Predict Student Dropout

B.Tech CSE AIML 2024 -25

Under the Supervision of : **Mr. Bikki Gupta Sir**

Name	Anurag Singh
Roll No	202401100400041
Date	22/04/2025

Introduction

The **Student Dropout Prediction** problem aims to identify students who are at risk of dropping out of school based on various factors such as **attendance**, **grades**, and **class participation**. Early identification of at-risk students allows for timely intervention, such as counseling or academic support, to help reduce dropout rates. This prediction can be achieved through machine learning classification algorithms.

In this report, we will apply a **Random Forest Classifier** to predict whether a student will drop out or not based on the features mentioned above. We will evaluate the performance of the model using common classification metrics like **accuracy**, **precision**, and **recall**. Additionally, we will visualize the model's performance through a **confusion matrix heatmap**.

Scope and Methodology

Scope:

- We are working with a **small dataset** consisting of only **10 students**.
- The dataset includes **three features**:
  1. **Attendance**: Percentage of classes attended by the student.
  2. **Grades**: Student's performance in academic assessments (out of 10).
  3. **Participation**: Level of engagement in class activities (percentage).
- The **target variable** is whether the student will drop out (binary classification: 0 = No, 1 = Yes).

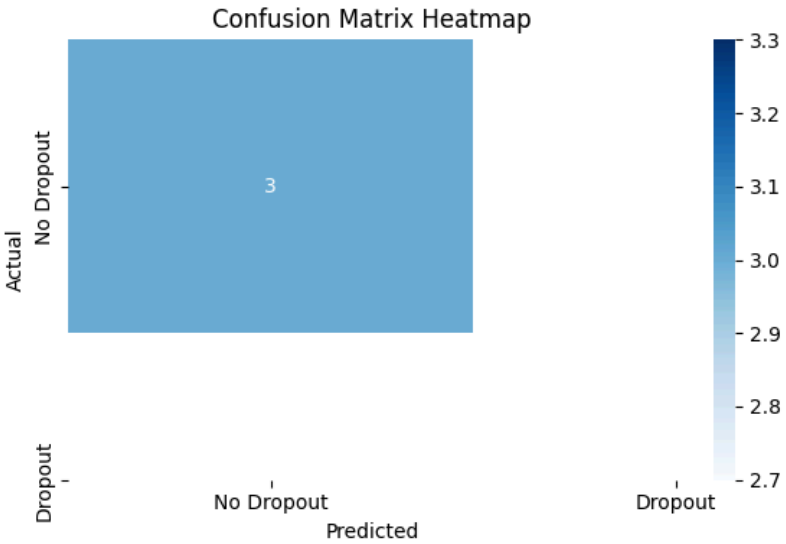
Methodology:

1. **Data Preprocessing**:
   - The dataset is first loaded and split into **features** (attendance, grades, participation) and the **target variable** (dropout status).
2. **Model Selection**:
   - We use the **Random Forest Classifier**, a popular and robust machine learning algorithm, to train the model. It works by constructing a multitude of decision trees and outputs the majority vote of these trees as the final prediction.
3. **Training and Evaluation**:
   - We split the data into **training** and **test** sets using a **70-30 split**.
   - The model is trained on the **training set** and predictions are made on the **test set**.
   - We calculate the **accuracy**, **precision**, and **recall** of the model to evaluate its performance.
4. **Confusion Matrix**:
   - A **confusion matrix** is generated to visually represent the model's performance. This is followed by a **heatmap** for better readability.

Output/Code

```python
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt

from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import confusion_matrix, accuracy_score, precision_score, recall_score

# Load the dataset from CSV file
data = pd.read_csv('student_dropout_sample.csv')  # Change path if needed


# Input features
X = data[['attendance', 'grades', 'participation']]

# Target variable
y = data['dropout']


# Split into training and testing sets (70% train, 30% test)
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.3, random_state=42
)
# Split into training and testing sets (70% train, 30% test)
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.3, random_state=42
)
```

```python
# Initialize and train the classifier
model = RandomForestClassifier(random_state=42)
model.fit(X_train, y_train)


# Predict on test data
y_pred = model.predict(X_test)


# Calculate evaluation metrics
accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred, zero_division=0)
recall = recall_score(y_test, y_pred, zero_division=0)

# Print metrics
print("Accuracy:", accuracy)
print("Precision:", precision)
print("Recall:", recall)


# Generate confusion matrix
cm = confusion_matrix(y_test, y_pred)

# Plot heatmap
plt.figure(figsize=(6, 4))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues',
            xticklabels=['No Dropout', 'Dropout'],
            yticklabels=['No Dropout', 'Dropout'])
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.title('Confusion Matrix Heatmap')
plt.tight_layout()
plt.show()
```




```
Accuracy: 1.0
Precision: 0.0
Recall: 0.0
```

References/ Credits

1. **Scikit-learn Library**
   Website: https://scikit-learn.org
   Used for model training, prediction, and evaluation (Random Forest, metrics, train-test split).
2. **Pandas & NumPy**
   - https://pandas.pydata.org
   - https://numpy.org
     Utilized for data handling and generation of synthetic datasets.
3. **Seaborn & Matplotlib**
   - https://seaborn.pydata.org
   - https://matplotlib.org
     Used to create the heatmap of the confusion matrix and data visualizations.
4. **Dataset**
   - Custom synthetic dataset generated for educational and demonstration purposes.
5. **Project Guidance**
   - Concept inspired by real-world educational data analytics projects focusing on student retention and dropout prediction.