1. **Task Description**

In this task, the goal was to perform advanced data cleaning techniques on the Titanic dataset. The focus was on handling missing values, applying fuzzy string matching for identifying potential duplicates in the **Name** column, and removing those duplicates to ensure the dataset was clean and consistent.

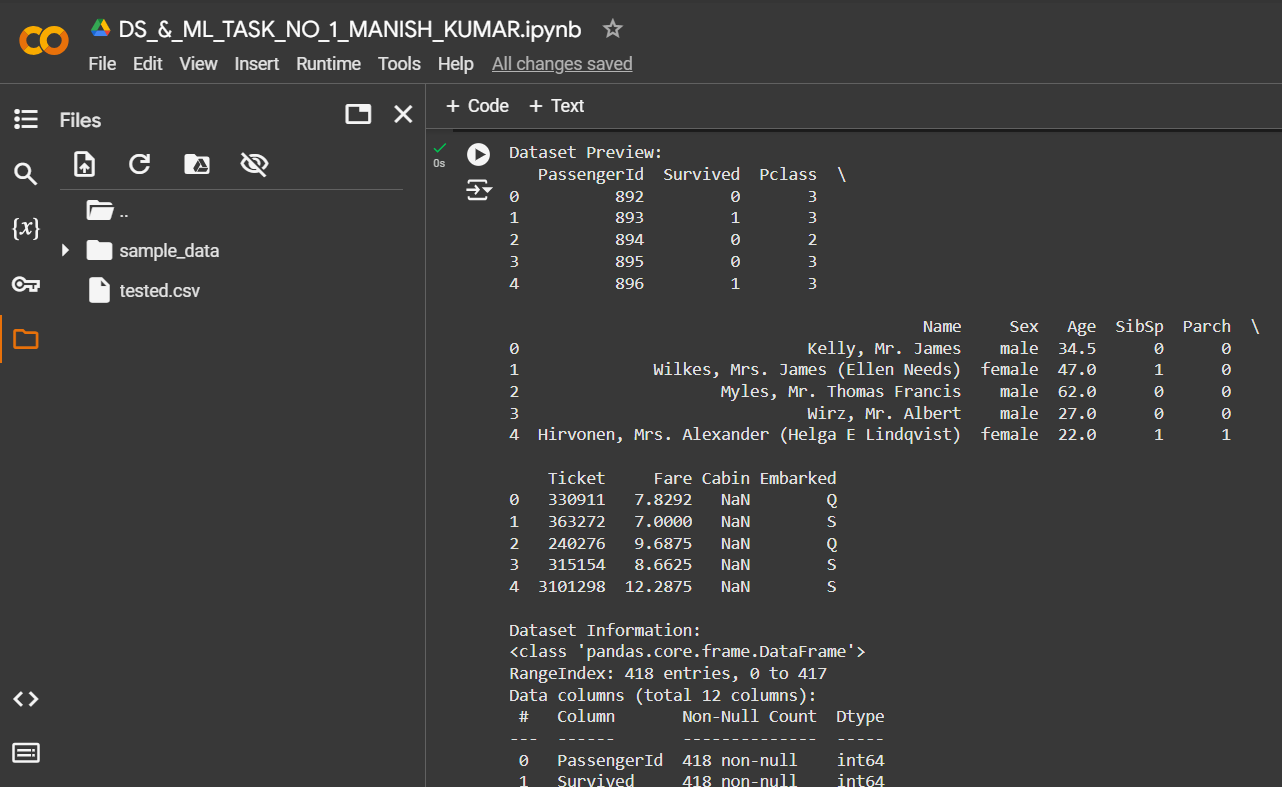
The main steps involved:

* Handling missing values in columns like **Age**, **Fare**, and **Cabin**.
* Using **FuzzyWuzzy**, a library for fuzzy string matching, to identify similar names in the **Name** column.
* Removing duplicates based on fuzzy matching results.
* Saving the cleaned dataset as **cleaned\_data.csv**.

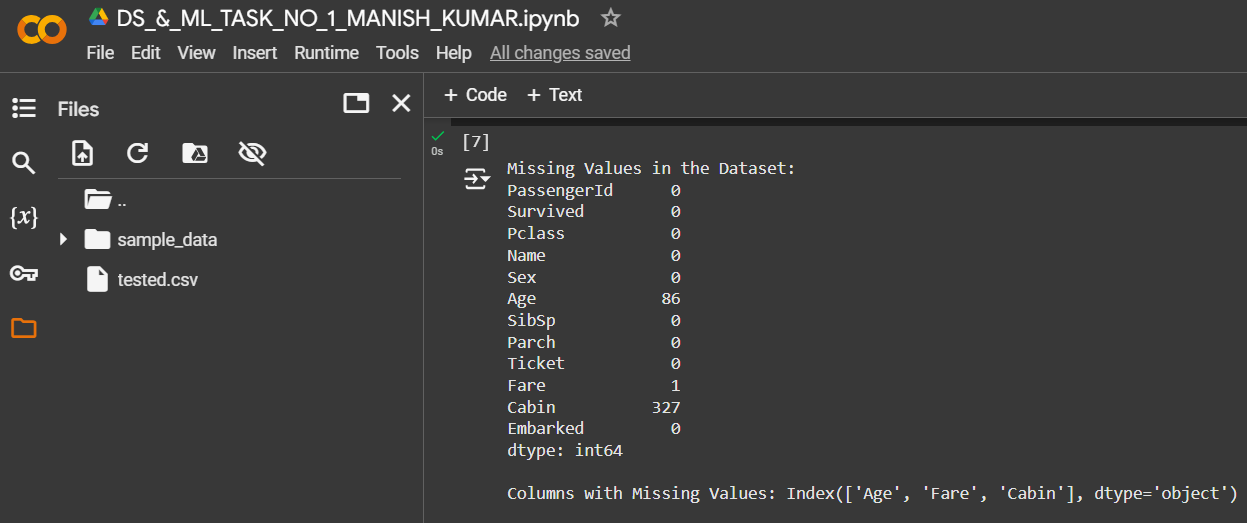
The task aimed to improve the quality of the dataset by addressing both missing data and duplicate entries, which are common issues in real-world datasets.

1. **Attach Screenshot of Output**

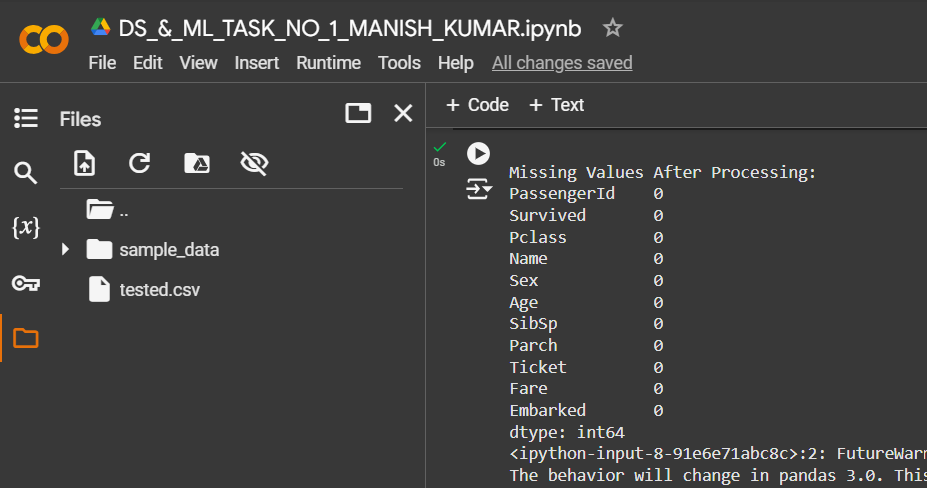
* **Dataset Preview:**



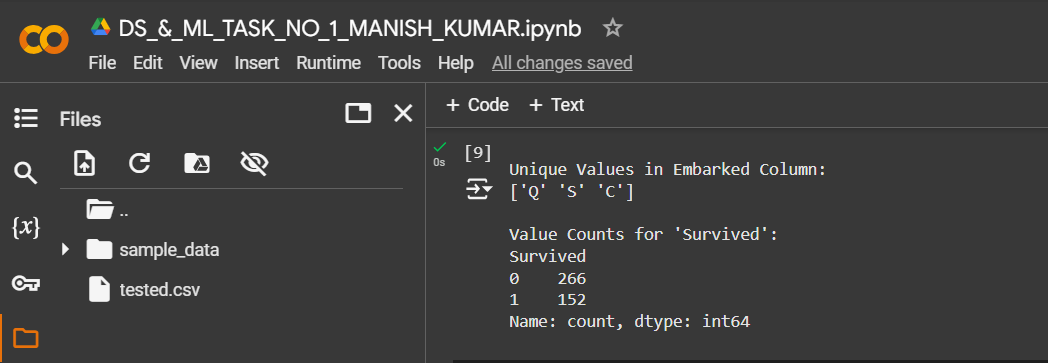
* **Missing values in the Dataset:**



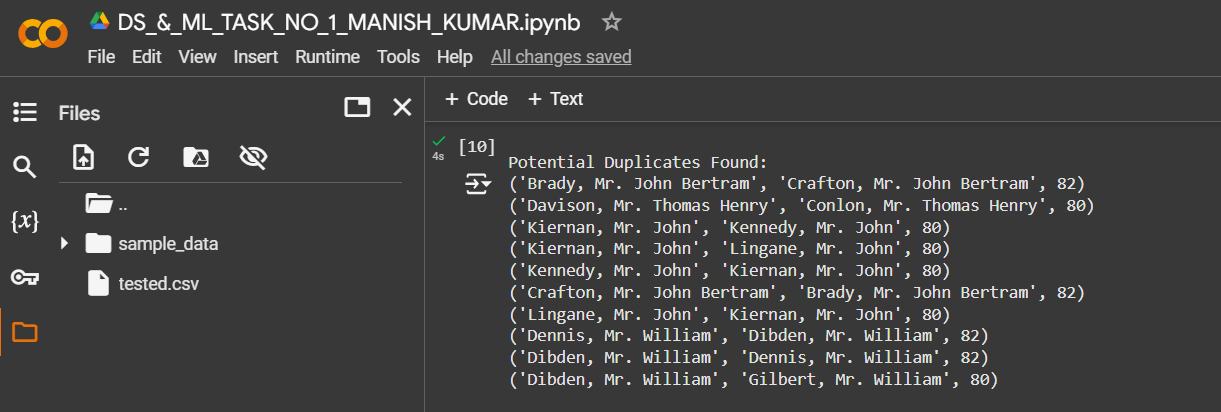
* **Missing values after Processing:**



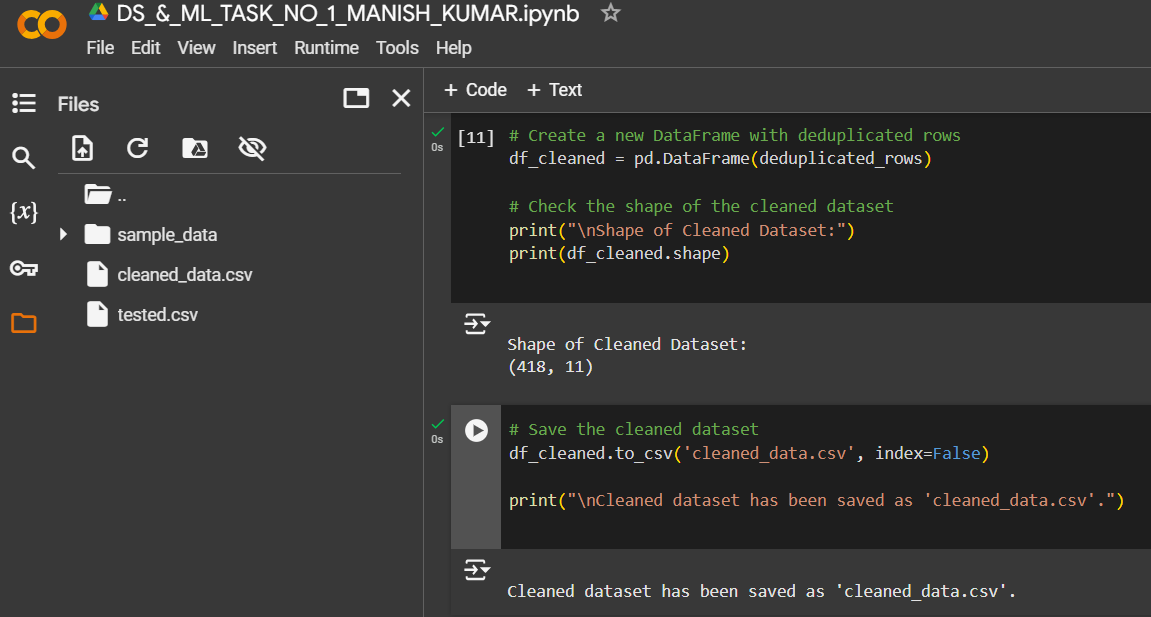
* **Unique Values:**



* **Potential Duplicates Found:**



* **Cleaned Dataset**



1. **Describe Widget/Algorithm Used in Task**

**Algorithm Used: Fuzzy String Matching (FuzzyWuzzy Library)**

**Fuzzy String Matching** is a technique used to compute the similarity between two strings, particularly useful for identifying and resolving inconsistencies like typos or variations in textual data. In this task, the **FuzzyWuzzy** Python library was employed to identify potential duplicates in the **Name** column of the Titanic dataset. Below is an elaboration of the algorithm and its implementation:

**Core Concept: Levenshtein Distance**

* **Levenshtein Distance**, also known as edit distance, measures the number of single-character edits (insertions, deletions, substitutions) required to change one string into another.
* FuzzyWuzzy leverages this concept to compute a **similarity ratio** (percentage match) between two strings..

**FuzzyWuzzy Methods Used**

* **fuzz.token\_sort\_ratio**:
  + This method tokenizes the string (splits it into words), sorts the tokens, and then compares the sorted tokens. This helps in comparing strings where the word order might differ.
  + Example:
    - String 1: *"John Smith"*
    - String 2: *"Smith John"*
    - fuzz.token\_sort\_ratio will return a **100% match**.
* **Threshold**:
  + A similarity threshold of **80%** was set to identify potential duplicates. Names with a similarity score of **80 or above** were flagged for deduplication.

**Steps in the Process**

1. **Data Preprocessing**:
   * **Handling Missing Values**:
     + **Age**: Replaced missing values with the column mean.
     + **Fare**: Replaced the single missing value with the column median.
     + **Cabin**: Dropped the column due to a high percentage of missing data.
   * Ensured all necessary preprocessing steps (e.g., removing unnecessary whitespace in names) were completed.
2. **Fuzzy Matching**:
   * Applied **FuzzyWuzzy** to the **Name** column.
   * Compared each name in the dataset to every other name.
   * Generated a list of name pairs with a similarity score >= 80.
3. **Deduplication**:
   * Reviewed flagged duplicates manually or programmatically to determine whether they represented true duplicates.
   * Retained only the first occurrence of each duplicate name and removed the rest.

**Libraries/Tools Used**

1. **Pandas**: Used for dataset manipulation, including filling missing values, dropping unnecessary columns, and exporting the cleaned data.
2. **FuzzyWuzzy**: Provided powerful tools for fuzzy string matching, particularly for comparing text fields like names.
3. **Python**: Implemented the entire workflow, including preprocessing, fuzzy matching, and deduplication.

**Benefits of the Algorithm**

* **Accuracy**: Handles slight variations in text effectively, making it ideal for datasets with inconsistent naming conventions.
* **Efficiency**: While comparing every record to every other record could be computationally expensive, FuzzyWuzzy streamlines the process with token-based comparisons.
* **Customizable**: Threshold values and methods can be adjusted based on the dataset's specific requirements.

**\*\*\* The End \*\*\***