



## **CC5067NI-Smart Data Discovery**

**60% Individual Coursework**

**2023-24 Spring**

**Student Name: Asmi Bajracharya**

**London Met ID: 22068053**

**College ID: np01cp4a220363**

**Assignment Due Date: Monday, May 13, 2024**

**Assignment Submission Date: Sunday, May 12, 2024**

**Word Count: 3057**

*I confirm that I understand my coursework needs to be submitted online via MySecondTeacher under the relevant module page before the deadline in order for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a marks of zero will be awarded.*

## **Acknowledgements**

I would like to express my heartfelt gratitude to Islington college for giving me this opportunity to learn about this course. I would also like to express my gratitude to our lecturer, Mr. Dipeshor Silwal, for helping me understand this course as well as the coursework.

I would also like to thank our teacher, Mr. Alish KC, for guiding and supporting me throughout this coursework. His dedication towards teaching and his willingness to answer questions made the learning experience more valuable.

I would also like to thank everyone who helped me complete this coursework on time. A special thanks to all the lecturers and teachers for not just helping me for this coursework but also making this course of valuable learning experience.

Thank you,

Asmi Bajracharya.

**Abstract**

This coursework presents the analysis of data science salaries. It covers various topics of data analysis and uses Python as the base programming language. This coursework consists of a huge data set of data science salaries which contains information such as job title, residence, salary, experience level, work year and more. In this coursework, we are required to understand the data and analyze it in order to find interesting patterns or trends. By using Python as the main language and libraries such as pandas and matplotlib we were able to not just analyze data but also create bar graphs, histograms, and boxplots to visualize data.

## Table of Contents

Introduction .....	1
Aims.....	1
Objectives.....	1
1.    Data understanding.....	2
2.    Data Preparation.....	4
2.1.    Write a python program to load data into pandas DataFrame. ....	4
2.2.    Write a python program to remove unnecessary columns i.e., salary and salary currency.....	6
2.3.    Write a python program to remove the NaN missing values from updated dataframe.....	8
.....	8
2.4.    Write a python program to check duplicates value in the dataframe. ....	9
2.5.    Write a python program to see the unique values from all the columns in the dataframe.....	11
2.6.    Rename the experience level columns as below.....	13
3.    Data Analysis .....	15
.....	15
3.1.    Write a Python program to show summary statistics of sum, mean, standard deviation, skewness, and kurtosis of any chosen variable. ....	19
3.2.    Write a Python program to calculate and show correlation of all variables.	21
4.    Data Exploration.....	22
4.1.    Write a python program to find out top 15 jobs. Make a bar graph of sales as well.	22
4.2.    Which job has the highest salaries? Illustrate with bar graph. ....	25

4.3. Write a python program to find out salaries based on experience level. Illustrate it through bar graph. ....	27
4.4. Write a Python program to show histogram and box plot of any chosen different variables. Use proper labels in the graph. ....	29
Conclusion .....	31
References.....	32

## Table of Figures

Figure 1 Importing pandas and loading data into pandas dataframe. ....	4
Figure 2 Data set (df) .....	4
Figure 3 Dataframe (df) continued. ....	5
Figure 4 Dataframe before removing unnecessary columns. ....	6
Figure 5 Dataframe before removing unnecessary columns continued.....	6
Figure 6 Dataframe after removing unnecessary columns. ....	7
Figure 7 Dataframe after removing unnecessary columns continued. ....	7
Figure 8 Removing the NaN missing values from updated dataframe and checking the dataframe. ....	8
Figure 9 Dataframe continued. ....	8
Figure 10 Checking duplicates in the dataframe. ....	9
Figure 11 Duplicates in the dataframe continued. ....	9
Figure 12 Duplicates in the dataframe continued. ....	10
Figure 13 checking unique values from all the columns in the dataframe. ....	11
Figure 14 Unique values from all the columns in the dataframe continued. ....	11
Figure 15 Unique values from all the columns in the dataframe continued. ....	12
Figure 16 Renaming the experience level column. ....	13
Figure 17 Checking if the values were renamed. ....	14
Figure 18 Checking if the values were renamed. ....	14
Figure 19 Removing the duplicate values. ....	15
Figure 20 Checking the dataframe. ....	16
Figure 21 Removing data inconsistencies.....	16
Figure 22 Checking if all the data inconsistencies are removed.....	18
Figure 23 Python program to show the sum of salary in USD and mean of salary in USD.....	19
Figure 24 Python program to show the standard deviation, skewness, and kurtosis of salary in USD. ....	20
Figure 25 Python program to calculate and show correlation of all variables.....	21
Figure 26 Python program to find out top 15 jobs. ....	22

Figure 27 importing matplotlib.pyplot.....	23
Figure 28 Bar graph of top 15 jobs. ....	23
Figure 29 Bar graph of top 15 jobs x axis. ....	24
Figure 30 Python program to find out the highest salaries. ....	25
Figure 31 Python program to make the bar graph for top five highest paid salaries. ....	25
Figure 32 Bar graph of top 5 highest paid salaries. ....	26
Figure 33 python program to find out salaries based on experience level.....	27
Figure 34 Bar graph of mean salary by experience level. ....	28
Figure 35 Python program to show histogram of salary_in_usd.....	29
Figure 36 Histogram of salary_in_usd.....	29
Figure 37 Box plot for salary_in_usd.....	30
Figure 38 Box Plot of salary_in_usd.....	30

**Table of table**

Table 1 Data set information table. ....	3
--	---



## Introduction

This is our first ever coursework of Smart Data Discovery. In this coursework, we are supposed to analyse the data of the data set which is given to us. The name of the data set is DataScienceSalaries which contains different information about the salaries of individuals that are involved in this field and factors that influence their salaries. We are going to analyse the dataset and understand the data given to us.

## Aims

This coursework aims to analyse the salaries of individual working in data science using Python and the DataScienceSalaries dataset.

## Objectives

The objective of this coursework is to:

- Analyse the dataset.
- Fix data inconsistencies and duplicates.
- Make different charts and graphs to understand the data better.
- Find out correlations between data and
- Summarize key statistics of the dataset.

## 1. Data understanding

For this coursework, we were given a dataset, DataScienceSalaries, which contains the data of salaries in the fields of data science. A dataset is a collection of data that is generally related to each other and is typically in a systematic format (Sheldon, 2024). Data sets are used for various purposes such as data analysis, forecasting, building AI, etc.

Dataset for this project contains various information such as work year, experience level, employment type, job title, salary, salary currency, salary in USD, employee residence, remote work ratio, company location, and company size. All these factors influence the salary levels. This dataset contains 3755 rows and 11 columns.

Having 3755 rows and 11 columns means that the data set has a large amount of data. This leaves room for duplicate data and data inconsistencies. While checking all the data, the dataset does not seem perfect and seems to contain duplicates and data inconsistencies and a few unnecessary rows.

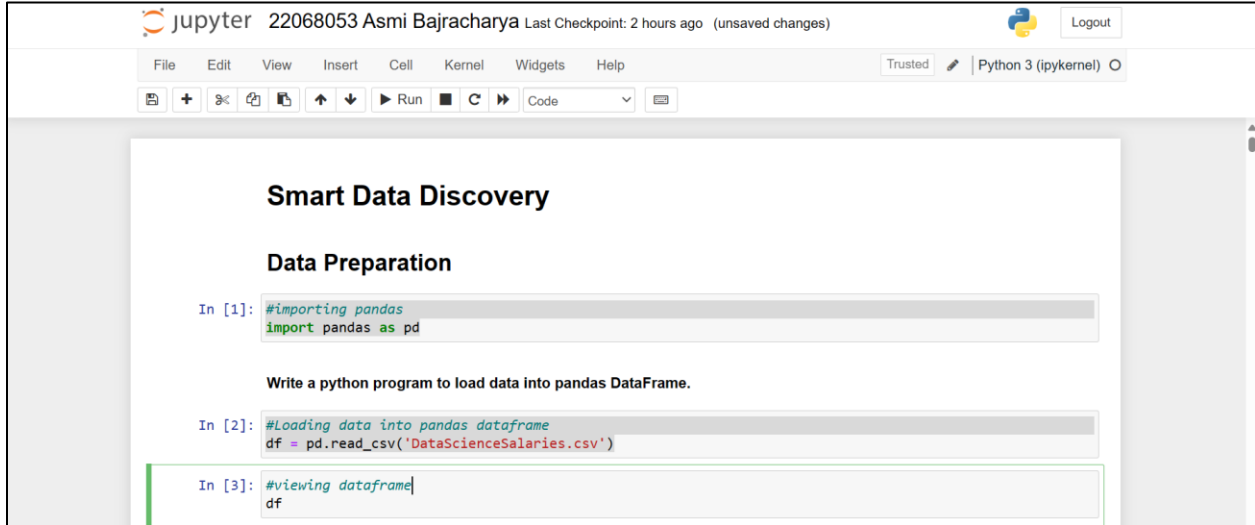
Here is the table which summarizes the dataset:

S. No	Column Name	Description	Data Type
1	work_year	This column contains the year in which data was recorded.	Integer
2	experience_level	This column contains the experience level of each individual. For example, Entry level (EN), etc.	String
3	employment_type	This column contains the type of employment of an individual. For example, full time (FT), or part time (PT) etc.	String
4	job_title	This column contains the name of the job. For example, Data Analyst, etc.	String
5	salary	This column contains the salaries of individuals.	Integer
6	salary_currency	This column contains the currency of the salaries. For example, EUR, USD, etc.	String
7	salary_in_usd	This column contains the salaries in USD.	Integer
8	employee_residence	This column contains the location of the employee's residence. For example, US, CA, etc.	String
9	remote_ratio	This column contains the ratio of work done remotely compared to onsite.	Integer
10	company_location	This column contains the location of the company.	String
11	company_size	This column contains the size of the company. For example, S for small, Medium (M), or Large (L).	String

*Table 1 Data set information table.*

## 2. Data Preparation

### 2.1. Write a python program to load data into pandas DataFrame.



The image shows a Jupyter Notebook interface with the title 'Smart Data Discovery'. The notebook contains three code cells. The first cell imports pandas as 'pd'. The second cell loads a CSV file named 'DataScienceSalaries.csv' into a DataFrame named 'df'. The third cell is partially visible, showing the start of a command to view the DataFrame.

```

In [1]: #importing pandas
import pandas as pd

Write a python program to load data into pandas DataFrame.

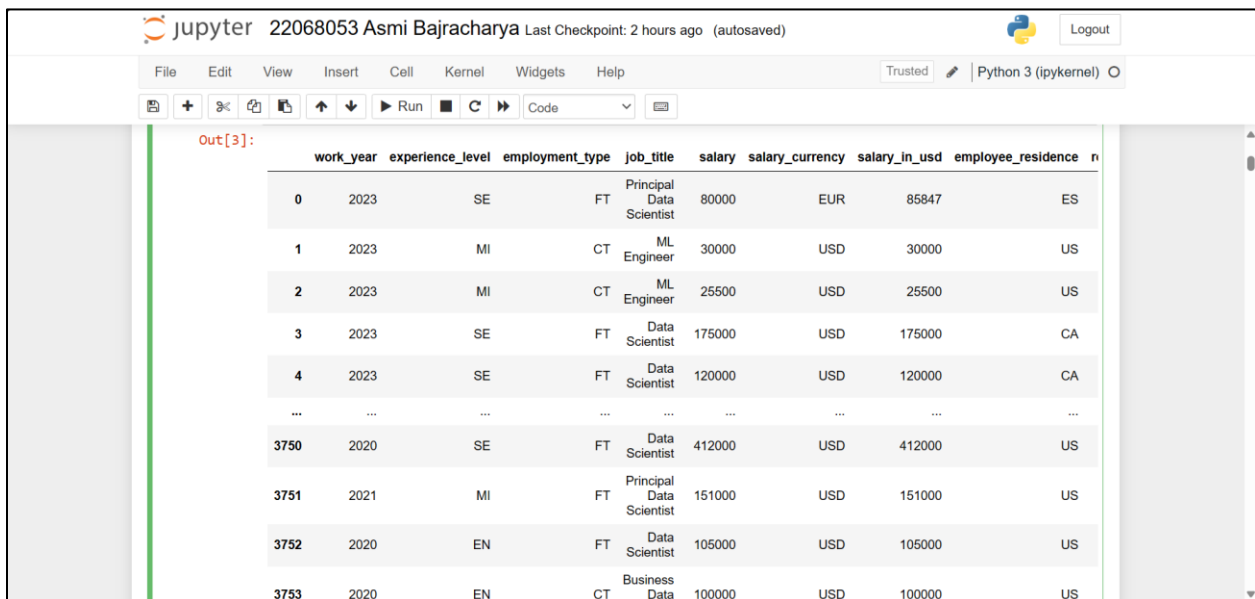
In [2]: #Loading data into pandas dataframe
df = pd.read_csv('DataScienceSalaries.csv')

In [3]: #viewing dataframe
df

```

Figure 1 Importing pandas and loading data into pandas dataframe.

Here, the first step is to import pandas before loading it into a dataframe. The data set is loaded from a CVS file to a pandas dataframe and is stored in a variable, df. Now, the next step is to view the dataframe.

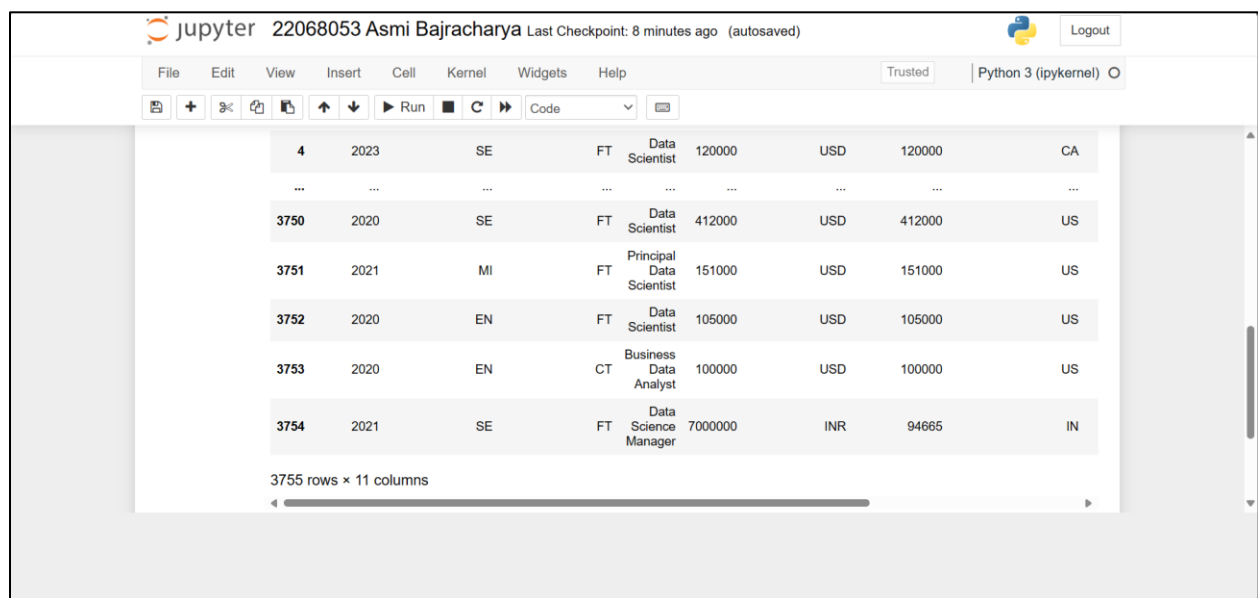


The image shows the output of the third code cell in the Jupyter Notebook. It displays a table with columns: work\_year, experience\_level, employment\_type, job\_title, salary, salary\_currency, salary\_in\_usd, employee\_residence, and n. The table contains 10 rows of data, with some rows truncated by ellipses.

	work_year	experience_level	employment_type	job_title	salary	salary_currency	salary_in_usd	employee_residence	n
0	2023	SE	FT	Principal Data Scientist	80000	EUR	85847	ES	
1	2023	MI	CT	ML Engineer	30000	USD	30000	US	
2	2023	MI	CT	ML Engineer	25500	USD	25500	US	
3	2023	SE	FT	Data Scientist	175000	USD	175000	CA	
4	2023	SE	FT	Data Scientist	120000	USD	120000	CA	
...	...	...	...	...	...	...	...	...	
3750	2020	SE	FT	Data Scientist	412000	USD	412000	US	
3751	2021	MI	FT	Principal Data Scientist	151000	USD	151000	US	
3752	2020	EN	FT	Data Scientist	105000	USD	105000	US	
3753	2020	EN	CT	Business Data	100000	USD	100000	US	

Figure 2 Data set (df)

As we can see in the figure above, this dataframe contains various columns and multiple rows containing different types of data. The dataframe consists of columns which includes work year, experience level, employment type, job title, salary, salary currency, salary in USD, employee residence, remote work ratio, company location, and company size.



	year	experience level	employment type	job title	salary	salary currency	salary in USD	employee residence	remote work ratio	company location
4	2023	SE	FT	Data Scientist	120000	USD	120000	CA		
...	...	...	...	...	...	...	...	...	...	...
3750	2020	SE	FT	Data Scientist	412000	USD	412000	US		
3751	2021	MI	FT	Principal Data Scientist	151000	USD	151000	US		
3752	2020	EN	FT	Data Scientist	105000	USD	105000	US		
3753	2020	EN	CT	Business Data Analyst	100000	USD	100000	US		
3754	2021	SE	FT	Data Science Manager	7000000	INR	94665	IN		

3755 rows × 11 columns

*Figure 3 Dataframe (df) continued.*

In the above figure, we can see that there are 3755 rows and 11 columns. This is the entire dataframe of DataScienceSalaries. Here, the output consists of a data frame which contains various data of the salaries of individuals and various factors that influence the salary.

## 2.2. Write a python program to remove unnecessary columns i.e., salary and salary currency.

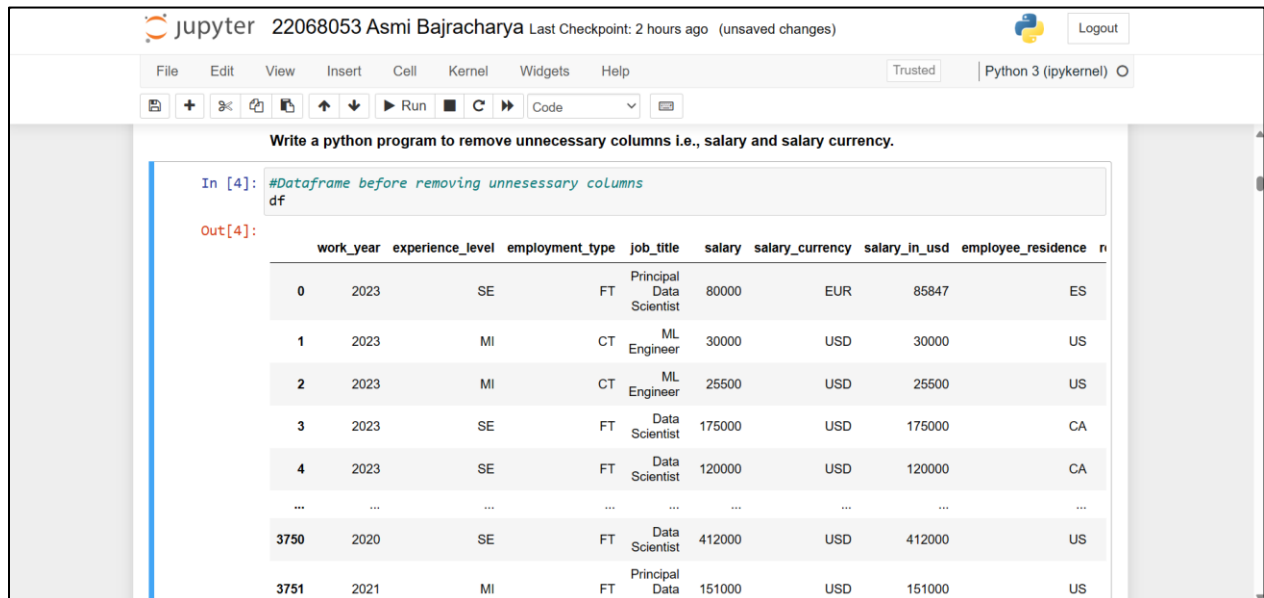


Figure 4 Dataframe before removing unnecessary columns.

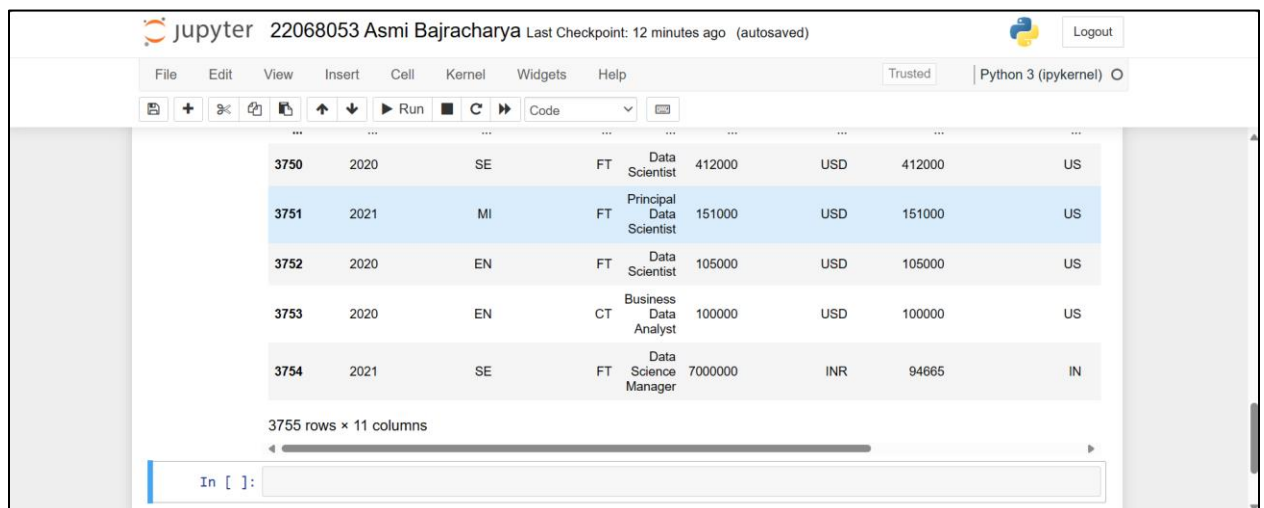


Figure 5 Dataframe before removing unnecessary columns continued.

As we can see in the above figure, the same data is repeating in three different columns. Salary and salary currency is not required as a column since there already exists a column salary\_in\_usd which contains the salary and currency.

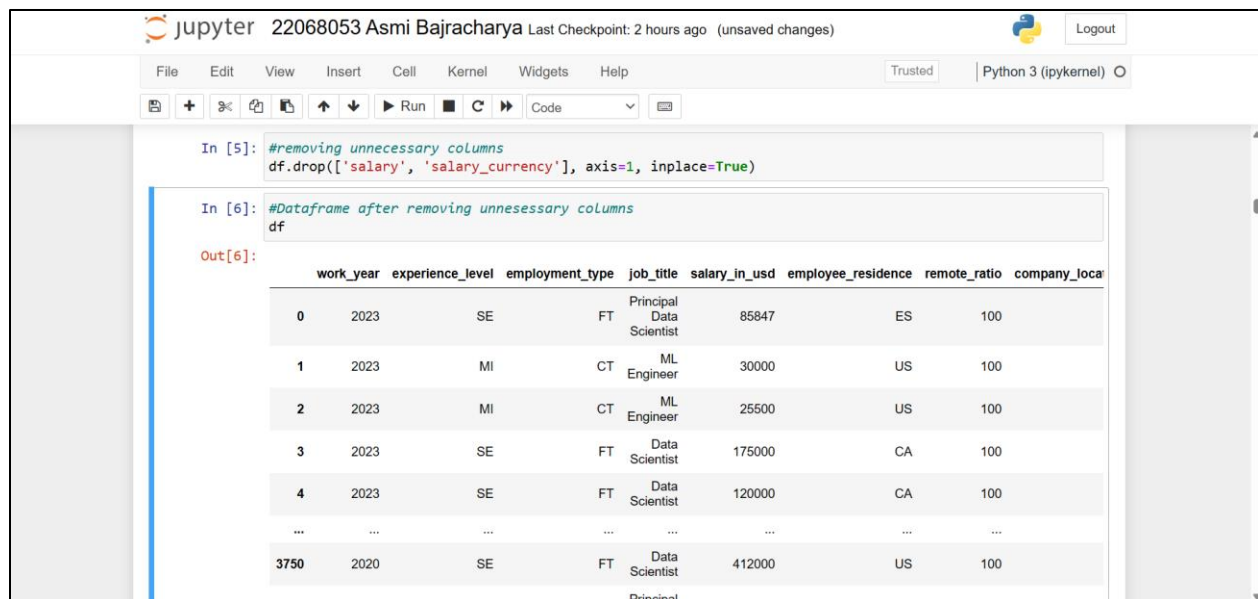


Figure 6 Dataframe after removing unnecessary columns.

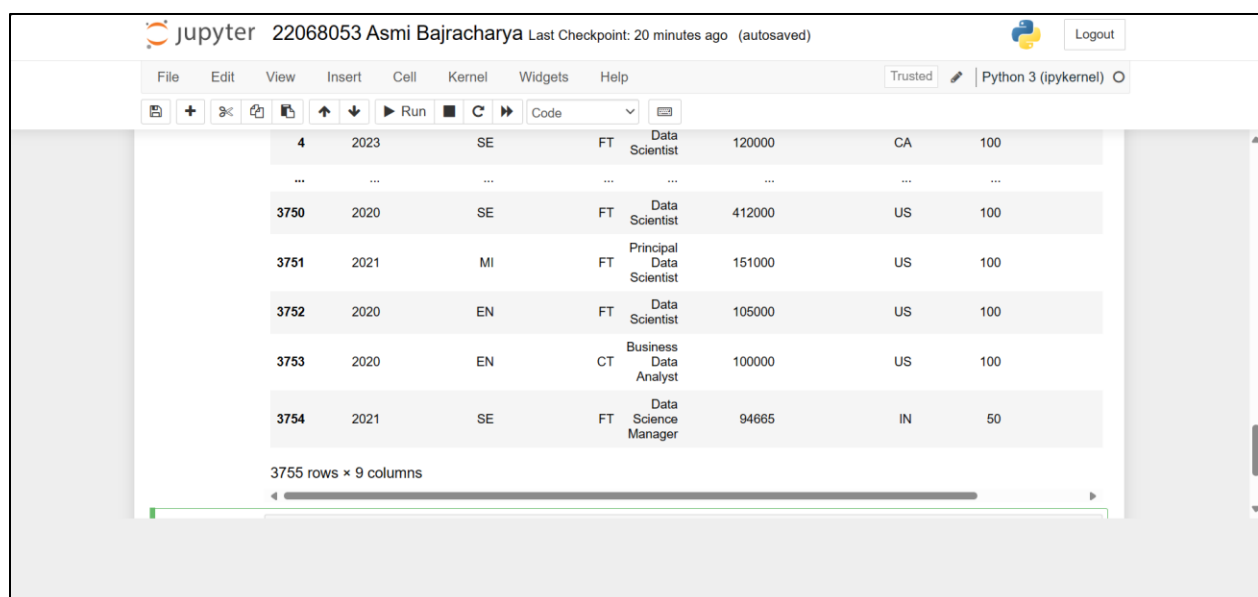
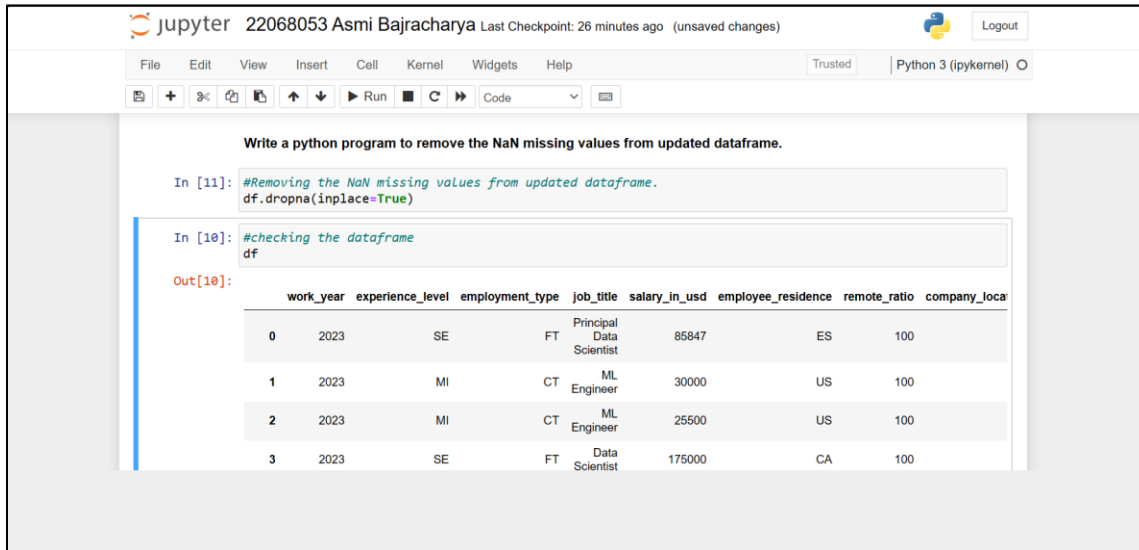


Figure 7 Dataframe after removing unnecessary columns continued.

The unnecessary columns are removed by using the drop function which has removed the columns salary and salary currency from the data set. Here, in the code axis=1, refers to the column and if the column is salary or salary currency then the flag becomes true, and the column is dropped. Finally, the unnecessary columns are dropped and storage is saved.

### 2.3. Write a python program to remove the NaN missing values from updated dataframe.



The screenshot shows a Jupyter Notebook interface with the following content:

```

Write a python program to remove the NaN missing values from updated dataframe.

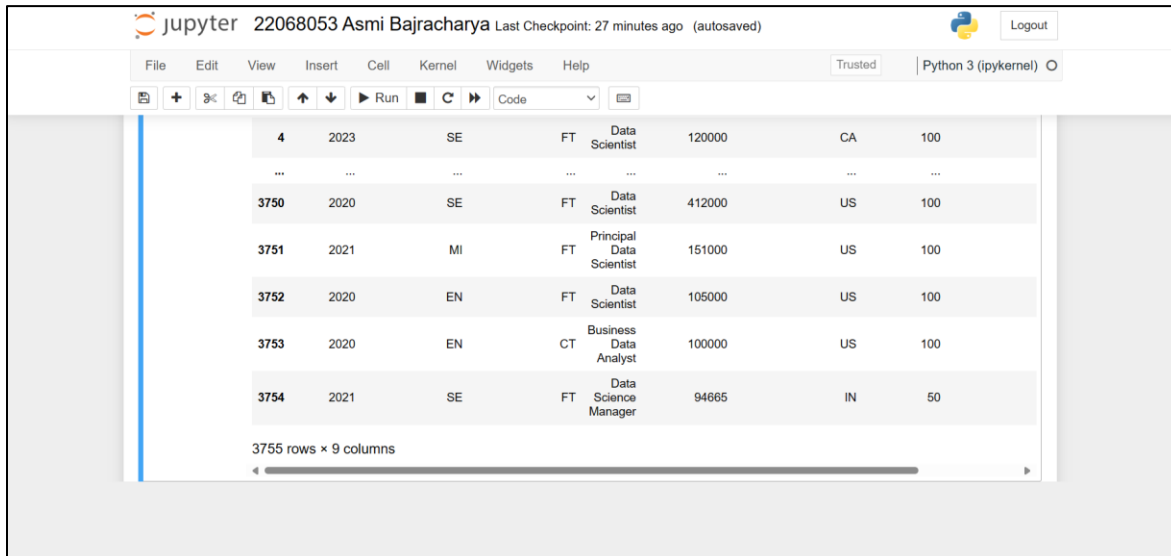
In [11]: #Removing the NaN missing values from updated dataframe.
df.dropna(inplace=True)

In [10]: #checking the dataframe
df
Out[10]:

```

	work_year	experience_level	employment_type	job_title	salary_in_usd	employee_residence	remote_ratio	company_location
0	2023	SE	FT	Principal Data Scientist	85847	ES	100	
1	2023	MI	CT	ML Engineer	30000	US	100	
2	2023	MI	CT	ML Engineer	25500	US	100	
3	2023	SE	FT	Data Scientist	175000	CA	100	

Figure 8 Removing the NaN missing values from updated dataframe and checking the dataframe.



The screenshot shows a Jupyter Notebook interface displaying a continuation of the dataframe. The output shows rows 4 through 3754, with a total of 3755 rows and 9 columns. The data is as follows:

	work_year	experience_level	employment_type	job_title	salary_in_usd	employee_residence	remote_ratio	company_location
4	2023	SE	FT	Data Scientist	120000	CA	100	
...	...	...	...	...	...	...	...	...
3750	2020	SE	FT	Data Scientist	412000	US	100	
3751	2021	MI	FT	Principal Data Scientist	151000	US	100	
3752	2020	EN	FT	Data Scientist	105000	US	100	
3753	2020	EN	CT	Business Data Analyst	100000	US	100	
3754	2021	SE	FT	Data Science Manager	94665	IN	50	

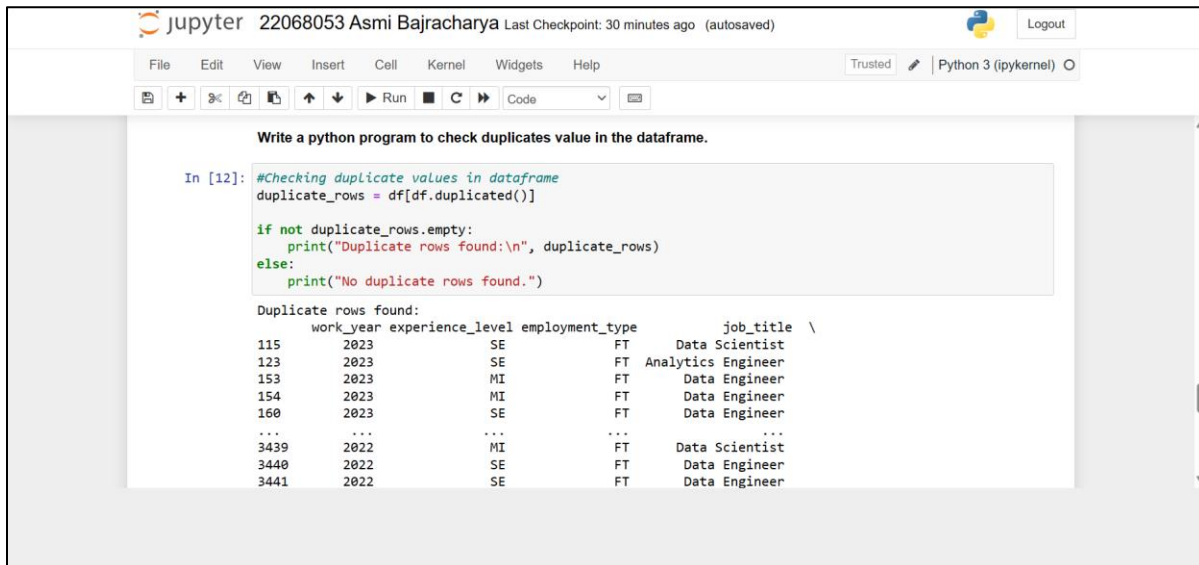
3755 rows x 9 columns

Figure 9 Dataframe continued.

Here, to remove the NaN missing values, we have used the `dropna()` function. However, this data frame does not seem to have any missing values. Hence, there are no NaN missing values in this data frame.



## 2.4. Write a python program to check duplicates value in the dataframe.



```

Write a python program to check duplicates value in the dataframe.

In [12]: #Checking duplicate values in dataframe
duplicate_rows = df[df.duplicated()]

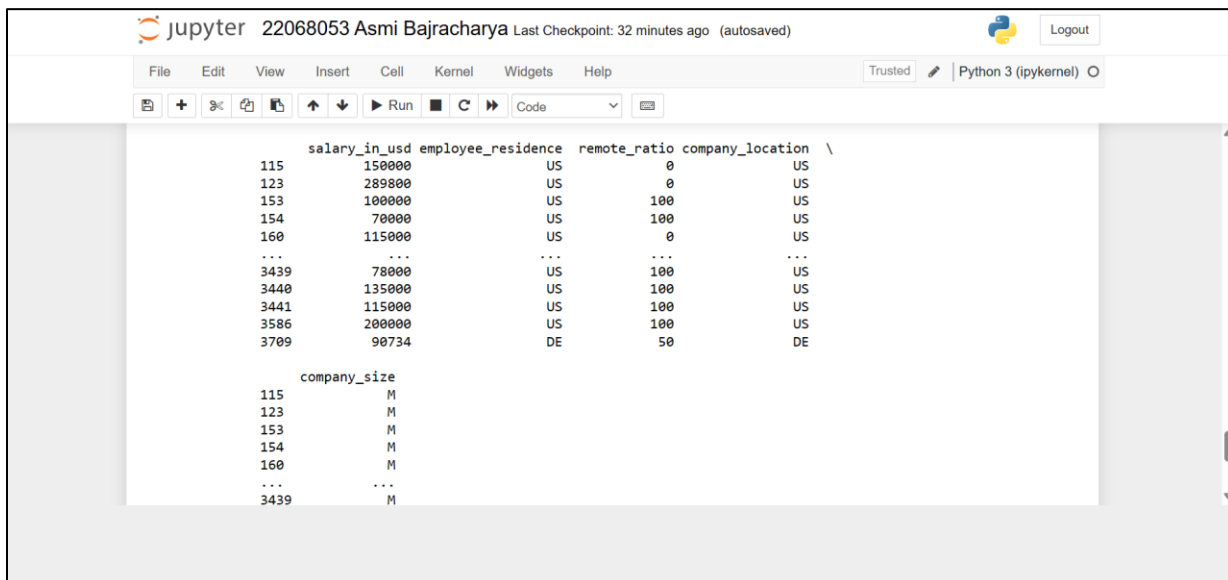
if not duplicate_rows.empty:
    print("Duplicate rows found:\n", duplicate_rows)
else:
    print("No duplicate rows found.")

Duplicate rows found:
  work_year  experience_level  employment_type  job_title \
115      2023                SE                FT  Data Scientist
123      2023                SE                FT  Analytics Engineer
153      2023                MI                FT  Data Engineer
154      2023                MI                FT  Data Engineer
160      2023                SE                FT  Data Engineer
...      ...                ...                ...
3439     2022                MI                FT  Data Scientist
3440     2022                SE                FT  Data Engineer
3441     2022                SE                FT  Data Engineer

```

Figure 10 Checking duplicates in the dataframe.

Here, we are checking the duplicate values in the data frame. We have used the function duplicated to check the duplicate values. Since, there are a lot of duplicate rows found, the output displays all the duplicate rows found in the data frame.



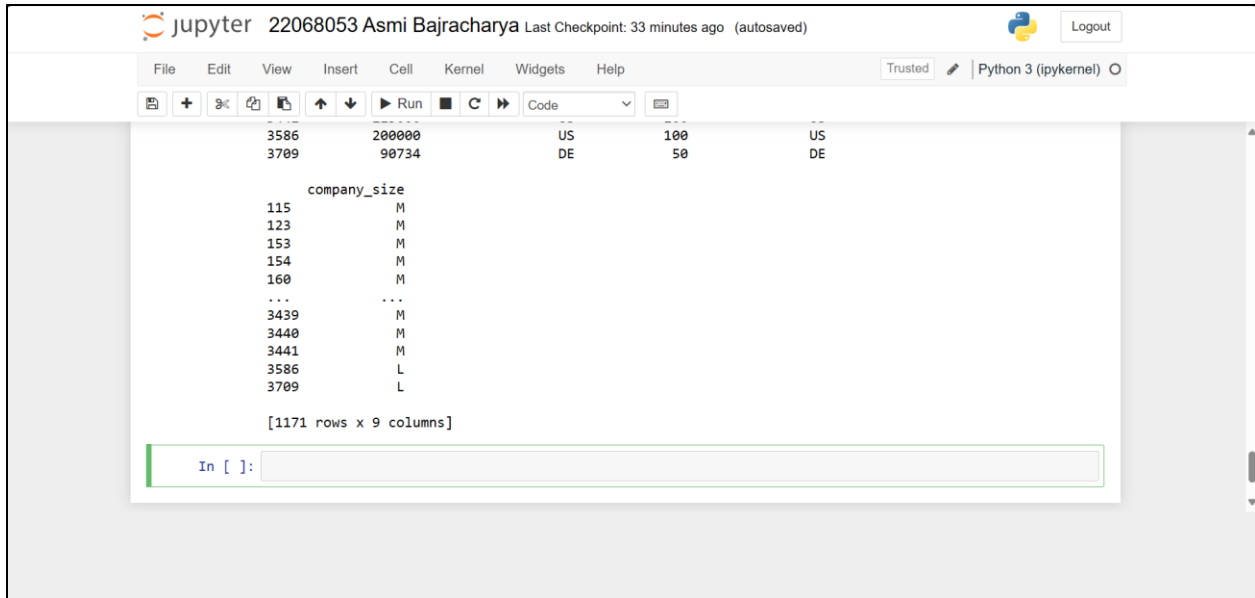
```

salary_in_usd  employee_residence  remote_ratio  company_location \
115      150000                US                0                US
123      289000                US                0                US
153      100000                US               100                US
154       70000                US               100                US
160      115000                US                0                US
...      ...                ...                ...
3439       78000                US               100                US
3440      135000                US               100                US
3441      115000                US               100                US
3586      200000                US               100                US
3709       90734                DE                50                DE

company_size
115      M
123      M
153      M
154      M
160      M
...      ...
3439     M

```

Figure 11 Duplicates in the dataframe continued.



A screenshot of a Jupyter Notebook interface. The top bar shows the Jupyter logo, the username '22068053 Asmi Bajracharya', and the last checkpoint time 'Last Checkpoint: 33 minutes ago (autosaved)'. The notebook is running on 'Python 3 (ipykernel)'. The code cell contains a DataFrame with 9 columns. The first two columns contain numerical values, and the remaining seven columns contain categorical data. The output shows a subset of the data, highlighting duplicate rows. The output is summarized as '[1171 rows x 9 columns]'.

```
3586      200000      US      100      US
3709      90734      DE       50      DE

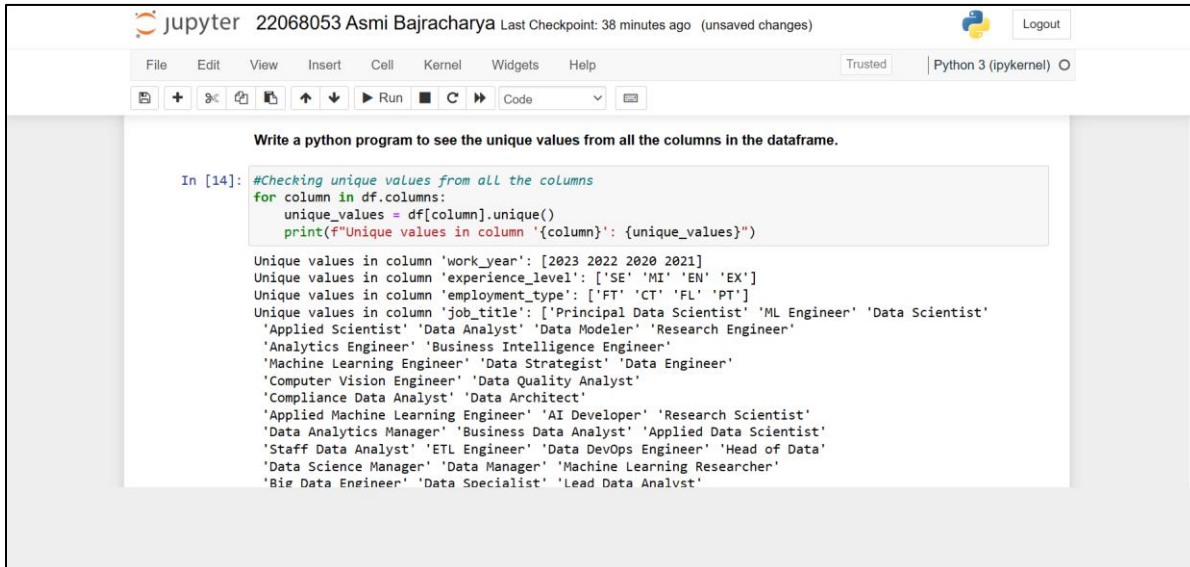
company_size
115      M
123      M
153      M
154      M
160      M
...
3439     M
3440     M
3441     M
3586     L
3709     L

[1171 rows x 9 columns]
```

Figure 12 Duplicates in the dataframe continued.

There are a total of 1171 duplicate rows found across 9 columns in this dataframe.

## 2.5. Write a python program to see the unique values from all the columns in the dataframe.



The screenshot shows a Jupyter Notebook interface with the following content:

```

jupyter 22068053 Asmi Bajracharya Last Checkpoint: 38 minutes ago (unsaved changes)
File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

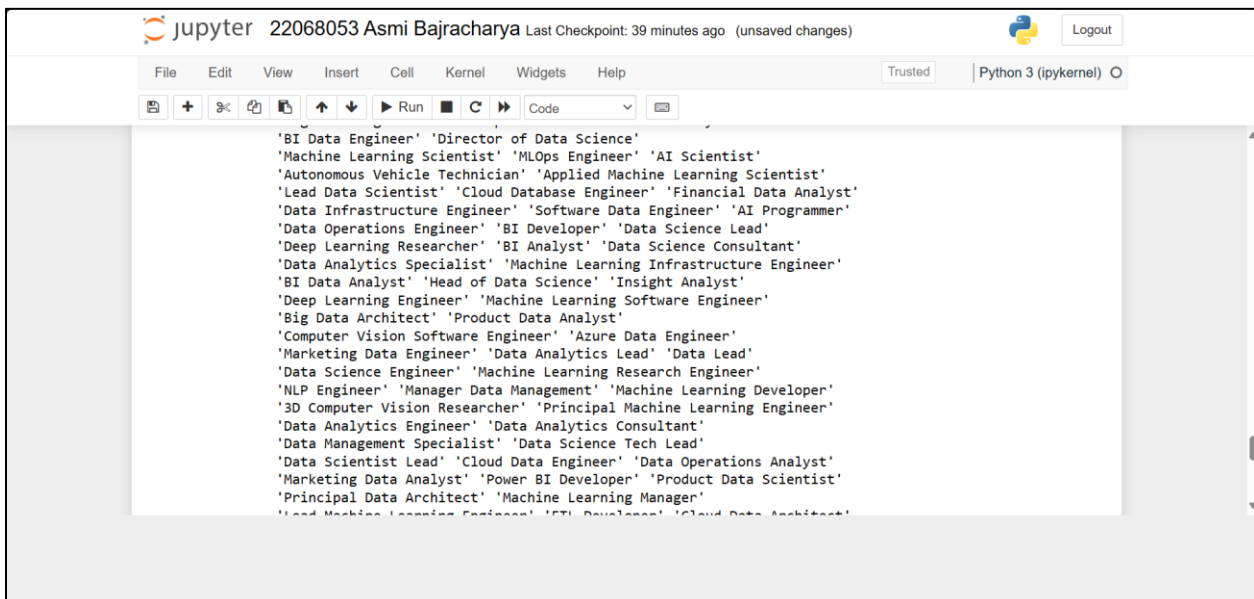
Write a python program to see the unique values from all the columns in the dataframe.

In [14]: #Checking unique values from all the columns
for column in df.columns:
    unique_values = df[column].unique()
    print(f"Unique values in column '{column}': {unique_values}")

Unique values in column 'work_year': [2023 2022 2020 2021]
Unique values in column 'experience_level': ['SE' 'MI' 'EN' 'EX']
Unique values in column 'employment_type': ['FT' 'CT' 'FL' 'PT']
Unique values in column 'job_title': ['Principal Data Scientist' 'ML Engineer' 'Data Scientist'
'Applied Scientist' 'Data Analyst' 'Data Modeler' 'Research Engineer'
'Analytics Engineer' 'Business Intelligence Engineer'
'Machine Learning Engineer' 'Data Strategist' 'Data Engineer'
'Computer Vision Engineer' 'Data Quality Analyst'
'Compliance Data Analyst' 'Data Architect'
'Applied Machine Learning Engineer' 'AI Developer' 'Research Scientist'
'Data Analytics Manager' 'Business Data Analyst' 'Applied Data Scientist'
'Staff Data Analyst' 'ETL Engineer' 'Data DevOps Engineer' 'Head of Data'
'Data Science Manager' 'Data Manager' 'Machine Learning Researcher'
'Big Data Engineer' 'Data Specialist' 'Lead Data Analyst']
  
```

Figure 13 checking unique values from all the columns in the dataframe.

Here we use the function, `unique()`, to check out all the unique values from all the columns in the data frame. We found out a lot of unique values from each column however, we can see there are a lot of data inconsistency in this dataframe.



The screenshot shows the continuation of the unique values from the previous figure:

```

'BI Data Engineer' 'Director of Data Science'
'Machine Learning Scientist' 'MLOps Engineer' 'AI Scientist'
'Autonomous Vehicle Technician' 'Applied Machine Learning Scientist'
'Lead Data Scientist' 'Cloud Database Engineer' 'Financial Data Analyst'
'Data Infrastructure Engineer' 'Software Data Engineer' 'AI Programmer'
'Data Operations Engineer' 'BI Developer' 'Data Science Lead'
'Deep Learning Researcher' 'BI Analyst' 'Data Science Consultant'
'Data Analytics Specialist' 'Machine Learning Infrastructure Engineer'
'BI Data Analyst' 'Head of Data Science' 'Insight Analyst'
'Deep Learning Engineer' 'Machine Learning Software Engineer'
'Big Data Architect' 'Product Data Analyst'
'Computer Vision Software Engineer' 'Azure Data Engineer'
'Marketing Data Engineer' 'Data Analytics Lead' 'Data Lead'
'Data Science Engineer' 'Machine Learning Research Engineer'
'NLP Engineer' 'Manager Data Management' 'Machine Learning Developer'
'3D Computer Vision Researcher' 'Principal Machine Learning Engineer'
'Data Analytics Engineer' 'Data Analytics Consultant'
'Data Management Specialist' 'Data Science Tech Lead'
'Data Scientist Lead' 'Cloud Data Engineer' 'Data Operations Analyst'
'Marketing Data Analyst' 'Power BI Developer' 'Product Data Scientist'
'Principal Data Architect' 'Machine Learning Manager'
'Lead Machine Learning Engineer' 'ETL Developer' 'Cloud Data Architect'
  
```

Figure 14 Unique values from all the columns in the dataframe continued.

The screenshot shows a Jupyter Notebook window with the title '22068053 Asmi Bajracharya'. The interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help) and a toolbar with icons for file operations, running cells, and code execution. The notebook is saved and trusted, and the kernel is Python 3 (ipykernel). The code cell contains the following text:

```
'Lead Data Engineer' 'Head of Machine Learning' 'Principal Data Analyst'
'Principal Data Engineer' 'Staff Data Scientist' 'Finance Data Analyst']
Unique values in column 'salary_in_usd': [ 85847 30000 25500 ... 28369 412000 94665]
Unique values in column 'employee_residence': ['ES' 'US' 'CA' 'DE' 'GB' 'NG' 'IN' 'HK' 'PT' 'NL' 'CH'
'CF' 'FR' 'AU'
'FI' 'UA' 'IE' 'IL' 'GH' 'AT' 'CO' 'SG' 'SE' 'SI' 'MX' 'UZ' 'BR' 'TH'
'HR' 'PL' 'KW' 'VN' 'CY' 'AR' 'AM' 'BA' 'KE' 'GR' 'MK' 'LV' 'RO' 'PK'
'IT' 'MA' 'LT' 'BE' 'AS' 'IR' 'HU' 'SK' 'CN' 'CZ' 'CR' 'TR' 'CL' 'PR'
'DK' 'BO' 'PH' 'DO' 'EG' 'ID' 'AE' 'MY' 'JP' 'EE' 'HN' 'TN' 'RU' 'DZ'
'IQ' 'BG' 'JE' 'RS' 'NZ' 'MD' 'LU' 'MT']
Unique values in column 'remote_ratio': [100 0 50]
Unique values in column 'company_location': ['ES' 'US' 'CA' 'DE' 'GB' 'NG' 'IN' 'HK' 'NL' 'CH' 'CF'
'FR' 'FI' 'UA'
'IE' 'IL' 'GH' 'CO' 'SG' 'AU' 'SE' 'SI' 'MX' 'BR' 'PT' 'RU' 'TH' 'HR'
'VN' 'EE' 'AM' 'BA' 'KE' 'GR' 'MK' 'LV' 'RO' 'PK' 'IT' 'MA' 'PL' 'AL'
'AR' 'LT' 'AS' 'CR' 'IR' 'BS' 'HU' 'AT' 'SK' 'CZ' 'TR' 'PR' 'DK' 'BO'
'PH' 'BE' 'ID' 'EG' 'AE' 'LU' 'MY' 'HN' 'JP' 'DZ' 'IQ' 'CN' 'NZ' 'CL'
'MD' 'MT']
Unique values in column 'company_size': ['L' 'S' 'M']
```

The input prompt 'In [ ]:' is visible at the bottom of the code cell.

Figure 15 Unique values from all the columns in the dataframe continued.

There are different number of unique values in each column. As we can see in the figure above there are four unique values in the column work year, experience level and employment type. There are 93 unique values in the column job title, however, jobs are repeated and there is data inconsistency because of the spellings of the data. We need to fix it. There are 3 unique values remote ratio and 82 unique company location as well as employee residence, and finally there are three unique values in the company size.

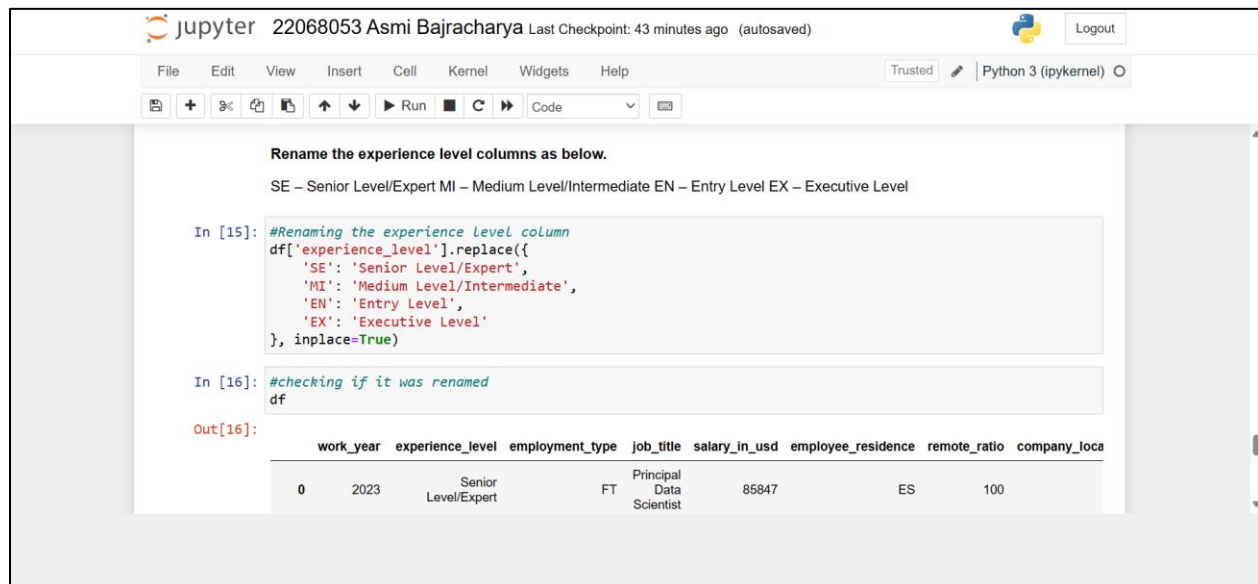
## 2.6. Rename the experience level columns as below.

**SE – Senior Level/Expert**

**MI – Medium Level/Intermediate**

**EN – Entry Level**

**EX – Executive Level**



The screenshot shows a Jupyter Notebook interface with the following content:

```

Rename the experience level columns as below.
SE – Senior Level/Expert MI – Medium Level/Intermediate EN – Entry Level EX – Executive Level

In [15]: #Renaming the experience level column
df['experience_level'].replace({
    'SE': 'Senior Level/Expert',
    'MI': 'Medium Level/Intermediate',
    'EN': 'Entry Level',
    'EX': 'Executive Level'
}, inplace=True)

In [16]: #checking if it was renamed
df
Out[16]:

```

	work_year	experience_level	employment_type	job_title	salary_in_usd	employee_residence	remote_ratio	company_location
0	2023	Senior Level/Expert	FT	Principal Data Scientist	85847	ES	100	

Figure 16 Renaming the experience level column.

Here, we have used the replace function to rename the values of the column. We have replaced 'SE' as senior level/expert, 'MI' as medium level/intermediate, 'EN' as entry level and 'EX' as executive level. Now to check if the values were renamed, we checked the updated dataframe.

Index	Year	Level	Experience	Job Type	Salary	Country	Count
1	2023	Medium Level/Intermediate	CT	ML Engineer	30000	US	100
2	2023	Medium Level/Intermediate	CT	ML Engineer	25500	US	100
3	2023	Senior Level/Expert	FT	Data Scientist	175000	CA	100
4	2023	Senior Level/Expert	FT	Data Scientist	120000	CA	100
...	...	...	...	...	...	...	...
3750	2020	Senior Level/Expert	FT	Data Scientist	412000	US	100
3751	2021	Medium Level/Intermediate	FT	Principal Data Scientist	151000	US	100
3752	2020	Entry Level	FT	Data Scientist	105000	US	100
3753	2020	Entry Level	CT	Business Data Analyst	100000	US	100

Figure 17 Checking if the values were renamed.

Index	Year	Level	Experience	Job Type	Salary	Country	Count
3750	2020	Senior Level/Expert	FT	Data Scientist	412000	US	100
3751	2021	Medium Level/Intermediate	FT	Principal Data Scientist	151000	US	100
3752	2020	Entry Level	FT	Data Scientist	105000	US	100
3753	2020	Entry Level	CT	Business Data Analyst	100000	US	100
3754	2021	Senior Level/Expert	FT	Data Science Manager	94665	IN	50

3755 rows x 9 columns

Figure 18 Checking if the values were renamed.

As we can see each and every value of the column is renamed correctly. All the four values have been renamed and it is easier to understand what the experience level is after the values were renamed.

### 3. Data Analysis

Before starting with the data analysis part, we must remove all the duplicate values and remove the data inconsistency so that we get the correct values for analysis.

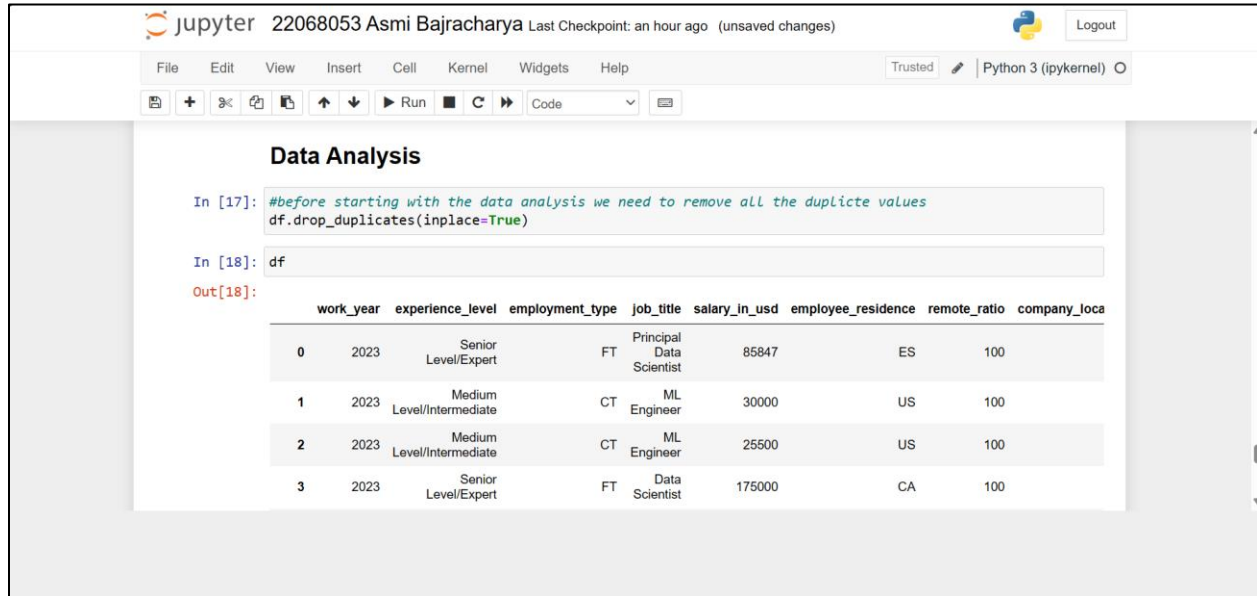
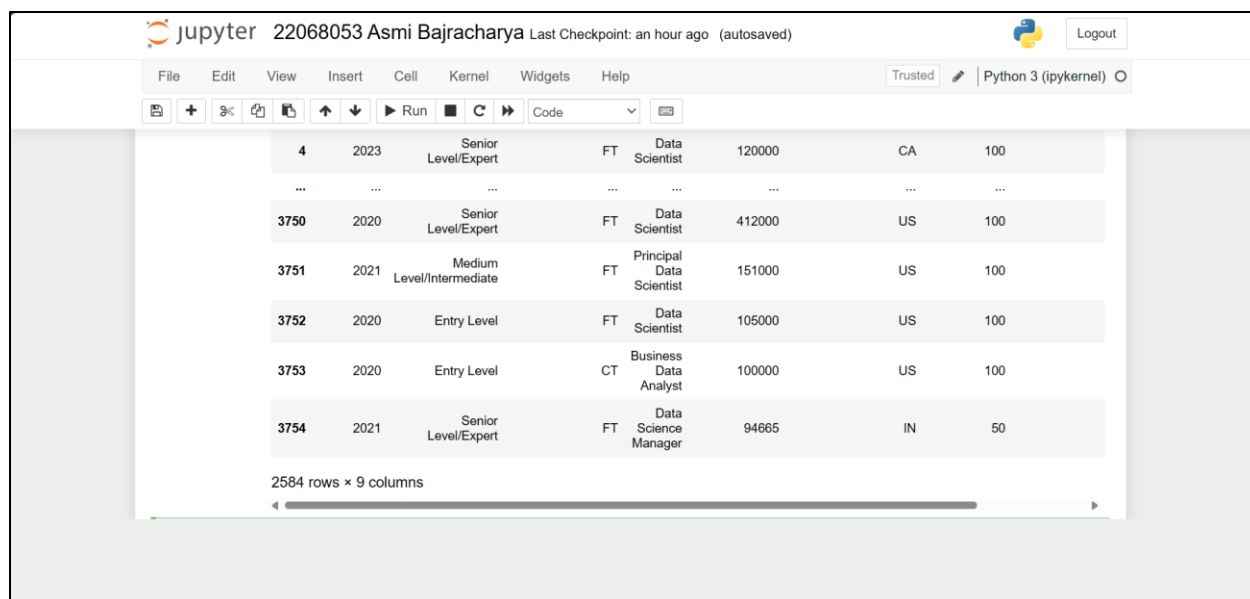


Figure 19 Removing the duplicate values.

Starting with dropping all the duplicates by using the drop function again, all the duplicates are removed. Checking the data frame to see if there are any duplicate values remaining.



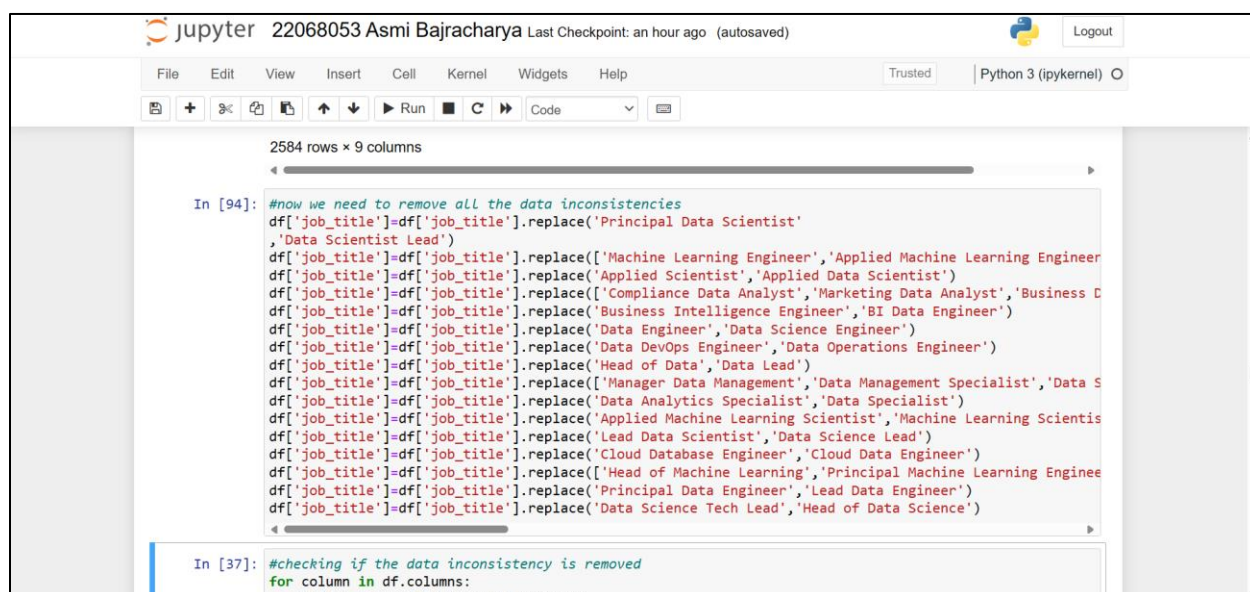
Index	Year	Level	Job Title	Type	Salary	Location	Score
4	2023	Senior Level/Expert	Data Scientist	FT	120000	CA	100
...	...	...	...	...	...	...	...
3750	2020	Senior Level/Expert	Data Scientist	FT	412000	US	100
3751	2021	Medium Level/Intermediate	Principal Data Scientist	FT	151000	US	100
3752	2020	Entry Level	Data Scientist	FT	105000	US	100
3753	2020	Entry Level	Business Data Analyst	CT	100000	US	100
3754	2021	Senior Level/Expert	Data Science Manager	FT	94665	IN	50

2584 rows x 9 columns

Figure 20 Checking the dataframe.

As we can see in the data frame level there are no duplicate data remaining.

Now, after removing all the duplicate values we must remove all the data inconsistencies also. As mentioned before, there is data inconsistency in the column, job title.



```

In [94]: #now we need to remove all the data inconsistencies
df['job_title'] = df['job_title'].replace('Principal Data Scientist',
'Data Scientist Lead')
df['job_title'] = df['job_title'].replace(['Machine Learning Engineer', 'Applied Machine Learning Engineer',
'Applied Scientist', 'Applied Data Scientist'])
df['job_title'] = df['job_title'].replace(['Compliance Data Analyst', 'Marketing Data Analyst', 'Business Data Analyst',
'Business Intelligence Engineer', 'BI Data Engineer'])
df['job_title'] = df['job_title'].replace('Data Engineer', 'Data Science Engineer')
df['job_title'] = df['job_title'].replace('Data DevOps Engineer', 'Data Operations Engineer')
df['job_title'] = df['job_title'].replace('Head of Data', 'Data Lead')
df['job_title'] = df['job_title'].replace(['Manager Data Management', 'Data Management Specialist', 'Data Science Specialist',
'Data Analytics Specialist', 'Data Specialist'])
df['job_title'] = df['job_title'].replace('Applied Machine Learning Scientist', 'Machine Learning Scientist')
df['job_title'] = df['job_title'].replace('Lead Data Scientist', 'Data Science Lead')
df['job_title'] = df['job_title'].replace('Cloud Database Engineer', 'Cloud Data Engineer')
df['job_title'] = df['job_title'].replace(['Head of Machine Learning', 'Principal Machine Learning Engineer',
'Principal Data Engineer', 'Lead Data Engineer'])
df['job_title'] = df['job_title'].replace('Data Science Tech Lead', 'Head of Data Science')

In [37]: #checking if the data inconsistency is removed
for column in df.columns:
    unique_values = df[column].unique()

```

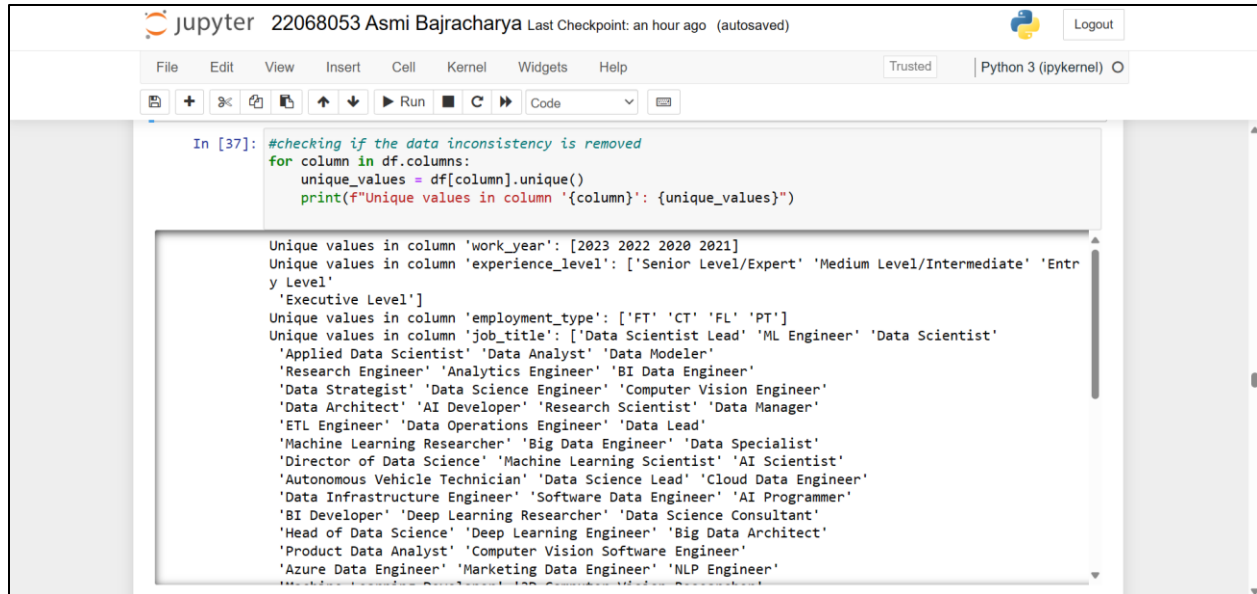
Figure 21 Removing data inconsistencies.



We have replaced similar jobs with different names into a same name. For example, principal data scientist and data scientist are both data scientists of senior level. Therefore, they can be categorized into the same job title. Similarly, all the machine learning engineers can be categorized into ML engineers. Applied scientist and applied data scientist are also similar so we have decided to categorize them as one. Business intelligence engineer and BI data engineer are both the same hence they are also categorized as one.

Similarly looking at the figure above we can see that data develop engineer and data operations engineer are quite similar, so they've been categorized as one. Head of data and data lead are also similar, so they are also categorized as one. All the data managers and data management specialists manage data hence, they are categorized as one as well. Machine learning scientists and applied machine learning scientists can also be categorized as one.

Lead data scientist and data science league sounds similar hence they're also categorized as one. All the different types of data analyst have been categorized as one for convenience. Cloud database engineer and cloud data engineer are also categorized as one. Principal data engineer and head of engineer is also categorized as one. Data science tech league and head of data science is also categorized as one.



The screenshot shows a Jupyter Notebook with the following code and output:

```
In [37]: #checking if the data inconsistency is removed
for column in df.columns:
    unique_values = df[column].unique()
    print(f"Unique values in column '{column}': {unique_values}")
```

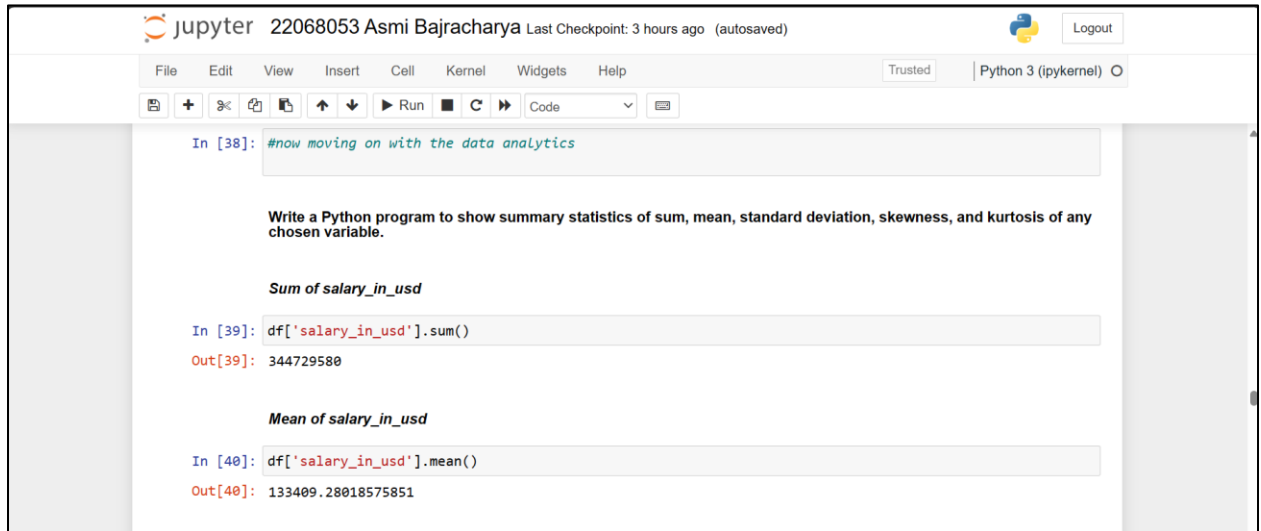
Unique values in column 'work\_year': [2023 2022 2020 2021]  
 Unique values in column 'experience\_level': ['Senior Level/Expert' 'Medium Level/Intermediate' 'Entry Level' 'Executive Level']  
 Unique values in column 'employment\_type': ['FT' 'CT' 'FL' 'PT']  
 Unique values in column 'job\_title': ['Data Scientist Lead' 'ML Engineer' 'Data Scientist' 'Applied Data Scientist' 'Data Analyst' 'Data Modeler' 'Research Engineer' 'Analytics Engineer' 'BI Data Engineer' 'Data Strategist' 'Data Science Engineer' 'Computer Vision Engineer' 'Data Architect' 'AI Developer' 'Research Scientist' 'Data Manager' 'ETL Engineer' 'Data Operations Engineer' 'Data Lead' 'Machine Learning Researcher' 'Big Data Engineer' 'Data Specialist' 'Director of Data Science' 'Machine Learning Scientist' 'AI Scientist' 'Autonomous Vehicle Technician' 'Data Science Lead' 'Cloud Data Engineer' 'Data Infrastructure Engineer' 'Software Data Engineer' 'AI Programmer' 'BI Developer' 'Deep Learning Researcher' 'Data Science Consultant' 'Head of Data Science' 'Deep Learning Engineer' 'Big Data Architect' 'Product Data Analyst' 'Computer Vision Software Engineer' 'Azure Data Engineer' 'Marketing Data Engineer' 'NLP Engineer']

Figure 22 Checking if all the data inconsistencies are removed.

After all the data inconsistencies were removed there are a total of 59 job titles that remain.

### 3.1. Write a Python program to show summary statistics of sum, mean, standard deviation, skewness, and kurtosis of any chosen variable.

Here, the chosen variable for this question is salary\_in\_usd.



The screenshot shows a Jupyter Notebook interface with the following content:

- Header: jupyter 22068053 Asmi Bajracharya Last Checkpoint: 3 hours ago (autosaved) Logout
- Menu: File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)
- Toolbar: Icons for file operations, cell navigation, and execution.
- Code Cell [38]: `#now moving on with the data analytics`
- Text: Write a Python program to show summary statistics of sum, mean, standard deviation, skewness, and kurtosis of any chosen variable.
- Section Header: **Sum of salary\_in\_usd**
- Code Cell [39]: `df['salary_in_usd'].sum()`
- Output [39]: 344729580
- Section Header: **Mean of salary\_in\_usd**
- Code Cell [40]: `df['salary_in_usd'].mean()`
- Output [40]: 133409.28018575851

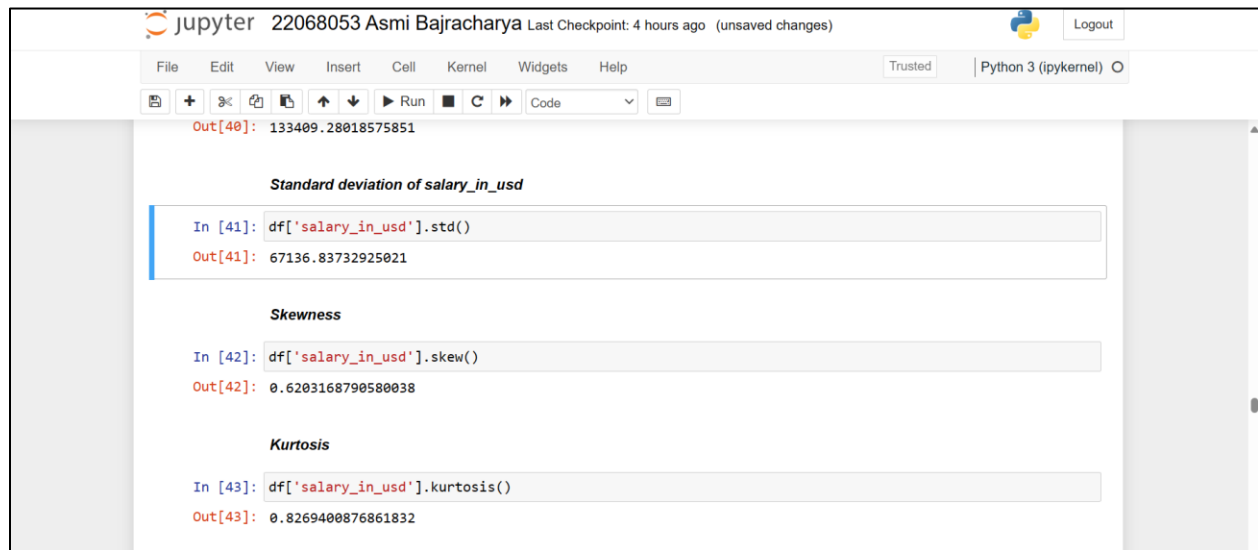
Figure 23 Python program to show the sum of salary in USD and mean of salary in USD.

#### Sum

We have used the `sum()` function to find out the sum of salary in USD of the data frame and the sum is 344729580.

#### Mean

Here, mean is the average salary and we have used the `mean()` function to find out the mean of salary in USD and the mean is 133409.28018575851.



The screenshot shows a Jupyter Notebook interface with the following content:

```
jupyter 22068053 Asmi Bajracharya Last Checkpoint: 4 hours ago (unsaved changes)
File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)
Out[40]: 133409.28018575851

Standard deviation of salary_in_usd

In [41]: df['salary_in_usd'].std()
Out[41]: 67136.83732925021

Skewness

In [42]: df['salary_in_usd'].skew()
Out[42]: 0.6203168790580038

Kurtosis

In [43]: df['salary_in_usd'].kurtosis()
Out[43]: 0.8269400876861832
```

Figure 24 Python program to show the standard deviation, skewness, and kurtosis of salary in USD.

## Standard deviation

Here we have used the `std()` function, to find out the standard deviation and the standard deviation of salary in USD is 67136.83732925021.

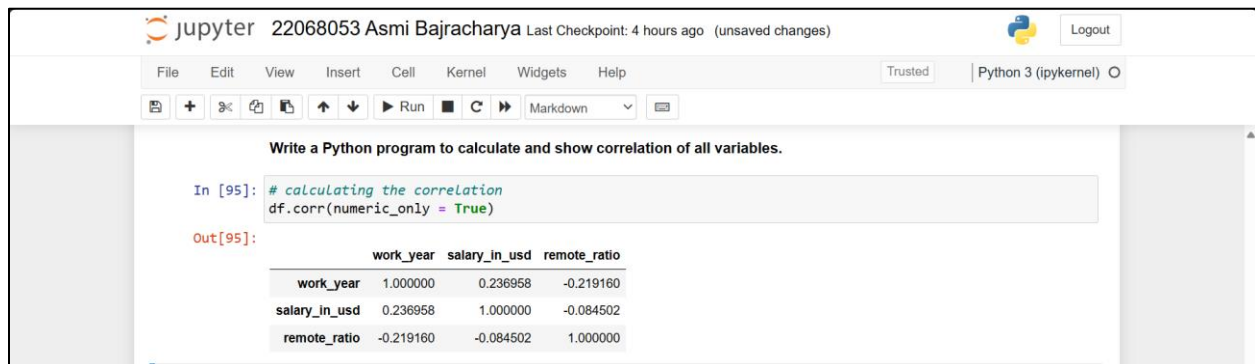
## Skewness

The measurement of asymmetry of a distribution is known as skewness (Turney, 2022). Here we have used `skew()` function to find out the skewness and the skewness is 0.6203168790580038.

## Kurtosis

The measurement of tailedness of distribution is known as Kurtosis (Turney, 2022). Here we have used the `Kurtosis()` function to find out the Kurtosis and the Kurtosis is 0.8269400876861832.

### 3.2. Write a Python program to calculate and show correlation of all variables.



```
Write a Python program to calculate and show correlation of all variables.
```

```
In [95]: # calculating the correlation
df.corr(numeric_only = True)
```

```
Out[95]:
```

	work_year	salary_in_usd	remote_ratio
work_year	1.000000	0.236958	-0.219160
salary_in_usd	0.236958	1.000000	-0.084502
remote_ratio	-0.219160	-0.084502	1.000000

Figure 25 Python program to calculate and show correlation of all variables.

Correlation measures how to variables change together. If one variable goes up while the other also moves up, they have a positive correlation whereas if one variable goes down and other goes up then they have negative correlation.

Here, work\_year and salary\_in\_usd have positive correlation of around 0.236958 whereas, work\_year and remote\_ratio have negative correlation of around -0.219160. Again, salary\_in\_usd and remote\_ratio have negative correlation of around -0.084502 and 1 here means perfect correlation with itself.

## 4. Data Exploration

**4.1. Write a python program to find out top 15 jobs. Make a bar graph of sales as well.**

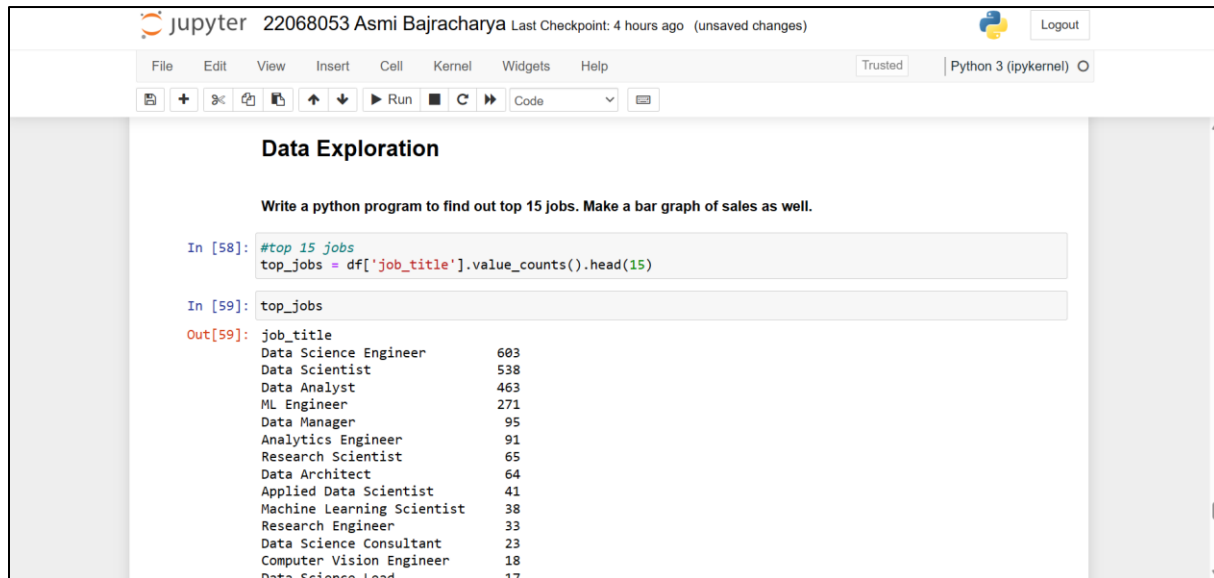


Figure 26 Python program to find out top 15 jobs.

Here, the column `job_title` is selected and `value_counts()` function counts how many times each unique job title appears in that column and finally `head(15)` function shows the top 15 most reoccurring jobs. The top 15 jobs in this dataframe are:

1. Data Science Engineer
2. Data Scientist
3. Data Analyst
4. ML Engineer
5. Data Manager
6. Analytics Engineer
7. Research Scientist
8. Data Architect
9. Applied Data Scientist
10. Machine Learning Scientist
11. Research Engineer

12. Data Science Consultant
13. Computer Vision Engineer
14. Data Science Lead
15. AI Scientist

Now, to meet the bar graph we need to import matplotlib.pyplot.

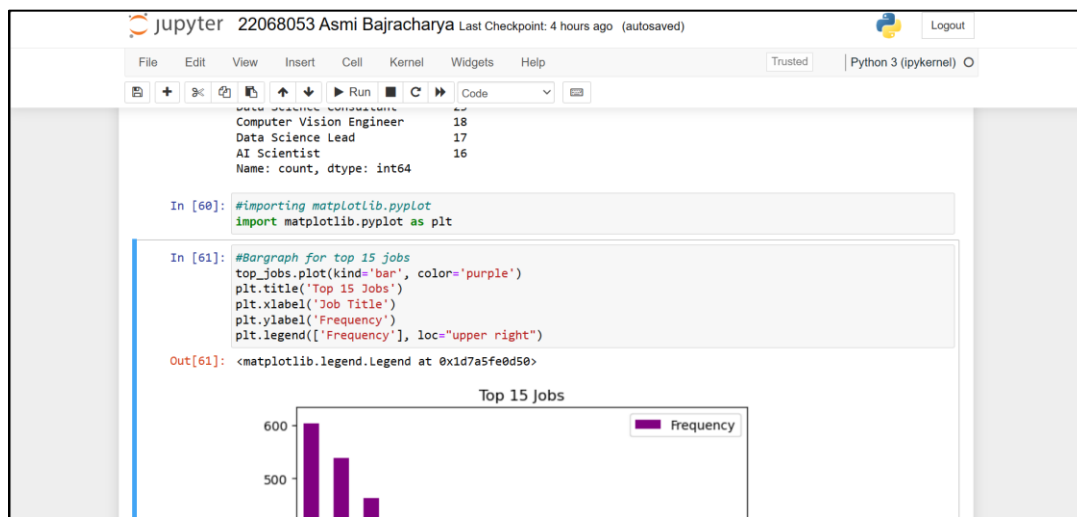


Figure 27 importing matplotlib.pyplot.

After importing matplotlib.pyplot, we are going to plot the bar graph. The topic of the paragraph is 'Top 15 jobs'. On the X axis we have the job title and on the Y axis we have the frequency of the repeated jobs.

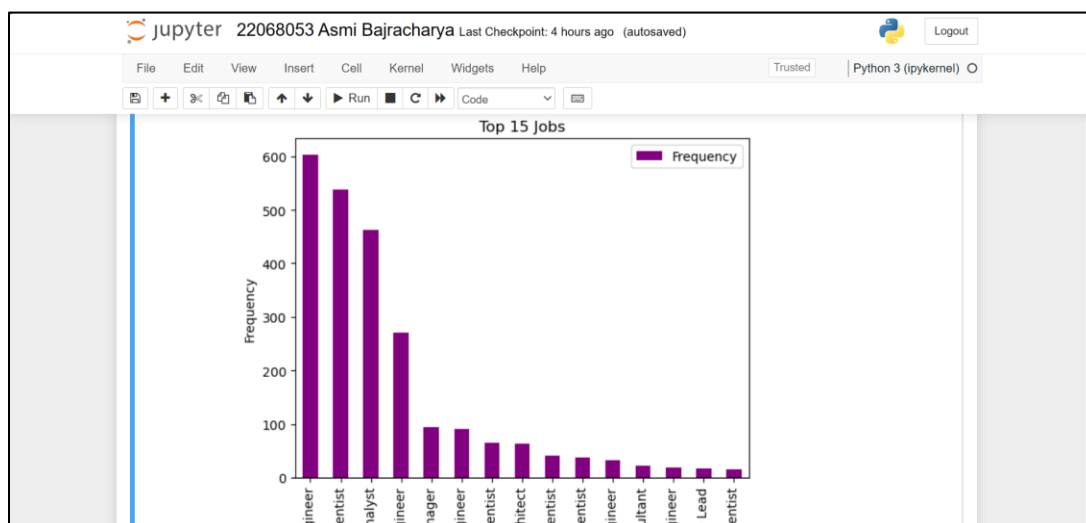


Figure 28 Bar graph of top 15 jobs.

Here, we can see that the frequency ranges from 0 to 600 and the job titles is shown in the figure below.

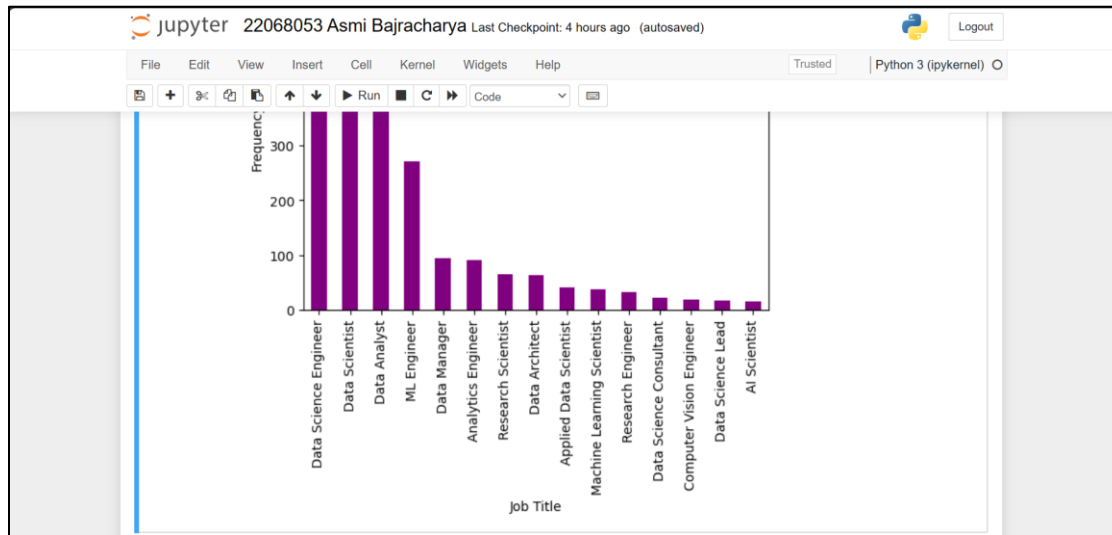


Figure 29 Bar graph of top 15 jobs x axis.

According to the bar graph data science engineer has the highest frequency of around 600 then the second is data scientist with the frequency of between 500 to 600 and similarly it is followed by data analyst. There is a significant gap between data analyst and ML engineer. And there is even more significant gap between ML engineer and data manager. Data manager is followed by research scientist, data architect, applied data scientist, machine learning scientist, research engineer, data science consultant, computer vision engineer, data science lead and finally AI scientist.



## 4.2. Which job has the highest salaries? Illustrate with bar graph.

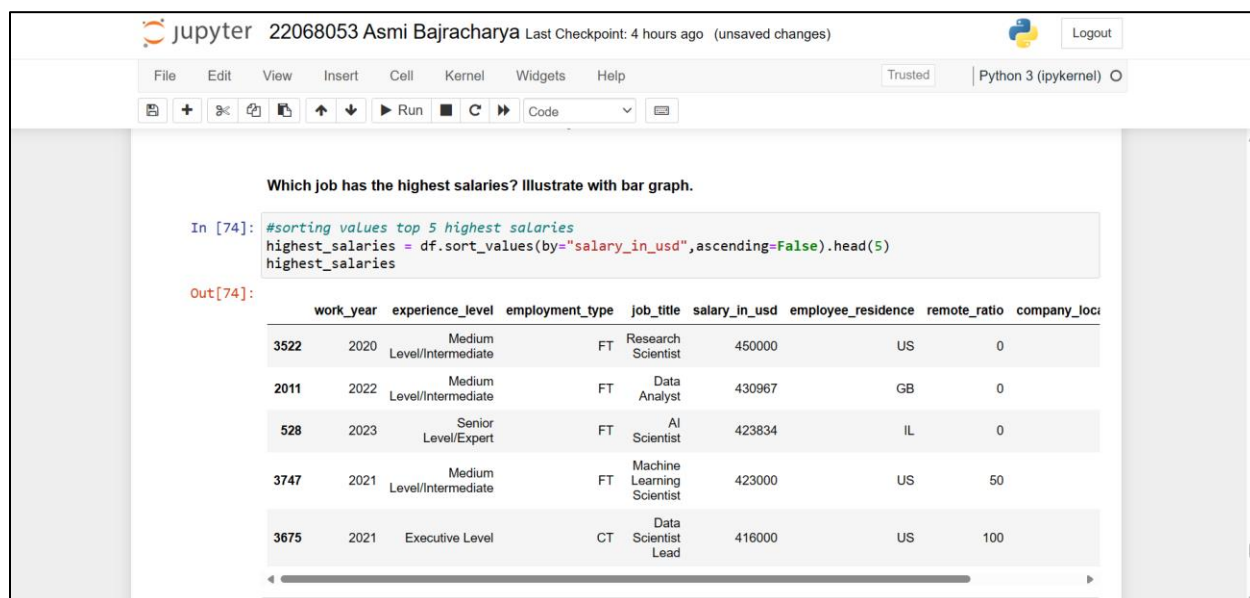


Figure 30 Python program to find out the highest salaries.

Here we're finding out the top five highest salaries in the data frame. We have sorted the values in descending order and use the head(5) function to find out the top five highest salaries. The highest salary is of research scientists with 450000, then it is Data Analyst with 430967, after that it is AI Scientist with 423834, then Applied Machine Learning Scientist with 423000 and finally Principal Data Scientist with 416000.

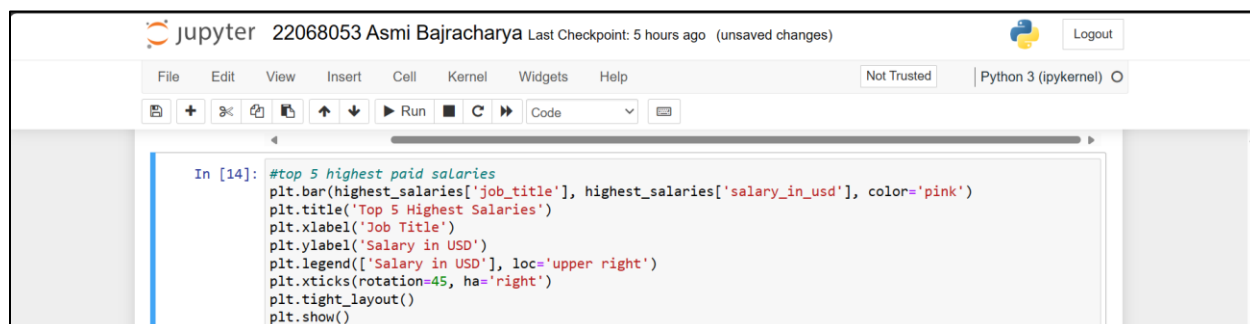


Figure 31 Python program to make the bar graph for top five highest paid salaries.

Here the title of this paragraph is 'Top 5 highest salaries' And on the X axis there are job title and on the Y axis there is salary in USD.

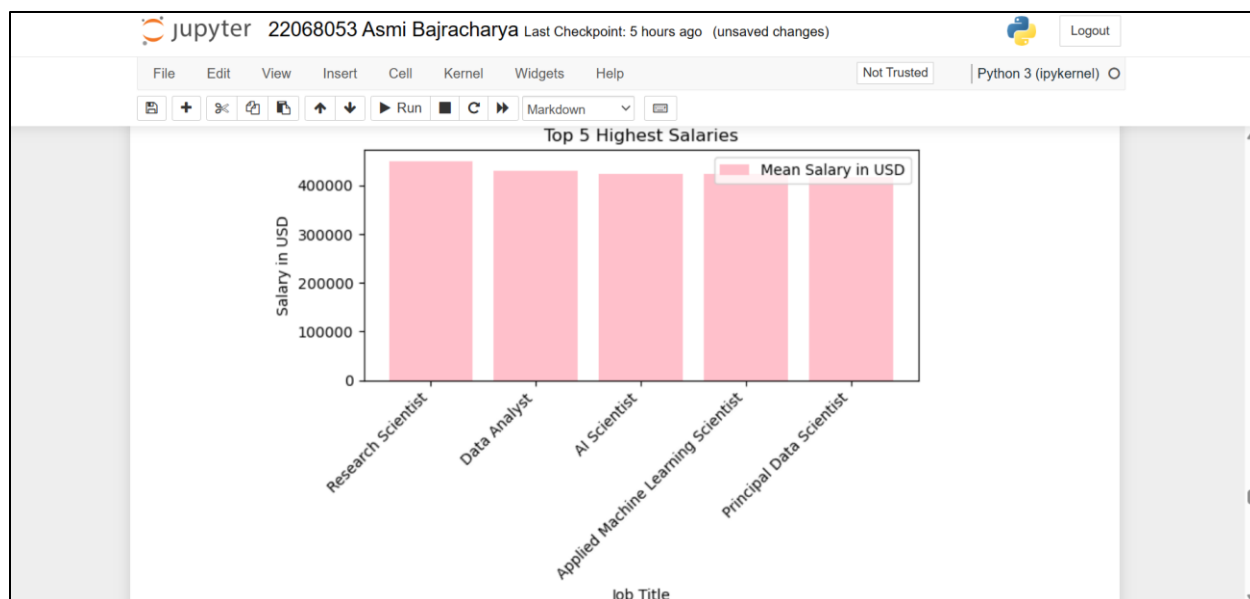
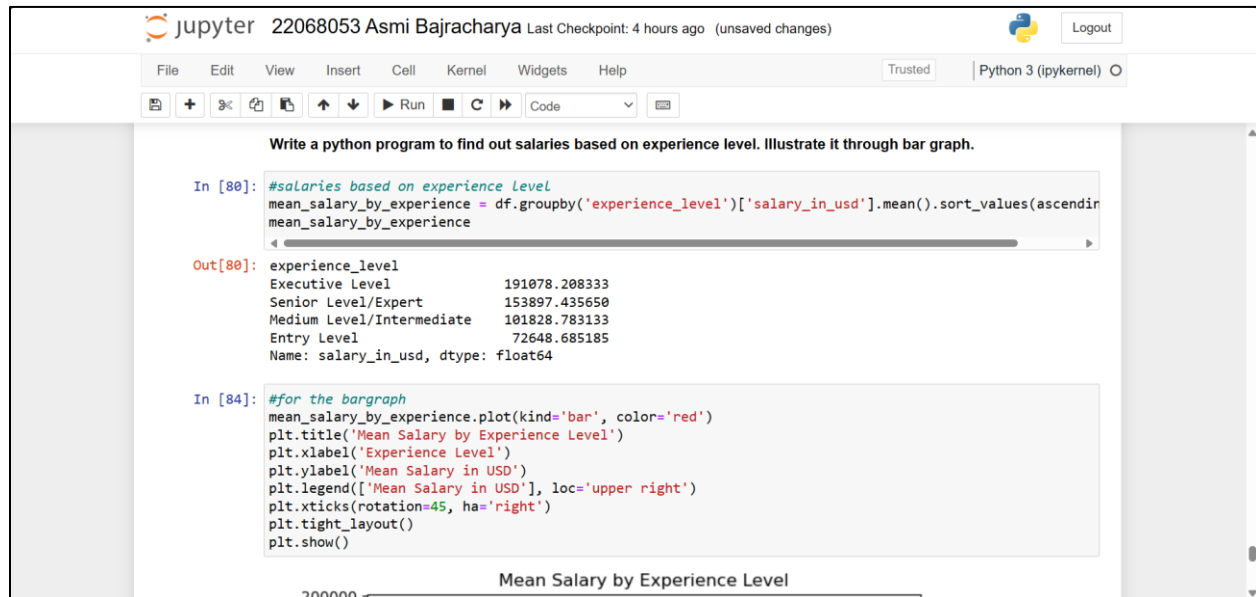


Figure 32 Bar graph of top 5 highest paid salaries.

In the above figure we can see the bar graph where research scientist has the most salary followed by data analyst call my AI scientist, applied machine learning scientist and principal data scientist. All these jobs have salaries over 400,000 USD.

**4.3. Write a python program to find out salaries based on experience level. Illustrate it through bar graph.**



*Figure 33 python program to find out salaries based on experience level.*

Here, to find out the salaries based on experience level we have grouped the salary in USD by experience level and after finding out the mean, we sorted it in descending order.

We found out that the executive level earns the most an entry level earns the least. Executive level earns an average of 191078.208333, then senior level/expert earns an average of 153897.435650, then medium level/ intermediate earns an average of 101828.783133, and lastly entry level earns an average of 72648.685185.

Now to plot it we have put the experience level on X axis and mean salary in USD in Y axis.

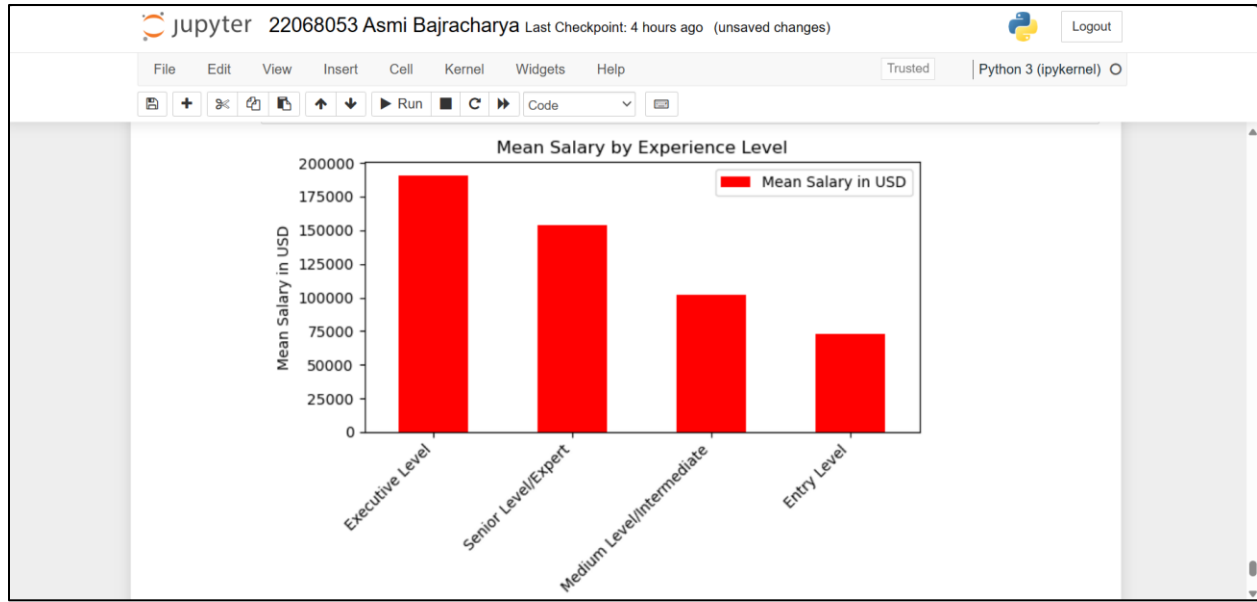
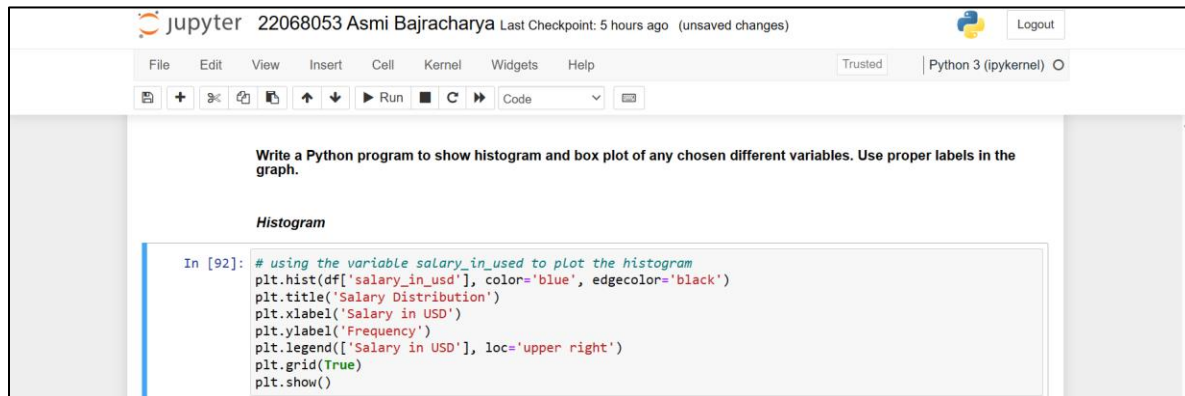


Figure 34 Bar graph of mean salary by experience level.

As we can see in the above figure, executive level earns the highest which is between 200,000 USD and 175,000 USD. Which is followed by senior level/expert then medium level/intermediate and lastly entry level.

#### 4.4. Write a Python program to show histogram and box plot of any chosen different variables. Use proper labels in the graph.

The variable chosen for making this histogram is salary\_in\_usd.



```
Write a Python program to show histogram and box plot of any chosen different variables. Use proper labels in the graph.

Histogram

In [92]: # using the variable salary_in_usd to plot the histogram
plt.hist(df['salary_in_usd'], color='blue', edgecolor='black')
plt.title('Salary Distribution')
plt.xlabel('Salary in USD')
plt.ylabel('Frequency')
plt.legend(['Salary in USD'], loc='upper right')
plt.grid(True)
plt.show()
```

Figure 35 Python program to show histogram of salary\_in\_usd.

Here, the title of the histogram is salary distribution and on the X axis we have salary in USD and on the Y axis we have frequency.

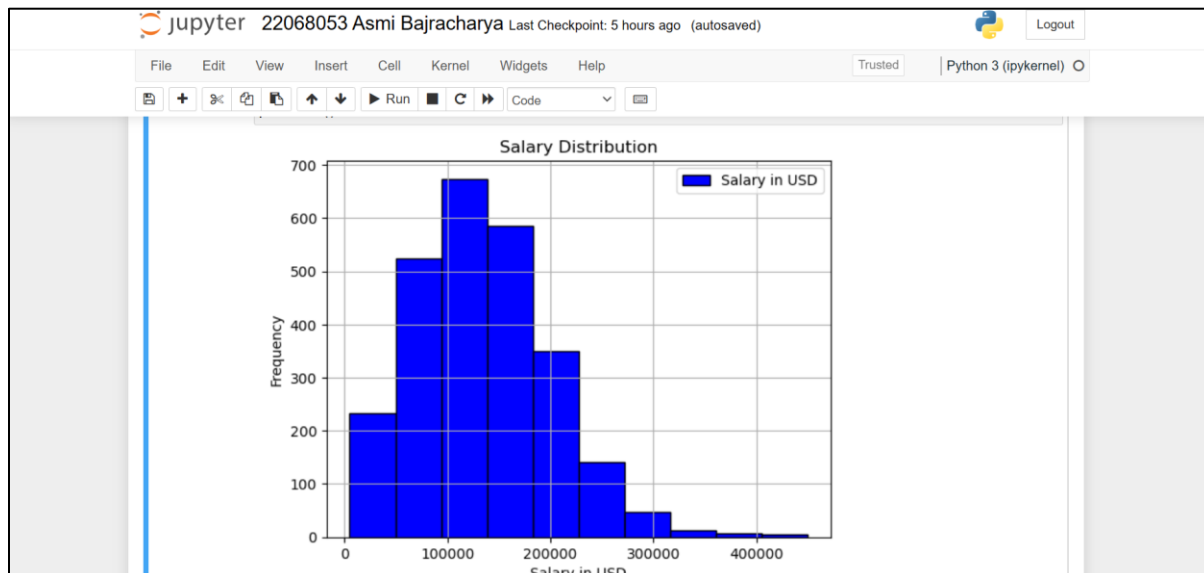


Figure 36 Histogram of salary\_in\_usd

Here, in this data set most people have the salary of around 100,000 USD on average. The salary of an average person ranges from 100,000 USD to 200,000 USD according to the histogram.

Salary in USD is chosen for the box plot also.



The image shows a Jupyter Notebook interface with a code cell. The code is as follows:

```

Boxplot

In [93]: #boxplot for salary in usd
plt.boxplot(df['salary_in_usd'], vert=False, patch_artist=True, showmeans=True)
plt.title('Box Plot of salary_in_usd ')
plt.xlabel('salary_in_usd')

Out[93]: Text(0.5, 0, 'salary_in_usd')

```

Figure 37 Box plot for salary\_in\_usd.

Here, the title of the box plot is 'Box Plot of salary\_in\_usd' and the X axis consist of the salary\_in\_usd.

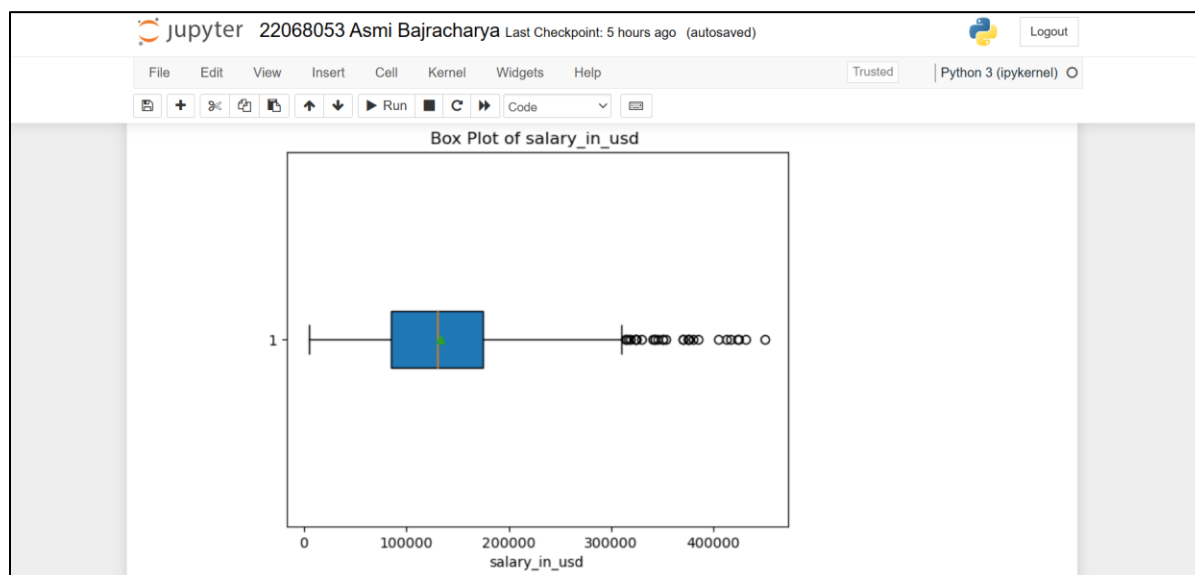


Figure 38 Box Plot of salary\_in\_usd

Here we can see the upper quartile, lower quartile and the mean as well as the outliers of salary\_in\_usd. The lower quartile ranges around 100,000 USD and the upper quartile ranges between 100,000 to 200,000 USD and the outliers lies over 300,000 USD.

## Conclusion

In this course work, we got to explore data science salaries using Python and DataScienceSalaries dataset. Firstly, we read the data carefully to understand the data resources. By using python as a programing language and pandas library, we started the preparation of the data by loading it into data frames then removing unnecessary columns and missing values as well as checking for unique and duplicate values

The second part was analyzing the data where we found out the sum, mean, standard deviation, skewness, and kurtosis of the data. Then we also found out the correlation of all the numeric values of the data. We properly analyzed the data in this part.

The third part of this coursework was data exploration. We found out the top 15 jobs as well as the highest paid salaries and illustrated them in a bar graph. We also found the salaries based on experience level and showed it in a bar graph. And lastly, we also made a histogram and box plot for the salaries in the dataset.

In conclusion, it is because of this coursework we got to learn to handle data and learnt valuable information on data understanding, preparation, analysis, and exploration. This contents of this coursework is not just limited to college but is also going to be very helpful in the future and I am very grateful that I got the opportunity to learn about this course.

## References

Sheldon, R., 2024. *What is a data set?*. [Online] Available at: <https://www.techtarget.com/whatis/definition/data-set> [Accessed 12 May 2024].

Turney, S., 2022. *Skewness | Definition, Examples & Formula*. [Online] Available at: <https://www.scribbr.com/statistics/skewness/> [Accessed 12 May 2024].

Turney, S., 2022. *What Is Kurtosis? | Definition, Examples & Formula*. [Online] Available at: [https://www.scribbr.com/statistics/kurtosis/#:~:text=Kurtosis%20is%20a%20measure%20of,\(thin%20tails\)%20are%20platykurtic.](https://www.scribbr.com/statistics/kurtosis/#:~:text=Kurtosis%20is%20a%20measure%20of,(thin%20tails)%20are%20platykurtic.) [Accessed 12 May 2024].