# Import required libraries

import pandas as pd

import numpy as np

from sklearn import svm

from sklearn.model\_selection import train\_test\_split

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

from sklearn.preprocessing import StandardScaler

import matplotlib.pyplot as plt

import seaborn as sns

# Load the Wine Quality dataset

url = "https://archive.ics.uci.edu/ml/machine-learning-databases/wine-quality/winequality-red.csv"

df = pd.read\_csv(url, sep=";")

# Drop the 'quality' column and create a new column 'quality\_class' based on the quality score

df['quality\_class'] = pd.cut(df['quality'], bins=[0, 4, 7, 10], labels=[0, 1, 2], include\_lowest=True)

# Define the features (X) and the target variable (y)

X = df.drop(['quality', 'quality\_class'], axis=1)

y = df['quality\_class']

# 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 using StandardScaler

scaler = StandardScaler()

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

X\_test\_scaled = scaler.transform(X\_test)

# Train an SVM model with a radial basis function (RBF) kernel

svm\_model = svm.SVC(kernel='rbf', C=1, gamma='scale')

svm\_model.fit(X\_train\_scaled, y\_train)

# Make predictions on the testing set

y\_pred = svm\_model.predict(X\_test\_scaled)

# Evaluate the model's performance

accuracy = accuracy\_score(y\_test, y\_pred)

print("Accuracy:", accuracy)

print("Classification Report:")

print(classification\_report(y\_test, y\_pred))

print("Confusion Matrix:")

print(confusion\_matrix(y\_test, y\_pred))

# Visualize the confusion matrix using a heatmap

plt.figure(figsize=(8, 6))

sns.heatmap(confusion\_matrix(y\_test, y\_pred), annot=True, cmap='Blues')

plt.xlabel("Predicted labels")

plt.ylabel("True labels")

plt.show()