Predicting Student Addiction with machine learning algorithm

Anand Kumar Dubey **3MSCDSA 23122005**

In [ ]:

```
import matplotlib.pyplot as plt   #importing necessory libraries for plottin
import seaborn as sns
```

In [ ]:

```
from sklearn.svm import SVC
from sklearn.preprocessing import StandardScaler

from sklearn.pipeline import make_pipeline
import numpy as np
```

In [ ]:

```
import pandas as pd
from sklearn.model_selection import train_test_split, GridSearchCV    #impo
from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, confusion_matrix, roc_auc
```

Loading and cleaning the Data

In [ ]:

```
#loading the data
test_df=pd.read_csv("student_addiction_dataset_test.csv")
train_df=pd.read_csv("student_addiction_dataset_test.csv")
```

In [ ]:

```
# #handeling missing values
# train_df.isnull().sum()
```

In [ ]:

```
# test_df.isnull().sum()
```

In [ ]:

```
train_df.columns
```

Out[ ]:

```
Index(['Experimentation', 'Academic_Performance_Decline', 'Social_Isolatio
n',
       'Financial_Issues', 'Physical_Mental_Health_Problems',
       'Legal_Consequences', 'Relationship_Strain', 'Risk_Taking_Behavior',
       'Withdrawal_Symptoms', 'Denial_and_Resistance_to_Treatment',
       'Addiction_Class'],
      dtype='object')
```

In [ ]:

```
#since most of the values in the dataset is non numerical we will
#convert them in numeric data first as non numeric is not suitable for machi
train_df['Experimentation']=train_df['Experimentation'].map({"Yes":1,"No":0
```

In [ ]:

```
train_df['Academic_Performance_Decline']=train_df['Academic_Performance_Dec
train_df['Social_Isolation']=train_df['Social_Isolation'].map({"Yes":1,"No"
train_df['Financial_Issues']=train_df['Financial_Issues'].map({"Yes":1,"No"
train_df['Physical_Mental_Health_Problems']=train_df['Physical_Mental_Healt
train_df['Legal_Consequences']=train_df['Legal_Consequences'].map({"Yes":1,'
train_df['Relationship_Strain']=train_df['Relationship_Strain'].map({"Yes":
train_df['Risk_Taking_Behavior']=train_df['Risk_Taking_Behavior'].map({"Yes'
train_df['Withdrawal_Symptoms']=train_df['Withdrawal_Symptoms'].map({"Yes":
train_df['Denial_and_Resistance_to_Treatment']=train_df['Denial_and_Resista
train_df['Addiction_Class']=train_df['Addiction_Class'].map({"Yes":1,"No":0
```

In [ ]:

```
#same thing we will do with test data
test_df.columns
```

Out[ ]:

```
Index(['Experimentation', 'Academic_Performance_Decline', 'Social_Isolatio
n',
       'Financial_Issues', 'Physical_Mental_Health_Problems',
       'Legal_Consequences', 'Relationship_Strain', 'Risk_Taking_Behavior',
       'Withdrawal_Symptoms', 'Denial_and_Resistance_to_Treatment',
       'Addiction_Class'],
      dtype='object')
```

In [ ]:

```
test_df['Experimentation']=test_df['Experimentation'].map({"Yes":1,"No":0})
test_df['Academic_Performance_Decline']=test_df['Academic_Performance_Declin
test_df['Social_Isolation']=test_df['Social_Isolation'].map({"Yes":1,"No":0
test_df['Financial_Issues']=test_df['Financial_Issues'].map({"Yes":1,"No":0
test_df['Physical_Mental_Health_Problems']=test_df['Physical_Mental_Health_
test_df['Legal_Consequences']=test_df['Legal_Consequences'].map({"Yes":1,"N
test_df['Relationship_Strain']=test_df['Relationship_Strain'].map({"Yes":1,"
test_df['Risk_Taking_Behavior']=test_df['Risk_Taking_Behavior'].map({"Yes":
test_df['Withdrawal_Symptoms']=test_df['Withdrawal_Symptoms'].map({"Yes":1,"
test_df['Denial_and_Resistance_to_Treatment']=test_df['Denial_and_Resistance
test_df['Addiction_Class']=test_df['Addiction_Class'].map({"Yes":1,"No":0})
```

In [ ]:

```
#Handeling Missing Values
train_df.isnull().sum()
```

Out[ ]:

```
Experimentation                       645
Academic_Performance_Decline          685
Social_Isolation                      677
Financial_Issues                      620
Physical_Mental_Health_Problems       665
Legal_Consequences                    686
Relationship_Strain                   632
Risk_Taking_Behavior                  613
Withdrawal_Symptoms                   653
Denial_and_Resistance_to_Treatment    654
Addiction_Class                         0
dtype: int64
```

In [ ]:

```
train_df = train_df.fillna(train_df.median())
```

In [ ]:

```
test_df=test_df.fillna(test_df.median())
```

In [ ]:

```
#since now the data is cleaned we can proceed for our machine learning mode
```

In [ ]:

```
# Define features and target variable
X_train = train_df.drop('Addiction_Class', axis=1)
y_train = train_df['Addiction_Class']
```

In [ ]:

```
# Initialize and train the model
rf_model = RandomForestClassifier()
rf_model.fit(X_train, y_train)
```

Out[ ]:

```
▾ RandomForestClassifier ⓘ ?

RandomForestClassifier()
```

In [ ]:

```
X_test = test_df.drop('Addiction_Class', axis=1)  # Remove the target varia
predictions = rf_model.predict(X_test)
```

```
# Optionally, print or save the predictions
test_df['Predicted_Addiction_Class'] = predictions
test_df.to_csv('test_predictions.csv', index=False)
```

In [ ]:

```
from sklearn.metrics import accuracy_score
true_labels = test_df['Addiction_Class']  # Assuming 'Addiction_Class' is th
predictions = test_df['Predicted_Addiction_Class']  # Assuming 'Predicted_Ac

# Calculate accuracy
accuracy = accuracy_score(true_labels, predictions)

print("Accuracy:", accuracy)
```

```
Accuracy: 0.7977871939736346
```

In [ ]: