import pandas as pd  
from sklearn.preprocessing import LabelEncoder, OneHotEncoder  
  
  
file\_path = 'sleep\_health\_lifestyle\_dataset.csv'   
data = pd.read\_csv(file\_path)  
  
  
label\_encoder = LabelEncoder()  
onehot\_encoder = OneHotEncoder(sparse\_output=False, drop='first')  
  
  
label\_columns = ['Gender', 'BMI Category', 'Sleep Disorder']  
for col in label\_columns:  
 data[f'{col}\_Encoded'] = label\_encoder.fit\_transform(data[col])  
  
  
onehot\_columns = ['Occupation']  
for col in onehot\_columns:  
 encoded = pd.DataFrame(  
 onehot\_encoder.fit\_transform(data[[col]]),  
 columns=onehot\_encoder.get\_feature\_names\_out([col])  
 )  
 data = pd.concat([data, encoded], axis=1)  
  
  
data[['BP\_Systolic', 'BP\_Diastolic']] = data['Blood Pressure (systolic/diastolic)'].str.split('/', expand=True).astype(int)  
  
  
columns\_to\_drop = label\_columns + onehot\_columns + ['Blood Pressure (systolic/diastolic)']  
data\_encoded = data.drop(columns=columns\_to\_drop, axis=1)  
  
  
print(data\_encoded.head())  
  
  
data\_encoded.to\_csv('encoded\_sleep\_health\_lifestyle\_dataset.csv', index=False)

from sklearn.preprocessing import StandardScaler  
import pandas as pd  
  
  
file\_path = 'encoded\_sleep\_health\_lifestyle\_dataset.csv'   
data\_encoded = pd.read\_csv(file\_path)  
  
  
scaler = StandardScaler()  
  
  
columns\_to\_scale = data\_encoded.columns.drop('Person ID')   
data\_encoded[columns\_to\_scale] = scaler.fit\_transform(data\_encoded[columns\_to\_scale])  
  
  
data\_encoded.to\_csv('standardized\_sleep\_health\_lifestyle\_dataset.csv', index=False)  
  
print("Standardized dataset saved as 'standardized\_sleep\_health\_lifestyle\_dataset.csv'")

from sklearn.model\_selection import train\_test\_split  
from sklearn.neighbors import KNeighborsClassifier  
from sklearn.metrics import accuracy\_score, classification\_report  
import pandas as pd  
  
  
file\_path = 'standardized\_sleep\_health\_lifestyle\_dataset.csv'  
data = pd.read\_csv(file\_path)  
  
  
data['Sleep\_Disorder\_Encoded'] = data['Sleep\_Disorder\_Encoded'].apply(lambda x: 1 if x > 0 else 0)  
  
  
X = data.drop(columns=['Sleep\_Disorder\_Encoded'])   
y = data['Sleep\_Disorder\_Encoded']   
  
  
X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42, stratify=y)  
  
  
k\_values = [3, 7, 11]  
results = {}  
  
  
for k in k\_values:  
 knn = KNeighborsClassifier(n\_neighbors=k)  
 knn.fit(X\_train, y\_train)  
 y\_pred = knn.predict(X\_test)  
 accuracy = accuracy\_score(y\_test, y\_pred)  
 print(f"K = {k}")  
 print(f"Accuracy: {accuracy}")  
 print(classification\_report(y\_test, y\_pred))

from sklearn.neural\_network import MLPClassifier  
from sklearn.metrics import accuracy\_score, classification\_report  
from sklearn.model\_selection import train\_test\_split  
import pandas as pd  
  
  
file\_path = 'standardized\_sleep\_health\_lifestyle\_dataset.csv'  
data = pd.read\_csv(file\_path)  
  
  
data['Sleep\_Disorder\_Encoded'] = data['Sleep\_Disorder\_Encoded'].apply(lambda x: 1 if x > 0 else 0)  
  
  
X = data.drop(columns=['Sleep\_Disorder\_Encoded']) # Özellikler  
y = data['Sleep\_Disorder\_Encoded'] # Hedef değişken  
  
  
X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42, stratify=y)  
  
  
hidden\_layer\_configurations = [  
 (32,), # 1 gizli katman, 32 nöron  
 (32, 32), # 2 gizli katman, her biri 32 nöron  
 (32, 32, 32) # 3 gizli katman, her biri 32 nöron  
]  
  
  
results = {}  
  
  
for hidden\_layers in hidden\_layer\_configurations:  
 mlp = MLPClassifier(hidden\_layer\_sizes=hidden\_layers, max\_iter=1000, random\_state=42)  
 mlp.fit(X\_train, y\_train)  
 y\_pred = mlp.predict(X\_test)  
 accuracy = accuracy\_score(y\_test, y\_pred)  
 print(f"Gizli Katman Yapılandırması: {hidden\_layers}")  
 print(f"Accuracy: {accuracy}")  
 print(classification\_report(y\_test, y\_pred))

from sklearn.model\_selection import train\_test\_split  
import pandas as pd  
  
  
file\_path = 'standardized\_sleep\_health\_lifestyle\_dataset.csv'  
data = pd.read\_csv(file\_path)  
  
  
data['Sleep\_Disorder\_Encoded'] = data['Sleep\_Disorder\_Encoded'].apply(lambda x: 1 if x > 0 else 0)  
  
  
X = data.drop(columns=['Sleep\_Disorder\_Encoded']) # Özellikler  
y = data['Sleep\_Disorder\_Encoded'] # Hedef değişken  
  
  
X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42, stratify=y)  
  
  
nb = GaussianNB()  
nb.fit(X\_train, y\_train)  
y\_pred = nb.predict(X\_test)  
accuracy = accuracy\_score(y\_test, y\_pred)  
  
  
print(f"Naive Bayes Model Accuracy: {accuracy}")  
print(classification\_report(y\_test, y\_pred))