****

**Assessment Report**

on

**“Student Dropout Prediction”**

submitted as partial fulfillment for the award of

**BACHELOR OF TECHNOLOGY**

**DEGREE**

SESSION 2024-25

in

**CSE(AI)**

By

Name : Ayush Gupta

Roll Number : 202401100300084

Section: B

**Under the supervision of**

“SHIVANSH PRASAD”

**KIET Group of Institutions, Ghaziabad**

📄 Student Dropout Prediction – Project Report

# 📌 Objective

Build a machine learning model to predict student dropout risk based on features like attendance, grades, and class participation.

# 📁 Dataset Overview

The dataset contains the following columns:

| Feature | Description |

|-----------------|----------------------------------------------|

| attendance | Attendance percentage (0–100) |

| grades | Academic average score (0–10) |

| participation | Class participation score (0–10) |

| dropout\_risk | Target: "yes" (1) or "no" (0) |

# 🧼 Data Preprocessing Steps

1. Uploaded dataset in Google Colab.

2. Checked for missing values (none found).

3. Converted 'dropout\_risk' to numeric ('yes' = 1, 'no' = 0).

# 📊 Features & Target

- Features: attendance, grades, participation

- Target: dropout\_risk (0 or 1)

# 🤖 Model Used

- Algorithm: Decision Tree Classifier

- Train/Test Split: 80/20

- Random State: 42 (reproducibility)

# 📈 Model Evaluation

Model Accuracy: 0.60

Includes classification report and a confusion matrix.

# 📋 Code Implementation (Colab)

# Install required libraries (Colab usually has these)

!pip install pandas scikit-learn matplotlib seaborn --quiet

# Step 1: Upload the dataset

from google.colab import files

uploaded = files.upload()

import pandas as pd

import io

# Load the uploaded file

filename = list(uploaded.keys())[0]

df = pd.read\_csv(io.BytesIO(uploaded[filename]))

# Step 2: Preview the data

print("First 5 rows:")

print(df.head())

# Step 3: Check for missing values

print("\nMissing values:")

print(df.isnull().sum())

# Step 4: Preprocess columns

features = ['attendance', 'grades', 'participation']

target = 'dropout\_risk'

df[target] = df[target].map({'yes': 1, 'no': 0})

# Step 5: Train/test split

from sklearn.model\_selection import train\_test\_split

X = df[features]

y = df[target]

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Step 6: Train Decision Tree

from sklearn.tree import DecisionTreeClassifier

model = DecisionTreeClassifier(random\_state=42)

model.fit(X\_train, y\_train)

# Step 7: Predict and evaluate

from sklearn.metrics import classification\_report, confusion\_matrix, accuracy\_score

y\_pred = model.predict(X\_test)

print("\nClassification Report:")

print(classification\_report(y\_test, y\_pred))

acc = accuracy\_score(y\_test, y\_pred)

print(f"Model Accuracy: {acc:.2f}")

# Confusion Matrix

import matplotlib.pyplot as plt

import seaborn as sns

cm = confusion\_matrix(y\_test, y\_pred)

sns.heatmap(cm, annot=True, fmt='d', cmap='Blues')

plt.title("Confusion Matrix")

plt.xlabel("Predicted")

plt.ylabel("Actual")

plt.show()

# Step 8: Manual Prediction

print("\n--- Predict Dropout Risk for a New Student ---")

attendance = float(input("Enter attendance (0-100): "))

grades = float(input("Enter grades (0-10): "))

participation = float(input("Enter participation score (0-10): "))

new\_student = pd.DataFrame([[attendance, grades, participation]], columns=features)

prediction = model.predict(new\_student)[0]

result = "❌ At Risk of Dropping Out" if prediction == 1 else "✅ Not at Risk"

print("\nPrediction for this student:", result)

# 📉 Limitations

- Accuracy is moderate (60%).

- Limited to 3 features only.

- No advanced tuning or ensemble methods applied.

# 🧠 Suggestions for Improvement

- Add more features (e.g., behavior, parental support).

- Use advanced models like Random Forest, XGBoost.

- Apply hyperparameter tuning and cross-validation.

# ✅ Conclusion

A complete ML pipeline was built using Colab to predict student dropout risk. While basic, it demonstrates data handling, model training, and prediction effectively.