



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 : Deepanshu Pal

Roll Number : 202401100300095

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.