Assesment Report

on

“Problem Statement”

submitted as partial fulfillment for the award of

BACHELOR OF TECHNOLOGY DEGREE

SESSION 2024-25

in

CSE AIML

By

Ity Gaur (202401100400101)

Under the supervision of

Sundeep Raj

KIET Group of Institutions, Ghaziabad

Affiliated to

Dr. A.P.J. Abdul Kalam Technical University, Lucknow

(Formerly UPTU)

May, 2025

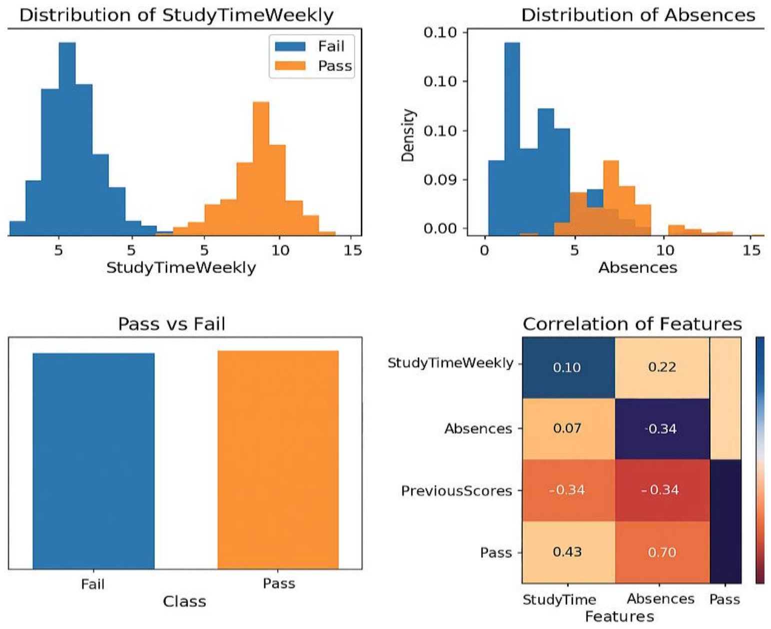
INTRODUCTION TO THE PROBLEM:

Problem Statement: Predicting Student Performance

In the context of education, predicting student performance is crucial for identifying students who may need additional support or intervention. This project focuses on predicting whether a student will pass or fail based on the following factors:

- **Attendance**: The number of classes a student has attended, which can reflect their engagement and understanding of the course material.
- **Previous Scores**: The student's historical academic performance, which is a strong predictor of future success.
- **Study Habits**: The time a student spends studying weekly, which can influence their preparation and academic success.

The goal of this project is to develop a classification model that predicts whether a student will pass or fail, helping educators identify students who might require additional support or intervention.



Methodologe:

The methodology used to solve this problem involves several steps, from data preprocessing to model evaluation. The steps are as follows:

1. Data Preprocessing

- **Data Cleaning**: The dataset was loaded from a CSV file (8. Student Performance Prediction.csv) and any missing or erroneous values were handled appropriately.
- **Feature Engineering**: A new binary column called Pass was created based on the GPA of the students. If the GPA was greater than or equal to 2.0, the student was labeled as "Pass" (1); otherwise, the student was labeled as "Fail" (0).
- **Feature Selection**: The relevant features selected for the prediction were **Study Time Weekly**, **Absences**, **Tutoring**, and **Parental Support**.
- **Data Transformation**: The features were scaled using **StandardScaler** to ensure that all features have similar ranges. This helps prevent certain features from dominating the model's learning process.

2. Model Selection

- **Logistic Regression** was chosen as the model for this binary classification problem. Logistic Regression is a simple and interpretable model commonly used for problems where the outcome is binary (Pass/Fail).
- **Logistic Regression** works by estimating the probability that a student will pass based on the input features, and it maps the output to one of two classes (0 or 1) using a logistic function.

3. Data Splitting

- The dataset was divided into **training** and **testing** sets using the train_test_split function. 80% of the data was used for training the model, and 20% was used for testing the model's performance.

4. Model Training and Hyperparameter Tuning

- The model was trained using the scaled training data. In this case, no hyperparameter tuning was done, but it could be added in future iterations to fine-tune the model's performance.

5. Model Evaluation

- The model's performance was evaluated using the following metrics:
    - **Accuracy**: This is the percentage of correct predictions out of all predictions. While useful, it can be misleading in the case of imbalanced datasets (i.e., a large number of students passing vs. failing).
    - **Confusion Matrix**: This matrix helped to understand the model's classification results by showing how many students were correctly classified as passing or failing, and how many were misclassified.
    - **Precision, Recall, and F1-Score**: These metrics were used to evaluate the model's ability to predict both passing and failing students. The F1-score provides a balanced measure between precision and recall, ensuring that both false positives and false negatives are minimized.
    - **Classification Report**: A comprehensive report that includes precision, recall, F1-score, and support for each class (Pass and Fail).

6. Visualization

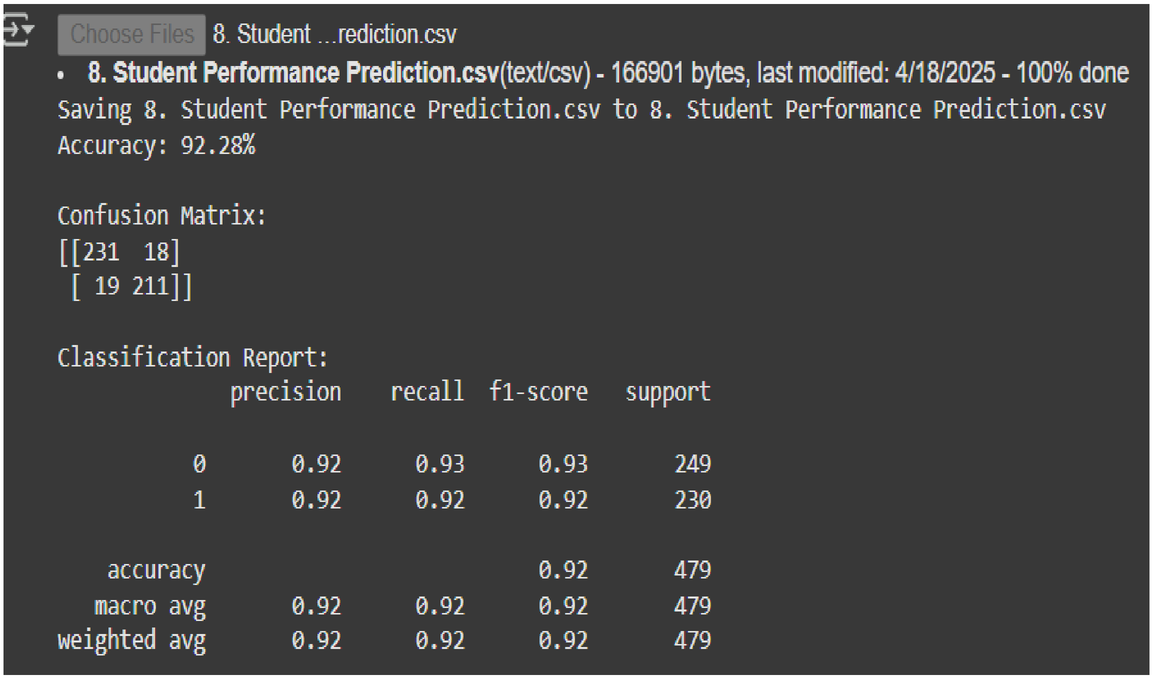
- The **Confusion Matrix** was visualized using a heatmap from the seaborn library. This helps us understand whether the model is better at predicting Pass or Fail students, and where improvements could be made.

CODE OF THE PROBLEM:

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
from sklearn.preprocessing import StandardScaler
import seaborn as sns
import matplotlib.pyplot as plt

# 1. Upload the file first
from google.colab import files
uploaded = files.upload()

# 2. Load the dataset
data = pd.read_csv('8. Student Performance Prediction.csv')  # Just the file name

# 3. Create Pass/Fail label based on GPA threshold
data['Pass'] = data['GPA'].apply(lambda x: 1 if x >= 2.0 else 0)  # 1 = Pass, 0 = Fail

# 4. Select features and target
features = ['StudyTimeWeekly', 'Absences', 'Tutoring', 'ParentalSupport']
X = data[features]
y = data['Pass']

# 5. Split into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# 6. Scale the features
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# 7. Train the Logistic Regression model
model = LogisticRegression()
model.fit(X_train_scaled, y_train)

# 8. Predict on test set
y_pred = model.predict(X_test_scaled)

# 9. Evaluate the model
```

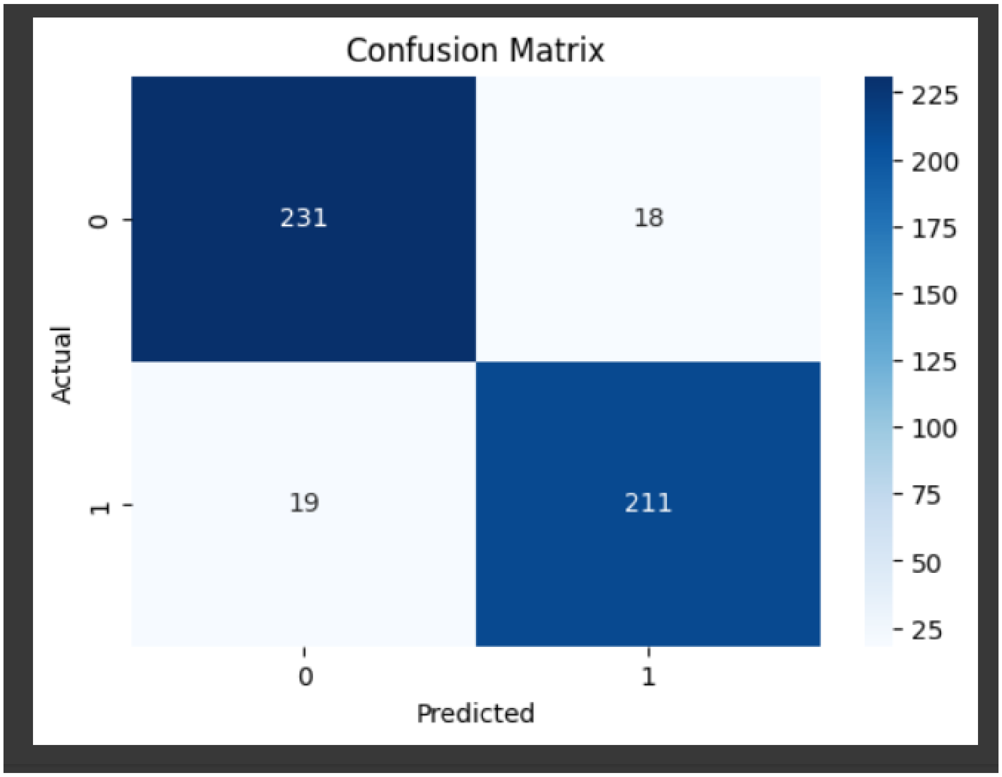
```
accuracy = accuracy_score(y_test, y_pred)
conf_matrix = confusion_matrix(y_test, y_pred)
class_report = classification_report(y_test, y_pred)
print(f"Accuracy: {accuracy*100:.2f}%")
print("\nConfusion Matrix:")
print(conf_matrix)
print("\nClassification Report:")
print(class_report)

# 10. Plot the Confusion Matrix
plt.figure(figsize=(6,4))
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Blues')
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.title('Confusion Matrix')
plt.show()
```

OUTPUT OF THE CODE:



- **8. Student Performance Prediction.csv**(text/csv) - 166901 bytes, last modified: 4/18/2025 - 100% done

```
Saving 8. Student Performance Prediction.csv to 8. Student Performance Prediction.csv
Accuracy: 92.28%

Confusion Matrix:
[[231  18]
 [ 19 211]]

Classification Report:
              precision    recall  f1-score   support

           0       0.92      0.93      0.93       249
           1       0.92      0.92      0.92       230

    accuracy                           0.92       479
   macro avg       0.92      0.92      0.92       479
weighted avg       0.92      0.92      0.92       479
```



References and Credits

1. **Datasets Used**:
   - The dataset used in this project is called Student Performance Prediction.csv and contains data about students' performance. It includes features like study time, attendance, and parental support.
2. **Libraries and Tools**:
   - **Python**: Programming language used for data manipulation, model training, and evaluation.
   - **Pandas**: Used for data preprocessing, including loading the CSV file and manipulating the data.
   - **Scikit-learn**: A library that provides tools for machine learning. It was used for splitting the data, scaling features, training the logistic regression model, and evaluating performance.
   - **Matplotlib** and **Seaborn**: Libraries used for visualizing the confusion matrix and performance metrics.
3. **External Content**:
   - Any external resources, datasets, or images used should be credited here. For example:
     - "Dataset obtained from the drive sent by teacher."
     - "Image from output .
4. **Acknowledgments**:
   - I would like to thank my teacher Mr Abhishek Shukla for providing guidance and feedback throughout the project.