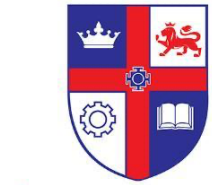
LONDON METROPOLITAN UNIVERSITY



islington college
(इस्लिङ्टन कलेज)

CC5067NI-Smart Data Discovery

60% Individual Coursework

2023-24 Spring

Student Name: Sanskriti Piya

London Met ID: 22067766

College ID: NP01CP4A220170

Assignment Due Date: Monday, May 13, 2024

Assignment Submission Date: Sunday, May 12, 2024

Word Count: 2419

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 mark of zero will be awarded.

TABLE OF CONTENTS

TABLE OF FIGURES

TABLE OF TABLES

1. DATA UNDERSTANDING

- The dataset serves as the establishment for all operations, strategies, and models that designers utilize to decipher them. A dataset is a sizeable collection of infromation points complied into a ssingle table. These days, datasets are utilized for an assortment of purposes in essentially each industry. These days, a part of colleges make their datasets accessible to the open, and websites like Kaggle and indeed GitHub release datasets that developers may utilize to induce the specified results (Geeksforgeeks, September 8, 2023). The significance of Python datasets lies in their ability to handle significant data volumes.Metadata is present in both Python records and record objects.Including the dataset object in your query can result in returning an index that is dependent on the rows and columns of these datasets (Educba, April 17,2023).

- Before embarking on the task, it's crucial to have a thorough understanding of your data set. The data set that has been provided to us contains information about data science salaries of various factors from 2020 to 2023. The various factors that can influence salary levels such as work year, experience level, employment level, job titles, salary, the currency of the salary, salary in USD, the employee's residence, remote ratio, company's location, and company's size. It's cardinal to ensure that data are rectified, cleaned, and organized before delving into in-depth data mining and analysis. This seems to involve annihilating duplication, dealing with missing values, standardizing formats, and more. Preparing the data to encourage facile analysis using a variation of tools and strategies is the main objective. This data set consists of work year, experience level, job title, salary,salary in usd,salary currency,company’s size, location of the company,residence of an employee and remote ratio. The descriptions of the columns are explained below with its following data types.

S.N	Column Name	Description	Data Type
1.	work_year	The work years that each employee has worked in are all stored in this column.	Int
2.	experience_level	The experience level of an employee, such as SE, MI, EN, and EX, is hoarded in the column .	Object
3.	employee_type	It holds the information of the employee_type data, which consists of CT and FT.	Object
4.	job_tilte	An employee's job title, such as "ML engineer," "data scientist," "data engineer," and so forth, is specified in this column.	Object
5.	salary	Employees' salaries are kept in the salary column.	Int
6.	salary_currency	This column holds an information about employee's salary in currency	object
7.	salary_in_usd	Employee's salaries are stored in USD in this column.	float
8.	employee_residence	This column contains the details about the employees' residences.	object
9.	remote_ratio		
10.	company_location	It specifies the location of the company	object
11.	company_size	This column contains the size of company such as L , M and S	Object

Table 1 : Description of the Data set

2. DATA PREPARATION

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

```
In [65]: import pandas as pd
         import numpy as np
         import matplotlib.pyplot as plt
```

```
In [66]: sal1=pd.read_csv("DataScienceSalaries.csv") # retrieves the datas from the CSV file
         salary_=pd.DataFrame(sal1)#replicates the contents of DataFrame sal1 into a new DataFrame called salary_.
         salary_
```

Figure 1 : Screenshot of program to load a dataset in the Data Frame.

Out[66]:

	work_year	experience_level	employment_type	job_title	salary	salary_currency	salary_in_usd	employee_residence	remote_ratio	company_location
0	2023	SE	FT	Principal Data Scientist	80000	EUR	85847	ES	100	ES
1	2023	MI	CT	ML Engineer	30000	USD	30000	US	100	US
2	2023	MI	CT	ML Engineer	25500	USD	25500	US	100	US
3	2023	SE	FT	Data Scientist	175000	USD	175000	CA	100	CA
4	2023	SE	FT	Data Scientist	120000	USD	120000	CA	100	CA
...	...	...	...	...	...	...	...	...	...	...
3750	2020	SE	FT	Data Scientist	412000	USD	412000	US	100	US
3751	2021	MI	FT	Principal Data Scientist	151000	USD	151000	US	100	US
3752	2020	EN	FT	Data Scientist	105000	USD	105000	US	100	US
3753	2020	EN	CT	Business Data Analyst	100000	USD	100000	US	100	US
3754	2021	SE	FT	Data Science Manager	7000000	INR	94665	IN	50	IN

3755 rows × 11 columns

Figure 2 : Screenshot of the output after loading the file

- Within the code given above, sal1=pd.read_csv("DataScienceSalaries.csv") Here, The DataScienceSalaries.csv file contains data that is being unsheathed from this code. A sal1 data frame is where the information is stored. Read_csv is the method for accessing / reading the CSV files. A new data frame named Salary_ is created by converting sal1 to the second line. Salary_ in the third line prints a complete data frame containing the information that has been extracted from the CSV file.

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

```
In [67]: salary_.drop(columns=['salary'], inplace=True)# eliminates the columns name salary from the dataframe.

In [68]: salary_
```

Figure 3 : Screenshot of program to remove unnecessary column i.e. salary.

Out[68]:

	work_year	experience_level	employment_type	job_title	salary_currency	salary_in_usd	employee_residence	remote_ratio	company_location	company_
0	2023	SE	FT	Principal Data Scientist	EUR	85847	ES	100	ES	
1	2023	MI	CT	ML Engineer	USD	30000	US	100	US	
2	2023	MI	CT	ML Engineer	USD	25500	US	100	US	
3	2023	SE	FT	Data Scientist	USD	175000	CA	100	CA	
4	2023	SE	FT	Data Scientist	USD	120000	CA	100	CA	
...	...	...	...	...	...	...	...	...	...	
3750	2020	SE	FT	Data Scientist	USD	412000	US	100	US	
3751	2021	MI	FT	Principal Data Scientist	USD	151000	US	100	US	
3752	2020	EN	FT	Data Scientist	USD	105000	US	100	US	
3753	2020	EN	CT	Business Data Analyst	USD	100000	US	100	US	
3754	2021	SE	FT	Data Science Manager	INR	94665	IN	50	IN	

3755 rows × 10 columns

Figure 4 : Screenshot of output after removing the salary column.

– By using the Pandas DataFrame Drop() method, we can delete any columns or rows Here salary column is being eliminated by specifying in place. By using inplace=True, the DataFrame is modified.After running this code, the salary column will be removed from the salary_ DataFrame. This process cannot be switched once it's executed.

```
In [69]: salary_.drop(columns=['salary_currency'], inplace=True)#removes the column name salary_currency from the dataframe.

In [70]: salary_
```

Figure 5 : Screenshot of a program to remove unnecessary column i.e. salary_currency.

Out[70]:

	work_year	experience_level	employment_type	job_title	salary_in_usd	employee_residence	remote_ratio	company_location	company_size
0	2023	SE	FT	Principal Data Scientist	85847	ES	100	ES	L
1	2023	MI	CT	ML Engineer	30000	US	100	US	S
2	2023	MI	CT	ML Engineer	25500	US	100	US	S
3	2023	SE	FT	Data Scientist	175000	CA	100	CA	M
4	2023	SE	FT	Data Scientist	120000	CA	100	CA	M
...	...	...	...	...	...	...	...	...	...
3750	2020	SE	FT	Data Scientist	412000	US	100	US	L
3751	2021	MI	FT	Principal Data Scientist	151000	US	100	US	L
3752	2020	EN	FT	Data Scientist	105000	US	100	US	S
3753	2020	EN	CT	Business Data Analyst	100000	US	100	US	L
3754	2021	SE	FT	Data Science Manager	94665	IN	50	IN	L

3755 rows × 9 columns

Figure 6: Screenshot of an output after removing salary in currency column.

– Columns define the column to be deleted as specified by this parameter and its name. When in place True is selected, the current data frame will modify. It optimizes data cleaning or pre-processing by eliminating redundant columns. Hereafter, the salary currency column will be detached from the salary_data frame.

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

```
In [71]: salary_.dropna()#excludes missing values from any rows in the dataframe.
```

Figure 7 : Screenshot of a program to remove NaN values from the updated dataframe

Out[71]:

	work_year	experience_level	employment_type	job_title	salary_in_usd	employee_residence	remote_ratio	company_location	company_size
0	2023	SE	FT	Principal Data Scientist	85847	ES	100	ES	L
1	2023	MI	CT	ML Engineer	30000	US	100	US	S
2	2023	MI	CT	ML Engineer	25500	US	100	US	S
3	2023	SE	FT	Data Scientist	175000	CA	100	CA	M
4	2023	SE	FT	Data Scientist	120000	CA	100	CA	M
...	...	...	...	...	...	...	...	...	...
3750	2020	SE	FT	Data Scientist	412000	US	100	US	L
3751	2021	MI	FT	Principal Data Scientist	151000	US	100	US	L
3752	2020	EN	FT	Data Scientist	105000	US	100	US	S
3753	2020	EN	CT	Business Data Analyst	100000	US	100	US	L
3754	2021	SE	FT	Data Science Manager	94665	IN	50	IN	L

3755 rows × 9 columns

Figure 8 : Screenshot of an output after removing the NaN values.

– In this picture above, dropna() function is used to eliminate NaN or missing values from a data frame. Nevertheless, the salary_ DataFrame depicted in the above illustration does not possess any NaN values or missing values.

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

```
In [72]: salary_[salary_.duplicated()]#pinpoints the duplicate values in the dataframe.
```

Figure 9 : Screenshot of a program to check duplicate values.

Out[72]:

	work_year	experience_level	employment_type	job_title	salary_in_usd	employee_residence	remote_ratio	company_location	company_size
115	2023	SE	FT	Data Scientist	150000	US	0	US	M
123	2023	SE	FT	Analytics Engineer	289800	US	0	US	M
153	2023	MI	FT	Data Engineer	100000	US	100	US	M
154	2023	MI	FT	Data Engineer	70000	US	100	US	M
160	2023	SE	FT	Data Engineer	115000	US	0	US	M
...	...	...	...	...	...	...	...	...	...
3439	2022	MI	FT	Data Scientist	78000	US	100	US	M
3440	2022	SE	FT	Data Engineer	135000	US	100	US	M
3441	2022	SE	FT	Data Engineer	115000	US	100	US	M
3586	2021	MI	FT	Data Engineer	200000	US	100	US	L
3709	2021	MI	FT	Data Scientist	90734	DE	50	DE	L

1171 rows × 9 columns

Figure 10 : Screenshot of an output of checking duplicate values.

– The duplicated () function detects duplicate lines in a Data frame. In the salary_ data frame, the function duplicated() is employed to identify any duplicate data. This approach helps to eliminate the possibility of duplicate data.

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

```
In [9]: sal2 = salary_.apply(pd.unique)#fabricates new dataframe name sal2 with unique values in each columns.

In [10]: sal2

Out[10]: work_year                                  [2023, 2022, 2020, 2021]
         experience_level                                  [SE, MI, EN, EX]
         employment_type                                   [FT, CT, FL, PT]
         job_title            [Principal Data Scientist, ML Engineer, Data S...
         salary_in_usd        [85847, 30000, 25500, 175000, 120000, 222200, ...
         employee_residence   [ES, US, CA, DE, GB, NG, IN, HK, PT, NL, CH, C...
         remote_ratio                                             [100, 0, 50]
         company_location     [ES, US, CA, DE, GB, NG, IN, HK, NL, CH, CF, F...
         company_size                                             [L, S, M]
         dtype: object
```

Figure 11: Screenshot of recognizing the unique values.

– According to the above image, extraction of unique elements is achieved through pd. unique(). An array is created that contains only the unique values for that row. By using the apply() method, a procedure can be applied to individual elements, rows, or columns inside a data frame. However, salary_.apply(pd.unique) is used to apply the pd.unique method to every column of DataFrame.The newly created DataFrame sal2 will have a unique value for each column of a data frame.

2.6. Rename the experience level columns

SE – Senior Level/Expert

MI – Medium Level/Intermediate

EN – Entry Level

EX – Executive Level

```
In [75]: salary_['experience_level'].replace({'SE':'Senior Level/Expert',
                                              'MI': 'Medium Level/Intermediate',
                                              'EN': 'Entry Level',
                                              'EX': 'Executive Level'}, inplace = True )#retitles the values in the dictionary
         salary_
```

Figure 12 : Screenshot of program to rename the experience level column.

Out[75]:

	work_year	experience_level	employment_type	job_title	salary_in_usd	employee_residence	remote_ratio	company_location	company_size
0	2023	Senior Level/Expert	FT	Principal Data Scientist	85847	ES	100	ES	L
1	2023	Medium Level/Intermediate	CT	ML Engineer	30000	US	100	US	S
2	2023	Medium Level/Intermediate	CT	ML Engineer	25500	US	100	US	S
3	2023	Senior Level/Expert	FT	Data Scientist	175000	CA	100	CA	M
4	2023	Senior Level/Expert	FT	Data Scientist	120000	CA	100	CA	M
...	...	...	...	...	...	...	...	...	...
3750	2020	Senior Level/Expert	FT	Data Scientist	412000	US	100	US	L
3751	2021	Medium Level/Intermediate	FT	Principal Data Scientist	151000	US	100	US	L
3752	2020	Entry Level	FT	Data Scientist	105000	US	100	US	S
3753	2020	Entry Level	CT	Business Data Analyst	100000	US	100	US	L
3754	2021	Senior Level/Expert	FT	Data Science Manager	94665	IN	50	IN	L

3755 rows × 9 columns

Figure 13 : Screenshot of output after renaming the experience level column

– According to the image illustrated above, Salary_['experience_level'] designates the experience_level column from the DataFrame Salary_.The intended value in the column selection is replaced with the specified value by using the. replace function.

3. DATA ANALYSIS

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

```
In [76]: salary_.drop_duplicates(inplace=True)

In [43]: salary_
```

Figure 14 : Screenshot of a program to remove all the duplicates values .

Out[43]:

	work_year	experience_level	employment_type	job_title	salary_in_usd	employee_residence	remote_ratio	company_location	company_size
0	2023	Senior Level/Expert	FT	Principal Data Scientist	85847	ES	100	ES	L
1	2023	Medium Level/Intermediate	CT	ML Engineer	30000	US	100	US	S
2	2023	Medium Level/Intermediate	CT	ML Engineer	25500	US	100	US	S
3	2023	Senior Level/Expert	FT	Data Scientist	175000	CA	100	CA	M
4	2023	Senior Level/Expert	FT	Data Scientist	120000	CA	100	CA	M
...	...	...	...	...	...	...	...	...	...
3750	2020	Senior Level/Expert	FT	Data Scientist	412000	US	100	US	L
3751	2021	Medium Level/Intermediate	FT	Principal Data Scientist	151000	US	100	US	L
3752	2020	Entry Level	FT	Data Scientist	105000	US	100	US	S
3753	2020	Entry Level	CT	Business Data Analyst	100000	US	100	US	L
3754	2021	Senior Level/Expert	FT	Data Science Manager	94665	IN	50	IN	L

2584 rows × 9 columns

Figure 15 :Screenshot of an output after removing the duplicate values

– The removal of redundant values from the data frame is necessary to calculate standard deviation, skewness, kurtosis, mean, total, and standard variation. Subsequently, to ensure the decency of data duplicate values are expunged using the drop.duplicate function before finding the result. In this case, duplicate rows are removed from the data frame salary_ by calling the function drop_duplicates() on the data frame salary_. Without the need for a new data frame, we can directly modify the data frame salary_ using inplace true.

i. Statistics of sum

```
In [59]: sum_of_variable = salary_['salary_in_usd'].sum()#computes the sum of the values in the salary_in_usd table.
         print("The sum of the variable is:", sum_of_variable) #prints the sum.

         The sum of the variable is: 344729580
```

Figure 16 : Screenshot of calculating the sum of salary_in_usd table.

- In the DataFrame Salary_, sum_of_variable = Salary_in_usd is used to compute the total sum of the values for the column salary_in_usd . .sum() method calculates the sum of the variable. Th calculated sum is stored in the variable name sum_of_the_variable and is printed through print() method.

ii. Statistics of mean

```
In [60]: mean_of_variable = salary_['salary_in_usd'].mean()#calculates the mean of the variable .
         print("The mean of the variable is:", mean_of_variable) #prints the mean

         The mean of the variable is: 133409.28018575851
```

Figure 17 : Screenshot of calculating the mean of salary_in_usd table.

- The line of code illustrated in the above picture calculates the mean of the values for the column salary_in_usd. .mean() is used to compute the mean of the variable and the value of the mean is printed using the print method.

iii. Statistics of standard deviation

```
In [61]: standard_deviation_of_variable = salary_['salary_in_usd'].std()#reckons the standard deviation of the variable.
         print("The standard deviation of the variable is:", standard_deviation_of_variable) #prints the standard deviation

         The standard deviation of the variable is: 67136.83732925021
```

Figure 18 : Screenshot of calculating the standard deviation of salary_in_usd.

- This line of code standard_deviation_of_variable = salary_['salary_in_usd'] shows the summary statistics of standard deviation . The value is then stored in a variable (standard_deviation_of_variable) . Print method is used to provide the output of standard deviation.

iv. Statistics of Skewness

```
In [62]: skewness_of_variable = salary_['salary_in_usd'].skew()#calculates the skewness of the values in salary_in_usd table.
         print("The skewness of the variable is:", skewness_of_variable)#gives the output of the skewness.

         The skewness of the variable is: 0.6203168790580038
```

Figure 19 : Screenshot of calculating the skewness of salay_in_usd.

– As depicted in the figure above, the skewness of the variable'salary_in.usd' situated inside the data frame salary_' is calculated through the skew() method. Following the skew() operation, print() is used to print the output.

v. Statistics of Kurtosis

```
In [63]: kurtosis_of_variable = salary_['salary_in_usd'].kurt()#computes the kurtosis of the variable.
         print("The kurtosis of the variable is:", kurtosis_of_variable)#gives the output of kurtosis.

         The kurtosis of the variable is: 0.8269400876861832
```

Figure 20 : Screenshot of calculating the kurtosis of salary_in_usd.

– This code is responsible for noting the kurtosis of a variable in the data frame "salary_" that has the name salary_in_usd. In contrast, the output of the kurtosis is printed using print() method.

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

```
In [58]: salary_.corr(numeric_only=True)
```

Out[58]:

	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 21 : Screenshot of calculating the correlation of the variables.

- In the DataFrame, this code is responsible for determining the correlation of the numeric columns of "salary_" and creating a correlation matrix. The data frame that will be used to perform the correlation is indicated by salary_.corr()When using this approach to determine column correlation, missing values (NaN) are typically excluded. The reason why numeric_only=True is that correlations are determined by using only numerical columns. Columns that are not numeric are excluded from correlation calculations.

4. DATA EXPLORATION

```
In [64]: salary_['job_title'].replace({'ML Engineer':'Machine Learning Engineer',
                                       'Applied Machine Learning Engineer':'Machine Learning Engineer',
                                       'Machine Learning Scientist':'Machine Learning Engineer',
                                       'Lead Machine Learning Engineer': 'Machine Learning Engineer',
                                       'Machine Learning Infrastructure Engineer':'Machine Learning Engineer',
                                       'Machine Learning Developer':'Machine Learning Engineer',
                                       'Machine Learning Manager' : 'Machine Learning Engineer',
                                       'Big Data Engineer': 'Data Engineer',
                                       'cloud Data Engineer': 'Data Engineer',
                                       'ETL Engineer': 'Data Engineer',
                                       'Data Devops Engineer': 'Data Engineer',
                                       'Azure Data Engineer':'Data Engineer',
                                       'Software Data Engineer':'Data Engineer',
                                       'Data Operation Engineer':'Data Engineer',
                                       'Computer Vision Engineer':'Data Engineer',
                                       'MLOps Engineer':'Data Engineer',
                                       'Analytics Engineer':'Data Engineer',
                                       'Applied Machine Learning Scientist':'Data Scientist',
                                       'Applied Scientist': 'Data Scientist',
                                       'Applied Data Scientist': 'Data Scientist',
                                       'Staff Data Scientist':'Data Scientist',
                                       'Principal Data Scientist' : 'Data Scientist',
                                       'Research Scientist':'Data Scientist',
                                       'Data Science Manager': 'Data Manager',
                                       'Data Analytics Manager': 'Data Manager',
                                       'Manager Data Management': 'Data Manager',
                                       'Business Data Analyst': 'Data Analyst',
                                       'Financial Data Analyst': 'Data Analyst',
                                       'Marketing Data Analyst': 'Data Analyst',
                                       'Staff Data Analyst' :'Data Analyst',
                                       'Data Quality Analyst' : 'Data Analyst',
                                       'Product Data Analyst': 'Data Analyst',
                                       'Lead Data Analyst': 'Data Analyst' }, inplace = True )#replaces the values.
salary_
```

Figure 22 : Screenshot of replacing the values.

Out[64]:

	work_year	experience_level	employment_type	job_title	salary_in_usd	employee_residence	remote_ratio	company_location	company_size
0	2023	Senior Level/Expert	FT	Data Scientist	85847	ES	100	ES	L
1	2023	Medium Level/Intermediate	CT	Machine Learning Engineer	30000	US	100	US	S
2	2023	Medium Level/Intermediate	CT	Machine Learning Engineer	25500	US	100	US	S
3	2023	Senior Level/Expert	FT	Data Scientist	175000	CA	100	CA	M
4	2023	Senior Level/Expert	FT	Data Scientist	120000	CA	100	CA	M
...	...	...	...	...	...	...	...	...	...
3750	2020	Senior Level/Expert	FT	Data Scientist	412000	US	100	US	L
3751	2021	Medium Level/Intermediate	FT	Data Scientist	151000	US	100	US	L
3752	2020	Entry Level	FT	Data Scientist	105000	US	100	US	S
3753	2020	Entry Level	CT	Data Analyst	100000	US	100	US	L
3754	2021	Senior Level/Expert	FT	Data Manager	94665	IN	50	IN	L

2584 rows × 9 columns

Figure 23: Screenshot of replaced values.

– Code above image shows values that are changed to maintain consistency with the data. Cleaning the data beforehand is necessary to produce the bar chart and histogram below as the data are inconsistent in the data frame. The above image displays data that has been purified to maintain its quality. Obtaining uncontaminated data leads to superior analysis and modelling outcomes. The act of cleaning data helps to reduce redundancy and simplify interpretation. According to the image illustrated above, Salary_['job_tilte'] designates the job_title column from the DataFrame Salary_.The intended value in the column selection is replaced with the specified value by using the. replace function. Figure 16 shows the images where the name of the job_title being replaced.

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

```
In [98]: #shows the top 15 jobs
         top_jobs=salary_['job_title'].value_counts().head(15)
         print(top_jobs)

         #Creates the bar graph
         plt.figure(figsize=(10,8))
         top_jobs.plot(kind='bar' , color='skyblue')
         plt.xlabel('Job Title')
         plt.ylabel('Sales')
         plt.title('Top 15 Jobs')
         plt.show
         plt.legend();
```

Figure 24 : Screenshot of a program to show top 15 jobs and to create a bar graph.

```
job_title
Data Engineer                727
Data Scientist               665
Data Analyst                 432
Machine Learning Engineer    292
Data Manager                  94
Data Architect                64
Research Engineer             33
Data Science Consultant       23
AI Scientist                  16
BI Data Analyst               15
Data Specialist               12
AI Developer                  11
Director of Data Science      11
BI Developer                  11
Head of Data                  10
Name: count, dtype: int64
```

Figure 25 : Screenshot of output of top 15 jobs

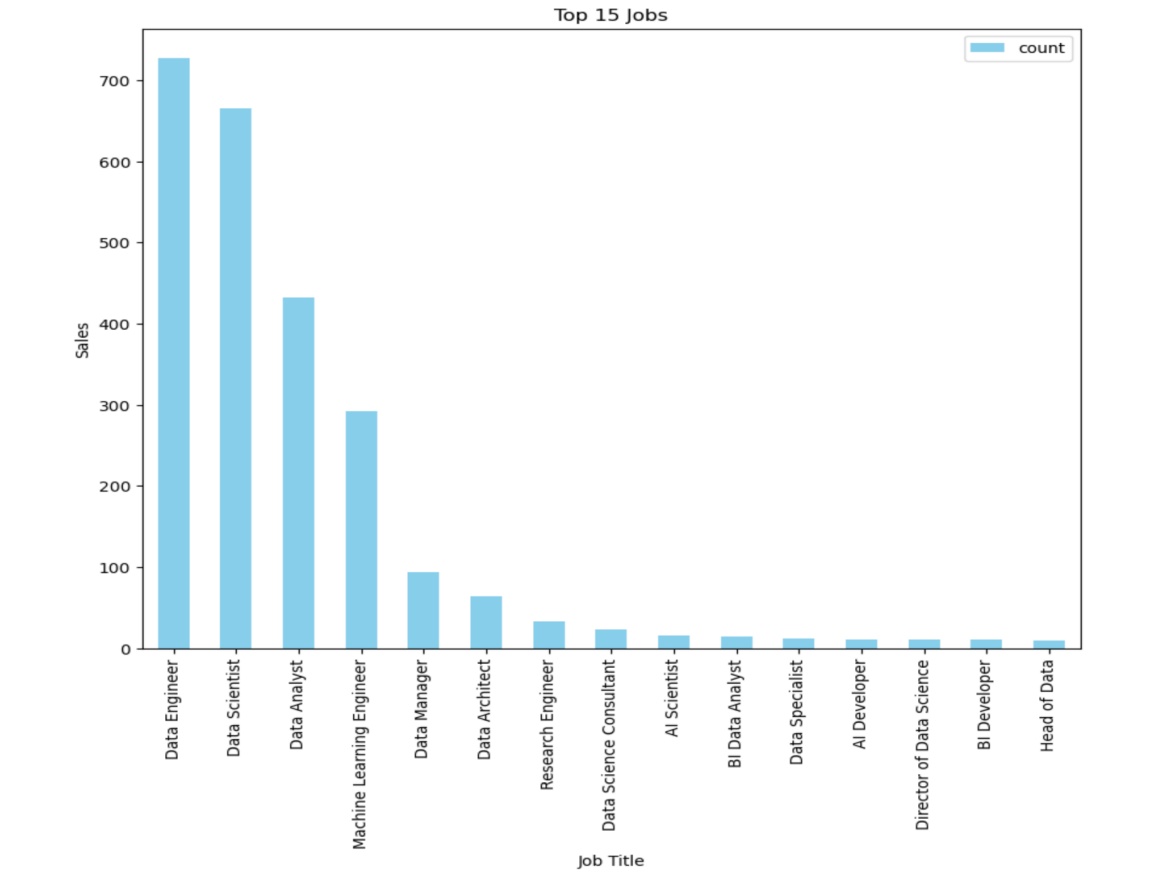


Figure 26 : Screenshot of output after plotting in the graph.

- In the salary_ DataFrame, the value_counts() function is used to calculate the frequency of job titles in the top-level column using the coding sequence Salary_['job_title'] = top_jobs.count_value().head(15). After that while creating a graph, Plt. figure employs the Figure() function of Matplotlib to generate a new plot representation with dimensions. The plot type is set to 'bar' and the color of the bar is defined by a function that sets colored sky blue. The xlabel() method in Matplotlib is used to label the plot's X-axis with the code "Job Title.Plt. ylabel() is utilized to label the Y-axis of the chart as 'Sales'. The Data engineer holds the highest amount of job title.
- By using the title() function in Matplotlib, one can use this line to transform the plot's title into a list of Top 15 Jobs. The plotting function plt. show() displays the result vital to understanding data visualization.

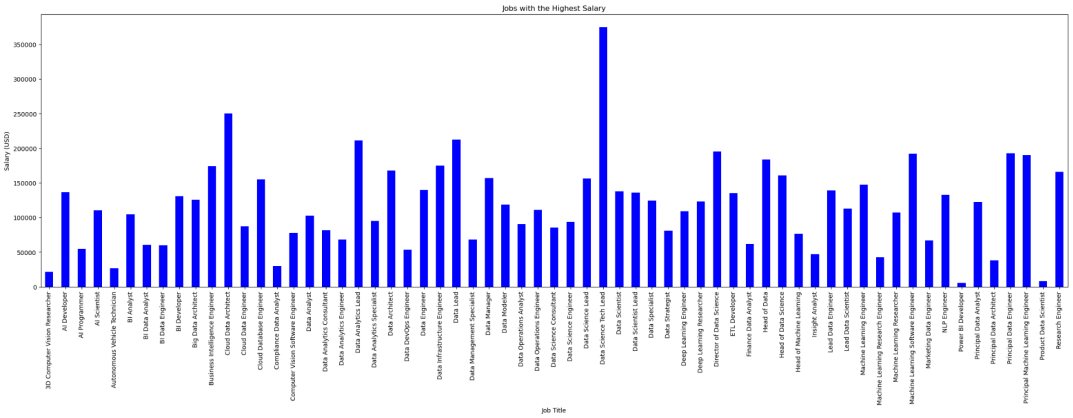
4.2. Which job has the highest salaries? Illustrate with bar graph.

```
In [116]: #It calculates the average salary for each job title
          _average_salary_of_job = salary_.groupby('job_title')['salary_in_usd'].mean()


          #this code finds the job with the highest salary
          highest_job_salary = _average_salary_of_job.idxmax()

          # Creates a bar graph
          plt.figure(figsize=(30, 8))
          _average_salary_of_job.plot(kind='bar',color='blue')
          plt.xlabel('Job Title')
          plt.ylabel('Salary (USD)')
          plt.title('Jobs with the Highest Salary')
          plt.show()
          print("The job with the highest_job_salary is:", highest_job_salary)
```

Figure 27 : Screenshot of a program to find out the jobs with highest salaries




```
The job with the highest_job_salary is: Data Science Tech Lead
```

Figure 28 : Screenshot of the output of the jobs with highest salaries.

- A bar chart displaying the average salary for all positions will be created by using the code to calculate the salary of each position and determine the highest average pay.The bar graph illustrates that Data Science Tech Lead is the job with the higest salary.
- The first line of code groups the DataFrame based on the job_title column to determine the average of the salary_in_usd column for each group. After computing the average salary for each position, the code utilizes the idxmax() function to determine the position with the highest average pay. This method yields the job title that matches the highest salary. By utilizing matplotlib and functions like xlabel(), plt. ylabel(), and plt. title()plot() with kind='bar', the average salary for

each position can be displayed in a bar chart. To sum up, plt. show portrays the bar graph and print functions print the highest salaries.

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

```
In [135]: #this line of code will calculate the mean salary for experience level
          salary_for_experience_level = salary_.groupby('experience_level')['salary_in_usd'].mean()
          salary_for_experience_level

Out[135]: experience_level
          Entry Level                  72648.685185
          Executive Level             191078.208333
          Medium Level/Intermediate   101828.783133
          Senior Level/Expert         153897.435650
          Name: salary_in_usd, dtype: float64
```

Figure 29 : Screenshot of a program to find out salaries based on experience level.

```
In [136]: # It will illustrate in a bar graph
          plt.figure(figsize=(10, 5))
          salary_for_experience_level.plot(kind='bar',color='red')
          plt.xlabel('Experience Level')
          plt.ylabel('Salary (USD)')
          plt.title('Salaries based on Experience Level')
          plt.show()
```

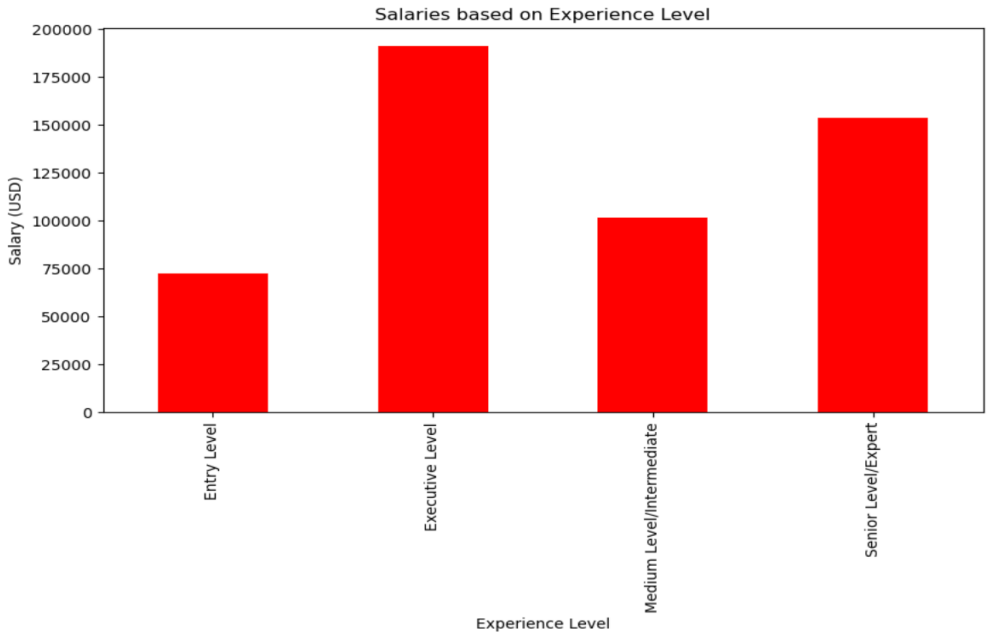


Figure 30: Screenshot of an illustration in graph of salaries based on experience level.

– The code shown in Figure 29 groups the data frame and determines the experience level-specific mean salary. In contrast, the graph illustration generates a bar graph according to experience level, specifically Entry Level, Executive Level, Medium Level, and Senior Level. According to the above image, among the Experience level, the Executive level has the highest salaries.

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

```
In [144]: #The following code will illustrate the histogram of salary_in_usd variable.

plt.figure(figsize=(10,5))
plt.hist(salary_ [ 'salary_in_usd'] , color='skyblue')
plt.xlabel('Salary (USD)')
plt.ylabel('Frequency')
plt.title('Histogram of Salary in USD')
plt.show()
```

Figure 31 : Screenshot of a program to show a histogram of Salary in USD.

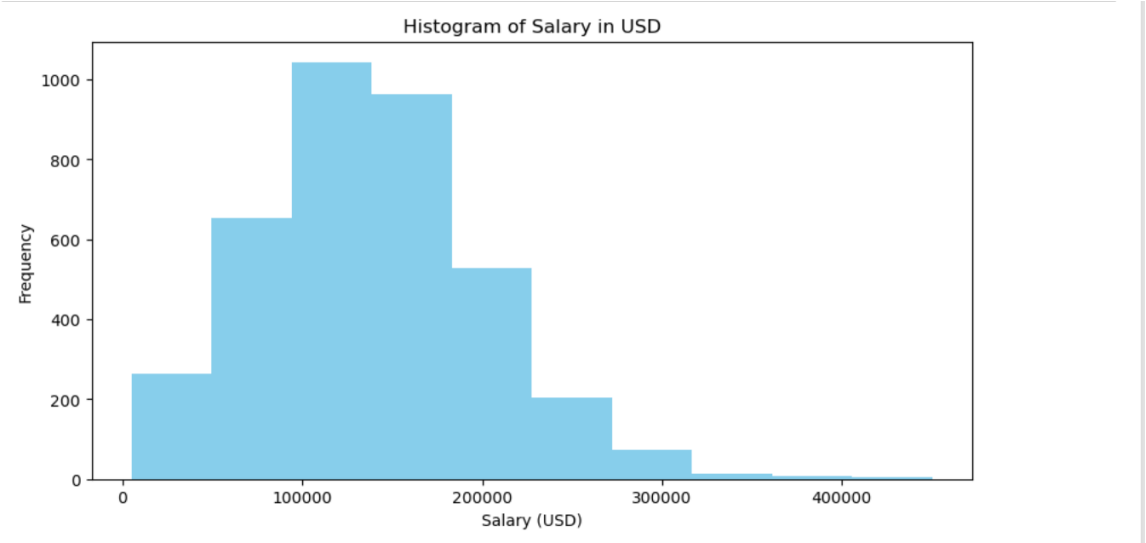


Figure 32 : Screenshot of the illustrating in histogram.

– This following code is used to illustrate the salaries in USD. First line of code (plt.figure(figsize) sets the size of the figure whereas plt.hist fabricates the plot of histogram. Then, plt.xlabel sets the label for x-axis (Salary (USD))while on the other hand plt.ylabel sets the level for y-axis(Frequency).Furthermore, plt.title collimates the title for the histogram. Moreover, plt.show() unveils the histogram

```
In [150]: #The following code shows the boxplot of salary_in_usd variable.

          plt.figure(figsize=(8,5))
          plt.boxplot(salary_['salary_in_usd'])
          plt.xlabel('Salary (USD)')
          plt.ylabel('Frequency')
          plt.title('Box Plot of Salary in USD')
          plt.show()
```

Figure 33 : Screenshot of a program for a box plot

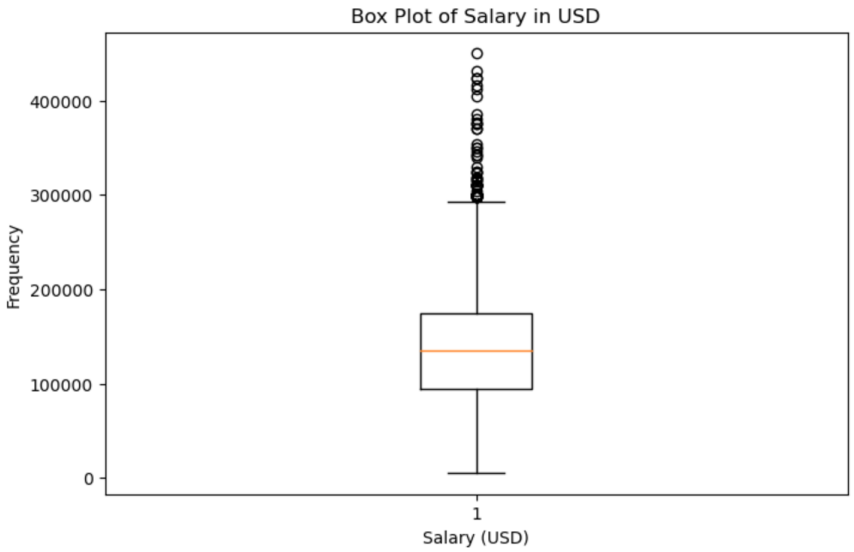


Figure 34: Screenshot of the illustration of the salary in usd through Box plot.

- The above figures shows the creation of Box plot for thr sakary in usd varaiable. Box plots are created with the functionplt.boxplot(salary_['salary_in_usd']). A box plot is created using the values of the DataFrame column salary_in_usd to show the distribution of these values. The labels and titles to boxplots can be attached using the plt.xlabel(), plt.ylabel(), and plt.title() functions. The box plot with the given labels and title is displayed by the plt.show function.

5. REFERENCES

Geeksforgeeks, September 8, 2023. *GeeksforGeeks.* [Online]
Available at: https://www.geeksforgeeks.org/what-is-dataset/
[Accessed May 2024].

Educba, April 17,2023. *Educba.* [Online]
Available at: https://www.educba.com/dataset-in-python/
[Accessed May 2024].