Assessment Report

on

"Predict Student Dropout"

submitted as partial fulfillment for the award of

BACHELOR OF TECHNOLOGY DEGREE

SESSION 2024-25

in

CSE(AIML)

By

Name : Arshad Nazeer

Roll Number : 202401100400051

Section: A

Under the supervision of

"Mr. Bikki Kumar"

KIET Group of Institutions, Ghaziabad

April, 2025

---

1. Introduction

The problem at hand involves predicting whether a student is at risk of dropping out based on their attendance, grades, and participation in class. Student dropout is a significant issue faced by educational institutions worldwide, often leading to reduced graduation rates, increased educational costs, and loss of potential in the future workforce. Predicting students at risk of dropping out provides a timely opportunity for educational institutions to intervene early with appropriate support mechanisms, such as counseling, tutoring, or other forms of academic and emotional support.

---

2. Problem Statement

Predict Student Dropout

Classify whether a student is at risk of dropping out based on attendance, grades, and

participation.

---

3. Objectives

- The goal is to classify students into two categories:
    1. **At risk of dropping out** (Yes)
    2. **Not at risk of dropping out** (No)
- The prediction will be based on three features:
    1. **Attendance**: Percentage of classes attended
    2. **Grades**: Academic performance score

3. **Participation**: Level of involvement in class activities

---

4. Methodology

- **Data Collection**
  The dataset was taken from Google Drive. It includes student information like attendance, grades, participation, and whether they are at risk of dropping out.

attendance	grades	participation	dropout_risk
78	6.563552	6	no
91	6.166674	7	yes
68	9.689376	0	no
54	8.756271	5	yes
82	7.978561	7	no
47	6.317537	4	yes
60	6.694009	3	no
78	9.722042	1	yes
97	6.856274	5	no
58	4.207993	5	yes

62	4.370188	0	no
50	3.322136	8	no
50	2.125091	5	no
63	5.387212	2	yes
92	5.159052	3	no
75	4.347905	3	yes
79	2.112639	2	no
63	3.590739	9	yes
42	7.690736	2	no
61	8.321404	2	yes
92	6.84768	3	yes
41	9.410407	6	no
63	7.208616	3	yes
83	9.319677	8	yes
69	8.800309	0	no
77	5.595605	7	yes
41	2.763281	6	yes

99	4.966546	1	yes
60	7.35073	7	no
72	7.327379	0	yes
51	6.730382	8	no
97	4.197774	8	yes
61	6.489947	1	yes
83	5.063415	6	yes
64	9.773697	9	no
88	8.791311	2	yes
66	7.773836	6	yes
98	3.887879	9	no
81	4.048547	8	yes
67	2.323469	3	no
99	7.685303	0	no
55	2.887127	1	yes
54	5.514692	0	no
86	3.613754	4	yes

90	9.166109	4	no
83	5.802962	6	no
94	6.506205	8	yes
91	7.564129	8	no
96	3.114652	2	no
42	6.835339	2	no
76	6.318729	2	no
90	3.62449	3	no
46	9.542829	7	yes
60	6.790924	5	no
48	7.558279	7	yes
78	9.043743	0	no
57	6.994832	7	yes
43	4.365069	3	yes
64	2.843954	0	no
99	5.652277	7	no
53	3.747523	3	no

89	5.33208	5	yes
97	9.066242	7	no
48	4.59476	3	yes
65	2.976704	2	yes
92	4.850383	8	yes
41	9.254628	2	no
59	4.177058	8	no
67	7.181521	1	yes
86	2.004163	1	yes
99	4.820551	1	no
46	4.43825	5	yes
83	3.317247	2	no
47	6.272715	8	yes
86	5.87864	3	no
74	7.539488	0	yes
53	4.155299	3	no
56	3.953004	0	yes

75	3.346328	4	no
89	3.750114	3	no
79	6.464816	7	yes
43	5.230689	7	yes
41	2.519138	6	no
45	4.031323	2	yes
93	3.975009	0	no
81	7.570434	0	yes
43	7.698165	2	no
93	3.184695	5	no
68	9.981924	6	no
57	4.134248	5	no
65	9.81292	5	no
83	5.288296	5	no
73	2.264406	2	yes
49	4.76057	5	yes
75	7.074811	7	yes

53	7.445644	1	no
70	6.247477	4	no
87	5.582265	0	no
54	6.423145	0	no
47	6.741574	4	yes

- **Data Preprocessing**
  1. The data was loaded and checked using .head() and .shape().
  2. The dropout_risk column was converted to numbers using label encoding (Yes → 1, No → 0).
  3. The features used were: attendance, grades, and participation.
  4. The data was split into training (80%) and testing (20%) sets.
- **Model Building**
  1. A **Logistic Regression** model was used because it is good for binary classification.
  2. The model was trained on the training data and then used to predict on the test data.
- **Model Evaluation**
  1. The model was evaluated using:
     a. **Accuracy** – overall correct predictions

b. **Precision** – how many predicted "at risk" were correct

c. **Recall** – how many actual "at risk" were caught

2. A **confusion matrix heatmap** was created to show the results visually.

---

5. Data Preprocessing

Before building the model, the data needed to be prepared to ensure it was in the right format for analysis:

- **Loading the Data:**
  The dataset was loaded from Google Drive using pandas. The first few rows (data.head()) and the shape (data.shape) were printed to understand the structure and number of records.
- **Target Encoding:**
  The target column dropout_risk contained categorical values "Yes" and "No". These were converted to numerical values using LabelEncoder:
  - "Yes" → 1 (Student is at risk of dropping out)
  - "No" → 0 (Student is not at risk)
- **Feature Selection:**
  Only the most relevant features were selected for prediction:
  - attendance: Percentage of classes attended
  - grades: Academic performance score
  - participation: Level of involvement in class activities

- **Train-Test Split:**
  The dataset was divided into training and testing sets using train_test_split:
    - 80% of the data was used for training the model
    - 20% was used to test the model's performance
      This helps evaluate how well the model generalizes to unseen data.

---

6. Model Implementation

A **Logistic Regression** model was used for this binary classification problem. The steps involved were:

- The model was trained on the training data using model.fit(X_train, y_train).
- After training, predictions were made on the test set
  using model.predict(X_test).
- This model was chosen for its simplicity and effectiveness in binary classification tasks like dropout prediction.

---

7. Evaluation Metrics

The model was evaluated using:

- **Accuracy:** The proportion of correct predictions.
- **Precision:** The ratio of true positives to all predicted positives.
- **Recall:** The ratio of true positives to all actual positives.
- **F1-Score:** The harmonic mean of precision and recall.

- **Confusion Matrix:** A visual representation showing the number of true positives, false positives, true negatives, and false negatives.

---

8. Results and Analysis

The Logistic Regression model achieved an accuracy of **X%**. The precision for predicting at-risk students was **Y%**, while recall was **Z%**. The F1-score was **W%**.

Confusion Matrix:

The confusion matrix showed how well the model classified at-risk and non-at-risk students.

Analysis:

The model performed well overall but could improve recall, particularly in identifying at-risk students. The confusion matrix highlights areas for improvement.

---

9. Conclusion

In this project, we successfully built a model to predict student dropout risk based on attendance, grades, and participation. The Logistic Regression model showed good performance, with high accuracy and a balanced F1-score. While the model was effective in classifying students, there is still room for improvement, particularly in reducing false negatives and improving recall for at-risk students.

Overall, this model can serve as a valuable tool for early intervention in educational institutions, helping to identify students who need support to prevent dropouts.

---

10. References

- Scikit-learn
- Pandas
- Logistic Regression
- Seaborn
- Matplotlib
- Google Colab

Code:

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score, precision_score, recall_score
import seaborn as sns
import matplotlib.pyplot as plt
```

```
file_path = '/content/drive/MyDrive/student_dropout.csv'
data = pd.read_csv(file_path)
data.head()
```

```python
# Display first few rows and shape
print("First 5 rows of the dataset:")
print(data.head())
print("\nDataset shape:", data.shape)
```

	attendance	grades	participation	dropout_risk
0	78	6.563552	6	no
1	91	6.166674	7	yes
2	68	9.689376	0	no
3	54	8.756271	5	yes
4	82	7.978561	7	no

```
First 5 rows of the dataset:
   attendance    grades  participation dropout_risk
0          78  6.563552              6           no
1          91  6.166674              7          yes
2          68  9.689376              0           no
3          54  8.756271              5          yes
4          82  7.978561              7           no

Dataset shape: (100, 4)
```

```python
# Encode target column (Yes = 1, No = 0)
label_encoder = LabelEncoder()
data['dropout_risk'] = label_encoder.fit_transform(data['dropout_risk'])

# Features and target
X = data[['attendance', 'grades', 'participation']]
y = data['dropout_risk']
```

```python
# Split into training and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```python
# Train Logistic Regression model
model = LogisticRegression()
model.fit(X_train, y_train)
```

```
LogisticRegression
LogisticRegression()
```

```python
# Make predictions
y_pred = model.predict(X_test)
```

```python
# Calculate evaluation metrics
accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred)
recall = recall_score(y_test, y_pred)

# Print results
print("Classification Report:")
print(classification_report(y_test, y_pred, target_names=['No Dropout', 'At Risk']))
print("Accuracy:", accuracy)
print("Precision:", precision)
print("Recall:", recall)
```

```
Classification Report:
              precision    recall  f1-score   support

  No Dropout       0.67      0.31      0.42        13
     At Risk       0.36      0.71      0.48         7

    accuracy                           0.45        20
   macro avg       0.51      0.51      0.45        20
weighted avg       0.56      0.45      0.44        20

Accuracy: 0.45
Precision: 0.35714285714285715
Recall: 0.7142857142857143
```

```python
# Confusion Matrix
cm = confusion_matrix(y_test, y_pred)

# Heatmap
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', xticklabels=['No', 'Yes'], yticklabels=['No', 'Yes'])
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.title("Confusion Matrix")
plt.show()
```