Data Science Job analysis

August 17, 2023

DATASET LINK :https://www.kaggle.com/datasets/niyalthakkar/data-science-jobs-analysis

[1]:
```
# Write this command on top to autocomplete text , it improves working speed
%config Completer.use_jedi = False
```

1 IMPORT LIBRARIES

[2]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

[3]:
```
df = pd.read_csv("ds.salaries.csv")
df.head()
```

[3]:

	Unnamed: 0	work_year	experience_level	employment_type	job_title	salary	salary_currency	salary_in_usd	employee_residence	remote_ratio	company_location	company_size
0	0	2020	MI	FT	Data Scientist	70000	EUR	79833	DE	0	DE	L
1	1	2020	SE	FT	Machine Learning Scientist	260000	USD	260000	JP	0	JP	S
2	2	2020	SE	FT	Big Data Engineer	85000	GBP	109024	GB	50	GB	M
3	3	2020	MI	FT	Product Data Analyst	20000	USD	20000	HN	0	HN	S
4	4	2020	SE	FT	Machine Learning Engineer	150000	USD	150000	US	50	US	L

2 IDENTIFYING MISSING AND DUPLICATE VALUES

[4]:
```
df.isnull().sum()
```

[4]:
```
Unnamed: 0            0
work_year             0
experience_level      0
employment_type       0
job_title             0
salary                0
salary_currency       0
salary_in_usd         0
employee_residence    0
remote_ratio          0
company_location      0
company_size          0
dtype: int64
```

[5]:
```
df.duplicated().sum()
```

[5]:
```
0
```

3 FEATURE SELECTION

[6]:
```
df.drop(columns = ['Unnamed:␣
↪0','salary','salary_currency','employee_residence','company_size'], inplace␣
↪= True)
```

[7]:
```
df.head()
```

[7]:

	work_year	experience_level	employment_type	job_title	salary_in_usd	remote_ratio	company_location
0	2020	MI	FT	Data Scientist	79833	0	DE
1	2020	SE	FT	Machine Learning Scientist	260000	0	JP
2	2020	SE	FT	Big Data Engineer	109024	50	GB
3	2020	MI	FT	Product Data Analyst	20000	0	HN
4	2020	SE	FT	Machine Learning Engineer	150000	50	US

4 DATA UNDERSTANDING

[8]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 607 entries, 0 to 606
Data columns (total 7 columns):
 #   Column            Non-Null Count  Dtype 
---  ------            --------------  ----- 
 0   work_year         607 non-null    int64 
 1   experience_level  607 non-null    object
 2   employment_type   607 non-null    object
 3   job_title         607 non-null    object
 4   salary_in_usd     607 non-null    int64 
 5   remote_ratio      607 non-null    int64 
 6   company_location  607 non-null    object
dtypes: int64(3), object(4)
memory usage: 33.3+ KB
```

[9]:
```
df.describe()
```

[9]:

	work_year	salary_in_usd	remote_ratio
count	607.000000	607.000000	607.00000
mean	2021.405272	112297.869852	70.92257
std	0.692133	70957.259411	40.70913
min	2020.000000	2859.000000	0.00000
25%	2021.000000	62726.000000	50.00000
50%	2022.000000	101570.000000	100.00000
75%	2022.000000	150000.000000	100.00000
max	2022.000000	600000.000000	100.00000

4.1 Observation based on above figure:

4.1.1 1) min salary is 2859 USD

4.1.2 2) avg salary is 1,12,297 USD

4.1.3 3) max salary is 6,00,000 USD

[10]:
```
# CATEGORICAL VARIABLE UNDERSTANDING
df['experience_level'].value_counts()
```

[10]:
```
SE    280
MI    213
EN     88
EX     26
Name: experience_level, dtype: int64
```

[11]:
```
df['employment_type'].value_counts()
```

[11]:
```
FT    588
PT     10
CT      5
FL      4
Name: employment_type, dtype: int64
```

[12]:
```
# FULL FORM OF ABBREVIATIONS
#     'SE': 'Senior',
#     'MI': 'Mid',
#     'EN': 'Entry',
#     'EX': 'Executive'

#     'FT': 'Full-time',
#     'PT': 'Part-time',
#     'CT': 'Contract',
#     'FL': 'Freelance
```

[13]:
```
df['remote_ratio'].value_counts()
```

[13]:
```
100    381
0      127
50      99
Name: remote_ratio, dtype: int64
```

5 PREPROCESSING

[14]:
```
# TRANSFORMING ABBREVIATIONS INTO FULL FORM FOR BETTER UNDERSTANDING
df['employment_type'] =  df['employment_type'].map({"FT":"Full Time","PT":"Part␣
 ↪Time","CT":"Contract","FL":"Freelance"})
df['experience_level'] =  df['experience_level'].map({"SE":"Senior","MI":
 ↪"Mid","EN":"Entry","EX":"Executive"})

# CONVERTING REMOTE_RATIO VALUES INTO WORK TYPE NAMES LIKE : REMOTE,HYBRID AND␣
 ↪ONSITE
df['remote_ratio'] = df['remote_ratio'].map({100:"Remote",0:"Onsite",50:
 ↪"Hybrid"})
```

[15]:
```
df.head()
```

[15]:

	work_year	experience_level	employment_type	job_title	salary_in_usd	remote_ratio	company_location
0	2020	Mid	Full Time	Data Scientist			
1	2020	Senior	Full Time	Machine Learning Scientist			
2	2020	Senior	Full Time	Big Data Engineer			
3	2020	Mid	Full Time	Product Data Analyst			
4	2020	Senior	Full Time	Machine Learning Engineer			

```
0          79833       Onsite              DE
1         260000       Onsite              JP
2         109024       Hybrid              GB
3          20000       Onsite              HN
4         150000       Hybrid              US
```

[16]:
```
# NOW LETS CONVERT THE LOCATIONS CODE INTO FULL NAME
# INSTALL CONVERTER FOR THIS TASK

! pip install country_converter
```

```
Requirement already satisfied: country_converter in c:\users\asus vivobook
14\anaconda3\lib\site-packages (1.0.0)
Requirement already satisfied: pandas>=1.0 in c:\users\asus vivobook
14\anaconda3\lib\site-packages (from country_converter) (1.4.2)
Requirement already satisfied: pytz>=2020.1 in c:\users\asus vivobook
14\anaconda3\lib\site-packages (from pandas>=1.0->country_converter) (2021.3)
Requirement already satisfied: numpy>=1.18.5 in c:\users\asus vivobook
14\anaconda3\lib\site-packages (from pandas>=1.0->country_converter) (1.21.5)
Requirement already satisfied: python-dateutil>=2.8.1 in c:\users\asus vivobook
14\anaconda3\lib\site-packages (from pandas>=1.0->country_converter) (2.8.2)
Requirement already satisfied: six>=1.5 in c:\users\asus vivobook
14\anaconda3\lib\site-packages (from python-
dateutil>=2.8.1->pandas>=1.0->country_converter) (1.16.0)
```

[17]:
```
import country_converter as coco
```

[18]:
```
# NOTE: IT AUTOMATICALLY DETECTS CODE AND CONVERT INTO FULL NAME OF COUNTRIES
df['company_location'] = coco.convert(df['company_location'],to =␣
 ↪'name_short',not_found = None)
```

[19]:
```
df.head()
```

[19]:
```
   work_year experience_level employment_type                   job_title  \
0       2020              Mid       Full Time              Data Scientist
1       2020           Senior       Full Time  Machine Learning Scientist
2       2020           Senior       Full Time           Big Data Engineer
3       2020              Mid       Full Time        Product Data Analyst
4       2020           Senior       Full Time   Machine Learning Engineer

   salary_in_usd remote_ratio company_location
0          79833       Onsite          Germany
1         260000       Onsite            Japan
2         109024       Hybrid   United Kingdom
3          20000       Onsite         Honduras
4         150000       Hybrid    United States
```

[20]:
```
df.rename(columns={'salary_in_usd': 'salary'}, inplace=True)
```

BELOW IS THE FINAL CLEANED DATA

[21]:
```
df.head()
```

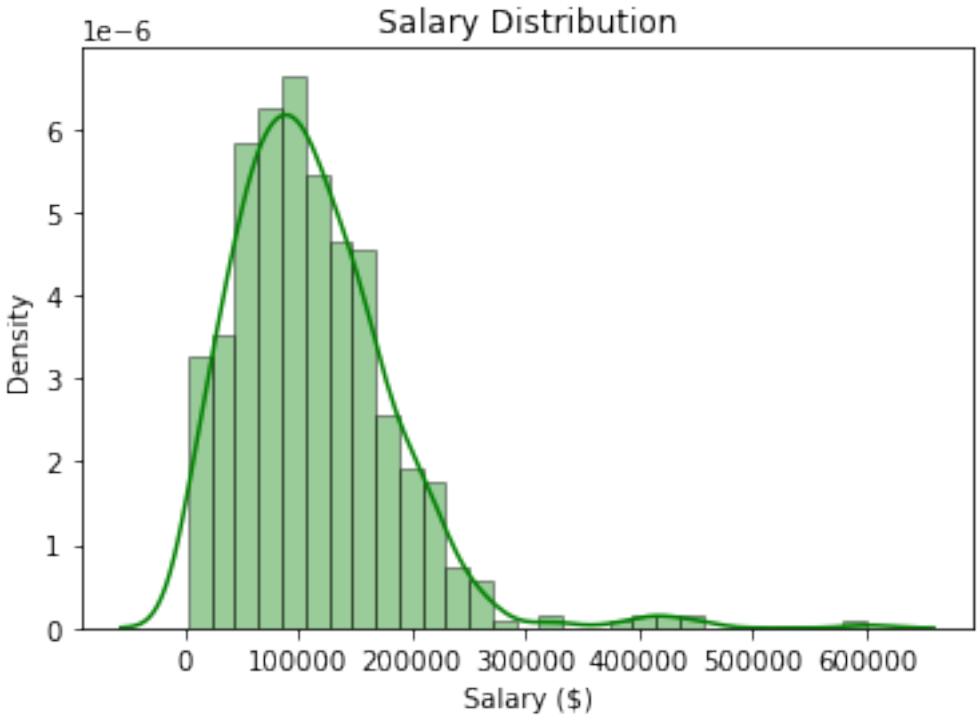
[21]:

	work_year	experience_level	employment_type	job_title	salary	remote_ratio	company_location
0	2020	Mid	Full Time	Data Scientist	79833	Onsite	Germany
1	2020	Senior	Full Time	Machine Learning Scientist	260000	Onsite	Japan
2	2020	Senior	Full Time	Big Data Engineer	109024	Hybrid	United Kingdom
3	2020	Mid	Full Time	Product Data Analyst	20000	Onsite	Honduras
4	2020	Senior	Full Time	Machine Learning Engineer	150000	Hybrid	United States

6 PERFORMING EDA (Exploratory Data Analysis)

[22]:
```
# LETS PLOT THE SALARY DISTRIBUTION

sns.distplot(df['salary'],kde=True,color="green",hist_kws={"edgecolor":"black"})
plt.title("Salary Distribution")
plt.xlabel("Salary ($)")
plt.show()
```



6.0.1 Obsevation based on above plot :

1) There are few people whose salary is greater than 3L US Dollar

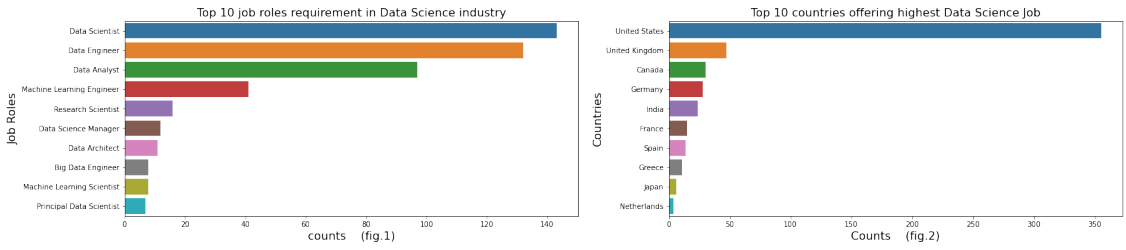
7

QUES1 : what are top 10 countries offering highest data science job?

QUES 2: what are the top 10 job openings in data science?

[23]:
```
# what are top 10 countries offering highest data science job?
plt.figure(figsize=(25,5))
plt.subplot(1,2,2)
top_country_count = df['company_location'].value_counts()[:10]
sns.barplot(top_country_count,top_country_count.index)
plt.title("Top 10 countries offering highest Data Science␣
 ↪Job",fontdict={"fontsize":16})
plt.xlabel("Counts    (fig.2)",fontdict={"fontsize":16})
plt.ylabel("Countries",fontdict={"fontsize":16})

# what are the top 10 job openings in data science?
plt.subplot(1,2,1)
top_jobs_count = df['job_title'].value_counts()[:10]
sns.barplot(top_jobs_count,top_jobs_count.index)
plt.title("Top 10 job roles requirement in Data Science␣
 ↪industry",fontdict={"fontsize":16})
plt.xlabel("counts    (fig.1)",fontdict={"fontsize":16})
plt.ylabel("Job Roles",fontdict={"fontsize":16})
plt.show()
```



7.0.1 Observations based on fig.1

1) Data Scientist is top 1st job role offered by data science

2) Data Engineer is 2nd top role among all job roles in data science

3) No. of openings in Data Analyst role are less than Data Eengineer

7.0.2 Observations based on fig.2

1) US is top 1st country offering highest data science job

2) UK is top 2nd country offering data science job

3) India is top 5th country offering data science job

QUES3: WHAT ARE THE TOP 10 COUNTRIES PAYING HIGHEST SALARY?

QUES4: WHAT ARE THE TOP 10 JOB ROLES THE HIGHEST AVERAGE SALARY?

[24]:
```
# lets group the salary based on country and then find the mean
# arrange all the data in descending order then apply slicing on top 10 values

top_country_salary = df.groupby('company_location')['salary'].agg('mean').
 ↪sort_values(ascending = False)[:10]
top_country_salary
```

[24]:
```
company_location
Russia                  157500.000000
United States           144055.261972
New Zealand             125000.000000
Israel                  119059.000000
Japan                   114127.333333
Australia               108042.666667
Iraq                    100000.000000
United Arab Emirates    100000.000000
Algeria                 100000.000000
Canada                   99823.733333
Name: salary, dtype: float64
```

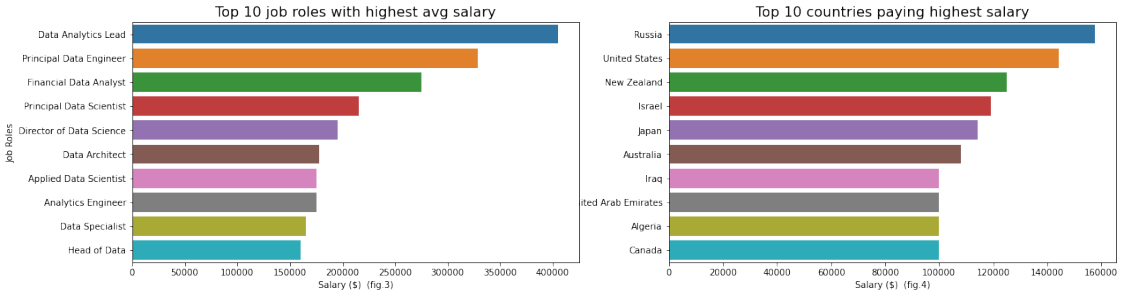
[25]:
```
# Top 10 countries paying highest avg salaries

plt.figure(figsize=(20,5))
plt.subplot(1,2,2)
sns.barplot(top_country_salary , top_country_salary.index)
plt.title("Top 10 countries paying highest salary",fontdict={"fontsize":16})
plt.xlabel("Salary ($)  (fig.4)",fontdict={"fontsize":10})
plt.ylabel("Countries",fontdict={"fontsize":10})

# Top 10 job roles with highest avg salary

plt.subplot(1,2,1)
top_jobtitle_salary = df.groupby('job_title')['salary'].mean().
 ↪sort_values(ascending = False)[:10]
```

```
sns.barplot(top_jobtitle_salary, top_jobtitle_salary.index)
plt.title("Top 10 job roles with highest avg salary",fontdict={"fontsize":16})
plt.xlabel("Salary ($)  (fig.3)",fontdict={"fontsize":10})
plt.ylabel("Job Roles",fontdict={"fontsize":10})
plt.show()
```



8 Observation based on fig.3

1) Data Analytics Lead is the top 1st role with the highest avg salary 4L US Dollar

2) Principal Data Engineer, Financial Data Analyst, Principal Data Scientist and Director of Data Science are among top 5 highest avg salary job roles.

9 Observation based on fig.4

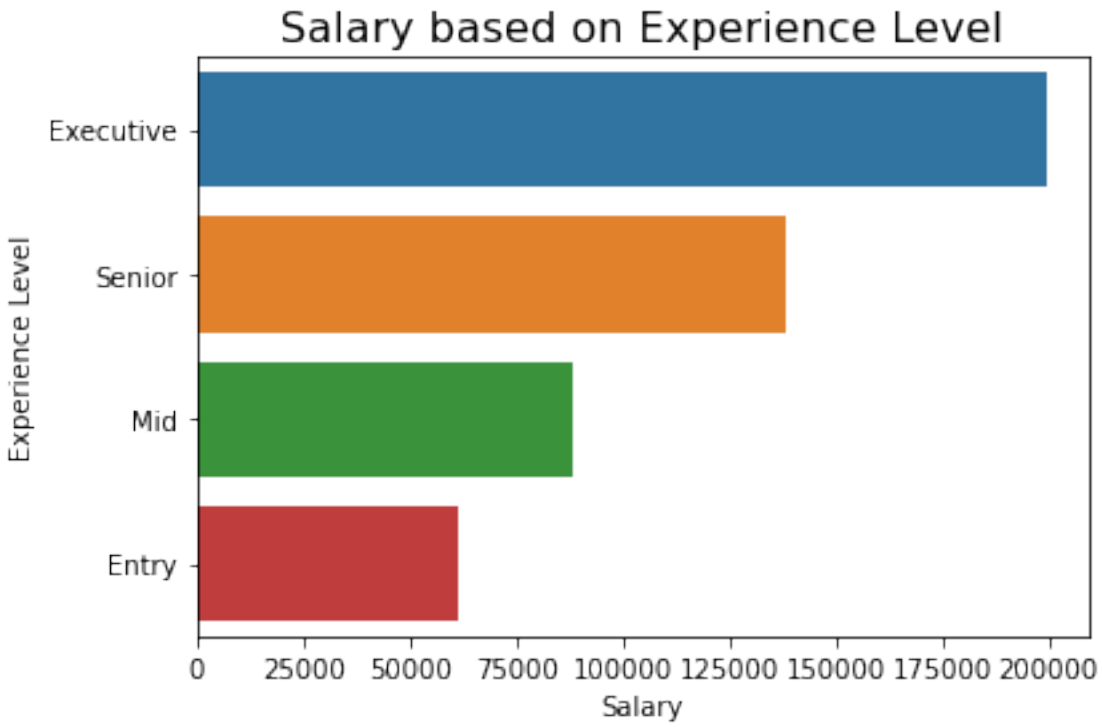
1) Russia is the top 1st country pays highest avg salary for data science role

2) The US, New Zealand, Israel, and Japan are among the top 5 countries with the highest average salaries for data science roles

[26]:
```
# Salary based on Experience level

experienceVssalary = df.groupby('experience_level')['salary'].mean().
  ↪sort_values(ascending=False)
```

[27]:
```
sns.barplot(experienceVssalary,experienceVssalary.index)
plt.xlabel("Salary",fontdict={"fontsize":10})
plt.ylabel("Experience Level",fontdict={"fontsize":10})
plt.title("Salary based on Experience Level",fontdict={"fontsize":16})
plt.show()
```



QUES: WHICH TYPE OF WORK HAVING THE LARGEST JOB VACCANCIES ?

[28]:
```
df['remote_ratio'].value_counts()
```

[28]:
```
Remote    381
Onsite    127
Hybrid     99
Name: remote_ratio, dtype: int64
```

10 Observations:

1) The category of remote work has the largest count of job vacancies