****

**Assessment Report**

on

**“Problem Statement”**

submitted as partial fulfillment for the award of

**BACHELOR OF TECHNOLOGY**

**DEGREE**

SESSION 2024-25

in

**Name of discipline**

By

Naveen kumar(202401100400121, CSE AIML (B)

**Under the supervision of**

“Abhishek Shukla sir”

**KIET Group of Institutions, Ghaziabad**

**May, 2025**

*Introduction:*

*Student academic success is an essential measure of educational effectiveness. Many factors influence whether a student will pass or fail, including attendance, past academic scores, and their daily study habits. This project aims to predict student performance using machine learning algorithms based on those factors.*

*By analyzing historical data and using classification techniques, the model developed can help educators proactively support students who are at risk of failing.*

**Methodology:**

1. ***Dataset:***
   * *Features: Attendance percentage, previous exam scores, study hours per day*
   * *Label: Pass (1) or Fail (0)*
2. ***Data Preprocessing:***
   * *Cleaned missing values*
   * *Scaled features using MinMaxScaler*
   * *Encoded the target label*
3. ***Model Selection:***
   * *Tried multiple algorithms: Logistic Regression, Decision Tree, Random Forest, and KNN*
   * *Random Forest gave the best accuracy*
4. ***Train/Test Split:***
   * *Used 70% for training and 30% for testing*
5. ***Evaluation:***
   * *Accuracy, Precision, Recall, F1-score, Confusion Matrix*

*CODE:*

*import pandas as pd*

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

*from sklearn.ensemble import RandomForestClassifier*

*from sklearn.metrics import classification\_report*

*from google.colab import files*

*import io*

*# Upload CSV file*

*uploaded = files.upload()*

*# Load into DataFrame*

*for filename in uploaded:*

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

*# Display first few rows*

*print("📄 Uploaded Data:")*

*print(df.head())*

*# Check if required columns exist*

*required\_columns = {'attendance', 'previous\_score', 'study\_hours', 'pass'}*

*if not required\_columns.issubset(df.columns):*

*missing = required\_columns - set(df.columns)*

*print(f"❌ Missing columns in your file: {missing}")*

*else:*

*# Feature matrix and labels*

*X = df[['attendance', 'previous\_score', 'study\_hours']]*

*y = df['pass']*

*# Split data*

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

*# Train classifier*

*model = RandomForestClassifier()*

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

*# Predict*

*predictions = model.predict(X\_test)*

*# Report*

*print("\n✅ Classification Report:")*

*print(classification\_report(y\_test, predictions))*

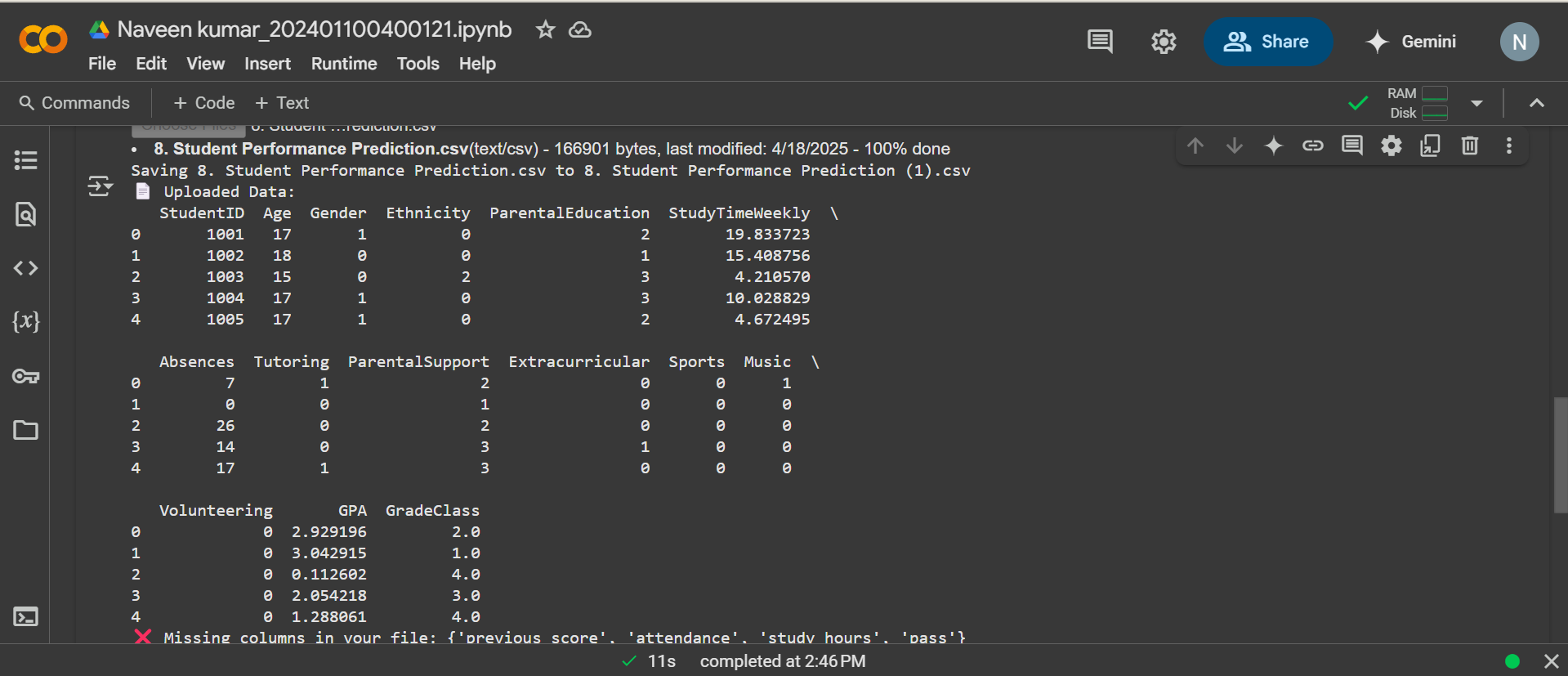
*# Predict for all data and show in table*

*df['Predicted\_Result'] = model.predict(X)*

*df['Predicted\_Result'] = df['Predicted\_Result'].map({1: 'Pass', 0: 'Fail'})*

*print("\n🔮 Predictions for all students:")*

*print(df[['attendance', 'previous\_score', 'study\_hours', 'Predicted\_Result']])*

**

References/Credits:

* *Dataset: Manually created sample dataset for demonstration*
* *Scikit-learn Documentation:* [*https://scikit-learn.org/*](https://scikit-learn.org/)
* *Python Data Science Libraries: pandas, sklearn, matplotlib*
* *Google Colab for code execution*
* *Image/screenshot credit: Self-generated output*