Reflective Document

Student Id:24038688

## Part 1

**Describe your approach to this part of the assessment**

In order to complete the activities in this part of the evaluation, the problem was carefully analyzed by breaking it down into smaller, manageable parts. So, the main purpose was to develop with the Python coding that could handle CSV data efficiently, ensuring robustness and reliability in different situations.

First one is Planning and preparation. The ability to read particular columns, load data into memory, and compute correlations between columns were among the essential requirements which were determined. Therefore, potential edge cases were expected to include non-numeric data, missing columns, and empty files. And also, error-handling systems were designed with this in mind.

On the other-hand, in order to extract the header and related data, validate the column index, and check for empty files, a function for reading a single column was put into place. For loading all the data into memory, a dictionary was employed, with column names serving as keys and data lists as values. So that, this method made it possible to retrieve and process data efficiently. Then, a mathematical formula served as the basis for the correlation computations, which were further verified to make sure the data was non-empty and numeric.

**Did you find it easy or difficult? Why?**

First of all, the function was somewhat difficult to develop and it took extra work to make sure the function handled a variety of edge circumstances, even if the fundamental logic for reading and processing the CSV files were simple. For example:

* For reading the content of the file and then, utilizing Python’s string manipulation tools to divide the rows into columns were the simple parts of the assessment.
* For managing the edge situations which presented difficulties, including dealing with faulty indices, empty files, and making sure that resources were property cleaned away following file access.

**What problems did you encounter?**

There are the main three issues which I encountered in part 1 of the assessment. They are:

1. **Resolving invalid column indices:** when invalid indices were supplied, and then, the function’s original absence of checks for out-of-range indices resulted in IndexError exceptions.
2. **Empty files:** Attempting to process an empty file resulted in unexpected behavior since the headers and data lines were absent.
3. **Memory Usage with Large Files:** reading all lines at once with file.readlines() could potentially lead to memory issues with large files.

**How did you overcome them?**

There are three main solutions which put into practice.

* **Index Validation:** To make sure the supplied column index is legitimate; range check was included. An IndexError with a detailed message is raised if the index is out the range.
* **Empty File Check:** To raise a ValueError in the event that the file is empty, and then, a validation step was introduced. So, This guarantees that in such cases, the function fails elegantly.
* **Optimised Memory Usage:** Iterating through the file line by line rather than using readlines() which could improve memory usage for larger datasets, though it is not necessary for tiny files. Therefore, this method was recorded for optimization in the future.

**Identify any strengths/weaknesses of the approach you took**

There are some strengths and weaknesses of the approach for the part 1 of the assessments.

**Strengths:**

* Data may be extracted from any CSV files using this modular and reusable procedure.
* Resilience against a range of edge circumstances which is safeguarded by the incorporation of strong error handling.
* Having a well-organized and flexible approach makes the code easier to read and maintain.

**Weaknesses:**

* Limitations: Using readlines() to load every line into memory could lead to issues when working with large files.
* Debugging in larger projects, not logging error messages and only printing them can create challenges when troubleshooting problems.

**How could this approach be improved?**

* **Logging:** use a logging tool instead of print statements to systemically record errors for debugging.
* **Chunk Based File Reading:** to handle larger datasets more efficiently, modify the process to read the file in smaller chunks.
* **Adaptability:** enhance the function to support a customizable delimiter, making it adaptable to non-standard CSV formats.
* **Unit Testing:** implement unit tests to verify the function under various conditions, including edge cases.

**Suggest an alternative approach you could have taken**

There are alternatives methods for part 1 of the assessments. They are:

* **Using Libraries:** the method might be made simpler by using the pandas (python library for data manipulation) or Python’s CSV module. For example, the read\_csv() function in pandas help to reduce boilerplate code by directly extracting columns using column names and indices.
* **Streaming Processing:** to save memory, the function could process lines as they are read rather than reading the file into memory.

## Part 2

**Describe your approach to this part of the assessment**

It started off with pattern finding in datasets, cleaning the data and verifying a hypothesis which was a part of data-analysis process. Then, I carried on with the essential things like loading, merging, visualizations, identifying/removing outliers which is done using python libraries (SciPy) like pandas, seaborn and running statistical tests.

Also, I merged the datasets, cross-verified the differences, removed outliers to keep the accuracy, and handled any missing values to find out the relationship between scores and final results. Even more, I did exploratory data analysis to create a few visualizations to illustrate main relationships by identifying trends and patterns in the data. Finally, hypothesis testing was conducted to assess how statistically significant the scores vary across different categories.

We had to adopt this process because of the emphasis on data accuracy, reliability and dependability, and valuable insights,through automation to minimize the need for manual work.

**Ease or Difficulty and Why?**

Overall, the task was a bit challenging because of its complex nature.Part of the requirement is attention to details, effective data management and a deep knowledge of a statistical methods.Loading and cleaning of data seem simple because of my experience with panda, but certain tasks, like handling mismatches and conducting pairwise hypothesis tests, posed some difficulties.

**Problems and Solutions**

1.Data mismatches: I encountered this issue between the scores and final\_result columns.This was addressed by creating a customer filter function which corrected some rows that could not align according to specific score ranges.

2.Outliers: The presence of extreme values in the score and click\_events made visualization and statistical analysis difficult and complicated. I removed these outliers adopting interquartile range range methods which helped to improve the analysis significantly.

3. Sample Size:There was this issue of insufficient sample Size of data to carry out hypothesis testing.This was handled by noting the observation and avoiding making conclusions based on it.

4.Missing values: Some missing values in the click\_events column could twist analysis.I had to replace NaNs with 0 since it is a sensible default for missing interaction data.

**Identify any strengths/weaknesses of the approach you took**

The Part 2 assessments displayed both strengths and weaknesses.

**Strengths:**

* The process of cleaning the data from datasets was systematic, ensured reproducibility and efficiency, which concentrated on analyzing only the most relevant information.
* Visualizations clearly showed understandings of key trends and relationships.
* Meticulous and careful cleaning steps ensured good level of data quality because the merged dataset was consistent and free from outliers
* Well structured hypothetical testing which ensured appropriate checks for assumptions like sample size and distributions.

**Weaknesses:**

* Limitation in flexibility if ranges change because the score ranges for final\_results were hardcoded.
* There was emphasis on predefined hypotheses, limiting exploratory insights beyond preset comparison (such as pass vs fail).

**How could this approach be improved?**

* **Dynamic Rules:** Enhancement of adaptability by replacing hardcoded score ranges with a configuration parameters.
* **Advanced Statistical Analysis:** Deploying non-parametric tests when assumptions of homogeneity of variance are violated in T-tests.
* **Visualization:** Use interactive visualization tools (such as Tableau and Plotly) to provide more flexible user exploration of the data and more dynamic and detailed analysis.
* **Source of data:** Since the current research was restricted to two datasets, it may have hampered the ability to identify deeper patterns. Therefore, incorporate additional data sources to enhance insights.
* Explore more robust merging through alternative joins to make sure that all valuable information is included during merging process

**Alternative Approaches**

* **Usage of Clustering analysis:** Adopting unsupervised machine Learning byusing techniques like K-means to discover the natural patterns without having any prior assumptions.
* **Regression Analysis:** to apply linear regression which can predict the relationships between constant variables.
* **Adopting Bootstrapping for Hypothesis Testing:** Use bootstrapping to calculate confidence intervals instead of T-tests, making the analysis more robust to small sample sizes and non-normal distributions.
* **Relational database:** employing structured querying by loading data into relational database like PostgreSQL
* **Multiple Techniques:** for a more detailed data analysis that exposes deeper insights and more accurate results, use a mix of various methods.