**Glassdoor Dashboard Documentation**

**Step 1: I handled each column one by one.**

* **Job title:**

There were too much noise, abbreviation, and distracting information, Like (sr., jr., full time, part time, some data and numbers), in order to handle such case:

1. Capitalize the first letter of each word.
2. Replace wild char and punctuation mark with nothing.
3. Graphical user interface

   Description automatically generatedI created a conditional column.

Graphical user interface, application

Description automatically generated

* **Company name:**

The company rating was stuck to the end of the company name.

1. Replace the dot with equal sign because there were some company names that contains dot.
2. Split the column with the equal sign from the most right.
3. I changed the output column of the split process to number and replaced the null with an outlier.
4. I added a custom column, in which I checked if the spited column does contain a number from (0-9) then get the main column from 0 until the length of the string without the last 2 char as they will be the equal sign and the number, else return the value in the main column.
5. Finally replace the equal with dot.

Graphical user interface, text, application, email

Description automatically generated

* **Headquarters / Size / Founded / Type of ownership / Sector / Revenue / Competitors**

1. replace -1 with Unknown.

* **Seniority:**

1. Replace the abbreviation with the whole word. (sr.: senior, jr.: junior)

* **Age:**

This column contains an outlier, and its data is not reliable, so I did calculated the q1 & q3 in order to replace all values that are below the q1 with q1 value and replace all values that are greater than q3 with q3 value.

Box and whisker chart

Description automatically generated with low confidence

* **Location / Headquarter:**

1. Split into state and city and trim the text.

* Size:

1. Split by non-digit to digit.
2. Remove text and keep the min and max number of employees.
3. Add a conditional column if the max is null then let it with the min.

Graphical user interface, application, Teams

Description automatically generated

* **Revenue:**

1. Replace the less than 1 million with 999 thousand.
2. Add a conditional column to track the company revenue if it’s in billion || millions || thousand.

Graphical user interface, application

Description automatically generated

1. Split by non-digit to digit.
2. Remove text and keep the min and max revenue.
3. Add a conditional column if the max is null then let it with the min.



* **Add a conditional column to track the company size by the number of employees.**

Graphical user interface, application

Description automatically generated

* **Skills:**

1. Duplicate the eda and keep the job title column and columns related to skills.
2. Unpivot skills column
3. Group by job title and skills and sum the value of repetition.

* **Job description:**

1. Duplicate the eda and keep the job title column and Job description
2. Replace wild char and punctuation mark with nothing.
3. Split the Job description column.
4. Unpivot the Job description column.
5. Trim the Job description column.
6. Merge the Job description column with the stop words in a left anti join so I get only the descriptive words in job description column.

* **Competitors:**

1. Duplicate the eda and keep the job title column, company name and competitors.
2. Split the competitor’s column and the unpivot it, to get the company and its competitors each at a row.

* **Measures:**

1. Count of Company = DISTINCTCOUNT(eda\_data\_edit[Company\_Name ])
2. Count of total Jobs = COUNT(eda\_data\_edit[Job\_ID])
3. Count of open jobs = CALCULATE(COUNT(eda\_data\_edit[employer\_provided]), FILTER(eda\_data\_edit, eda\_data\_edit[employer\_provided]= 0))
4. count of filled job = [Count of total Jobs]-[Count of open jobs]

* **Derived Column:**

1. MaxSalary = eda\_data\_edit[max\_salary]\*1000
2. MinSalary = eda\_data\_edit[min\_salary]\*1000

* **Aggregated table**

1. Aggregate = GROUPBY(eda\_data\_edit, eda\_data\_edit[Job\_Title\_clean], "Average", AVERAGEX(CURRENTGROUP(), eda\_data\_edit[MinSalary]+  eda\_data\_edit[MaxSalary]))

Graphical user interface, application, table

Description automatically generated

The challenges I went through:

Hence the data is web scraped so it is containing many non-reliable data like outliers and wild characters, moreover its poorly structured so went through a process of cleaning and transforming the data in a better way. One of main challenges where that most of the data is in a text format and I rarely worked on such a one so it was hard to get insights and build a story.