DATA SCIENCE AI & ML Internship

Module & Milestone Project – Final Presentation

TOPIC:STUDENT PERFORMANCE ANALYSIS SYSTEM

PRESENTED BY:POORNIMA

Pioneered by

COMED KARES
Community Centric Education

ERA
FOUNDATION

COMED KARES
Community Centric Education

Problem Statement

Schools and colleges manage large student data. Manual calculation of marks and results takes time. Errors may occur in manual calculations and an automated system to analyze performance easily.

COMED KARES
Community Centric Education

System Architecture

Input: Student data in CSV file.
Processing: Python program using Pandas & NumPy.
Analysis: Calculate Total, Average, Result.
Output: Performance report and insights.

COMED KARES
Community Centric Education

Hardware Components Used

- Computer or Laptop.
- Minimum 4GB RAM recommended.
- Keyboard and Mouse for operation.
- No special hardware required.

COMED KARES
Community Centric Education

Control Logic & Decision Flow

Read CSV file using pd.read_csv().

- Calculate Total and Average marks.
- If Average ≥ 50 → Pass.
- If Average < 50 → Fail.
- Identify highest marks using idxmax().

COMED KARES
Community Centric Education

Final Outcome

Automated student result generation.

- Quick identification of toppers.
- Clear Pass/Fail classification.
- Better understanding of attendance impact.

COMED KARES
Community Centric Education

Final Outcome

```python
import pandas as pd

data = {
    "Name": ["Alice", "Bob", "Charlie", "David", "Eva"],
    "Math": [78, 45, 88, 60, 35],
    "Science": [85, 50, 92, 58, 40],
    "English": [90, 40, 84, 65, 30],
    "Attendance": [92, 80, 95, 85, 70]
}

df = pd.DataFrame(data)
df.to_csv("students.csv", index=False)

print("students.csv created successfully!")
```

[6]

```
students.csv created successfully!
```

```python
import pandas as pd
import numpy as np

# --- 1. CSV Reading and DataFrame Basics ---
# Read the student data from a CSV file into a pandas DataFrame
try:
    df = pd.read_csv('student_data.csv')
```

COMED KARES
Community Centric Education

Final Outcome

```
try:
    df = pd.read_csv('student_data.csv')
    print("Original DataFrame:")
    print(df.head())
    print("-" * 30)
except FileNotFoundError:
    print("Error: student_data.csv not found. Please create the file.")
    exit()

# New Derived Columns (Average Marks per Student) ---
# Calculate the average score for each student using pandas .mean(axis=1)
# axis=1 ensures the mean is calculated across columns (subjects) for each row (student)
subject_cols = ['Math', 'Science', 'English']
df['Average_Score'] = df[subject_cols].mean(axis=1)
print("\nDataFrame with Average Scores:")
print(df[['Name', 'Average_Score']].head())
print("-" * 30)

#Pass/Fail Classification (Logic & Conditions) ---
# Define a passing threshold
PASS_THRESHOLD = 60

# Use a Python function and apply it to the DataFrame to classify students
def classify_pass_fail(score):
    """Classifies a score as Pass or Fail based on a threshold."""
    if score >= PASS_THRESHOLD:  # Python logic/condition
        return 'Pass'
    else:
        return 'Fail'

# Create a new column 'Pass_Status' using the classify_pass_fail function and .apply()
df['Pass_Status'] = df['Average_Score'].apply(classify_pass_fail)
print("\nDataFrame with Pass/Fail Status:")
print(df[['Name', 'Average_Score', 'Pass_Status']].head())
print("-" * 30)

#NumPy (Average, Max, Min Analysis Examples) ---
# Use NumPy functions for overall analysis
```

COMED KARES
Community Centric Education

Final Outcome

```
# Use NumPy functions for overall analysis
overall_avg = np.mean(df['Average_Score'])
overall_max = np.max(df['Average_Score'])
overall_min = np.min(df['Average_Score'])

print("\nOverall Performance Analysis (using NumPy):")
print(f"Average Score of all students: {overall_avg:.2f}")
print(f"Maximum Average Score: {overall_max:.2f}")
print(f"Minimum Average Score: {overall_min:.2f}")
print("-" * 30)

#Filtering (Pass/Fail Students) ---
# Filter DataFrame for only 'Pass' students
passed_students = df[df['Pass_Status'] == 'Pass']
print("\nStudents who Passed:")
print(passed_students[['Name', 'Average_Score']])
print("-" * 30)

#Analysis Example: Subject-wise Topper ---
# Find the topper for each subject using pandas groupby and idxmax
for subject in subject_cols:
    topper_index = df[subject].idxmax()
    topper_name = df.loc[topper_index, 'Name']
    topper_score = df.loc[topper_index, subject]
    print(f"Topper in {subject}: {topper_name} (Score: {topper_score})")
print("-" * 30)

# Analysis Example: Attendance vs. Marks ---
# Analyze correlation between attendance and average marks to understand their relationship
correlation = df['Attendance_Days'].corr(df['Average_Score'])
print("\nAttendance vs. Marks Analysis:")
print(f"Correlation between Attendance Days and Average Score: {correlation:.2f}")
if correlation > 0.7: # Python condition for strong correlation
    print("Conclusion: There is a strong positive correlation, suggesting higher attendance is linked to higher marks.")
else:
    print("Conclusion: The correlation is weak or moderate, other factors might be more influential.")
print("-" * 30)
```

COMED KARES
Community Centric Education

RESULTS:

```
Student Data:
      Name  Math  Science  English  Attendance
0    Alice    78       85       90          92
1      Bob    45       50       40          80
2  Charlie    88       92       84          95
3    David    60       58       65          85
4      Eva    35       40       30          70

NumPy Analysis:
Overall Average: 62.666666666666664
Maximum Marks: 92
Minimum Marks: 30

Passed Students:
      Name  Math  Science  English  Attendance    Average Result
0    Alice    78       85       90          92  84.333333   Pass
2  Charlie    88       92       84          95  88.000000   Pass
3    David    60       58       65          85  61.000000   Pass

Failed Students:
   Name  Math  Science  English  Attendance  Average Result
1  Bob    45       50       40          80     45.0   Fail
4  Eva    35       40       30          70     35.0   Fail

Subject-wise Toppers:
...
1      Bob    45       50       40          80  45.000000   Fail
2  Charlie    88       92       84          95  88.000000   Pass
3    David    60       58       65          85  61.000000   Pass
```

COMED KARES
Community Centric Education

Learnings & Skills Gained

- Understanding of Python basics.
- Working with NumPy and Pandas.
- Data cleaning and analysis skills.
- Logical thinking and decision making.

COMED KARES
Community Centric Education

Thank you