Predicting A Student's Adaptability Level in Online Education

SHUBHAANGI VERMA (RA1911026030026)

TANISHK ARORA (RA1911026030019)

AYUSH RAWAT (RA1911026030045)

RISHIKA SHARMA (RA1911026030046)

SRM
INSTITUTE OF SCIENCE AND TECHNOLOGY
(Deemed to be University u/s 3 of UGC Act, 1956)
DELHI-NCR CAMPUS, GHAZIABAD (U.P)

GUIDE NAME :- Dr. ANAND PANDEY

CSE – A

MAY 2023

Motivation

- Increasing importance of online education since Covid-19.
- Factors affecting adaptability level in online education.
- Different students faced different difficulties.
- Predicting the adaptability level beforehand helps improve it to get an optimal level.

Introduction

The Covid-19 pandemic has forced a massive shift towards online learning, leading to an increased emphasis on the adaptability of students to this new mode of education. The importance of understanding the factors that impact a student's adaptability to online learning cannot be overstated. With this in mind, our team has set out to develop a classifier that can analyze various parameters related to a student's personal and academic background, as well as their learning habits, and determine how these factors relate to their level of adaptability in online learning.

LITERATURE REVIEW

- Impact of technology on virtual learning system.
- Multiple research papers published to study different factors.
- Researchers studied the improvement of online education model.
- Comparison between offline and online education system.
- Similar trends in on-campus vs off-campus performances.
- Difficulties faced in online education systems
- Rural areas vs Urban Areas : The former faced more challenges Lots of barriers : technological, communication, financial, etc.
- Impact of pandemic on the global education system.
- According to one research, better learning in online education systems.

EXISTING PLAN

- One of the major problems that students face with online education is the lack of face to-face interaction with their teachers and peers. In traditional in-person learning, students can interact with their teachers and classmates in real-time, ask questions, and get feedback instantly. However, in online learning, students are limited to virtual interactions, which can be less engaging and less effective than face-to-face interactions. Moreover, students who are not comfortable with technology may struggle to participate in online classes and communicate with their teachers and peers.
- Another issue that students face with online education is the lack of structure and routine. In traditional in-person learning, students have a set schedule and routine that helps them stay on track and manage their time effectively.
- However, with online learning, students often have to create their own schedule and manage their time independently, which can be challenging for some students. Students who struggle with self-discipline and time management may find it difficult to stay motivated and focused in an online learning environment.

PROPOSED SOLUTION

There are many different machine learning models that can be used to predict student adaptability in online learning. The choice of model depends on the specific problem, data, and available resources. We are going to use these machine learning models for predicting student adaptability in online learning:

- **Decision Tree**: This model is often used for classification problems and can be used to predict whether a student is likely to drop out of an online course or succeed. Decision trees can be easy to interpret and can handle categorical and continuous data.

- **Random Forest**: This model is an extension of decision trees and can be used for both classification and regression problems. Random forests can handle missing data, noisy data, and can capture nonlinear relationships.

- **Support Vector Machine (SVM):** This model is a powerful classifier that can handle high-dimensional data and non-linear relationships. SVM can be used to predict student success or engagement in online courses.

- **Artificial Neural Networks:** This model is highly flexible and can be used for both classification and regression problems. Neural networks can capture complex relationships between input features and output variables and can handle large datasets.

- **K Nearest Neighbours:** K-nearest neighbors (KNN) is a non-parametric algorithm used for both classification and regression tasks in machine learning. It is a simple and intuitive algorithm that works by finding the "nearest" data points to a given input based on some distance metric. Once the nearest neighbors are identified, the algorithm makes a prediction based on the class or value of those neighbors.

- **Gaussian Naive Bayes:** Gaussian Naive Bayes is a probabilistic algorithm used for classification tasks in machine learning. It is a simple and efficient algorithm that is based on the Bayes theorem and assumes that the features are conditionally independent given the class variable.

- **Logistic Regression:** This model is often used for binary classification problems and can be used to predict whether a student will drop out of an online course or not. Logistic regression is simple to implement, can handle continuous and categorical data, and provides interpretable coefficients.

METHODOLOGIES

Data preprocessing is a procedure of getting ready the raw information and making it suitable fora device gaining knowledge. It's the first and important step even as creating a Machine Learning Model. When creating a machine learning model, it is not continually a case that we encounter the smooth and formatted records. And while doing any operation with data, it's miles obligatory to easy it and put in a formatted way. So for this, we use data preprocessing project. Real-world data is typically complex and may contain various types of noise, missing values, and formatting inconsistencies that render it unsuitable for use in machine learning models. Therefore, it is essential to perform data preprocessing tasks to clean, transform, and prepare the data for analysis. By doing so, the accuracy and efficiency of machine learning models can be significantly improved, leading to more reliable results and better decision-making.

DATASET DESCRIPTION

- The dataset Contains categorical data.
- Some of the features contain binary values for eg. Yes/no, boy/girl.
- As our data is categorical, no outlier is observed.

Initial Dataset

Age	Education Level	Institution Type	IT Student	Location	Load-shedding	Financial Condition	Internet Type	Network Type	Class Duration	Self Lms	Devic
21-25	University	Non Government	No	Yes	Low	Mid	Wifi	4G	3-6	No	Ta
21-25	University	Non Government	No	Yes	High	Mid	Mobile Data	4G	1-3	Yes	Mobi
16-20	College	Government	No	Yes	Low	Mid	Wifi	4G	1-3	No	Mobi
11-15	School	Non Government	No	Yes	Low	Mid	Mobile Data	4G	1-3	No	Mobi
16-20	School	Non Government	No	Yes	Low	Poor	Mobile Data	3G	0	No	Mobi
...	...	...	...	...	...	...	...	...	...	...	
16-20	College	Non Government	No	Yes	Low	Mid	Wifi	4G	1-3	No	Mobi
16-20	College	Non Government	No	No	High	Mid	Wifi	4G	3-6	No	Mobi
11-15	School	Non Government	No	Yes	Low	Mid	Mobile Data	3G	1-3	No	Mobi
16-20	College	Non Government	No	No	Low	Mid	Wifi	4G	1-3	No	Mobi
11-15	School	Non Government	No	Yes	Low	Poor	Mobile Data	3G	1-3	No	Mobi

lumns

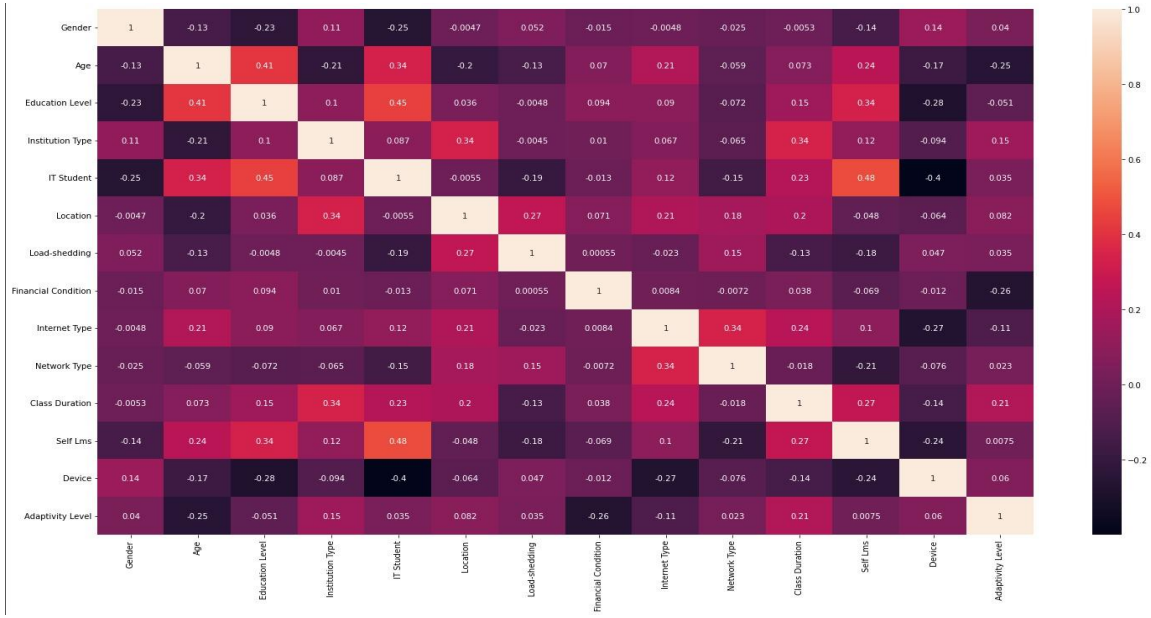
Data Exploration

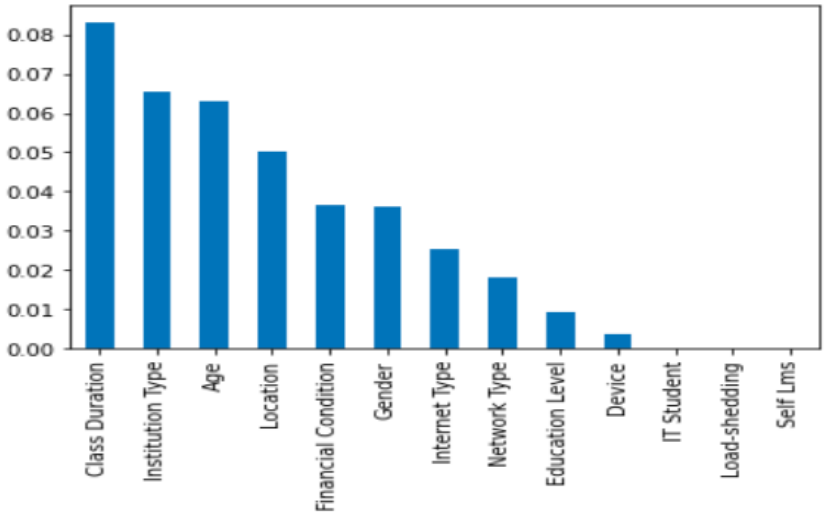
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1205 entries, 0 to 1204
Data columns (total 14 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   Gender               1205 non-null   object
 1   Age                  1205 non-null   object
 2   Education Level      1205 non-null   object
 3   Institution Type     1205 non-null   object
 4   IT Student           1205 non-null   object
 5   Location             1205 non-null   object
 6   Load-shedding        1205 non-null   object
 7   Financial Condition  1205 non-null   object
 8   Internet Type        1205 non-null   object
 9   Network Type         1205 non-null   object
 10  Class Duration       1205 non-null   object
 11  Self Lms             1205 non-null   object
 12  Device               1205 non-null   object
 13  Adaptivity Level     1205 non-null   object
dtypes: object(14)
memory usage: 131.9+ KB
```

DATASET PREPROCESSING

- No Null Values observed.
- String values in the dataset have been scaled to integers.
- Features like "Load Shedding" and "Self LMS" have very low correlation w.r.t target variable and low information gain hence are dropped while selecting features to improve model performance.





MODEL TRAINING
1) LOGISTIC REGRESSION

```
#Implementing Logistic Regression

model = lm.LogisticRegression(max_iter=100)
y_pred = model.fit(X_train, y_train).predict(X_test) #training the model

accuracy = accuracy_score(y_pred, y_test)
target_names = ['low adaptivity', 'moderate adaptivity', 'high adaptivity']

print("accuracy is: ", accuracy)
print(classification_report(y_test, y_pred, target_names=target_names))
```

```
accuracy is:  0.6473029045643154
                    precision    recall  f1-score   support

     low adaptivity       0.80      0.22      0.35        18
moderate adaptivity       0.66      0.56      0.60       104
    high adaptivity       0.64      0.79      0.70       119

           accuracy                           0.65       241
          macro avg       0.70      0.52      0.55       241
       weighted avg       0.66      0.65      0.63       241
```

2) Gaussian Naive Bayes

```
#Implementing Gaussian Naive Bayes
from sklearn.naive_bayes import GaussianNB
model = GaussianNB()
y_pred = model.fit(X_train, y_train).predict(X_test) #training the model

accuracy = accuracy_score(y_pred, y_test)
print("accuracy is: ", accuracy)
print(classification_report(y_test, y_pred, target_names=target_names))
```

```
accuracy is:  0.6307053941908713
                    precision    recall  f1-score   support

     low adaptivity       0.32      0.39      0.35        18
moderate adaptivity       0.65      0.62      0.64       104
    high adaptivity       0.67      0.67      0.67       119

          accuracy                           0.63       241
         macro avg       0.55      0.56      0.55       241
      weighted avg       0.64      0.63      0.63       241
```

3) Decision Trees

```
#Implementing Decision Tree
from sklearn import tree
clf = tree.DecisionTreeClassifier(criterion="entropy",max_depth=10)
clf = clf.fit(X_train, y_train)
y_pred=clf.predict(X_test)
accuracy = accuracy_score(y_pred, y_test)
print("accuracy is: ", accuracy)
print(classification_report(y_test, y_pred, target_names=target_names))
```

```
accuracy is:  0.8298755186721992
                    precision    recall  f1-score   support

     low adaptivity       0.72      0.72      0.72        18
moderate adaptivity       0.86      0.84      0.85       104
    high adaptivity       0.82      0.84      0.83       119

           accuracy                           0.83       241
          macro avg       0.80      0.80      0.80       241
       weighted avg       0.83      0.83      0.83       241
```

4) Random Forest

```
#Implementing Random Forest classifier
from sklearn.ensemble import RandomForestClassifier

model = RandomForestClassifier(criterion="entropy",n_estimators = 100, max_depth =
model.fit(X_train, y_train)
y_pred = model.predict(X_test)

accuracy = accuracy_score(y_pred, y_test)
print("accuracy is: ", accuracy)
print(classification_report(y_test, y_pred, target_names=target_names))
```

```
accuracy is:  0.8672199170124482
                    precision    recall  f1-score   support

     low adaptivity       1.00      0.61      0.76        18
moderate adaptivity       0.88      0.85      0.86       104
    high adaptivity       0.85      0.92      0.88       119

          accuracy                           0.87       241
         macro avg       0.91      0.79      0.83       241
      weighted avg       0.87      0.87      0.87       241
```

5) Support Vector Machine

```python
## Implementing SVM
from sklearn.svm import SVC
clf = SVC(kernel='poly',degree=7)
clf.fit(X_train, y_train)
y_pred = clf.predict(X_test)

accuracy = accuracy_score(y_pred, y_test)
print("accuracy is: ", accuracy)
print(classification_report(y_test, y_pred, target_names=target_names))
```

```
accuracy is:  0.8589211618257261
                    precision    recall  f1-score   support

     low adaptivity       0.62      0.72      0.67        18
moderate adaptivity       0.87      0.93      0.90       104
    high adaptivity       0.90      0.82      0.85       119

          accuracy                           0.86       241
         macro avg       0.79      0.82      0.81       241
      weighted avg       0.86      0.86      0.86       241
```

6) Multi -Layer Perceptron Training

```
#Implementing Artificial Neural Network
from sklearn.neural_network import MLPClassifier
clf = MLPClassifier(hidden_layer_sizes=(512,256,128,64), learning_rate='adaptive',random_state=1, max_iter=1000)
```

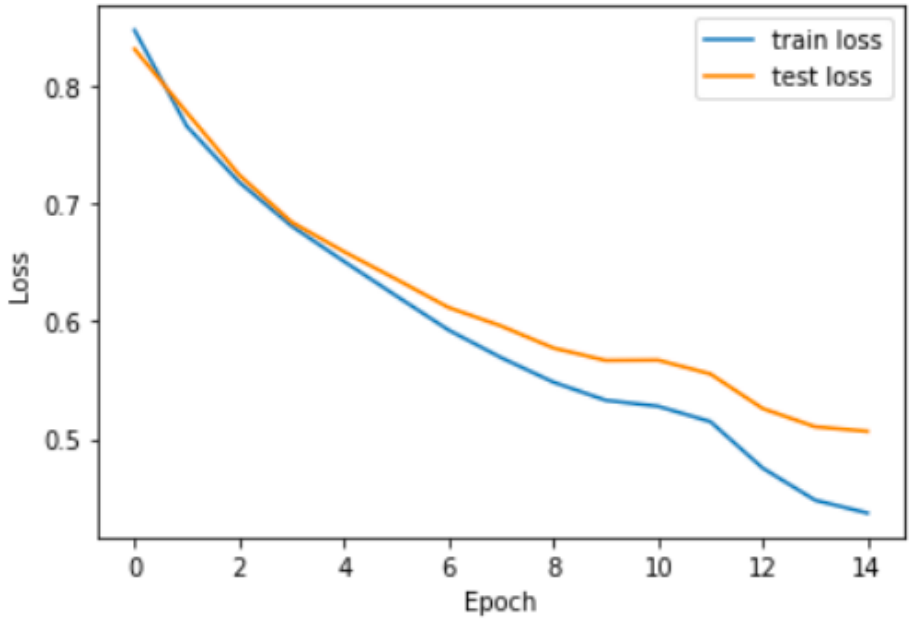
```
[ ] from sklearn.metrics import log_loss
```

```
[ ] classes=np.unique(y_train)
    train_loss=[]
    test_loss=[]
    for epoch in range(15):
        clf.partial_fit(X_train, y_train,classes)
        train_loss.append(log_loss(y_train,clf.predict_proba(X_train)))
        test_loss.append(log_loss(y_test,clf.predict_proba(X_test)))

    epoch=range(15)
    plt.plot(epoch, train_loss, label = 'train loss')
    plt.plot(epoch, test_loss, label = 'test loss')
    plt.xlabel("Epoch")
    plt.ylabel("Loss")
    plt.legend()
    plt.show()
```

MLP

Loss Plot For MLP



Accuracy For MLP

```
[ ]  print("Training accuracy: ", clf.score(X_train, y_train))
     print("Testing accuracy: ",clf.score(X_test, y_test))

     Training accuracy:  0.8215767634854771
     Testing accuracy:  0.8257261410788381
```

K-Nearest Neighbors

```
#Implementing K-Nearest Neighbours
from sklearn.neighbors import KNeighborsClassifier
clf = KNeighborsClassifier(n_neighbors=3)
clf.fit(X_train, y_train)
y_pred=clf.predict(X_test)
accuracy = accuracy_score(y_pred, y_test)
print("accuracy is: ", accuracy)
print(classification_report(y_test, y_pred, target_names=target_names))
```

```
accuracy is:  0.8174273858921162
                    precision    recall  f1-score   support

     low adaptivity       0.87      0.72      0.79        18
moderate adaptivity       0.85      0.77      0.81       104
    high adaptivity       0.79      0.87      0.83       119

           accuracy                           0.82       241
          macro avg       0.84      0.79      0.81       241
       weighted avg       0.82      0.82      0.82       241
```

RESULTS

By training the above models on the given dataset, we can say that Random Forests give the best performance amongst all the classifiers used for prediction as it has more accuracy, precision, recall, F1-score than other classifiers. Below is a table which can be used for comparison between different models.

Models	Class Name	Accuracy	Precision	Recall	F-1
Linear Regression	Low	69.71%	0.88	0.31	0.54
	Moderate		0.74	0.56	0.64
	High		0.66	0.87	0.75
Naive Bayes	Low	70.12%	0.58	0.61	0.59
	Moderate		0.72	0.61	0.66
	High		0.70	0.80	0.75
Random Forest	Low	86.72%	1.00	0.67	0.80
	Moderate		0.88	0.85	0.86
	High		0.84	0.92	0.88

CONCLUSION & FUTURE SCOPE

- Tried to forecast the student's adaptability level using ML models.
- Used classifiers such as Logistic Regression, Gaussian NB, Decision Trees, Random Forest, Support Vector Machine, ANN and KNN.
- Since the data is categorical so decision trees work well and hence random forest(ensemble learning of DTs) works the best for the given dataset.
- Work done would beneficial for the educational decision makers to improve the quality of education.

REFERENCES

1) Dataset: https://www.kaggle.com/datasets/mdmahmudulhasansuzan/studentsadaptability-level-in-online-education

2) https://ccrc.tc.columbia.edu/media/k2/attachments/adaptability-to-onlinelearning.pdf

3) https://eric.ed.gov/?id=EJ1175336

4) https://ijonse.net/index.php/ijonse/article/view/49

5) https://www.researchgate.net/publication/349601508_Barriers_to_Online_Learning_in_the_Time_of_COVID19_A_National_Survey_of_Medical_Students_in_the_Philippines

6) https://www.frontiersin.org/articles/10.3389/fpsyg.2021.637776/full

7) https://educationaltechnologyjournal.springeropen.com/articles/10.1186/s4123 9-021-00252-3