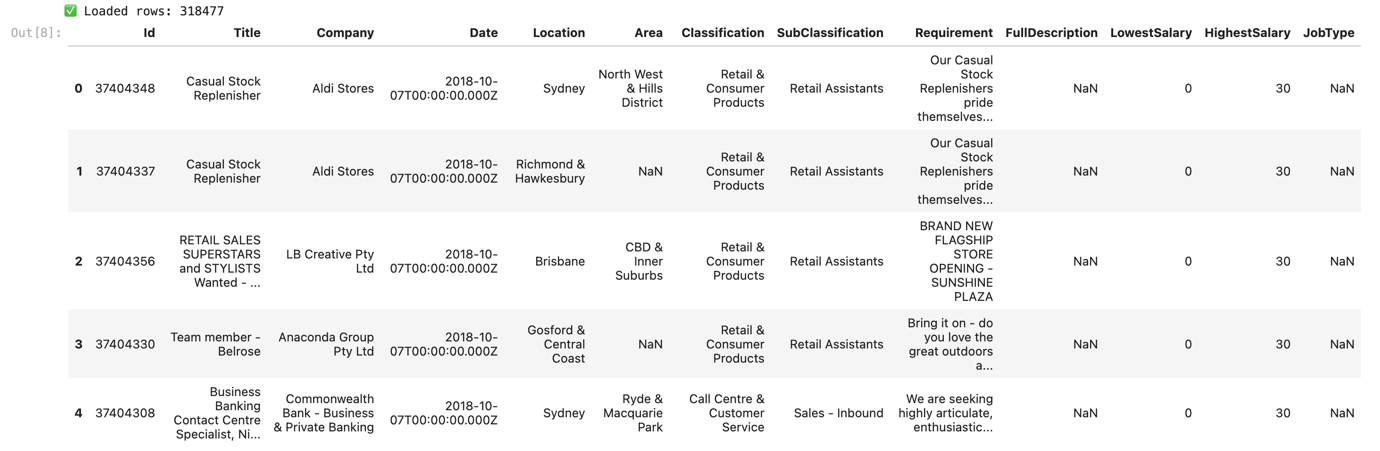
**Part 1: Data Exploration Report**

**(1) Describe the dataset**

The dataset contains job advertisements with a variety of features including job titles, companies, locations, descriptions, and salary ranges. It appears to have been extracted from a job portal or employment listings. It includes 318,477 rows and 13 columns after skipping bad lines. It is provided in CSV format and was loaded using Python's pandas library. The key columns include: Id, Title, Company, Date, LowestSalary, HighestSalary, JobTypeRequirement, FullDescription, while Location, Area, Classification, SubClassification contain many missing values.

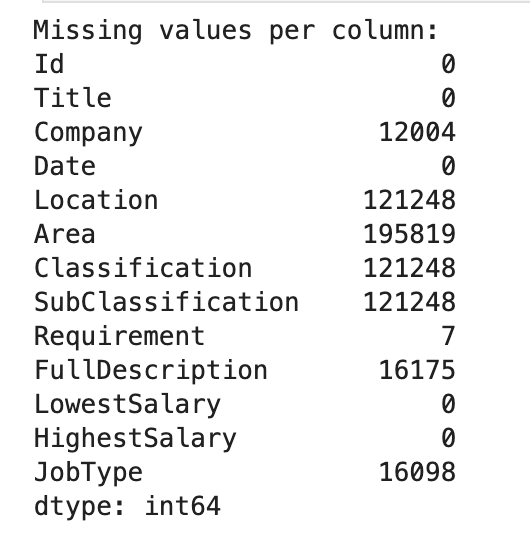


*Figure 1 Sample Data Frame from 318477 rows*

<Jeh description part Describe the dataset>

Personally, I have decided to neglect **FullDescription** and **JobType** from analysis because of their lower significance and missing values (NaN) around 16000 rows. However, there’s no need to drop those missing rows using df\_clean = df.dropna() as it would affect the overall data shape.

Columns such as Location, Area, Classification, and SubClassification have substantial missing data (over 100,000 entries), and I believe including them could introduce bias, so there is no need to directly analyse them. Therefore, I do not use them in key analyses, but could benefit inference for some missing value in Company**.**

****

*Figure 2 Missing Values per Column*

In contrast, columns I expect to analyse like:

* Title and Company are very important, as they allow for deep analysis across various job sectors and hiring practices.
* Requirement is also highly valuable, as it can be matched with job titles and companies to explore qualification trends.
* LowestSalary and HighestSalary are critical numerical features for understanding job compensation.

<Chan: Describe the steps you used for data preparation and preprocessing>

**(2) Describe the steps for data preparation and preprocessing**

**Loading the data** The dataset is loaded using pandas:

import pandas as pd

df **=** pd**.**read\_csv("data\_v1.csv", engine**=**"python", on\_bad\_lines**=**'skip')

**Initial structure verification**

Before any preprocessing, I ensured that each row had exactly 13 columns — matching the expected format of the dataset. Invalid rows were abandoned during analysis to maintain initial structural consistency and validity. I verified that all rows had 13 columns, then checked any rows with missing data for consideration:

print("All columns:", df**.**columns**.**tolist())

print("Missing values per column:")

print(df**.**isnull()**.**sum())

**Flexible missing data handling**  
Rather than dropping all rows with any missing data, I used a **selective strategy and create a new value** for better analysis:

1. For core fields like Title, Company, Requirement, and Salary, I kept rows even with some missing values, especially when the missing portion was small or could be compensated with inference from related features.
   1. For example, Company had around 12000 missing values, but those rows might still be useful if Location, Classification, or Requirement are present — allowing us to approximate company type or industry.
   2. This opens the possibility of creating derived features (such as CompanyGroupID) to group similar job types or hiring patterns.
2. For Requirement, which is highly valuable and only has a very small number of missing values (7 rows), and I believe it will not affect the large-scale analysis.
3. For fields with excessive missing datalike Location, Area, Classification, and SubClassification (each missing in over 120k rows), I chose not to use them directly for correlation or prediction, but rather for general descriptive statistics (such as distribution by area or job type, if available) and inference to gathering CompanyGroupID.

**(3) What is your hypothesis about the analysis outcome?**

I hypothesize that:

1. **Job titles and companies** are strong indicators of salary range. (This expectation is based on the understanding that salary levels vary significantly across industries and roles - for example, IT and Mining jobs generally offer higher compensation compared to Retail or Customer service.
2. By analyzing **Requirement** fields in connection with job titles, we may identify expected qualifications for certain roles or company preferences.
3. **~~Location-based trends~~** ~~may still be observable in a general sense (e.g., percentage of listings per region), however I avoid linking it to salary or job title due to a huge missing data. (Optional)~~