

Data Collection and Preprocessing Phase

Date	15 March 2024
Team ID	739666
Project Title	Student Adaptability Level of Online Education
Maximum Marks	6 Marks

Data Exploration and Preprocessing Template

Identifies data sources, assesses quality issues like missing values and duplicates, and implements resolution plans to ensure accurate and reliable analysis.

Section	Description
Data Overview	Provide an overview of the data used to assess student adaptability, including sources like LMS logs, assessment scores, and survey data. Include basic statistics, dimensions, and structure of the data.
Univariate Analysis	Explore individual variables that indicate student adaptability, such as average time spent on the platform, quiz scores, participation in discussions, etc. Analyze their mean, median, mode, and distribution.
Bivariate Analysis	Investigate relationships between two variables, such as the correlation between time spent on the platform and quiz scores, or between participation in discussions and overall performance. Use correlation matrices, scatter plots, and other visualizations.
Multivariate Analysis	Identify patterns and relationships involving multiple variables to understand the factors contributing to student adaptability. Use techniques like PCA (Principal Component Analysis) or cluster analysis to uncover hidden trends.
Outliers and Anomalies	Identify and treat outliers or anomalies in the data that could skew the results. For example, extremely high or low engagement scores that are not representative of typical student behavior.

Data Preprocessing Code Screenshots

Loading Data

```
[27]: ##### IMPORT LIBRARIES
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')

[28]: df = pd.read_csv('students_adaptability_level_online_education.csv') ##path of the data set
```

Handling Missing Data

```
[38]: ## Finding the missing values
df.isnull().sum()

[38]: Gender      0
Age             0
Education Level  0
Institution Type 0
IT Student      0
Location        0
Load-shedding   0
Financial Condition 0
Internet Type   0
Network Type    0
Class Duration  0
Self Lms        0
Device          0
Adaptivity Level 0
dtype: int64
```

Data Transformation

```
[35]: from sklearn.preprocessing import LabelEncoder
en = LabelEncoder()
for column in df.columns:
    df[column] = en.fit_transform(df[column])

[36]: df['Age'].tail(20)

[36]: 1185    1
1186    1
1187    2
1188    5
1189    3
1190    4
1191    1
1192    2
1193    3
1194    1
1195    3
1196    2
1197    3
1198    2
1199    3
1200    2
1201    2
1202    1
1203    2
1204    1
Name: Age, dtype: int32
```

Feature Engineering

```
[34]: ### Finding the unique values present in the dataset
dict={}
for column in df.columns:
    unique_values=df[column].unique()
    dict[column]=unique_values
dict

[34]: {'Gender': array(['Boy', 'Girl'], dtype=object),
'Age': array(['21-25', '16-20', '11-15', '26-30', '6-10', '1-5'], dtype=object),
'Education Level': array(['University', 'College', 'School'], dtype=object),
'Institution Type': array(['Non Government', 'Government'], dtype=object),
'IT Student': array(['No', 'Yes'], dtype=object),
'Location': array(['Yes', 'No'], dtype=object),
'Load-shedding': array(['Low', 'High'], dtype=object),
'Financial Condition': array(['Mid', 'Poor', 'Rich'], dtype=object),
'Internet Type': array(['Wifi', 'Mobile Data'], dtype=object),
'Network Type': array(['4G', '3G', '2G'], dtype=object),
'Class Duration': array(['3-6', '1-3', '0'], dtype=object),
'Self Lms': array(['No', 'Yes'], dtype=object),
'Device': array(['Tab', 'Mobile', 'Computer'], dtype=object),
'Adaptivity Level': array(['Moderate', 'Low', 'High'], dtype=object)}

[35]: from sklearn.preprocessing import LabelEncoder
en = LabelEncoder()
for column in df.columns:
    df[column] = en.fit_transform(df[column])
```

Save Processed Data

```
[4]: df.to_csv('students_adaptability_level_online_education.csv', index=False)

[5]: df.to_excel('students_adaptability_level_online_education.xlsx', index=False)
```