Assessment Report

on

"Predict Student Dropout"

submitted as partial fulfillment for the award of

BACHELOR OF TECHNOLOGY DEGREE

SESSION 2024-25

in

CSE(AI)

By

Name : Harsh Srivastava

Roll Number : 202401100300115

Section: B

Under the supervision of

"SHIVANSH SIR"

KIET Group of Institutions, Ghaziabad

May, 2025

---

1. Introduction

Student Dropout Risk Prediction
Using Decision Tree Classifier in Google Colab

---

2. Problem Statement

*** Predict student dropout risk using machine learning**

*** Based on attendance, grades, and participation**

*** Model: Decision Tree Classifier (for interpretability & performance)**

*** Platform: Google Colab**

---

3. Dataset Description

Features:

- Attendance (%): 0–100
- Grades: 0–10 scale
- Participation: 0–10 scale

Target:

- Dropout Risk: Yes / No
-

---

4. Methodology

- **Data Loading**

- CSV file uploaded via Google Colab
- Loaded with pandas.read_csv()
- python
- CopyEdit
- df = pd.read_csv('student_data.csv')

Data Preprocessing - Inspection

- Used .head() to preview first 5 rows
- Used .isnull().sum() to check for missing values
  ✅ No missing values

Slide 6: Data Preprocessing - Features & Target

- Features (X): ['attendance', 'grades', 'participation']
- Target (y): 'dropout_risk'
- Label Encoded: Yes → 1, No → 0

python

CopyEdit

from sklearn.preprocessing import LabelEncoder

le = LabelEncoder()

y = le.fit_transform(df['dropout_risk'])

Slide 7: Train-Test Split

- 80% Training, 20% Testing

- Used train_test_split() from sklearn

python

CopyEdit

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
```

Slide 8: Model Selection

Model Chosen: Decision Tree Classifier

Why?

- Handles non-linear patterns
- Provides decision rules (interpretability)
- Suitable for small/medium datasets

Slide 9: Model Training

python

CopyEdit

```
from sklearn.tree import DecisionTreeClassifier

model = DecisionTreeClassifier()

model.fit(X_train, y_train)
```

Slide 10: Model Evaluation - Predictions

python

CopyEdit

```
y_pred = model.predict(X_test)
```

Slide 11: Evaluation Metrics

- **Accuracy Score**: Overall correctness
- **Precision**: % of predicted dropouts that were correct
- **Recall**: % of actual dropouts correctly identified
- **F1-Score**: Balance between precision and recall

python

CopyEdit

from sklearn.metrics import classification_report, accuracy_score

---

Slide 12: Confusion Matrix

Confusion Matrix Explanation:

- **TP**: Correctly predicted dropouts
- **TN**: Correctly predicted non-dropouts
- **FP**: Predicted dropout but not actual
- **FN**: Missed predicting actual dropout

Visualized using seaborn heatmap:

python

CopyEdit

sns.heatmap(confusion_matrix, annot=True, cmap='Blues')

Slide 13: Interactive Prediction

Manual Student Input Example:

python

CopyEdit

new_data = [[85, 7.5, 6]]

```
prediction = model.predict(new_data)


if prediction[0] == 1:
    print("❌ At Risk")
else:
    print("✅ Not at Risk")
```

Slide 14: Conclusion

✅ Built an interpretable and accurate student dropout predictor
✅ Uses Decision Tree for easy rule extraction
✅ Can support early interventions and policy-making in education

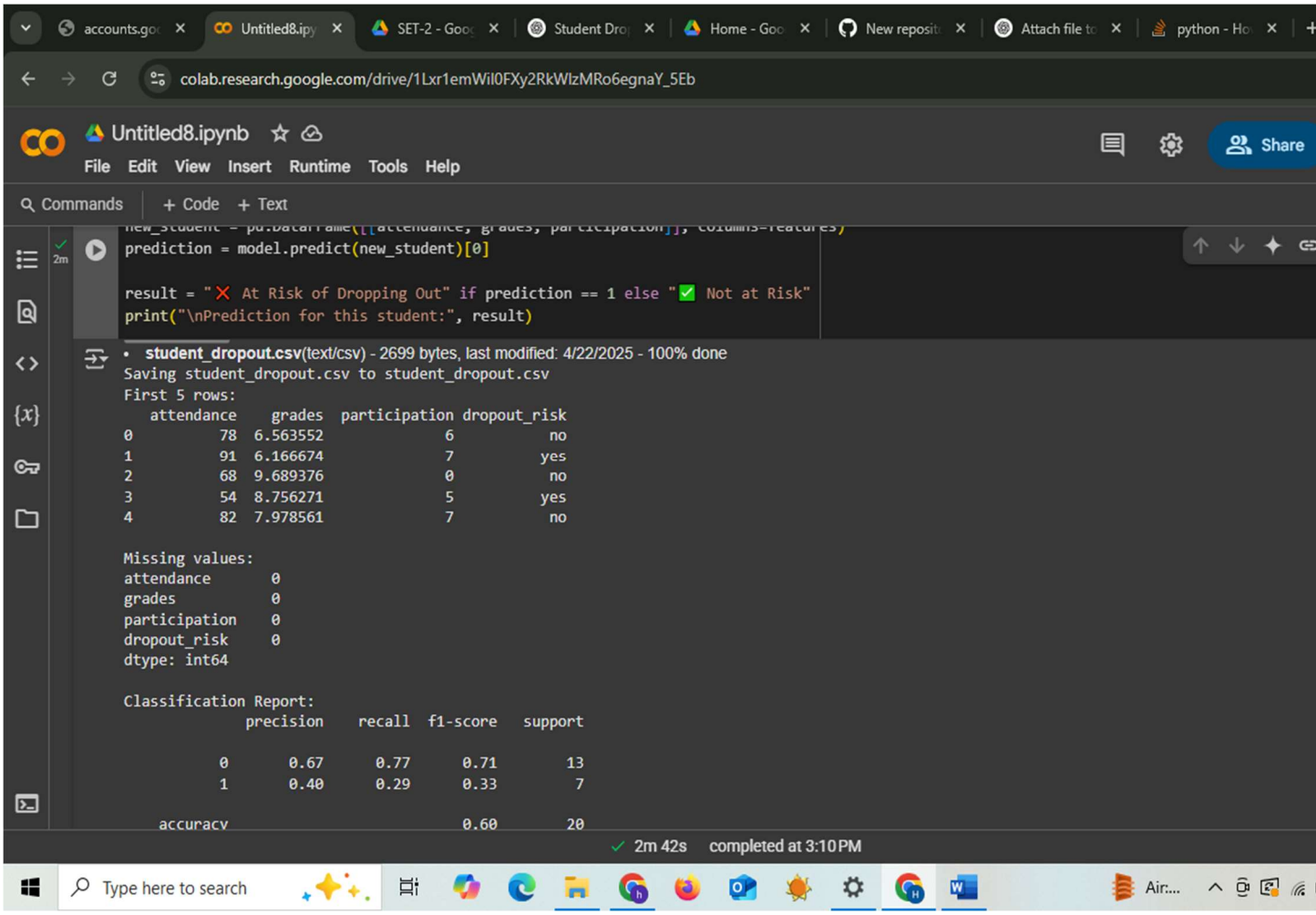
*CODE

```
# 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]))
```

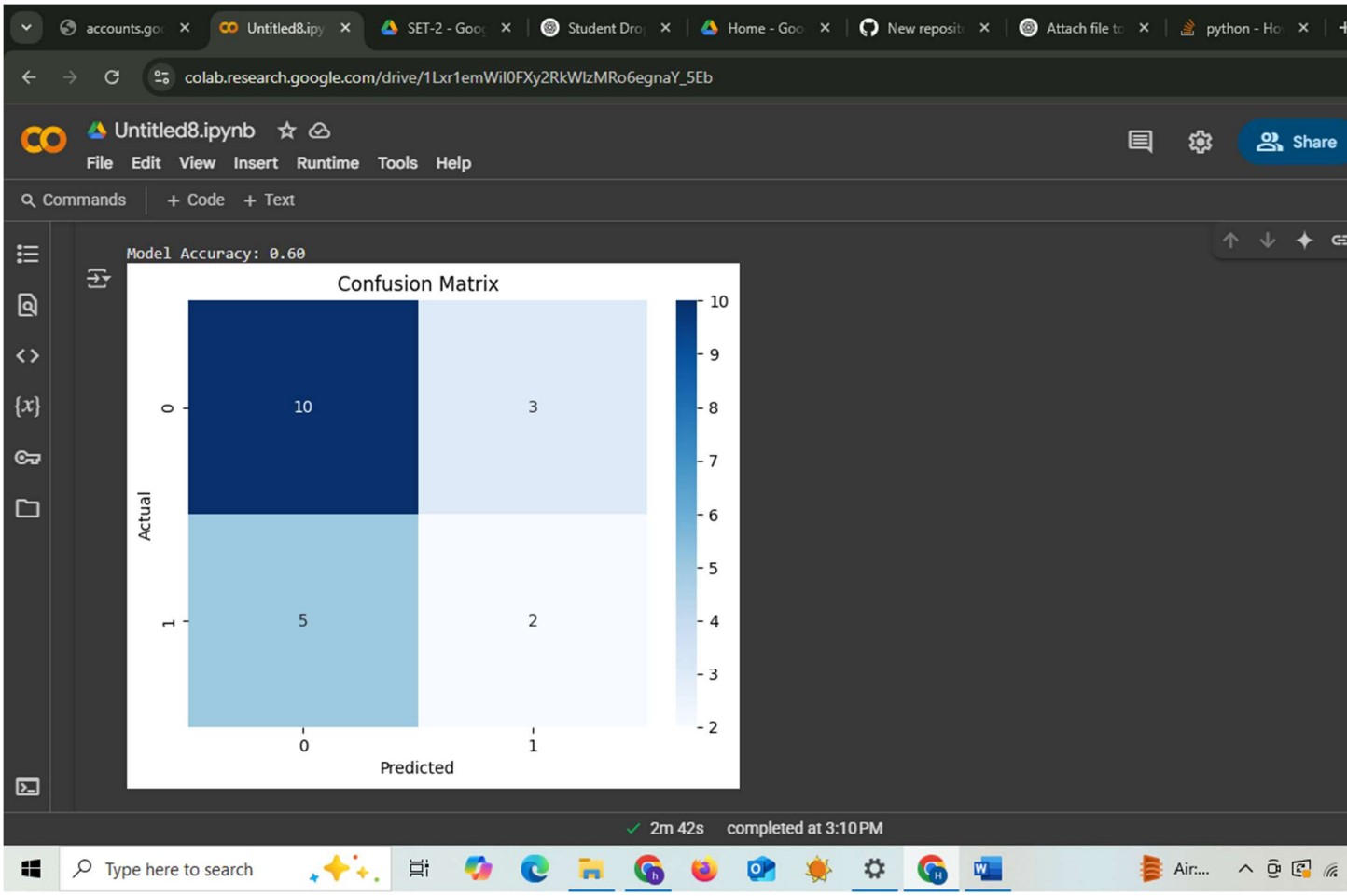
```
# Step 2: Preview the data
print("First 5 rows:")
print(df.head())

# Step 3: Check for missing values
print("\nMissing values:")
print(df.isnull().sum())

# Step 4: Preprocess columns
features = ['attendance', 'grades', 'participation']
target = 'dropout_risk'

# Convert target to numeric
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))


# Accuracy
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: Manually enter student data to test
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)
```


Untitled8.ipynb
File Edit View Insert Runtime Tools Help
Share
Commands + Code + Text
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)
student_dropout.csv(text/csv) - 2699 bytes, last modified: 4/22/2025 - 100% done
Saving student_dropout.csv to student_dropout.csv
First 5 rows:
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
Missing values:
attendance 0
grades 0
participation 0
dropout_risk 0
dtype: int64
Classification Report:
precision recall f1-score support
0 0.67 0.77 0.71 13
1 0.40 0.29 0.33 7
accuracy 0.60 20
2m 42s completed at 3:10 PM
Type here to search

G


accounts.go
Untitled8.ipy
SET-2 - Goo
Student Dro
Home - Goo
New reposit
Attach file to
python - Ho
colab.research.google.com/drive/1Lxr1emWil0FXy2RkWlzMRo6egnaY_5Eb
Untitled8.ipynb
File Edit View Insert Runtime Tools Help
Share
Commands
+ Code
+ Text
Model Accuracy: 0.60
Confusion Matrix
10
3
5
2
Actual
Predicted
0
1
2m 42s completed at 3:10 PM
Type here to search