

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/369453742>

# Visualization and Explorative Data Analysis

Article · March 2023

---

CITATIONS

0

---

READS

154

2 authors, including:



[Sreeramana Aithal](#)

Srinivas University India

1,275 PUBLICATIONS 14,541 CITATIONS

SEE PROFILE

# Visualization and Explorative Data Analysis

Dr. Tarak Hussain<sup>1</sup>, Dr. PS Athal<sup>2</sup>

<sup>1</sup>PDF Scholar, Institute of Computer Science & Information Science, Srinivas University, Mangalore, India

<sup>2</sup>Professor, Institute of Computer Science & Information Science Srinivas University, Mangalore, India

---

## ABSTRACT

Data need to be analyzed so as to produce good result. Using the result decision can be generated. For example recommendation system, ranking of the page, demand forecasting, prediction of purchase of the product etc. There are some leading companies where the review of the customer plays a great role to analyze the factor which influences the review rating. We have used exploratory data analysis (EDA) where data interpretations can be done in row and column format. We have used python for data analysis. It is object oriented, interpreted and interactive programming language. It is open source with rich sets of libraries like pandas, MATplotlib, seaborn etc. We have used different types of charts and various types of parameter to analyze data science job salarydata sets taken from kaggle.

**Keywords:** Exploratory Data Analysis, Kaggle, Visualization.

---

## INTRODUCTION

Exploratory data analysis (EDA) is an approach to analyze and summarize data in order to gain insights and identify patterns or trends. It is often the first step in data analysis and is used to understand the structure of the data, detect outliers and anomalies, and inform the selection of appropriate statistical models.

It is an approach to analyze data that involves exploring and summarizing the data to understand its characteristics, properties, and underlying structure. It is an iterative and flexible process that allows data analysts to gain insights into the data and identify patterns, trends, and relationships that may not be immediately apparent.

EDA involves a variety of techniques, including visualization, statistical summaries, and hypothesis testing, to help analysts gain a better understanding of the data. The goal of EDA is to uncover interesting features of the data and generate hypotheses that can be tested with further analysis.

One of the key benefits of EDA is that it helps analysts identify potential problems with the data, such as missing or incorrect values, outliers, or inconsistencies. By detecting these issues early on, analysts can take steps to address them and ensure the accuracy and reliability of their analyses.

Some of the common techniques used in EDA include:

- a) Visualization:** Visualizing data using plots and graphs is an effective way to explore the data and identify patterns and relationships. Common visualization techniques include scatter plots, box plots, histograms, and heat maps.
- b) Statistical summaries:** Calculating summary statistics such as mean, median, variance, and correlation coefficients can provide insights into the data and help identify patterns and relationships.
- c) Hypothesis testing:** Testing hypotheses about the data can help analysts determine whether observed patterns or relationships are statistically significant or simply due to chance.
- d) Data transformation:** Transforming data through normalization, standardization, or other techniques can help improve the accuracy of analysis and facilitate comparison across different datasets.

Overall, EDA is an important step in the data analysis process that helps analysts gain a deeper understanding of the data and generate hypotheses that can guide further analysis. By exploring the data in a systematic and iterative way, analysts can identify patterns, relationships, and potential issues that may not be immediately apparent, and develop more accurate and reliable models and insights.

Here we have taken the data from kaggle containing jobs description of computers(the most common jobs)

## LITERATURE REVIEW

Exploratory data analysis (EDA) is an important first step in the data analysis process that involves exploring and summarizing data to identify patterns and relationships. EDA can help data analysts gain a deeper understanding of the data, detect potential problems or errors, and generate hypotheses that can guide further analysis.

Numerous studies have highlighted the importance of EDA in data analysis. In a study published in the journal "Statistical Science," John W. Tukey, one of the pioneers of EDA, argued that EDA was essential to understanding data and identifying patterns that might not be apparent through more formal statistical methods. Tukey emphasized the need for flexible, iterative, and exploratory approaches to data analysis that allowed analysts to generate hypotheses and explore the data in a systematic and flexible way.

Another study published in the journal "Computational Statistics and Data Analysis" examined the effectiveness of different visualization techniques in EDA. The study found that visualizations, such as scatter plots and box plots, were effective in identifying patterns and relationships in the data, and that interactive visualizations could be particularly useful for exploring large datasets.

In addition to these studies, numerous books and articles have been written on EDA and its importance in data analysis. For example, "Exploratory Data Analysis" by Tukey provides a comprehensive overview of EDA techniques and their applications in various fields. Other works, such as "Data Analysis Using Regression and Multilevel/Hierarchical Models" by Gelman and Hill, emphasize the importance of EDA in model selection and development.

Overall, the literature highlights the importance of EDA in data analysis and emphasizes the need for flexible and iterative approaches to exploring and summarizing data. By generating hypotheses and exploring the data in a systematic and iterative way, analysts can gain a deeper understanding of the data and develop more accurate and reliable models and insights.

## METHODOLOGY

### 1. Data Exploration:

Data exploration is the process of examining and understanding data to gain insights and identify patterns or relationships. It involves using various techniques and tools to analyze data, including statistical analysis, visualization, and summarization.

**2. Data Cleaning:** Data cleaning is the process of identifying and correcting errors, inconsistencies, and inaccuracies in a dataset. It involves several tasks, including handling missing data, correcting data format, detecting and removing outliers, dealing with duplicates, and addressing inconsistencies in data values.

We used imputing missing data using techniques such as mean imputation, removing duplicates, standardizing formats, and correcting errors in data values.

**3. Data Modeling:** It refers to the process of creating a mathematical representation of the data that can be used to make predictions or classify new data points. The goal of data modeling in machine learning is to create a model that accurately captures the patterns and relationships in the data and can be used to make predictions on new, unseen data.

We used statistical method of linear regression model It involves several steps, including:

- Data preparation: cleaning, transforming, and scaling the data to prepare it for modeling.
- Feature selection: selecting the most relevant features or variables to include in the model.
- Model selection: choosing the appropriate model or algorithm for the data and problem.
- Model training: using the data to train the model and adjust its parameters.

Once the model has been trained and evaluated, it can be used to make predictions on new, unseen data.

**5. Results:** We can visualize large amount of complex data by the use of chart, graph and tables. Human brain can process information using chart, graphs. It is an easy way to convey the concept. It can identify the area which needs improvement. It can clarify the factor very well.

The sunburst chart illustrates the distribution of data science roles. The largest segment is Data Scientist at 28.5%, followed by Data Engineer at 21.5%, Data Analyst at 14.1%, and Machine Learning Engineer at 6.8%. The Machine Learning Engineer segment is further subdivided into many smaller roles, including Data Architect, Data Science Manager, Research Scientist, ML Engineer, AI Scientist, Machine Learning Scientist, BI Data Analyst, BI Developer, Data Science Consultant, Business Data Analyst, Big Data Engineer, Applied Machine Learning Scientist, Data Operations Engineer, Director of Data Science, Principal Data Scientist, Head of Data, Data Analytics Manager, Data Specialist, Data Engineering Manager, Machine Learning Infrastructure Engineer, Lead Data Engineer, Data Analytics Engineer, Data Manager, Machine Learning Developer, Bi Analyst, Cloud Data Analyst, Data Operations Analyst, Corporate Vision Software Engineer, Head of Data Science, and Data Analytics Engineer.

Page | 13

```
df.describe()
```

[75]:

	work_year	salary	salary_in_usd	remote_ratio
count	1342.000000	1.342000e+03	1342.000000	1342.000000
mean	2021.725782	2.427670e+05	123075.044709	63.822653
std	0.570913	1.089676e+06	65992.159472	45.252542
min	2020.000000	2.324000e+03	2324.000000	0.000000
25%	2022.000000	8.000000e+04	75000.000000	0.000000
50%	2022.000000	1.300000e+05	120000.000000	100.000000
75%	2022.000000	1.751000e+05	164996.000000	100.000000
max	2023.000000	3.040000e+07	600000.000000	100.000000

Figure 2. Shows statistical summary of data frame.

df.describe() is a method in Pandas library of Python that provides a statistical summary of a DataFrame object. The output of df.describe() includes the count, mean, standard deviation, minimum value, 25th percentile (Q1), median (50th percentile), 75th percentile (Q3), and maximum value for each numerical column in the DataFrame.

```
df.sample(n=10)
```

[10]:

	work_year	experience_level	employment_type	job_title	salary	salary_currency	salary_in_usd	employee_residence	remote_ratio	company_location	company_size
791	2022	MI	FT	Data Engineer	45000	EUR	47379	GR	100	GR	M
414	2022	SE	FT	Data Scientist	191475	USD	191475	US	100	US	M
239	2022	SE	FT	Data Scientist	198440	USD	198440	US	100	US	M
813	2022	MI	FT	Data Engineer	40000	GBP	49542	GB	0	GB	M
248	2022	SE	FT	Data Scientist	198440	USD	198440	US	0	US	L
423	2022	SE	FT	Data Engineer	300000	USD	300000	US	0	US	M
355	2022	SE	FT	Data Analyst	113000	USD	113000	US	100	US	M
608	2022	SE	FT	Data Science Manager	140100	USD	140100	US	100	US	L
1232	2021	EX	FT	Data Science Consultant	59000	EUR	69741	FR	100	ES	S
129	2022	SE	FT	Data Analyst	130050	USD	130050	US	100	US	M

Figure 3. shows random sample of n rows from given dataframe df

```
df.pivot_table(index='salary', columns='work_year', aggfunc={'salary': 'mean'})
```

[11]:

work_year	2020	2021	2022	salary
salary				
2324	NaN	NaN	2324.0	NaN
4000	NaN	4000.0	NaN	NaN
5000	NaN	5000.0	NaN	NaN
7500	NaN	NaN	7500.0	NaN
8000	8000.0	NaN	8000.0	NaN
...	...	...	...	...
7000000	NaN	7000000.0	NaN	7000000.0
7500000	NaN	NaN	7500000.0	NaN
8500000	NaN	8500000.0	NaN	NaN
11000000	11000000.0	11000000.0	NaN	NaN
30400000	NaN	30400000.0	NaN	NaN

437 rows x 4 columns

Figure 4. shows pivot table of data work year and salary

```
[88]: df.head
```

```
[88]: <bound method NDFrame.head of work_year experience_level employment_type job_title \
0      2022      MI      FT Machine Learning Engineer
1      2022      MI      FT Machine Learning Engineer
2      2022      MI      FT Data Scientist
3      2022      MI      FT Data Scientist
4      2022      MI      FT Data Scientist
...      ...      ...      ...      ...
1337    2023      EN      PT AI Scientist
1338    2023      MI      FT Data Analyst
1339    2023      MI      FT Data Analyst
1340    2023      MI      FT Data Scientist
1341    2023      MI      FT Data Engineer

      salary salary_currency salary_in_usd employee_residence remote_ratio \
0      130000      USD      130000      US      0
1       90000      USD       90000      US      0
2      120000      USD      120000      US     100
3      100000      USD      100000      US     100
4       85000      USD       85000      US     100
...      ...      ...      ...      ...      ...
1337     12000      USD       12000      BR     100
1338     75000      USD      75000      US      0
1339     62000      USD      62000      US      0
1340     73000      USD      73000      US      0
1341     38400      EUR      45391      NL     100

      company_location company_size
0      US      M
1      US      M
2      US      M
3      US      M
4      US      M
...      ...      ...
1337    US      S
1338    US      L
1339    US      L
1340    US      L
1341    NL      L

[1342 rows x 11 columns]>
```

Figure 5. shows first n rows of data framed

#### IV. Training Data

In machine learning, the term "training data" refers to a dataset that is used to train a machine learning algorithm or model. The training data is used to teach the algorithm or model how to recognize patterns, make predictions or classify new data based on the features or characteristics of the data.

The training data consists of a set of labeled examples, where each example contains a set of input features and the corresponding output or target value. The input features represent the independent variables or predictors, while the target value represents the dependent variable or response variable that the model is trying to predict.

During the training process, the algorithm or model is fed with the training data and it adjusts its parameters or weights to minimize the difference between its predicted output and the actual output of the training examples. The goal of the training process is to learn a function or model that can generalize well to unseen data, meaning that it can make accurate predictions on new, previously unseen data.

After the training process is complete, the trained model can be used to make predictions on new data that it has not seen before, as long as the input features of the new data are similar to those of the training data.

```

from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.4, random_state=42)
X_train.shape, X_test.shape, y_train.shape, y_test.shape

[117...] ((805, 1), (537, 1), (805, 10), (537, 10))

```

Figure 6. shows train data under study.

```

:
print("Min || Max (X_train)")
print(f"{min(X_train)}\t{max(X_train)}")

Min || Max (X_train)
[2020] [2023]

```

Figure 6. Shows maximum & minimum value of salary in different years.

**V. Data visualization** is a powerful technique that helps to analyze and communicate complex data sets visually. It involves creating visual representations of data in the form of charts, graphs, maps, and other interactive displays that make it easier to understand and extract insights from the data.

There are many benefits of data visualization for analysis, including:

**Spotting trends and patterns:** Visualizing data can help to identify trends and patterns that may not be apparent from a table of numbers or raw data. This can lead to new insights and discoveries.

**Communicating complex data:** Visualization can help to simplify complex data and make it easier to communicate to others. This is particularly useful when dealing with large or complex data sets.

**Identifying outliers and anomalies:** Visualization can help to identify outliers or anomalies in the data that may be difficult to spot in raw data.



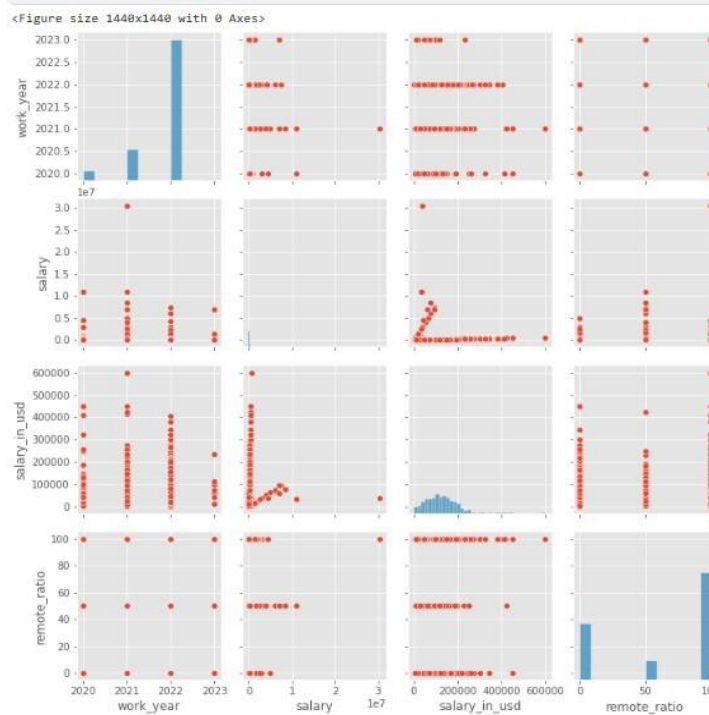


Figure 7. Shows visualization of pairplot of variables.

```
[14]: array([<AxesSubplot:ylabel='salary_in_usd'>], dtype=object)
```

top 15 salary\_in\_usd

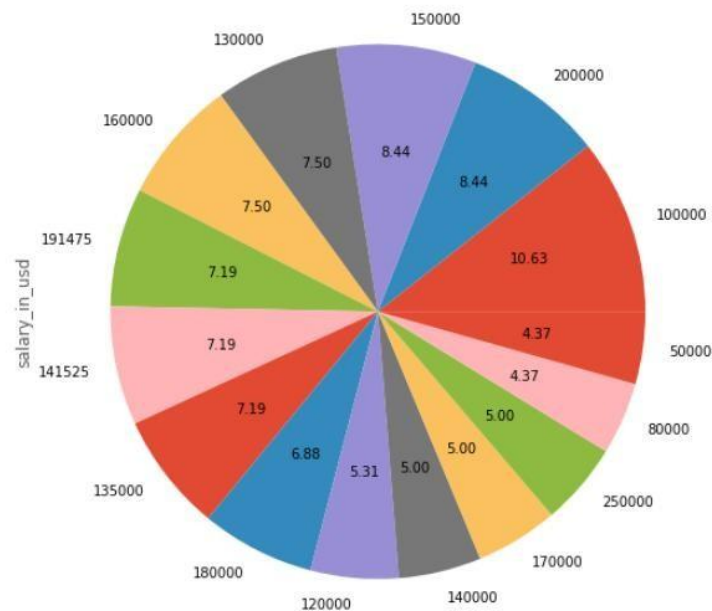


Figure 8. Shows top 15 salary in USD.



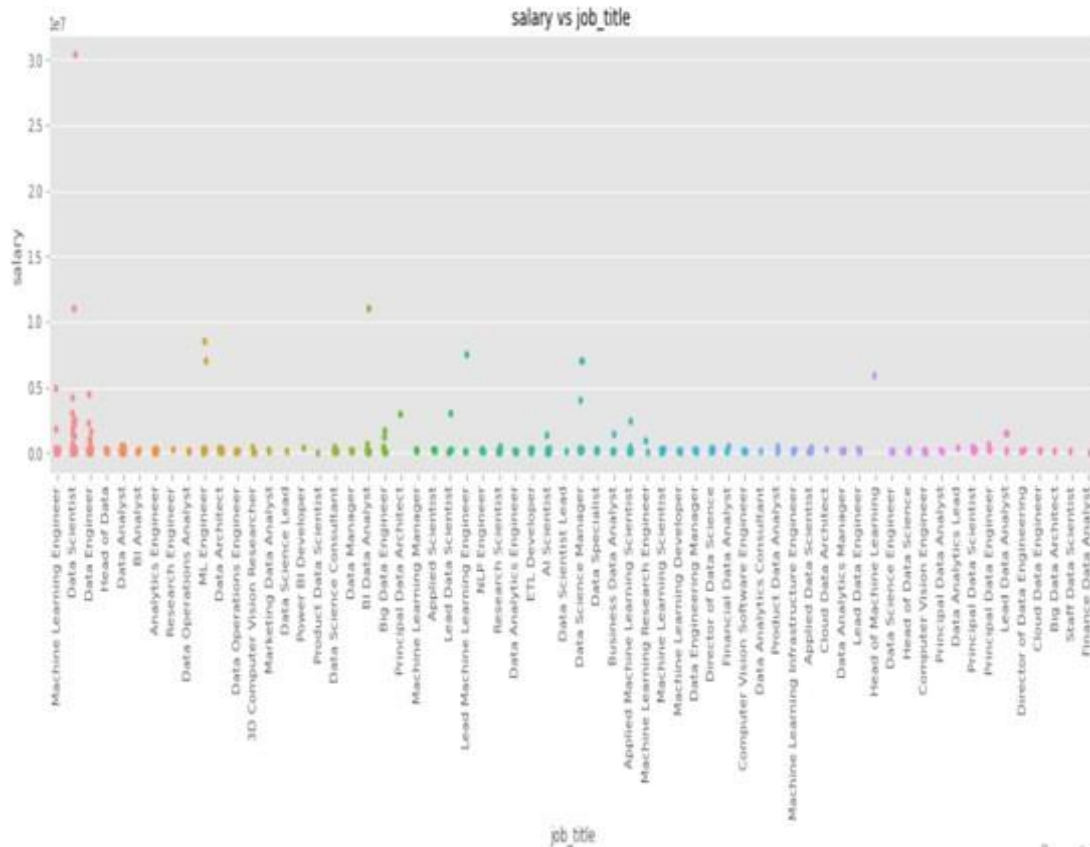


Figure 9. Shows strip plot of job title vs salary of employee.

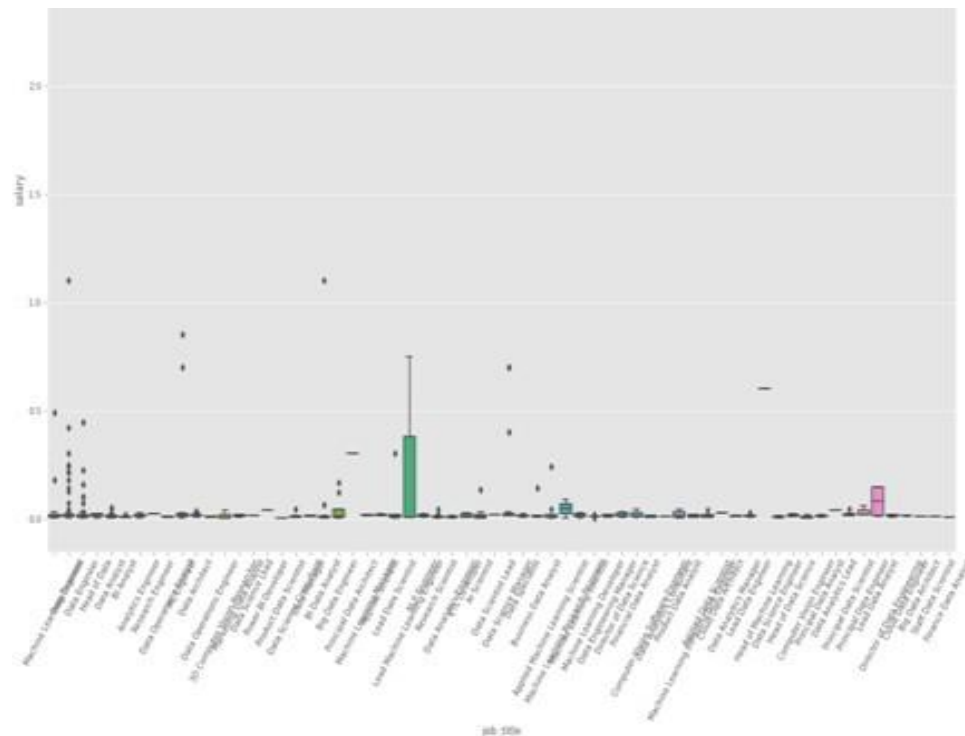


Figure 10. Shows box plot of job title vs salary of employee.

## VI. Evaluation of Experimental Data

### a) Experimental Results

We used the python code to evaluate the performance of different models using three common regression metrics: mean squared error (MSE), mean absolute error (MAE), and R-squared score (R2 score). The predictions dictionary likely contains predictions made by different regression models for a test set of data, and y\_test is the corresponding true target variable values.

The code iterates through each item in the predictions dictionary and prints the name of the model along with its evaluation metrics. The mean\_squared\_error, mean\_absolute\_error, and r2\_score functions are imported from the sklearn.metrics module and are used to calculate the corresponding metrics for each model's predictions.

The output will show the evaluation metrics for each model, allowing the user to compare their performances and choose the best model.

```
formodel_name, preds in predictions.items():  
print(f"\t=== {model_name} ===")  
print(f"> MSE : {mean_squared_error(y_test, preds)}")  
print(f"> MAE : {mean_absolute_error(y_test, preds)}")  
print(f"> R2 score : {r2_score(y_test, preds)}")  
print()
```

#### Linear regression

```
>MSE : 714714581567.9695  
>MAE : 235448.8678080483  
> R2 score : -0.003968404508760681
```

#### Logistic regression

```
>MSE : 738187015911.2216  
>MAE : 186376.9422718808  
> R2 score : -0.03694042308131085
```

#### Ridge

```
>MSE : 714621117588.9155  
>MAE : 235315.1919060143  
> R2 score : -0.003837114502442285
```

#### Lasso

```
>MSE : 714714270748.344  
>MAE : 235448.42423582167  
> R2 score : -0.0039679678965920395
```

#### Elastic net

```
>MSE : 708599645989.8032  
>MAE : 222772.72690546844  
> R2 score : 0.004621321060594652
```

#### Linear Regression

```
lr = LinearRegression()  
lr.fit(X_train,y_train)  
lr.score(X_test,y_test)
```

```
-0.003968404508760681
```

## CONCLUSION

We used python programming language for exploratory data analysis. The developer can understand the code .it offers a variety of libraries and some of them uses great visualization tool. Visualization process can make it easier to create the clear report

The data we are using taken from **data sciencejob salary** data set from kaggle.

1. In the first step we have imported the Pandas libraries. numpy packages.
2. After that we have imported fairly large salary details CSV file as a data frame df. It gives the data sets in the form of rows and column. We have used head() method to return top 5 rows of the data frame or series.
3. We have to choose the right visualization method. When visualizing individual variables, it is important to first understand what type of variable we are dealing with. This will help us find the right visualization method for that variable .for this we have imported Matplot lib, seaborn library packages. We have used df.dtypes to list the data for each column.

### Data Frame

```
<class 'pandas.core.frame.DataFrame'>
```

RangeIndex: 1342 entries, 0 to 1341

Data columns (total 11 columns):

#	Column	Non-Null Count	Dtype
---	--------	----------------	-------

0	work year	1342 non-null	int64
1	experience level	1342 non-null	object
2	employment type	1342 non-null	object
3	job title	1342 non-null	object
4	salary	1342 non-null	int64
5	salary currency	1342 non-null	object
6	salary_in_usd	1342 non-null	int64
7	employee residence	1342 non-null	object
8	remote ratio	1342 non-null	int64
9	company location	1342 non-null	object
10	company size	1342 non-null	object

dtypes: int64(4), object(7)

memory usage: 115.5+ KB

We use predict method of the model object to generate a prediction for the input yrExp. The predicted salary is stored in the predSalary variable, and then printed to the console using the print function. The predicted salary is rounded to one decimal place using the np.round function.

```
defsingle_prediction(model,yrExp):
```

```
predSalary = model.predict(np.array(yrExp).reshape(-1,1))
```

```
print("Predicted Salary :: ", np.round(predSalary[0],1))
```

### Single prediction (lr, 1.5)

**Predicted Salary:: 493095803.9**

In this article we have explained the detail about explorative data analysis. We have used the language python programming language for implementation.. We have implemented different library packages of python. We got the required result taking different parameter. In future we will use more data sets and other functions to get the clear idea related to exploratory data analysis

## REFERENCES

- [1]. Aindrila Ghosh, Mona Nashaat, James Miller, Shaikh Quader, and Chad Marston, "A Comprehensive Review of Tools for Exploratory Analysis of Tabular Industrial Datasets," Visual Informatics, Volume 2, Issue 4, December 2018, pp. 235-253.
- [2]. John T. Behrens, "Principles and Procedures of Exploratory Data Analysis," Psychological Methods, 1997, Vol. 2, No. 2, pp.131-160.

- [3]. ChokeyWangmo, "An Exploratory Study on Bank Lending to SME Sector in Bhutan," International Journal of Scientific & Technology Research, volume 6, issue 11, November 2017, pp. 47-51.
- [4]. Matthew Ntow-Gyamfi and Sarah SerwaaBoateng, "Credit Risk and Loan Default among Ghanaian Banks: An Exploratory Study," Management Science Letters, Vol. 3, 2013, pp.753–762.
- [5]. X. Francis Jency, V. P. Sumathi, Janani Shiva Sri, "An Exploratory Data Analysis for Loan Prediction Based on Nature of the Clients," International Journal of Recent Technology and Engineering (IJRTE), Volume-7 Issue-4S, November 2018, pp.176-179.
- [6]. K. UlagaPriya, S. Pushp, K. Kalaivani, A. Sartiha, "Exploratory Analysis on Prediction of Loan Privilege for Customers using Random Forest," International Journal of Engineering & Technology, Vol.
- [7]. "Exploratory Data Analysis" by John W. Tukey (1977).
- [8]. "Data Analysis with Open Source Tools" by Philipp K. Janert (2010).
- [9]. "Python for Data Analysis" by Wes McKinney (2017).
- [10]. "R Graphics Cookbook" by Winston Chang (2013).