DATA SCIENCE

PROBLEM STATEMENT NO. 50

BASIC CSV FILE ANALYZER

CONTENT

Problem Statement

Why this?

Use of Modules

Use of Error Handling

Use of try method

Data Cleaning and Saving it

PROBLEM STATEMENT

To design Python script that reads a CSV file and gives quick insights.

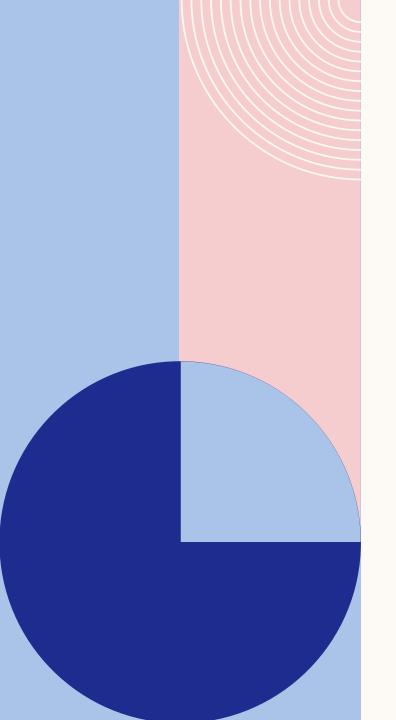
Expected Outcome:

Show file structure (columns, types), Show top 5 and bottom 5 rows, Count of unique values per column, Save cleaned data to a new CSV

Input:

Any CSV file





WHY THIS?

The reason I chose this problem statement was because it stood out to me more as compared to other problems. In my opinion CSV analyzer solve issues of many who want a quick insight to their CSV file.

USE OF MODULES

During the course of learning about Data Science on GUVI, I learned about the concept of importing required modules and thus applied appropriate modules in my python script.

```
CSV File Analyzer.py

1 import pandas as pd
2 import os
3
```

USE OF ERROR HANDLING

During the course of learning about Data Science on GUVI, I learned about the concept of error handling and thus thought of applying in my python script.

USE OF TRY METHOD

During the course of learning about Data Science on GUVI, I learned about the concept of try method and thus thought of applying in my python script.

DATA CLEANING AND SAVING IT

With GUVI videos and external help, I was able to complete a major part where I got stuck a few times and that was saving the cleaned file.

Scratched my head and got angry but got over with it within the time period. Since this was a part of marking rubric I paid more attention to it, similarly I tried sticking to the rubric as much as possible for other parts also.

```
# 6. Save cleaned data to a new CSV
# Basic cleaning: dropping duplicate rows
initial rows = len(df)
df cleaned = df.drop duplicates()
rows after cleaning = len(df cleaned)
print(f"\n--- Data Cleaning ---")
if initial rows > rows after cleaning:
    print(f"Removed {initial rows - rows after cleaning} duplicate row(s).")
    print("No duplicate rows found or removed.")
# Construct new file name
base name = os.path.basename(file path)
dir name = os.path.dirname(file path)
new file name = f"cleaned data {base name}"
output file path = os.path.join(dir name, new file name)
df cleaned.to csv(output file path, index=False)
print(f"Cleaned data saved to: '{output file path}'")
print("\n--- Analysis Complete ---")
```

THANK YOU

From: Nisarg Kumar Gharde

*I hope the presentation was up to your expectation and sorry for any inconvenience, if caused.