**Detailed Summary of My Data Cleaning Tasks**

I have a dataset containing the results of a survey conducted by the Stack Overflow website. (<https://survey.stackoverflow.co/2024/> ) The main dataset consists of 65,437 rows and 114 columns. In addition, there is another dataset showing the schema, which consists of 87 rows and 6 columns.

Although the raw data is somewhat organized, there are many null/NaN values.

Besides that, the survey results mostly consist of string values. For this reason, I will need to perform some data manipulation to extract meaningful insights from these string values during the analysis phase.

Some columns contain multiple values that I will need to evaluate for analysis as part of the data structuring process. At this point, I will first determine the group of data I want to analyze and then use Python - Pandas Library to organize the necessary values. I will store the data I plan to analyze for each dashboard in separate CSV files.

**1. Main Data Groups Creation:**

In my data cleaning process, the first major task was to create several main data groups from the survey dataset. I achieved this by selecting specific columns of interest and categorizing them into distinct data groups for better organization and analysis. Here's a brief overview of the groups I created:

* **SUMMARY:** This group includes essential respondent information such as their ResponseId, DevType, RemoteWork, Industry, Country, Age, EdLevel, YearsCode, LearnCode, and JobSat.
* **CODE LEARNING:** This group focuses on how respondents learned to code, represented by their ResponseId and LearnCode columns.
* **TECH CHOICE:** This group captures the technologies admired by respondents, including their preferred programming languages (LanguageAdmired), databases (DatabaseAdmired), platforms (PlatformAdmired), web frameworks (WebframeAdmired), operating systems (OpSysPersonal use), and AI tools (AISearchDevAdmired).
* **AI:** This group gathers information about respondents' opinions on AI, with columns such as AISelect, AISent, AIAcc, AIComplex, AIThreat, AIBen, and AIToolCurrently Using.

After creating these groups, I saved each one as separate CSV files to facilitate future analysis.

**2. Null Values Analysis and Column Separation:**

Next, I focused on understanding and processing null values within my datasets. Here are the functions I developed to achieve this:

* **getting\_null\_info(csv\_file, path):** This function analyzes a CSV file, loads it into a pandas DataFrame, and prints out information about the number of null (missing) and non-null (not missing) values for each column in the dataset. This information is crucial for identifying data quality issues and guiding subsequent data cleaning steps.
* **sep\_and\_bool(csv\_file, path, column\_name):** This function processes columns containing semicolon-separated strings in a CSV file. It creates new DataFrame columns with boolean values indicating the presence of each unique value from the original column. This transformation helps in better analyzing and utilizing multi-valued categorical data. The function then replaces the original file with the processed one to maintain data consistency.
* **detect\_semicolon(csv\_file, path):** This function reads a CSV file and checks for the presence of semicolons (;) in any column of the first 20 rows. It returns a list of columns that contain semicolons, which helps in identifying columns that need special processing.
* **cleansing\_csv(csv\_file, path):** This function orchestrates the data cleaning process by identifying columns with semicolons and passing them to the sep\_and\_bool function for further processing. It ensures that all relevant columns are properly transformed and cleaned.

**3. Automating Data Cleansing for Multiple Files:**

To streamline my data cleaning workflow, I implemented a script that automates the cleansing process for multiple files:

* I scanned a specified directory for files starting with 'df'.
* I applied the cleansing functions to each identified file.
* This automation ensures consistent and efficient processing of all relevant files, saving time and reducing the risk of errors.

**4. Top Column Selection Based on True Counts:**

In this step, I focused on identifying the most relevant columns in selected CSV files based on the count of 'True' values. Here's how the function works:

* **getting\_top(path, csv):** This function loads a CSV file into a pandas DataFrame and calculates the count of 'True' values for each column, excluding ResponseId. Based on the total number of columns, it retains the top columns with the highest true counts:
  + If the file has 40 or more columns, it keeps the top 15 columns.
  + If the file has between 20 and 39 columns, it keeps the top 10 columns.
  + The function removes the less significant columns and replaces the original file with the updated one to ensure data relevance and efficiency.

**5. Scaling and Null Value Replacement:**

The final step of my data cleaning process involved adjusting the scale of the "JobSat" column and replacing null values. Here are the details:

* **Scaling:** I adjusted the "JobSat" column from a 10-point scale to a 100-point scale to standardize the responses. I also identified and scaled other columns containing 10-point scale answers accordingly.
* **Null Value Replacement:** To improve data handling, I replaced zero values with nan values in the dataset, making it easier to analyze and manage the data.