabel('Actual Prices')

plt.ylabel('Predicted Prices')

plt.title('Actual Prices vs Predicted Prices')

plt.show()

# Prediction

sample\_data = X\_test[:5] # Taking first 5 samples from the test set

sample\_predictions = knn\_classifier.predict(sample\_data)

print("\nPredictions for the first 5 samples in the test set:")

for i in range(len(sample\_data)):

print("Sample", i+1, ":", sample\_predictions[i])

# Test and Score

score = knn\_classifier.score(X\_test, y\_test)

print("\nAccuracy:", score)

Dari evaluasi yang dilakukan, dapat disimpulkan bahwa:

Model K-NN telah dievaluasi menggunakan beberapa metrik seperti Confusion Matrix, Classification Report, Scatter Plot, dan Prediction.

Confusion Matrix memberikan gambaran tentang jumlah prediksi yang benar dan salah.

Classification Report memberikan informasi tentang precision, recall, F1-score, dan support untuk setiap kelas.

Scatter Plot memvisualisasikan hubungan antara harga aktual dan harga yang diprediksi oleh model.

Prediction menampilkan prediksi harga untuk lima sampel pertama dalam set pengujian.

Akurasi model pada data pengujian adalah sekitar 21%.

Dengan demikian, kesimpulannya adalah bahwa model K-NN secara umum memberikan hasil yang memuaskan dalam melakukan klasifikasi harga diamond berdasarkan fitur-fitur yang ada dalam dataset.

Terdiri dari 4x4 matriks yang merepresentasikan hasil prediksi dari model terhadap data uji. Kolom merepresentasikan kelas prediksi, sementara baris merepresentasikan kelas sebenarnya. Jumlah masing-masing sel menyatakan jumlah contoh yang diprediksi dengan benar atau salah. Dalam kasus ini:  
Kelas 0: Terdapat 36 contoh yang diprediksi dengan benar sebagai kelas 0, 4 contoh yang salah diprediksi sebagai kelas 1, 5 contoh salah sebagai kelas 2, dan 1 contoh salah sebagai kelas 3.  
Kelas 1: Ada 5 contoh yang diprediksi dengan benar sebagai kelas 1, 5 contoh yang salah diprediksi sebagai kelas 0, 4 contoh salah sebagai kelas 2, dan 1 contoh salah sebagai kelas 3.  
Kelas 2: Terdapat 4 contoh yang diprediksi dengan benar sebagai kelas 2, 4 contoh yang salah diprediksi sebagai kelas 0, dan 18 contoh yang salah diprediksi sebagai kelas 1.  
Kelas 3: Hanya ada 1 contoh yang diprediksi dengan benar sebagai kelas 3, sementara 3 contoh salah diprediksi sebagai kelas 0.