import numpy as np import pandas as pd import matplotlib.pyplot as plt import seaborn as sns from sklearn.model selection import train test split from sklearn.ensemble import RandomForestClassifier from sklearn.metrics import accuracy score # loading the dataset to a Pandas DataFrame wine dataset = pd.read csv('D:\Machine learning\Wine quality prediction\winequality-red.csv') wine_dataset.head() volatile citric free sulfur fixed residual total sulfur chlorides pH sulphates alcohol quality density acidity acidity acid sugar dioxide dioxide 0 7.4 0.00 0.076 0.9978 3.51 9.4 5 0.70 1.9 11.0 34.0 0.56 1 9.8 7.8 0.88 0.00 0.098 0.9968 5 2.6 25.0 67.0 3.20 0.68 0.04 2 0.9970 3.26 5 7.8 0.76 2.3 0.092 15.0 54.0 0.65 9.8 3 11.2 0.28 0.56 1.9 0.075 17.0 60.0 0.9980 3.16 0.58 9.8 6 4 7.4 0.70 0.00 1.9 0.076 0.56 9.4 5 11.0 34.0 0.9978 3.51 In [4]: #number of columns of rows and dataset wine dataset.shape (1599, 12)Out[4]: wine_dataset.isnull().sum() Out[7]: fixed acidity 0 volatile acidity 0 citric acid 0 residual sugar 0 chlorides 0 free sulfur dioxide 0 total sulfur dioxide 0 density 0 sulphates 0 alcohol 0 quality 0 dtype: int64 wine_dataset.describe() free sulfur total sulfur fixed volatile residual citric acid chlorides density pН sulphates acidity acidity dioxide dioxide sugar count 1599.000000 1599.000000 1599.000000 1599.000000 1599.000000 1599.000000 1599.000000 1599.000000 1599.000000 1599.000000 1599 mean 8.319637 0.527821 0.270976 2.538806 0.087467 15.874922 46.467792 0.996747 3.311113 0.658149 1(std 1.741096 0.179060 0.194801 1.409928 0.047065 10.460157 32.895324 0.001887 0.154386 0.169507 min 4.600000 0.120000 0.000000 0.900000 0.012000 1.000000 6.000000 0.990070 2.740000 0.330000 25% 7.100000 0.390000 0.090000 1.900000 0.070000 7.000000 22.000000 0.995600 3.210000 0.550000 ć **50**% 7.900000 0.520000 0.260000 2.200000 0.079000 14.000000 38.000000 0.996750 3.310000 0.620000 1(**75**% 9.200000 0.640000 0.420000 2.600000 0.090000 21.000000 62.000000 0.997835 3.400000 0.730000 11 15.500000 0.611000 2.000000 max 15.900000 1.580000 1.000000 72.000000 289.000000 1.003690 4.010000 14 In [9]: # no of values for each quality sns.catplot(x='quality', data = wine_dataset, kind='count') Out[9]: <seaborn.axisgrid.FacetGrid at 0x22c75b6ce80> 700 -600 500 400 count 300 200 100 6 quality # volatile vs quality plot =plt.figure(figsize=(5,5)) sns.barplot(x='quality',y='volatile acidity',data = wine_dataset) Out[12]: <AxesSubplot:xlabel='quality', ylabel='volatile acidity'> 1.0 0.8 volatile acidity 0.6 0.4 0.2 0.0 4 Ś 6 quality # Correlation # Positive Correlation # Negative Correlation correlation= wine_dataset.corr() plt.figure(figsize=(10,10)) sns.heatmap(correlation,cbar=True, square=True, fmt='.2f',annot=True,cmap='Blues') Out[61]: <AxesSubplot:> 1.0 - 0.8 0.11 0.09 -0.15 -0.11 -0.68 -0.06 0.12 fixed acidity 1.00 -0.26 0.00 0.06 -0.01 0.08 0.02 -0.26 -0.20 -0.39 volatile acidity -0.261.00 -0.55- 0.6 1.00 -0.06 0.04 0.11 0.67 -0.55-0.54citric acid 0.11 0.00 1.00 0.06 -0.09 0.01 0.04 0.01 residual sugar - 0.4 chlorides 0.09 0.06 0.06 1.00 0.01 0.05 -0.27-0.22 -0.13 free sulfur dioxide -0.15 -0.01 -0.06 0.01 1.00 0.67 -0.02 0.07 0.05 -0.07 0.2 total sulfur dioxide - -0.11 0.07 -0.07 0.04 0.08 0.04 0.05 0.67 1.00 -0.21 -0.19 - 0.0 0.67 0.02 -0.02 0.07 1.00 -0.34 -0.50 -0.17 density pH - -0.68 -0.54-0.09 -0.27 0.07 -0.07 -0.34 1.00 -0.20 -0.06- -0.2 0.01 0.05 0.04 0.09 -0.26-0.20 1.00 sulphates 0.11 0.04 -0.22 -0.07 -0.21 0.09 alcohol - -0.06 -0.20 -0.50 1.00 0.48 -0.4 -0.13 -0.05 -0.19 quality 0.12 -0.390.01 -0.17-0.06 0.48 1.00 chlorides fixed acidity free sulfur dioxide total sulfur dioxide alcohol volatile acidity residual sugal density 돐 sulphates - -0.6 # Data Preprocessing X= wine_dataset.drop('quality',axis=1) Χ total sulfur fixed volatile citric free sulfur residual chlorides pH sulphates alcohol density acidity acidity acid dioxide dioxide sugar 0.00 34.0 0.99780 0 7.4 0.700 1.9 0.076 11.0 3.51 0.56 9.4 1 0.880 0.00 0.098 0.99680 0.68 7.8 2.6 25.0 3.20 9.8 67.0 2 7.8 0.760 0.04 2.3 0.092 15.0 54.0 0.99700 3.26 0.65 9.8 3 0.280 0.56 0.075 0.99800 11.2 1.9 17.0 60.0 3.16 0.58 9.8 4 7.4 0.700 0.00 1.9 0.076 0.99780 3.51 11.0 34.0 0.56 9.4 1594 6.2 0.600 0.08 0.090 44.0 0.99490 3.45 0.58 10.5 2.0 32.0 1595 5.9 0.550 0.10 2.2 0.062 39.0 51.0 0.99512 3.52 0.76 11.2 1596 0.510 0.076 29.0 6.3 0.13 2.3 40.0 0.99574 3.42 0.75 11.0 1597 5.9 0.645 0.12 2.0 0.075 32.0 44.0 0.99547 3.57 0.71 10.2 1598 6.0 0.310 18.0 0.47 3.6 0.067 42.0 0.99549 3.39 0.66 11.0 1599 rows × 11 columns Y= wine_dataset['quality'].apply(lambda y_value: 1 if y_value>6 or 2 if y_value==6 or 3 y_value<6) File "<ipython-input-1-1a91e140e5ca>", line 1 Y= wine_dataset['quality'].apply(lambda y_value: 1 if y_value>6 or 2 if y_value==6 or 3 y_value<6) SyntaxError: invalid syntax 0 0 0 2 0 3 0 4 0 1594 0 1595 0 1596 0 1597 1598 Name: quality, Length: 1599, dtype: int64 #Train Test split In [24]: X_train, X_test, Y_train, Y_test= train_test_split(X, Y, test_size=0.2, random_state=3) # Model training # Random Forest Classifier model= RandomForestClassifier() In [28]: model.fit(X_train,Y_train) Out[28]: RandomForestClassifier() In [31]: # Model Evaluation # Accuracy Score X_test_prediction= model.predict(X_test) In [34]: test_data_accuracy = accuracy_score(X_test_prediction,Y_test) test_data_accuracy Out[35]: 0.93125 $input_data = (7.5, 0.5, 0.36, 6.1, 0.071, 17.0, 102.0, 0.9978, 3.35, 0.8, 10.5)$ In [47]: # changing the input data to a numpy array input_data_as_numpy_array= np.asarray(input data) # reshape the data as we are predicting the label for only one instance input_data_reshaped= input_data_as_numpy_array.reshape(1,-1) In [48]: prediction=model.predict(input_data_reshaped) if (prediction[0] == 1): In [49]: print('Good Quality Wine') print('Bad Quality') Bad Quality