LONDON METROPOLITAN UNIVERSITY

Islington college
(इस्लिङ्टन कलेज)

CC5067NI-Smart Data Discovery

60% Individual Coursework

2023-24 Spring

Student Name: Sachida Paudel

London Met ID: 22068732

College ID: NP01CP4A220006

Assignment Due Date: Monday, May 13, 2024

Assignment Submission Date: Monday, May 13, 2024

Word Count: 3097

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.

Table of Contents

Table Of Figures

Table of Tables

1. Data Understanding:

Data understanding also called data exploration is the process that involves gaining understanding of a dataset to understand its structure, contain, texture and characteristics. It is an initial step in the data analysis life cycle, and it plays a major role in providing informed decision on how to proceed with data analysing effectively (Verjus & Gigandet, 2022).

The given dataset of the coursework is about a company information whose work year is from the year of 2020 to 2023. The company consists of experience level of the staff be it a Senior level Engineer or a medium – level role or an Entry level role and an Executive level of engineering. The employment type in the company includes a combination of full-time workers, part-time workers, contract workers and freelancers. The company contains of principal data scientist, Machine Learning 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, 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.

The data set also consists of salary in USD for each job title. The residence of the employees are from Spain, United States, Canada, Germany, United Kingdom, Nigeria, India, Hong Kong, Portugal, Netherlands, Switzerland, Central African Republic, France, Australia, Finland, Ukraine, Ireland, Israel, Ghana, Austria, Colombia, Singapore, Sweden, Slovenia, Mexico, Uzbekistan, Brazil, Thailand, Croatia, Poland, Kuwait, Vietnam, Cyprus, Argentina, Armenia, Bosnia and Herzegovina, Kenya, Greece, North Macedonia, Latvia, Romania, Pakistan, Italy, Morocco, Lithuania, Belgium, American Samoa, Iran, Hungary, Slovakia, China, Czech Republic, Costa Rica, Turkey, Chile, Puerto Rico, Denmark, Bolivia, Philippines, Dominican Republic, Egypt, Indonesia, United Arab Emirates, Malaysia, Japan, Estonia, Honduras, Tunisia, Russia, Algeria, Iraq, Bulgaria, Jersey, Serbia, New Zealand, Moldova, Luxembourg, Malta.

The dataset also contains companies work remote ratio rate where remote ratio 100 means completely remote, 50 means partly remote and 0 means fully on-site. In addition to it, the dataset also involves company location that includes Spain, United States, Canada, Germany, United Kingdom, Nigeria, India, Hong Kong, Netherlands, Finland, Switzerland, Ukraine, Colombia, Australia, Sweden, Ireland, Central African Republic, France, Italy, Brazil, Portugal, Mexico, Singapore, Thailand, Indonesia, Denmark, Norway, Malaysia, Poland, Hungary, Japan, Argentina, Philippines, Serbia, Belgium, Austria, Russia, South Africa, Egypt, Turkey, Czech Republic, United Arab Emirates, Morocco, Saudi Arabia, South Korea, Qatar, Kenya, Greece, Vietnam, Pakistan, Israel, Romania, Iran, Algeria, Finland, Slovenia, Luxembourg, Thailand, Croatia, Bosnia and Herzegovina, Slovakia, Egypt, Latvia, Lithuania, Estonia, Iceland, Serbia, Tunisia, Ukraine, Chile, Peru, Ecuador, Venezuela, Costa Rica, Uruguay, Paraguay, Dominican Republic, El Salvador, Guatemala, Jamaica, Haiti, Nicaragua, Belize, Guyana, Suriname, Trinidad and Tobago, Bahamas, Barbados, Antigua and Barbuda, Dominica, Saint Kitts and Nevis, Saint Lucia, Saint Vincent and the Grenadines, Grenada, Anguilla, Saint Martin, Bermuda, British Virgin Islands, Cayman Islands, Turks and Caicos Islands, Puerto Rico, U.S. Virgin Islands, Martinique, Guadeloupe, Saint Barthélemy, Saint Pierre and Miquelon, Wallis and Futuna, French Polynesia, New Caledonia, French Southern Territories, Mayotte, Réunion, Seychelles,

Guadeloupe, Saint Barthélemy, Saint Pierre and Miquelon, Wallis and Futuna, French Polynesia, New Caledonia, French Southern Territories, Mayotte, Réunion, Seychelles.

It also includes the company size where “S” means Small, “M” means medium and “L” means large.

Serial Number	Column Name	Description	Data Type
1	work_year	It refers to the specific year when the task was carried out. In the given dataset the work_year was between 2020-2023.	The datatype for the column name work_year is **int64**.
2	experience_level	It refers to the experience_level of the data scientist that are categorised into different levels.	The datatype of the column name experience_level is **object**.
3	employment_type	It refers to the type of employment for the data scientists in the company be it full time, part time, flexible, contract or internship.	The datatype of the column name employment_type is **object.**
4	job_title	It refers to the specific title of the job in reference to the data scientist’s role.	The datatype of the column name job_title is **object**.
5	salary	It refers to the salary	The datatype of the

		provided to the specific role of data scientists.	column name salary is **int64.**
6	salary_currency	It refers to the currency in which the salary is provided.	The datatype of the column name salary_currency is **object.**
7	salary_in_usd	It denotes the salary provided in USD to the data scientists	The datatype of the column name salary_in_usd is **float64.**
8	employee_residence	It refers to the residence of the employees that works in the company.	The datatype of the column name employee_residence is **object.**
9	remote_ratio	It denotes the company remote ratio rate of the employees where one-hundred means fully remote, fifty means partly remote and zero means fully in-site.	The datatype id the column name remote_ratio is **int64.**
10	company_location	It refers to the location of where the companies are located	The datatype of the column name company_location is **object.**

11	company_size	Denotes the size of the companies located in various locations.	The datatype of the column name company_size is **object.**

Table 1: Explaining Datasets and its datatypes.

2. Data Preparation

The process of getting raw data ready for further processing and analysis is known as Data Preparation. The major steps involved in data preparation includes gathering, refining, and organizing data into a format that is suitable for ML algorithms. using dedicated data is necessary to streamline and enhance this process (Amazon Web Services, 2024).

2.1 Write a python program to load data into pandas Data Frame

```
In [1]: import pandas as pd
        import matplotlib.pyplot as plt
```

Figure 1: Import pandas and matplotlib module.

Before loading the data into pandas' data frame pandas is import using the line" import panda as pd" and pd is assigned alias. Pandas is as a well-known python library utilised for both data analysis and manipulation. Without using pandas anything related to pandas like the pandas data frame cannot be used in the Jupiter notebook. Matplotlib is imported as "import matplotlib. pyplot as plt" and is assigned with the alias plt. This is mainly used for visualising and creating plots in python and jupyter notebook.

```
In [2]: #2 Data Preparation
        # reading the DataScienceSalaries CSV file
        df = pd.read_csv("DataScienceSalaries.csv")
        df
```

Out[2]:

	work_year	experience_level	employment_type	job_title	salary	salary_currency	salary_in_usd	employee_residence	remote_ratio	company_location	c
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: Read the CSV file and load it in the data frame and display it

The code snippet "df = pd.read_csv(DataScienceSalaried.csv)" is employed to read a CSV file and store its contents in a data frame. Upon execution of this code, the data from the CSV file is read and presented within the Jupyter Notebook environment, as shown in the figure above.

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

```
In [3]: #dropping the salary and salary_currency from the columns
        df = df.drop(['salary', 'salary_currency'], axis = 1)
        df
```

Out[3]:

	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 3: Remove salary and salary_currency column.

In the figure above, the column named salary and salary_currency is dropped because a column named salary_in_usd already exist which is a combination of both salary and salary_currency. By using the code “df = df.drop([‘salary, salary_currency’], axis = 1)” the salary and salary_currency column is removed as shown In the figure above. Also, axis = 1 indicated that the operation must be carried out in the column.

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

```
In [4]: #Checking if any NaN missing values exist
        df.isnull().sum()

Out[4]: work_year            0
        experience_level     0
        employment_type      0
        job_title            0
        salary_in_usd        0
        employee_residence   0
        remote_ratio         0
        company_location     0
        company_size         0
        dtype: int64

In [5]: df.dropna(inplace=True)

In [6]: df

Out[6]:
```

	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 4: Check NaN values and drop if one exists

In the above figure, the first code is written to check if the NaN missing value exists. The code written is “df.isnull().sum()”. When displayed no NaN value is found. Also, the code “df.drop(inplace = True)” is used to remove remove the NaN missing values. But since there was no NaN missing value nothing was removed.

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

```
In [7]: #checking duplicate values in dataframe
        duplicate_values = df[df.duplicated()]
        duplicate_values
```

Out[7]:

	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 5: Check duplicate values in dataframe

In the figure above, all the duplicate values in the dataframe is checked using the code “duplicate_values = df[df.duplicated()]”. When the line is executed all the duplicate values in the data is displayed as shown in the figure.

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

```
In [8]: #Displaying all the name of the columns in the dataframe
        df.columns

Out[8]: Index(['work_year', 'experience_level', 'employment_type', 'job_title',
               'salary_in_usd', 'employee_residence', 'remote_ratio',
               'company_location', 'company_size'],
              dtype='object')

In [9]: #Showing the unique values from all the columns of DataScienceSalaries
        for columns in df.columns:
            print(df[columns].unique())

[2023 2022 2020 2021]
['SE' 'MI' 'EN' 'EX']
['FT' 'CT' 'FL' 'PT']
['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'
 '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'
 'Lead Data Engineer' 'Head of Machine Learning' 'Principal Data Analyst'
 'Principal Data Engineer' 'Staff Data Scientist' 'Finance Data Analyst']
[ 85847  30000  25500 ...  28369 412000  94665]
['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']
[100   0  50]
['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']
['L' 'S' 'M']

Type Markdown and LaTeX: α²
```

Figure 6: Display unique value of all the columns from the data frame.

Firstly, before checking the unique value from all the columns the name of all the columns in the dataframe is displayed using the code “df.columns”. After the name of all the columns was displayed a code for showing the unique values for the columns was displayed using the code “for columns in df.columns: print(df[columns].unique())” and after the line was executed the values of all the unique columns was displayed as shown in the figure above.

2.6 Rename the experience level columns as below.

SE – Senior Level/Expert

MI – Medium Level/Intermediate

EN – Entry Level

EX – Executive Level

```
In [10]: #renaming the experience level column
rename_data_values = {
    'SE': 'Senior Level/Expert',
    'MI': 'Medium Level/Intermediate',
    'EN': 'Entry Level',
    'EX': 'Executive Level'
}
df.replace(rename_data_values, inplace = True)
```

```
In [11]: df
```

Out[11]:

	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 7: Rename the experience level column

In the figure above the SE, MI, EN and EX experience level column is renamed as **Senior Level/Expert, Medium Level/Intermediate, Entry Level, Executive Level** by using the code:

"rename_data_values = {

'SE': 'Senior Level/Expert',

'MI': 'Medium Level/Intermediate',

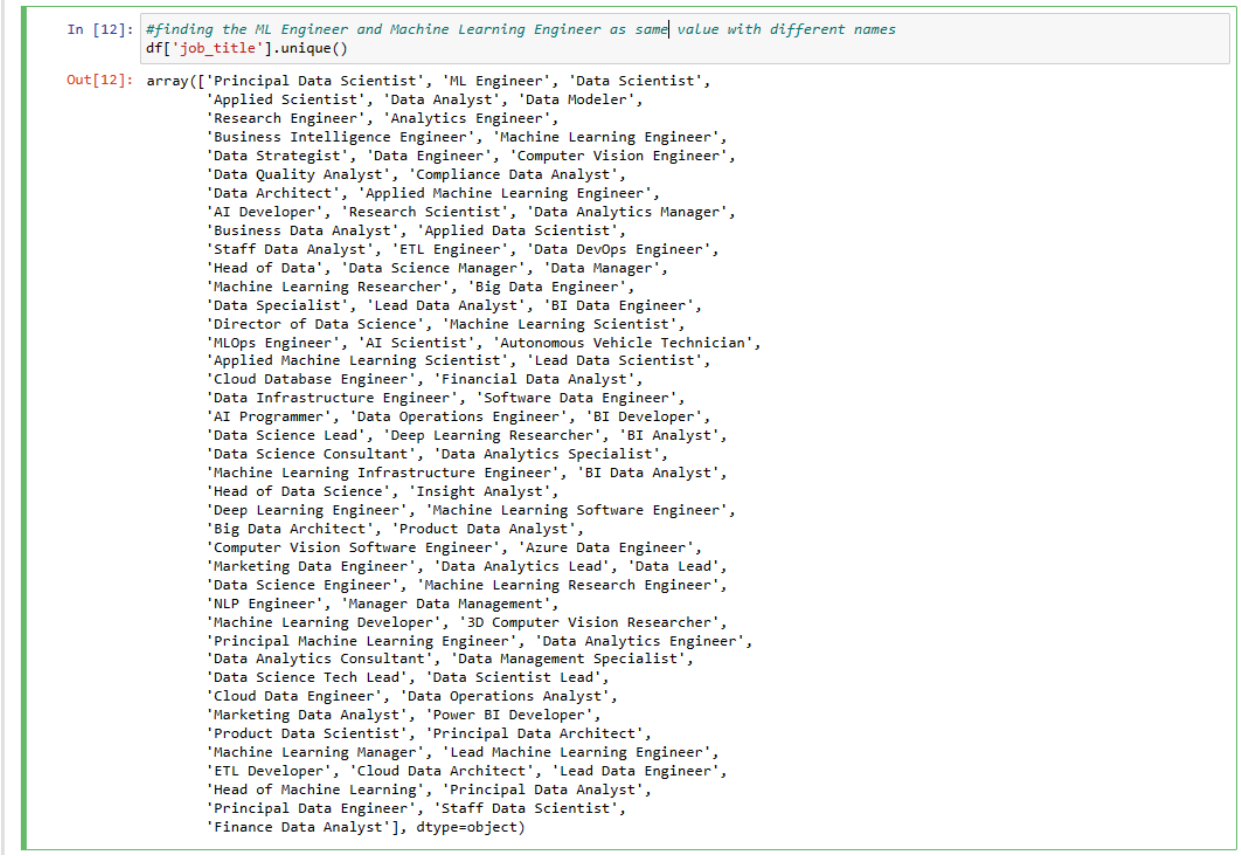
'EN': 'Entry Level',

'EX': 'Executive Level'

}

df.replace(rename_data_values, inplace = True)".

When the code is executed the specific column is renamed and displayed as shown in the figure above.



```
In [12]: #finding the ML Engineer and Machine Learning Engineer as same value with different names
         df['job_title'].unique()

Out[12]: array(['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', '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', 'Lead Data Engineer',
                'Head of Machine Learning', 'Principal Data Analyst',
                'Principal Data Engineer', 'Staff Data Scientist',
                'Finance Data Analyst'], dtype=object)
```

Figure 8: Finding the ML Engineer and Machine Learning Engineer job_title as same value with different names.

Again, when the uniqueness of the column named "job_title" was search the same values "ML Engineer" and "Machine Learning Engineer" were found with different name but they were the same as shown in the figure. So, there was a need to rename them too.

In [13]:
```
#replacing ML Engineer to Machine Learning engineer
rename = {
    'ML Engineer': 'Machine Learning Engineer'
}
df.replace(rename, inplace = True)
df
```

Out[13]:

	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	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	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 9: Renaming the column name "ML Engineer" to "Machine Learning Engineer"

The value of “ML Engineer ” was renamed to “Machine Learning Engineer” by using the code “rename = {

'ML Engineer': 'Machine Learning Engineer'

}

df.replace(rename, inplace = True)

Thus, the value was renamed and displayed as shown in the figure.

3. Data Analysis

Data analysis is a thorough process that involves inspecting, examining, purifying, transforming, and modelling data to extract valuable insights, making informed decision and support various tasks. It involves various techniques and methodologies to interpret data originating from diverse sources, which may be in structured or unstructured format (Crabtree, 2023).

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

```
In [14]: #The variable chose for calcuting the sum, mean, standard deviation, skewness and kurtosis is 'salary_in_usd'

         df1_sum = df['salary_in_usd'].sum()
         df1_sum

Out[14]: 516576814
```

Figure 10: Summary statistics of sum.

The variable that was chosen for performing the summary statistics of sum was “salary_in_usd”. The line of code “df1_sum = df['salary_in_usd'].sum()” calculates the sum of the value in the column name “salary_in_usd” and assigned it to the variable named “df1_sum” as shown in the figure above. After the code was executed, the value was displayed below.

```
In [15]: #Calculating mean for the value in "salary_in_usd"
         df1_mean = df['salary_in_usd'].mean()
         df1_mean

Out[15]: 137570.38988015978
```

Figure 11: Summary statistics of mean.

The variable that was chosen for performing the summary statistics of mean was “salary_in_usd”. The line of code “df1_mean = df['salary_in_usd'].mean()” calculates the mean of the value in the column name “salary_in_usd” and assigned it to the variable named “df1_mean” as shown in the figure above. After the code was executed, the value was displayed.

```
In [16]: #Calculating standard deviation for the value in "salary_in_usd"
         df1_standard_deviation = df['salary_in_usd'].std()
         df1_standard_deviation

Out[16]: 63055.625278224084
```

Figure 12: Summary Statistics for standard deviation.

The variable that was chosen for performing the summary statistics of standard deviation was “salary_in_usd”. The line of code “df1_standard_deviation = df['salary_in_usd'].std()” calculates the standard deviation of the value in the column name “salary_in_usd” and assigned it to the variable named “df1_standard_deviation” as shown in the figure above. After the code was executed, the value was displayed.

```
In [17]: #Calculating skewness for the value in "salary_in_usd"
         df1_skewness = df['salary_in_usd'].skew()
         df1_skewness

Out[17]: 0.5364011659712974
```

Figure 13: Summary Statistics for Skewness.

The variable that was chosen for performing the summary statistics of skewness was “salary_in_usd”. The line of code “df1_skewness = df['salary_in_usd'].skew()” calculates the skewness of the value in the column name “salary_in_usd” and assigned it to the variable named “df1_ skewness” as shown in the figure above. After the code was executed, the value was displayed.

```
In [18]: #Calculating kurtosis for the value in "salary_in_usd"
         df1_kurtosis = df['salary_in_usd'].kurtosis()
         df1_kurtosis

Out[18]: 0.8340064594833612
```

Figure 14: Summary Statistics for Kurtosis

The variable that was chosen for performing the summary statistics of Kurtosis was “salary_in_usd”. The line of code “df1_kurtosis = df['salary_in_usd']. kurtosis ()” calculates the kurtosis of the value in the column name “salary_in_usd” and assigned it to the variable named “df1_ kurtosis” as shown in the figure above. After the code was executed, the value was displayed.

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

```
In [19]: #calculating the correlation of all variable
         df.select_dtypes(include=['int64', 'float64']).corr()
```

Out[19]:

	work_year	salary_in_usd	remote_ratio
work_year	1.00000	0.228290	-0.236430
salary_in_usd	0.22829	1.000000	-0.064171
remote_ratio	-0.23643	-0.064171	1.000000

Figure 15: Calculating correlation of all variables.

The line of codes “df.select_dtypes(include=['int64', 'float64']).corr()” does the work of calculating the correlation matrix for the datatype integer and float in the data frame. It only allows to calculate the correlation of the said datatypes and show the correlation matrix between them. After the code was executed, the correlation matrix between **work_year, salary_in_usd and remote_ratio** was displayed as shown in the figure. The value of remote_ratio **one** is the highest, which means they are the most correlated.

4. Data Exploration

Data exploration is the first step in data analysis that involves utilization of data visualization tools and statistical methods to uncover dataset initial pattern and characteristics. In the Data exploration phase, the raw data goes through examination in both manual process and automated exploration methods to inspect the dataset visually. identify patterns and ascertain connections between various variables (Robinson, et al., 2023).

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

```
In [20]: #4. Data Exploration
         #finding top 15 jobs using job_title and salary by shorting the salary_in_usd value in ascending order
         top15_jobs = df['job_title'].value_counts().head(15)
         top15_jobs

Out[20]: job_title
         Data Engineer                1040
         Data Scientist                840
         Data Analyst                  612
         Machine Learning Engineer     323
         Analytics Engineer            103
         Data Architect                101
         Research Scientist             82
         Applied Scientist              58
         Data Science Manager           58
         Research Engineer              37
         Data Manager                   29
         Machine Learning Scientist     26
         Data Science Consultant        24
         Data Analytics Manager         22
         Computer Vision Engineer       18
         Name: count, dtype: int64
```

Figure 16: Finding top 15 jobs

The line of codes “top15_jobs = df['job_title'].value_counts().head(15)” is used to count how many times each different job title occurs in the “job_title” column and it selects the top 15 job title that appears the most and assigned it in the variable named “top15_jobs”. After the code was executed, the top 15 jobs were displayed as shown in the figure.

```
In [42]: #plotting the bar graph of top 15 jobs with sales
         top15_jobs.plot(kind='bar', color='green')
         plt.title('Top 15 Jobs')
         plt.xlabel('Job Title')
         plt.ylabel('Count')
         plt.xticks(rotation=45,  ha='right')
         plt.yticks(rotation = 45)
         plt.show()
```

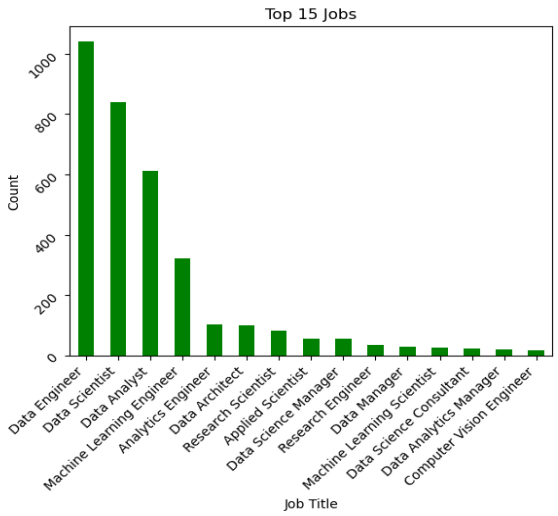


Figure 17: Plotting bar graph of top 15 jobs.

The line of codes “top15_jobs.plot(kind='bar', color='green')

plt.title('Top 15 Jobs'): used to set the title of the plot

plt.xlabel('Job Title'): Set the x-axis label to “Job Title”

plt.ylabel('Count'): Set the y-axis label to “Count”

plt.xticks(rotation=45, ha='right'): rotate 45 degree in x-axis label

plt.yticks(rotation = 45): Rotates y-axis label to 45 degree

plt.show() “ is used to show the plot of top 15 job with their respective counts.

When the code gets executed, the bar graph plot was shown with top 15 jobs where the top job being **Data Engineer** followed by data scientist, data analyst and so on.

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

```
In [22]: #Firstly calculating the salary of each job title and shorting it in descending order
         highestsalary = df.groupby('job_title')['salary_in_usd'].max().sort_values(ascending = False).head(15)
         highestsalary

Out[22]: job_title
         Research Scientist                    450000
         Data Analyst                          430967
         AI Scientist                          423834
         Applied Machine Learning Scientist    423000
         Principal Data Scientist              416000
         Data Scientist                        412000
         Data Analytics Lead                   405000
         Applied Data Scientist                380000
         Data Architect                        376080
         Data Science Tech Lead                375000
         Machine Learning Software Engineer    375000
         Director of Data Science              353200
         Applied Scientist                     350000
         Computer Vision Engineer              342810
         Machine Learning Engineer             342300
         Name: salary_in_usd, dtype: int64

In [23]: #showing the highest salary only
         highest_salary = highestsalary.max()
         highest_salary

Out[23]: 450000
```

Figure 18: Shorting salaries in descending order.

In the figure above, the salary for each job title is calculated and shorted in descending order. The line of code **"highestsalary = df.groupby('job_title')['salary_in_usd'].max().sort_values(ascending = False).head(15)"** does so by grouping the data frame by "job_title" column and selects the "salary_in_usd" column. When the code is executed, "job_title" **Research Scientist** was shown as the job with the highest salary of 450000.

```
In [24]: #plotting a bar diagram for job having highest salary
highestsalary.plot(kind='bar', color='red')
plt.title('Job having highest salary')
plt.xlabel('Job Title')
plt.ylabel('Salary (USD)')
plt.xticks(rotation=45, ha='right')
plt.show()
```

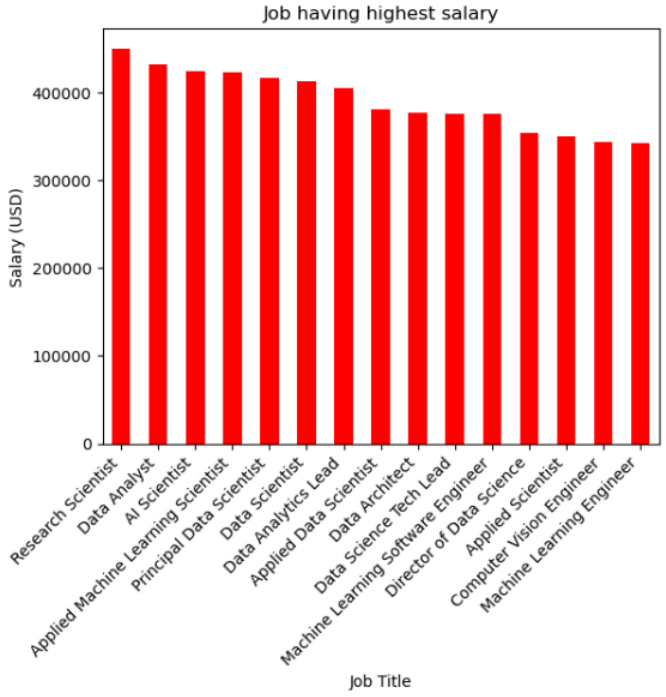


Figure 19: Bar graph for the job_title with highest salary.

The line of code “highestsalary.plot(kind='bar', color='red')

plt.title('Job having highest salary'): used to set the title of the plot

plt.xlabel('Job Title'): Set the x-axis label to “Job Title”

plt.ylabel('Salary (USD)'): Set the Y-axis label to “Salary (USD)”

plt.xticks(rotation=45, ha='right'): rotate 45 degree in x-axis label

plt.show()” is used to generate a bar plot with the highest salary of **“Research Scientist, 450000 ”.** When the code gets executed, a bar was plotted with x-axis and y-axis that shows the highest salary with the job title.

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

```
In [25]: #average salary based on experience level
         experience_level_salary = df.groupby('experience_level')['salary_in_usd'].mean()#mean() is used because it calculates the average
         experience_level_salary

Out[25]: experience_level
         Entry Level                  78546.284375
         Executive Level             194930.929825
         Medium Level/Intermediate   104525.939130
         Senior Level/Expert         153051.071542
         Name: salary_in_usd, dtype: float64
```

Figure 20: Salary based on experience_level

The line of code: "experience_level_salary = df.groupby('experience_level')['salary_in_usd'].mean()" is used to calculate the average value salary for each experience level. The code is used to group the data frame by experience_level column and select the salary_in_usd column.

```
In [26]: #Plotting a bar diagram for "Salary Based on experience level"
experience_level_salary.plot (kind = 'bar', color='black')
plt.xlabel('Experience Level')
plt.ylabel('Salary in USD')
plt.title('Salary based on experience level')
plt.xticks(rotation=45)
plt.show()
```

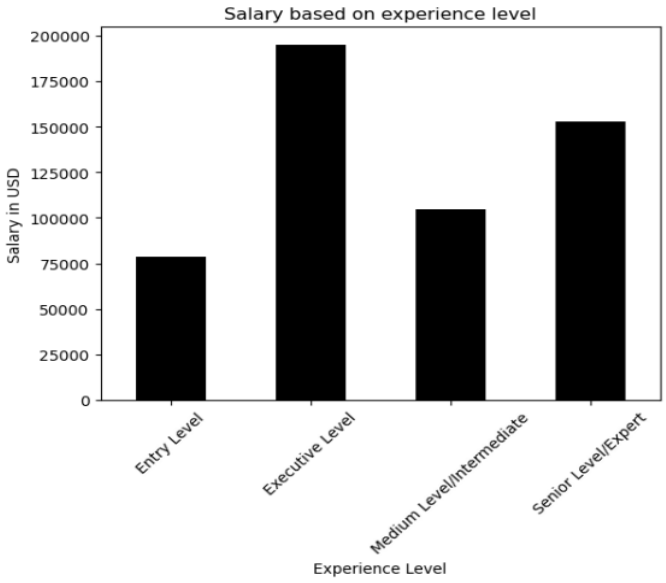


Figure 21: Bar Diagram of Salary based on experience_level.

From the bar graph it is shown that the “Executive level” experience_level salary is the highest around 200,000 USD. From the code “experience_level_salary.plot (kind = 'bar', color='black')

plt.xlabel('Experience Level'): Set the x-axis label to 'Experience Level'

plt.ylabel('Salary in USD'): Set the x-axis label to 'Salary in USD

plt.title('Salary based on experience level'): used to set the title of the plot

plt.xticks(rotation=45): rotate 45 degree in x-axis label

plt.show()” shows the salary based on different experience_level in the bar diagram.

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

```
In [44]: # Plotting histogram of salary_in_usd
plt.hist(df['salary_in_usd'], color='skyblue', edgecolor = 'black', alpha=0.9)
plt.title('Histogram for Salary in USD')
plt.xlabel('Salary USD')
plt.ylabel('Frequency')
plt.show()
```

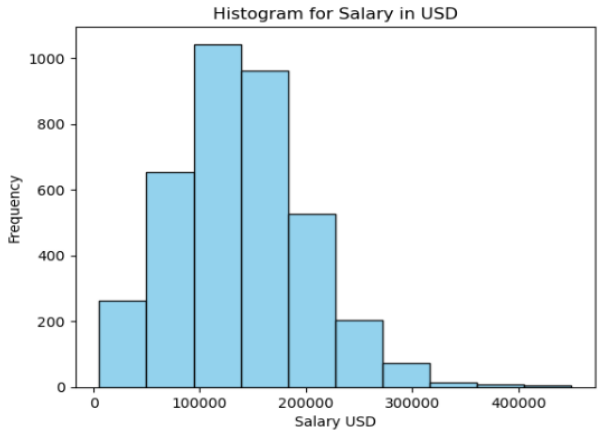


Figure 22: Histogram Plot for salary_in_usd

In the figure a histogram is plotted for the salary_in_usd column. The frequency in the histogram refers to the number of occurrences of each salary range. The line of code “plt.hist(df['salary_in_usd'], color='skyblue', edgecolor = 'black', alpha=0.9)

plt.title('Histogram for Salary in USD'): Title for histogram

plt.xlabel('Salary USD'): Set x-label to (“Salary USD”)

plt.ylabel('Frequency'): Set y-label to (“Frequency”)

plt.show()” plotted a histography for salary_in_usd column.

From the graph it is visible that the data clearly shows that for salaries between 0 and 50,000 USD, they occur roughly 250 times in the dataset. Moving up to the 50,000-100,000 USD range, a noticeable increase was seen, with salaries in the dataset appearing around 600 to 650 times. Moving further into higher income brackets, particularly between 100,000 and 200,000 USD, the frequency varies significantly, ranging from 500 to well over 1000 occurrences in the dataset. In the range of 200,000 to 300,000 USD, salaries are observed between 100 and 500 times in the dataset. However, it's interesting

to note that on reaching the 300,000-400,000 USD range, frequency of occurrences drop notably, ranging from around 100 to none in the dataset.

```
In [28]: # Plotting box plot of salary in usd
         plt.boxplot(df['salary_in_usd'])
         plt.title('Box Plot of Salary in USD')
         plt.xlabel('Salary in USD')
         plt.show()
```

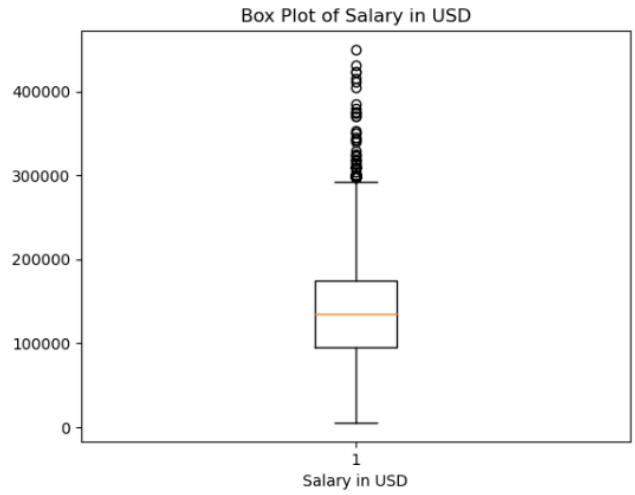


Figure 23: Box plot of salary in USD.

The figure shows the box plot of "salary_in_usd". The box plot is used in providing visual summary of important statistics like quartile and medians within the salary data.

The code "plt.boxplot(df['salary_in_usd'])

plt.title('Box Plot of Salary in USD')

plt.xlabel('Salary in USD')

plt.show()" is used to show the box plot of salary_in_usd.

The box plot in the figure above reveals several key insights. It shows that 50% of salaries fall within the range of approximately 100,000 USD to 200,000 USD, with the lower 25% (Q1) and upper 25% (Q3) defining the interquartile range. The median, depicted by the yellowish line, divides the data equally. Outliers, lying beyond the extended whisker, are noticeably skewed to one side, representing extreme values like the highest salaries in the context of the coursework.

5. Conclusion

The coursework deals with applying programming knowledge and skills to the task related to data analysis. The coursework involves the data science salary analysis which involves one's skills for problem – solving and critical thinking and evaluation. The main tasks of the coursework was to write a python program in jupyter notebook and to prepare a technical report on data understanding, data exploration and initial analysis of the given data set.

The primary task of the coursework was to obtain a better understanding of the elements and the data that influence the salaries of data scientists and to discover any regulatory or tendencies within the data. The data understanding, data preparation, data analysis and data exploration tasks were carried out and solved in jupyter notebook successfully. For the same values of the job titles with different names the renaming process was carried out. Top fifteen jobs' titles were shown, salary of the experience level was shown, correlation between all variable was shown, and the summary statistics of sum, mean, standard deviation, skewness, and kurtosis of the column name salary in USD was shown. Also, from the given dataset the histogram and box plot of salary in USD was shown, read, and analysed.

Thus, all the tasks related to data analysis were successfully carried out using python program. After the execution of the code, there was detailed understanding of the elements influencing the salaries of data scientists. Hence, after the successful completion of code the data was prepared for further data mining and analysis successfully.

6. References

Amazon Web Services, 2024. *What is Data Preparation?.* [Online]
Available at: https://aws.amazon.com/what-is/data-preparation/#:~:text=Data%20preparation%20is%20the%20process,exploring%20and%20visualizing%20the%20data.
[Accessed 11 may 2024].

Crabtree, M., 2023. *What is Data Analysis?.* [Online]
Available at: https://www.datacamp.com/what-is-data-analysis-expert-guide
[Accessed 12 may 2024].

Robinson, S., Hanna, K. T. & Biscobing, J., 2023. *Data Exploration.* [Online]
Available at: https://www.techtarget.com/searchbusinessanalytics/definition/data-exploration
[Accessed 12 may 2024].

Verjus, M. & Gigandet, Y., 2022. Project in Data Analytics for Decision Making. In: *Data understanding and preparation.* chicago: bookdown, pp. 12-52.