import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.tree import DecisionTreeClassifier

from sklearn.naive\_bayes import CategoricalNB

from sklearn.ensemble import RandomForestClassifier

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

dataframe = pd.read\_csv("predictiondata.csv")

dataframe.head()

dataframe = pd.read\_csv("sampledata.csv")

dataframe.head()

dataframe.info()

dataframe.describe()

dataframe.groupby('Overall Dificulty').describe()

dataframe.groupby('CGPA:').describe()

dataframe.hist(column='Overall Dificulty',by='CGPA:',bins=5,figsize=(12,8))

sns.countplot(x='Overall Dificulty',data=dataframe,palette='rainbow')

features=dataframe.drop(['Overall Dificulty'],axis=1).values

targets=dataframe['Overall Dificulty'].values

features\_train, targets\_train = features[0:400], targets[0:400]

features\_test, targets\_test = features[400:], targets[400:]

model=CategoricalNB()

model.fit(features\_train, targets\_train)

pred=model.predict(features\_test)

print(pred)

print(confusion\_matrix(targets\_test,pred))

print(classification\_report(targets\_test, pred))

nb\_accuracy=model.score(features\_test, targets\_test)\*100

print(nb\_accuracy)

rf = RandomForestClassifier(n\_estimators=20, random\_state=100)

rf.fit(features\_train, targets\_train)

y\_pred = rf.predict(features\_test)

print(y\_pred)

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

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

rf\_accuracy=rf.score(features\_test, targets\_test)\*100

print(rf\_accuracy)

classifier= DecisionTreeClassifier(criterion='entropy', random\_state=0)

classifier.fit(features\_train, targets\_train)

x\_pred = classifier.predict(features\_test)

print(x\_pred)

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

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

classifier\_accuracy=classifier.score(features\_test, targets\_test)\*100

print(classifier\_accuracy)

data = {'Random Forest' : rf\_accuracy, 'Naive Bayes' : nb\_accuracy, 'Decission Tree': classifier\_accuracy }

algorithms = data.keys()

results = data.values()

fig = plt.figure(figsize = (5, 5))

barlist = plt.bar(algorithms, results, width = 0.3)

barlist[0].set\_color('orange')

barlist[1].set\_color('green')

barlist[2].set\_color('blue')

plt.xlabel("Algorithm")

plt.ylabel("Accuracy")

plt.ylim(80, 100)

plt.title("Question paper analysis")

plt.show()











































