# HR Data Capstone Project

Submitted in Partial Fulfillment of requirements for the Award of certificate of

Post Graduate Program in Data Science and Business Analytics

**Capstone Project Report**

Submitted to

GREAT LAKES INSTITUTE OF MANAGEMENT

Global Mindset - Indian Roots

Submitted by:

**Satya Raga Sudha**

**Akashatra Sharma**

Under the guidance of

**V Surya Prakash Raju**

**Batch:** PGP DSBA LVC JAN2022

**Year of Completion**: 2022

## CERTIFICATE OF COMPLETION SIGNED BY MENTOR

This is to certify that the participants "Akashatra Sharma" & "Satya Raga Sudha" who are the students of Great Learning have successfully completed their project on **HR Data Capstone Project**.

This project is the record of authentic work carried out by them during the academic year 2022

Mentor

Date:

Place:

<<Email from mentor is also acceptable >>

# ACKNOWLEDGEMENTS

We are grateful to our respectable teacher, **Mr. V Surya Prakash Raju** who has been instrumental in guiding us through this project successfully. With his wisdom and knowledge, we were able to complete this report with ease under his supervision which was a very enriching experience for all of us!

We also would like to thank our teachers & professors whose advice & teachings helped make the processing part much smoother and easier than expected considering it was such an ambitious task from the start!

Team is combined when great mind come together and when blended together , they achieve greater heights. So, thanks to my partner and me who worked day & night on this project and made it complete with a success !

Lastly, without their help along the way, We're not sure if we could have made it here today so thanks go out as well to everyone else that contributed at some point or another during our journey on completing this remarkable undertaking together.

# LIST OF TABLES

## Contents

LIST OF TABLES ........ 4

LIST OF FIGURES ........ 6

GLOSSARY OF TERMS / ABBREVIATIONS ........ 8

EXECUTIVE SUMMARY ........ 9

To overcome such problems, if we use the prediction tools then we can predict the salary details of each employee recruited by the company, such that it will reduce the stress or work carried out by the HR team for negotiating the salary and avoid discrimination in the company because of the minimal human interference thus providing the organization ease in their respective work. 9

**1. Problem Statement** ........ 9

We have a problem statement related to an organization Delta Ltd. The HR Team of Delta Ltd. want to have a system, which can predict the salary of employees, which will lead to no discrimination & employee satisfaction based on their past data, easy to use, avoid manual judgement & effective tool with minimal involvement. ........ 9

We have a scope of developing a tool, which help them out in solving their issue & reduce their effort in salary calculation. It will be easy to use and avoid manual work out. ........ 9

The objective, we have here is to collect past data of all employees of Delta Ltd, which are presently used for estimation of Annual salary of an employee by HR Team. Then we understand the data, analyze the data & prepare a model to predict the salary of new employee with similar kind of profile & avoid manual judgement. At for the proper working of model, we'll test the model by comparing it with existing data as confirmation. ........ 9

**2. Data Description** ........ 9

We have collected data (25000 Applicants) from the HR Team of Delta Ltd. It contains 29 different parameter on which the salary judgement (Expected CTC) that is our target variable is processed. We have observed it contains both numerical & categorical data. ........ 9

Numerical data – There are 12 Parameters such as Index, Application ID, Total experience, Experience in field, passing years of graduation, PG & PHD, Current CTC, No. of companied worked, No. of publication, certification & expected CTC. ........ 9

Categorical data - Remaining 17 out of 29 are categorical data. Ordinal categorical data are – Education, Appraisal Rating and Designation. We do have Missing values in Department, Roles, Designation, education, education related columns. Most of the missing values have arisen due to freshers & under graduates. The fresher are outliers. Duplicate data was also checked and they were none to found. ........ 9

We performed all the necessary data descriptive stats and can be viewed in the ........ 9

- The Data Dictionary is present in the ........ 9

**3. Main Results** ........ 12

**4. Recommendations** ........................................................................................................ 13

Section 1: Introduction........................................................................................................ 14

- The HR team of Delta want to have a system, which can predict the salary of employees, which will lead to no discrimination & employee satisfaction based on their past data, easy to use, avoid manual judgement & effective tool with minimal involvement.................................. 14

- We have a scope of developing a tool, which help them out in solving their issue & reduce their effort in salary calculation. It will be easy to use and avoid manual work out. .................. 14

- The objective, we have here is to collect past data of all employees of Delta Ltd, which are presently used for estimation of Annual salary of an employee by HR Team. Then we understand the data, analyze the data & prepare a model to predict the salary of new employee with similar kind of profile & avoid manual judgement. At for the proper working of model, we'll test the model by comparing it with existing data as confirmation. ........................................... 14

**Sections 2 : EDA and Insights**........................................................................................ 15

Sections 3 : Model Development ........................................................................................ 24

Sections 4 : Final Recommendation ..................................................................................... 27

Bibliography ........................................................................................................................ 27

Appendix ............................................................................................................................. 28

# LIST OF FIGURES

Figure 1 : Data Info........10
Figure 2 : Data - Outliers Present........12
Figure 3 : Best MLR Model........13
Figure 4 : Univariate Analysis 1........15
Figure 5 : Univariate Analysis 2........16
Figure 6 : Univariate Analysis 3........16
Figure 7 : Univariate Analysis 4........17
Figure 8 : Univariate Analysis 5........17
Figure 9 : Univariate Analysis 6........17
Figure 10 : Univariate Analysis 7........17
Figure 11 : Univariate Analysis 8........18
Figure 12 : Count Plot 1........18
Figure 13 : Count Plot 2........19
Figure 14 : Count Plot 3........19
Figure 15 : Count Plot 4........19
Figure 16 : Count Plot 5........20
Figure 17 : Count Plot 6........20
Figure 18 : Correlation Matrix 1........21
Figure 19 : Correlation MATRIX 2........21
Figure 20 :Heatmap........22
Figure 21 : Pairplot........23
Figure 22 : Base Model........24
Figure 23 : Final Model........25
Figure 24 : Scatter Plot - Train Data Best Model MLR........25
Figure 25 : Scatter Plot - Test Data Best Model MLR........26
Figure 26 : Density Plot - Fitted(predicted) vs Actual Values........26
Figure 27 : Data Dictionary........28
Figure 28 : Null Values Check........29
Figure 29 : Null Values List........30
Figure 30 : Anomalies Check Categorical Data 1........30
Figure 31 : Anomalies Check Categorical Data 2........31
Figure 32 : Anomalies Check Numerical Data 1........31
Figure 33 : Anomalies Check Numerical Data 2........32
Figure 34 : Appendix Encoding 1........32
Figure 35 : Appendix Encoding 2........32
Figure 36 : Appendix Encoding 3........33
Figure 37 : Appendix Encoding 4........33
Figure 38 : Appendix Encoding 5........33
Figure 39 : Outlier Treatment 1........34
Figure 40 : Outlier Treatment 2........34
Figure 41 : Outlier Treatment 3........35
Figure 42 : Outlier Treatment 4........35
Figure 43 : Appendix After Outlier Treatment........35
Figure 44 : Ordinal Encoding 1........36

Figure 45 : Ordinal Encoding 2 ........................................................................................36
Figure 46 : Label Encoding 1 ...........................................................................................36
Figure 47 : Label Encoding 2 ...........................................................................................37
Figure 48 : Label Encoding 3 ...........................................................................................37
Figure 49 : Appendix Imputation .......................................................................................38
Figure 50 : VIF - Multi colinearity Check 1 ........................................................................38
Figure 51 : VIF Multi Colinearity Check 2 ..........................................................................39
Figure 52 : Appendix Residual Plot - Best Model MLR ......................................................39

# GLOSSARY OF TERMS / ABBREVIATIONS

1. HR Team - Human Resource Team
2. cat - Categorical Data
3. num - Numerical Data
4. df - Original Data frame "Expected_CTC.csv"
5. data - Copied Whole Original Data Frame df into this

# EXECUTIVE SUMMARY

Human Resource Team (HR Team) plays a vital role in determining a salary of an employee in the organization. There are many aspects and factors that are needed to be taken into consideration while doing this & even if slightest mistake in judgement is done, then it could affect the performance and analysis of the employee which will lead to dissatisfaction of the employee ultimately leaving the company. Therefore, HR team need to manage well to retain the talent in any organization. In the current situation, people are moving out of organizations frequently and thus the organization need replacements for ongoing projects as well as for new projects.

To overcome such problems, if we use the prediction tools then we can predict the salary details of each employee recruited by the company, such that it will reduce the stress or work carried out by the HR team for negotiating the salary and avoid discrimination in the company because of the minimal human interference thus providing the organization ease in their respective work.

## 1. Problem Statement

We have a problem statement related to an organization Delta Ltd. The HR Team of Delta Ltd. want to have a system, which can predict the salary of employees, which will lead to no discrimination & employee satisfaction based on their past data, easy to use, avoid manual judgement & effective tool with minimal involvement.
We have a scope of developing a tool, which help them out in solving their issue & reduce their effort in salary calculation. It will be easy to use and avoid manual work out.
The objective, we have here is to collect past data of all employees of Delta Ltd, which are presently used for estimation of Annual salary of an employee by HR Team. Then we understand the data, analyze the data & prepare a model to predict the salary of new employee with similar kind of profile & avoid manual judgement. At for the proper working of model, we'll test the model by comparing it with existing data as confirmation.

## 2. Data Description

We have collected data (25000 Applicants) from the HR Team of Delta Ltd. It contains 29 different parameter on which the salary judgement (Expected CTC) that is our target variable is processed. We have observed it contains both numerical & categorical data.

**Numerical data** – There are 12 Parameters such as Index, Application ID, Total experience, Experience in field, passing years of graduation, PG & PHD, Current CTC, No. of companied worked, No. of publication, certification & expected CTC.

**Categorical data** - Remaining 17 out of 29 are categorical data. Ordinal categorical data are – Education, Appraisal Rating and Designation. We do have Missing values in Department, Roles, Designation, education, education related columns. Most of the missing values have arisen due to freshers & under graduates. The fresher are outliers. Duplicate data was also checked and they were none to found.

We performed all the necessary data descriptive stats and can be viewed in the Appendix.

- The Data Dictionary is present in the Appendix. Can refer to it whenever needed.

- **Data Info**

  The data info gave us multiple information. They were as follows:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 25000 entries, 0 to 24999
Data columns (total 29 columns):
 #   Column                            Non-Null Count  Dtype
---  ------                            --------------  -----
 0   IDX                               25000 non-null  int64
 1   Applicant_ID                      25000 non-null  int64
 2   Total_Experience                  25000 non-null  int64
 3   Total_Experience_in_field_applied 25000 non-null  int64
 4   Department                        22222 non-null  object
 5   Role                              24037 non-null  object
 6   Industry                          24092 non-null  object
 7   Organization                      24092 non-null  object
 8   Designation                       21871 non-null  object
 9   Education                         25000 non-null  object
 10  Graduation_Specialization         18820 non-null  object
 11  University_Grad                   18820 non-null  object
 12  Passing_Year_Of_Graduation        18820 non-null  float64
 13  PG_Specialization                 17308 non-null  object
 14  University_PG                     17308 non-null  object
 15  Passing_Year_Of_PG                17308 non-null  float64
 16  PHD_Specialization                13119 non-null  object
 17  University_PHD                    13119 non-null  object
 18  Passing_Year_Of_PHD               13119 non-null  float64
 19  Curent_Location                   25000 non-null  object
 20  Preferred_location                25000 non-null  object
 21  Current_CTC                       25000 non-null  int64
 22  Inhand_Offer                      25000 non-null  object
 23  Last_Appraisal_Rating             24092 non-null  object
 24  No_Of_Companies_worked            25000 non-null  int64
 25  Number_of_Publications            25000 non-null  int64
 26  Certifications                    25000 non-null  int64
 27  International_degree_any          25000 non-null  int64
 28  Expected_CTC                      25000 non-null  int64
dtypes: float64(3), int64(10), object(16)
memory usage: 5.5+ MB
```

**Figure 1 : Data Info**

**Interpretations:**

- There 3 float data type, 10 integer data type & 16 object data type.

-Many variables were representing null values. Hence, they must be checked upon and a solution to it shall be found out.

-Refer to Appendix for checking which variables had null values.

- **Checking For Anomalies / Bad Data**

- We first separated the categorical and numerical variables and then check for anomalies/bad data.
- In categorical data, there should be alphabets and words present and for numerical data there should be numbers/integers/any numeric format. So after separating categorical & numerical data, when we checked for the different symbols such as **'$', '?'** in both data, we found out that there were 0 entries with it in both data set.
- As we all know that anomalies and outliers and two different things, so outliers were checked after anomalies were checked through.
- Refer to Appendix  for the better understanding on checking for anomalies.

**NOTE :** The original dataset **"df"**(name given in Jupyter notebook) we loaded is very precious and HR Team cannot afford to tamper with it during data pre processing so for the better safety, we copied the whole original data set into a new dataframe called as **"data".** After this we performed all data pre processing and model building on this new data set.

- **ANOVA Test**

- We converted the object data type into categorical data type before performing ANOVA test.
- We then performed **one way ANOVA Test** on all the relevant variables. We assumed Level of Significance = 0.05 by default as no other level of significance value was provided.
- **One way ANOVA Test** was performed in order to determine which variables are significant and which are insignificant.
- As we performed the ANOVA Test, many insignificant variables were to be found out and thus we dropped them. Although the significant variables were kept and will use them in the EDA & Model Building.
- Refer to Appendix for the detailed understanding of how **One Way ANOVA Test** was performed to find out the insignificant & significant variables.
- After conducting all this, we imputed null values in the significant variables with the relevant measures and proceeded further.

- **Encoding**

- As per the problem statement, it was a **Multi Linear Regression** type of problem. So, we need all the data in numerical format in order to build the best optimal model.
- We performed **'Label Encoding'** and **'Ordinal Encoding'** after the ANOVA Test to convert the categorical data into numerical data format.
- Refer to Appendix for the detailed information regarding which variables were encoded.

- **Outliers**

- We checked for the outliers in the data set and wherever necessary we treated them.
- However, some extreme values were also present in the dataset and after though checking/inspection, they were considered only as extreme values as they were logically meaningful and thus they were considered as outliers.
- Refer to Appendix for better understanding of treatment of outliers.
- **Outliers** can be checked in the graph as below :

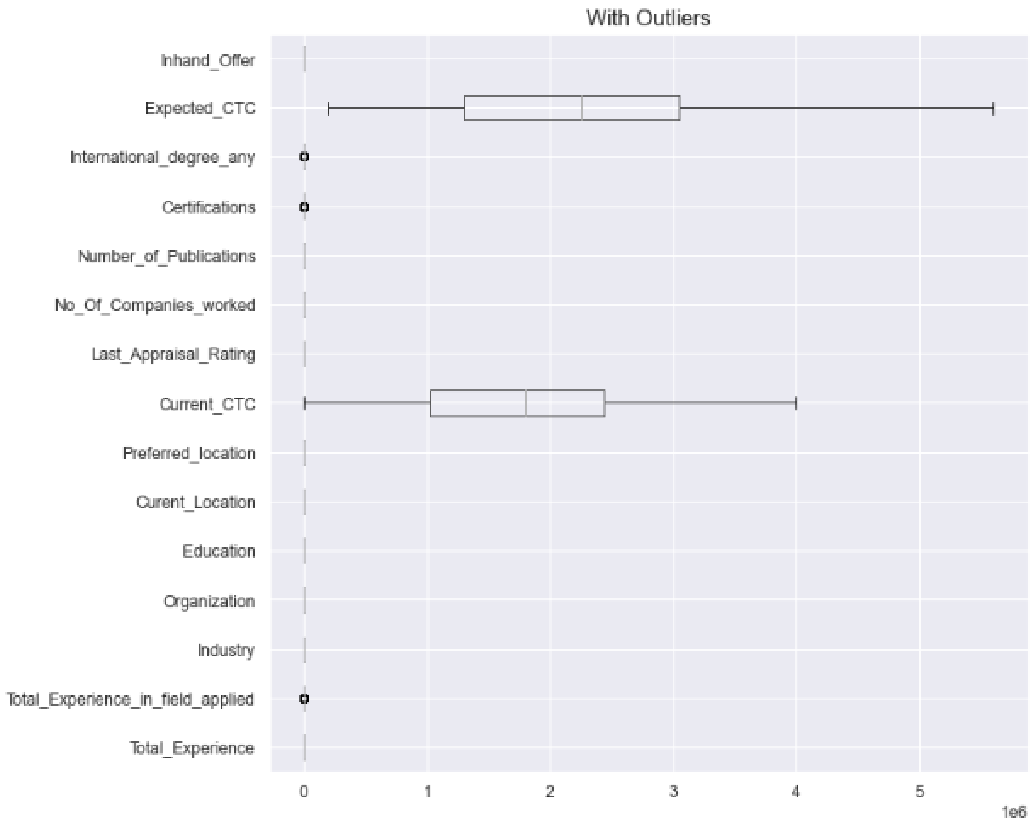


**Figure 2 : Data - Outliers Present**

- **EDA**

- We performed EDA after all the data pre processing and find some useful insights for the model building.
- Refer to EDA & Insights for more detailed understanding.

## 3. Main Results

- After EDA comes the model building. In this phase , we approached with the problem statement and objective that we need to build a **Multi Linear Regression Model** to find the solution.
- Then we proceeded to build MLR model. With the help of statsmodel library, we builded a base model and find out that it contained multi-colinearity which made the model overfitted. So, we used the hyper parameter tuning and builded multiple models till the time all the multi-colinearity was nowhere to be found in the model.
- After all the efforts , we finally builded the model which was perfect in all its way and its R squared value = 0.987 & Adj. R squared value = 0.987
- After building the best model, we then checked the validation of model by trying to run the model on test data.
- As a result, the model run successfully . Thereby, confirming that the model build is very good.
- In addition to it, we plotted scatter plot & Distplot for the best model and giving the graphical representation of the best model build.
- Refer to Model Development for more detailed information and understanding.

OLS Regression Results

| | | | |
|---|---|---|---|
| Dep. Variable: | Expected_CTC | R-squared: | 0.987 |
| Model: | OLS | Adj. R-squared: | 0.987 |
| Method: | Least Squares | F-statistic: | 1.892e+05 |
| Date: | Sun, 11 Dec 2022 | Prob (F-statistic): | 0.00 |
| Time: | 18:02:58 | Log-Likelihood: | -2.8998e+05 |
| No. Observations: | 21944 | AIC: | 5.800e+05 |
| Df Residuals: | 21934 | BIC: | 5.801e+05 |
| Df Model: | 9 | | |
| Covariance Type: | nonrobust | | |

| | coef | std err | t | P>\|t\| | [0.025 | 0.975] |
|---|---|---|---|---|---|---|
| Intercept | -1.201e+05 | 3777.338 | -31.792 | 0.000 | -1.27e+05 | -1.13e+05 |
| Industry | -7227.1469 | 281.269 | -25.695 | 0.000 | -7778.454 | -6675.840 |
| Organization | 3092.5457 | 195.549 | 15.815 | 0.000 | 2709.256 | 3475.836 |
| Education | 7.832e+04 | 870.851 | 89.941 | 0.000 | 7.66e+04 | 8e+04 |
| Current_CTC | 1.2291 | 0.001 | 1113.463 | 0.000 | 1.227 | 1.231 |
| Last_Appraisal_Rating | 7.48e+04 | 701.874 | 106.567 | 0.000 | 7.34e+04 | 7.62e+04 |
| No_Of_Companies_worked | -1.903e+04 | 580.300 | -32.800 | 0.000 | -2.02e+04 | -1.79e+04 |
| Number_of_Publications | 3039.9551 | 357.238 | 8.510 | 0.000 | 2339.743 | 3740.167 |
| International_degree_any | -1.148e+04 | 3290.858 | -3.487 | 0.000 | -1.79e+04 | -5025.612 |
| Inhand_Offer | 2.048e+04 | 2158.925 | 9.487 | 0.000 | 1.62e+04 | 2.47e+04 |

| | | | |
|---|---|---|---|
| Omnibus: | 11923.901 | Durbin-Watson: | 1.996 |
| Prob(Omnibus): | 0.000 | Jarque-Bera (JB): | 153498.521 |
| Skew: | 2.341 | Prob(JB): | 0.00 |
| Kurtosis: | 15.082 | Cond. No. | 8.61e+06 |

**Figure 3 : Best MLR Model**

## 4. Recommendations

- As per the model development and validation, we inferred some useful business insights with the help of which we are giving out point of business recommendations.
- These are as follows :
  - Current CTC is an important factor in determining the Expected CTC of an employee. So, the company should always keep in tab with the Current CTC of the candidate who is applying keeping in mind the role & department he/she is applying for.
  - Location is another important factor for the determination of Expected CTC. Nowadays, many employees prefer to work at their own preferred location because of many reasons including family, transportation etc. So, the company should come up with a strategy to attract the candidates who are well qualified so that they could neglect there preferred location and become willing to change their location. Strategies such as increased salary but would be incentives only(based on performance in the company). This kind of strategy is quite useful to attract candidates.
  - Employees who work earlier for big organization tends to become more successful , so company should also consider these types of candidates as their target to go for as they bring valuable knowledge and strategies that can help company grow to bigger heights.

## Section 1: Introduction

- The HR team of Delta want to have a system, which can predict the salary of employees, which will lead to no discrimination & employee satisfaction based on their past data, easy to use, avoid manual judgement & effective tool with minimal involvement.
- We have a scope of developing a tool, which help them out in solving their issue & **reduce their effort in salary calculation. It will be easy to use and avoid manual work out.**
- The objective, we have here is to collect past data of all employees of Delta Ltd, which are presently used for estimation of Annual salary of an employee by HR Team. Then we understand the data, analyze the data & prepare a model to predict the salary of new employee with similar kind of profile & avoid manual judgement. At for the proper working of model, we'll test the model by comparing it with existing data as confirmation.

- **Data Sources**

- We found the database from the internet websites such as Kaggle, Towards Data Science, Stack Overflow etc.
- According to the problem statement, we need to build a model that can predict the Expected CTC for the applying candidate in the company. So, the general approach for that type of problem should be Multi Linear Regression.
- Therefore, we need to build **Multi Linear Regression Model**.
- Before building the model, we need to do some data pre processing.

**NOTE :** The original dataset **"df"**(name given in Jupyter notebook) we loaded is very precious and HR Team cannot afford to tamper with it during data pre processing so for the better safety, we copied the whole original data set into a new dataframe called as **"data".** After this we performed all data pre processing and model building on this new data set.

- **Data Pre Processing**

- From the data , we inferred that there were several null values present in the dataset, so we needed to impute relevant values into them.
- For that, we performed **One Way ANOVA Test** on all the variables and checked whether they were significant or not.
- We converted the object data type into categorical data type before performing ANOVA test. We assumed Level of Significance = 0.05 by default as no other level of significance value was provided.
- The insignificant ones will be dropped off for better model building and null values will be imputed using relevant method in the significant variables.
- As we performed the ANOVA Test, many insignificant variables were to be found out and thus we dropped them. Although the significant variables were kept and will use them in the EDA & Model Building.
- Refer to Appendix for the detailed understanding of how **One Way ANOVA Test** was performed to find out the insignificant & significant variables.

- We know that **'Graduation_Specialization'** & **'University_Grad'** are correlated to **'Passing_Year_Of_Graduation'** and same for 'PG_Specialization' & **'PHD_Specialization'**.
- So,as **'Passing_Year_Of_Grad'** , **'Passing_Year_Of_PG'** & **'Passing_Year_Of_PHD'** were found to be insignificant that means there correlated variables must also become irrelevant for model building.
- As a result, the insignificant variables are not needed in the model building and thus they needed to be dropped from the table.
- After conducting all this, we imputed null values in the significant variables with the relevant measures

- **Imputation & Encoding**
- The significant variables found after conducting ANOVA test were being imputed by relevant measures.
- '**Curent_Location**' & '**Preferred_location**' were having many different unique values , so we grouped them into 3 ties namely Tier_1, Tier_2, Tier_3. After grouping them, we transformed them into numeric format.
- We conducted Label Encoding on **'Education'** & '**Inhand_Offer**' , so as to label them into higher to lower order numeric format.
- We did ordinal encoding on '**Last_Appraisal_Rating**' on it and convert it into numeric format. In addition to it we also used central tendency 'mode' on the same variable to impute the null values as the imputation was less than 1% only.
- We imputed on **'Industry'** & '**Organization**' using the central tendency 'mode' and after that we transformed both variables into numeric format using ordinal encoding.
- Outliers were also removed . Refer to Appendix about how Outlier treatment was conducted.
- Refer to Appendix for the better understanding of how imputation and encoding were done.
- After all data pre processing , we moved on to EDA for further analysis.

## Sections 2 : EDA and Insights

- **Univariate Analysis**
- We created dist plot and boxplots for all the variables. They are as follows :

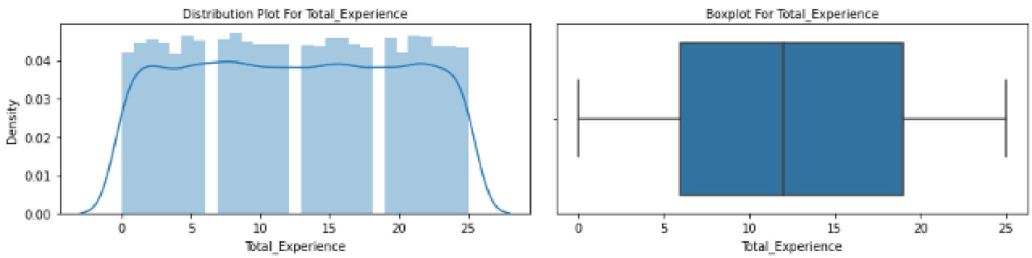


**Figure 4 : Univariate Analysis 1**

- It had no outliers present in it and seemed to closed to a normal distribution.

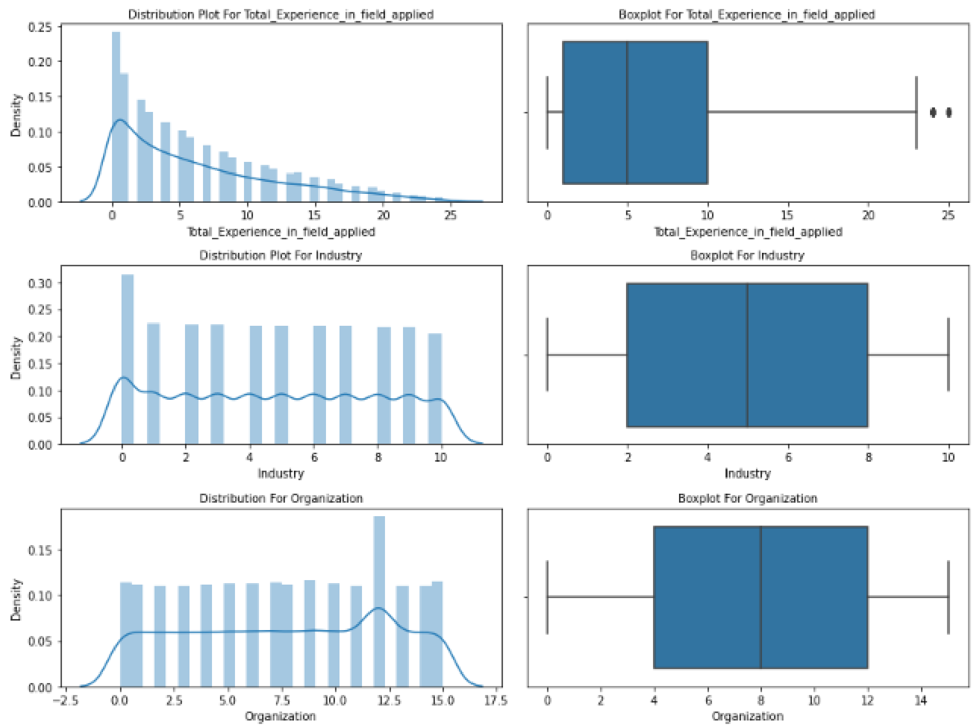


**Figure 5 : Univariate Analysis 2**

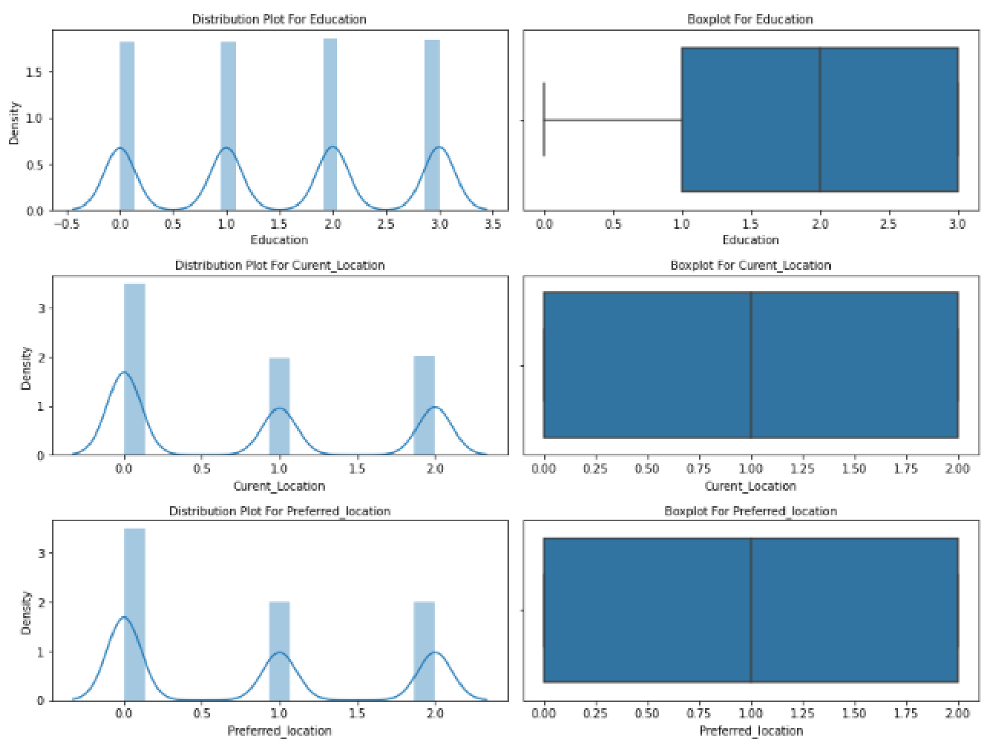


**Figure 6 : Univariate Analysis 3**

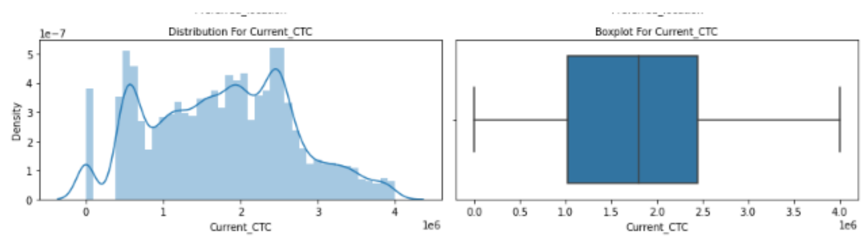


**Figure 7 : Univariate Analysis 4**

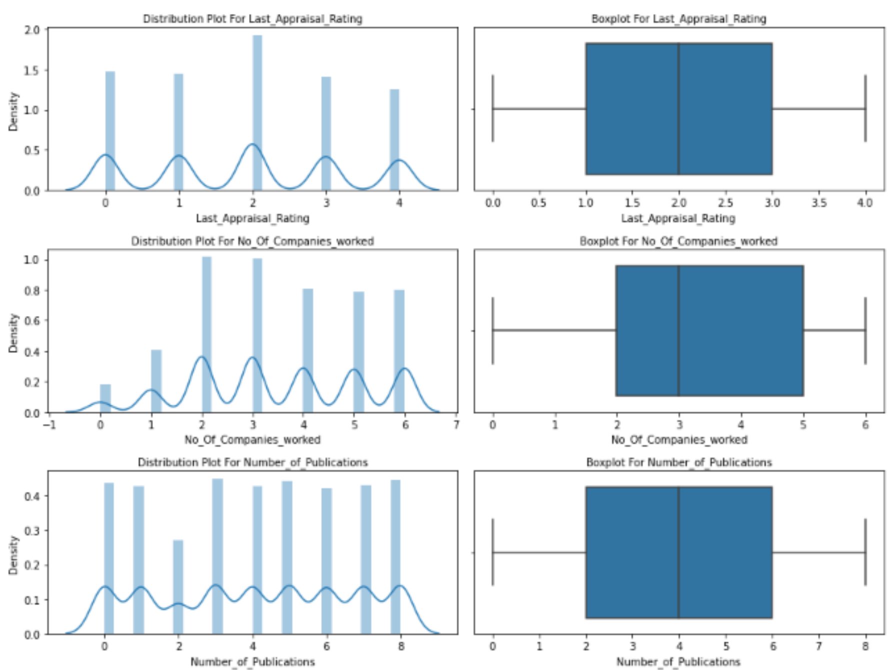


**Figure 8 : Univariate Analysis 5**

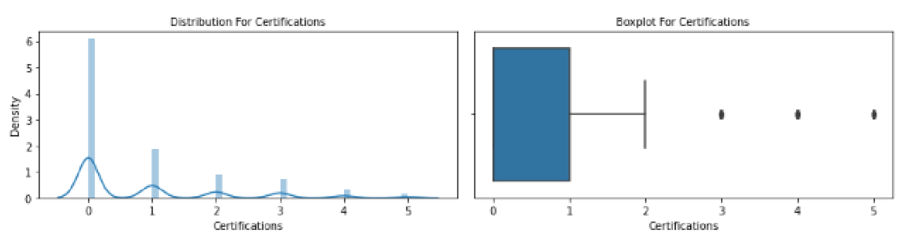


**Figure 9 : Univariate Analysis 6**

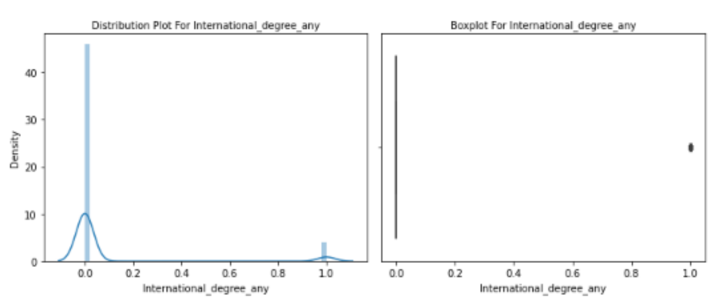


**Figure 10 : Univariate Analysis 7**

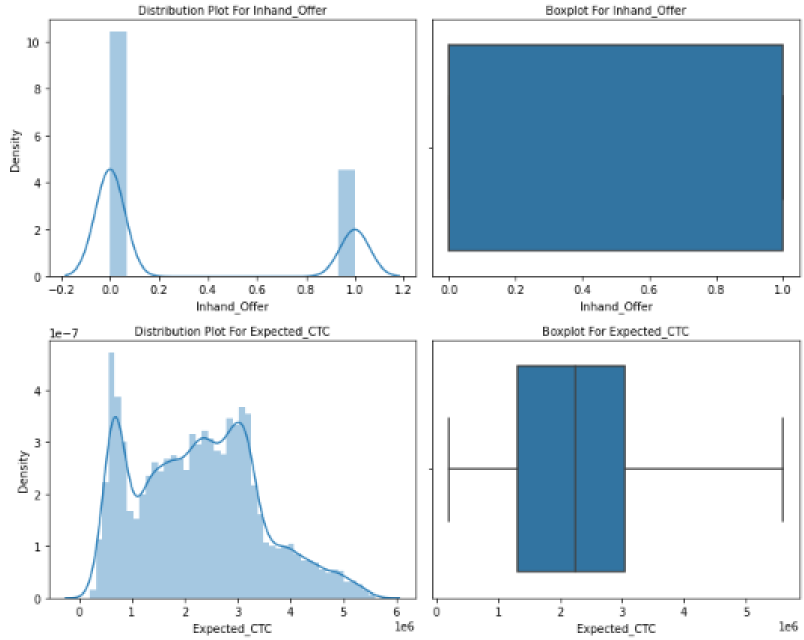


**Figure 11 : Univariate Analysis 8**

- The presence of outliers and influential cases can dramatically change the magnitude of regression coefficients and even the direction of coefficient signs (i.e., from positive to negative or vice versa).
- So, these outliers must be find out and shall be treated in order to perform linear regression.
- After inferring insights, it was confirmed that in **"International_degree_any"** no outliers were present. The dots representing them are actual values and that are 0 & 1 only. They cannot be treated as outliers

- **Bivariate Analysis**
- We created count plot for some feature variables.

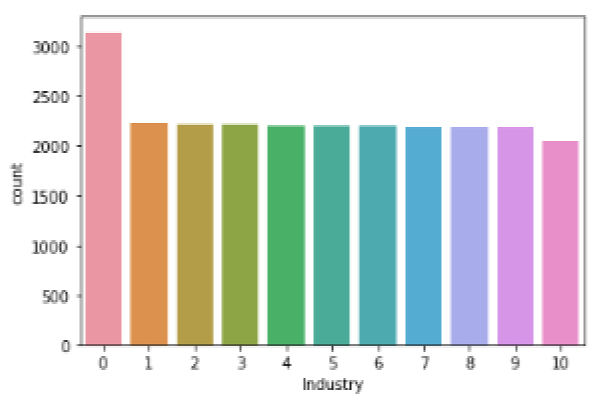


**Figure 12 : Count Plot 1**

- Industry '0' had maximum number of employees as compared to other industries.
- Industry '10' tends to have minimal employees coming from there while other remaining industries shows around same range of employees coming .

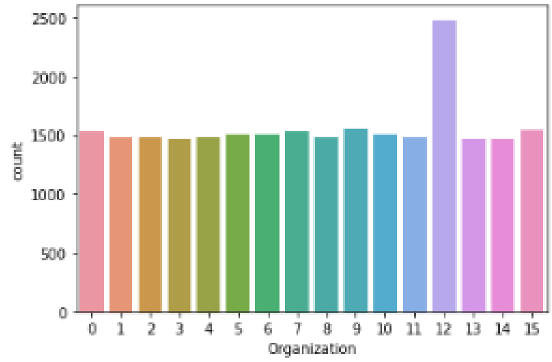


**Figure 13 : Count Plot 2**

- Organization ‘12’ had maximum number of employees as compared to other industries.

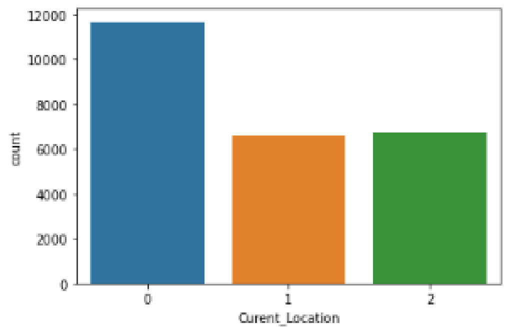


**Figure 14 : Count Plot 3**

- Most Employees tend to come from ‘Tier_1’/’0’ location
- The remaining two location that is ‘1’, ‘2’ tends to have almost same number of employees coming to that location as their company is there.

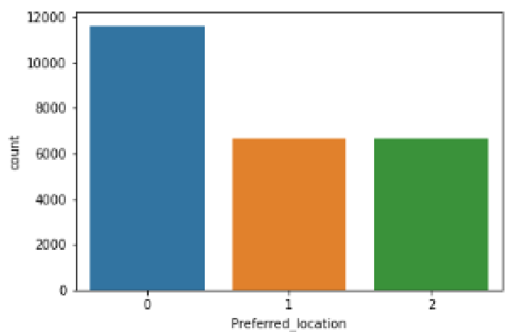


**Figure 15 : Count Plot 4**

- Most Employees tend to come from ‘Tier_1’/’0’ location
- The remaining two location that is ‘1’, ‘2’ tends to have almost same number of employees coming to that location as their company is there.

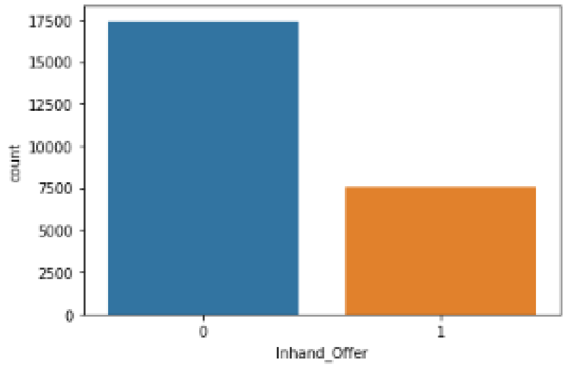


**Figure 16 : Count Plot 5**

- Majority of the employees from the data doesn't have Inhand offer with them.
- The employees having the Inhand offer are around 6500.

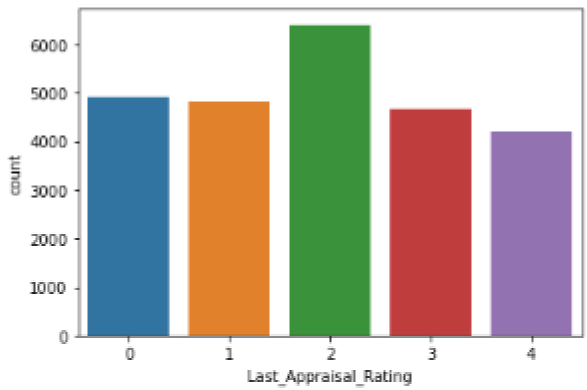


**Figure 17 : Count Plot 6**

- '2' shows the highest 'Last_Appraisal_Rating' as compared to others.
- Among all, only '4' was having the minimum count of employees having 'Last_Appraisal_Rating'.

- **Multivariate Analysis**
- **Multivariate analysis** (**MVA**) is a Statistical procedure for analysis of data involving more than one type of measurement or observation. It may also mean solving problems where more than one dependent variable is analyzed simultaneously with other variables . Multivariate analysis is one of the most useful methods to determine relationships and analyze patterns among large sets of data. It is particularly effective in minimizing bias if a structured study design is employed. However, the complexity of the technique makes it a less sought-out model for novice research enthusiasts. Therefore, although the process of designing the study and interpretation of results is a tedious one, the techniques stand out in finding the relationships in complex.
- We created correlation matrix for the data.

| | Total_Experience | Total_Experience_in_field_applied | Industry | Organization | Education | Curent_Location | Preferred_location | C |
|---|---|---|---|---|---|---|---|---|
| Total_Experience | 1.000000 | 0.645135 | 0.098728 | -0.073088 | 0.023036 | 0.007283 | -0.001081 | |
| Total_Experience_in_field_applied | 0.645135 | 1.000000 | 0.066946 | -0.043967 | 0.017786 | 0.003272 | -0.000822 | |
| Industry | 0.098728 | 0.066946 | 1.000000 | -0.059656 | -0.002715 | 0.006107 | 0.002121 | |
| Organization | -0.073088 | -0.043967 | -0.059656 | 1.000000 | -0.000150 | 0.002374 | 0.000992 | |
| Education | 0.023036 | 0.017786 | -0.002715 | -0.000150 | 1.000000 | 0.003801 | -0.007327 | |
| Curent_Location | 0.007283 | 0.003272 | 0.006107 | 0.002374 | 0.003801 | 1.000000 | 0.006788 | |
| Preferred_location | -0.001081 | -0.000822 | 0.002121 | 0.000992 | -0.007327 | 0.006788 | 1.000000 | |
| Current_CTC | 0.846476 | 0.548017 | 0.108649 | -0.077749 | 0.294165 | 0.012050 | 0.000174 | |
| Last_Appraisal_Rating | 0.053481 | 0.037002 | -0.004837 | -0.004811 | 0.006569 | 0.002255 | 0.000913 | |
| No_Of_Companies_worked | 0.398135 | 0.249045 | 0.116356 | -0.077736 | -0.001687 | -0.006081 | 0.002995 | |
| Number_of_Publications | -0.000494 | -0.010663 | 0.005859 | 0.000772 | -0.002820 | -0.004114 | 0.000998 | |
| Certifications | -0.001130 | -0.002814 | 0.009726 | -0.001450 | -0.500894 | -0.107803 | -0.000969 | |
| International_degree_any | 0.084072 | 0.043070 | 0.013353 | -0.010051 | 0.002334 | 0.002631 | -0.006128 | |
| Expected_CTC | 0.816593 | 0.529115 | 0.083108 | -0.061631 | 0.359005 | 0.012829 | -0.000271 | |
| Inhand_Offer | 0.057390 | 0.029298 | 0.055270 | -0.024219 | 0.013519 | -0.009507 | 0.004156 | |

**Figure 18 : Correlation Matrix 1**

| n | Current_CTC | Last_Appraisal_Rating | No_Of_Companies_worked | Number_of_Publications | Certifications | International_degree_any | Expected_CTC | Inhand_Offer |
|---|---|---|---|---|---|---|---|---|
| 1 | 0.846476 | 0.053481 | 0.398135 | -0.000494 | -0.001130 | 0.084072 | 0.816593 | 0.057390 |
| 2 | 0.548017 | 0.037002 | 0.249045 | -0.010663 | -0.002814 | 0.043070 | 0.529115 | 0.029298 |
| 1 | 0.108649 | -0.004837 | 0.116356 | 0.005859 | 0.009726 | 0.013353 | 0.083108 | 0.055270 |
| 2 | -0.077749 | -0.004811 | -0.077736 | 0.000772 | -0.001450 | -0.010051 | -0.061631 | -0.024219 |
| 7 | 0.294165 | 0.006569 | -0.001687 | -0.002820 | -0.500894 | 0.002334 | 0.359005 | 0.013519 |
| 8 | 0.012050 | 0.002255 | -0.006081 | -0.004114 | -0.107803 | 0.002631 | 0.012829 | -0.009507 |
| 0 | 0.000174 | 0.000913 | 0.002995 | 0.000998 | -0.000969 | -0.006128 | -0.000271 | 0.004156 |
| 4 | 1.000000 | 0.063031 | 0.379740 | -0.006399 | -0.143402 | 0.078774 | 0.986718 | 0.068238 |
| 3 | 0.063031 | 1.000000 | 0.034768 | -0.005386 | -0.008832 | 0.016846 | 0.148601 | 0.313313 |
| 5 | 0.379740 | 0.034768 | 1.000000 | 0.000608 | 0.012990 | 0.047270 | 0.343150 | 0.059160 |
| 8 | -0.006399 | -0.005386 | 0.000608 | 1.000000 | 0.018549 | 0.016419 | 0.001518 | 0.260928 |
| 9 | -0.143402 | -0.008832 | 0.012990 | 0.018549 | 1.000000 | 0.009298 | -0.173992 | 0.018207 |
| 8 | 0.078774 | 0.016846 | 0.047270 | 0.016419 | 0.009298 | 1.000000 | 0.074557 | 0.022363 |
| 1 | 0.986718 | 0.148601 | 0.343150 | 0.001518 | -0.173992 | 0.074557 | 1.000000 | 0.101582 |
| 6 | 0.068238 | 0.313313 | 0.059160 | 0.260928 | 0.018207 | 0.022363 | 0.101582 | 1.000000 |

**Figure 19 : Correlation MATRIX 2**

- After this, a heatmap was created w.r.t to correlation matrix for visualization.

- **Heatmap**

- A **Correlation Heatmap** is a rectangular representation of data and it repeats the same data description twice because the categories are repeated on both axis for computing analysis. Hence, the same result is obtained twice. A correlation heatmap that presents data only once without repetition that is categories are correlated only once is known as a **Triangle Correlation Heatmap**.

- Since data is symmetric across the diagonal from left-top to right bottom the idea of obtaining a triangle correlation heatmap is to remove data above it so that it is depicted only once. The elements on the diagonal are the parts where categories of the same type correlate.
- We created heat map with the help of correlation matrix

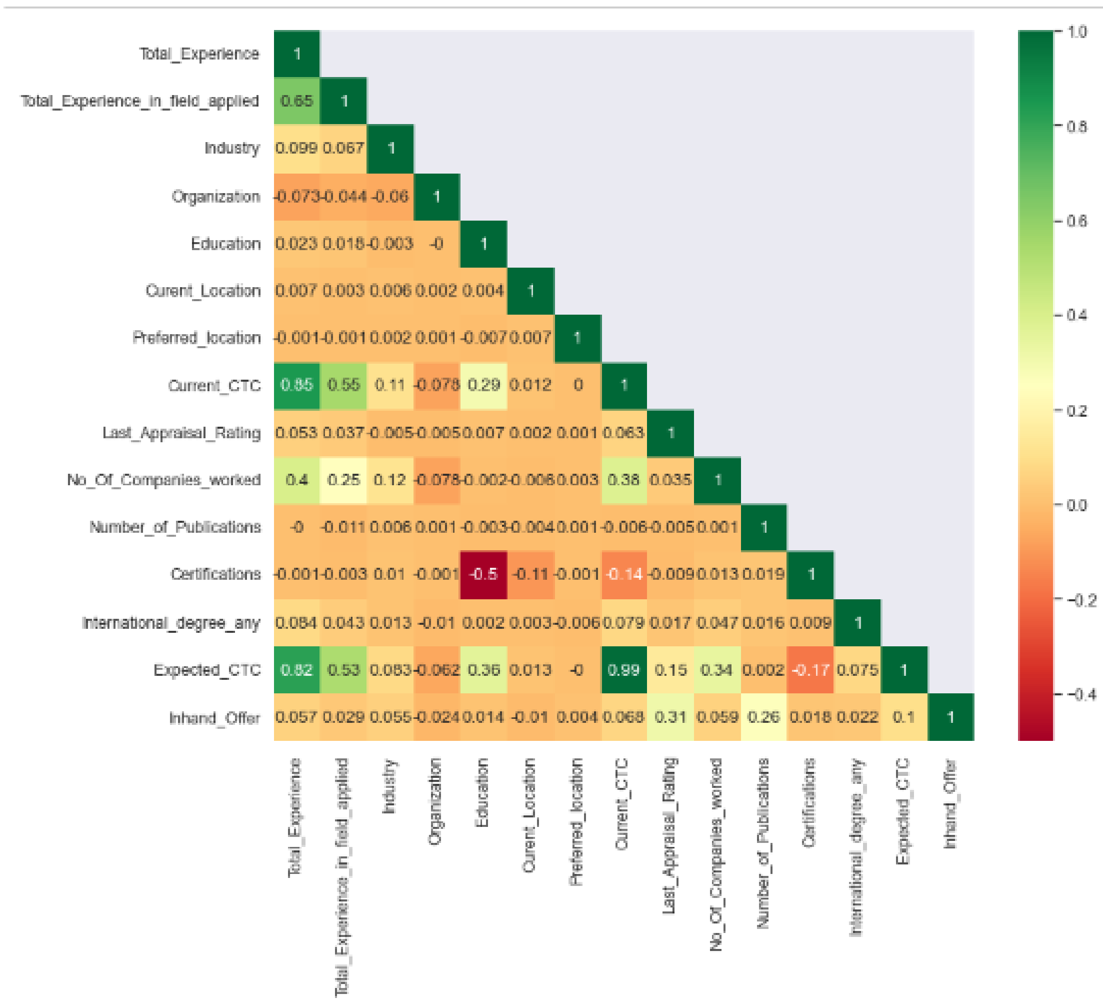


**Figure 20 :Heatmap**

- Heatmap shows that ‘Current_CTC’ & ‘Expected_CTC’ are highly correlated to each other with a value of 99% which indicates they are highly proportional to each other.
- While ‘Total_Experience’ and ‘Expected_CTC’ also show a high relation with a value of 82%.
- ‘Certifications’ & ‘Education’ are negatively correlated with a high value of 50%.
- ‘Total_Experience’ & ‘Expected_CTC’ also showed a very high correlation value of 85%.
- ‘Total_Experienced_in_field_applied’ had decent positive correlation with both ‘Current_CTC’ & ‘Expected_CTC’ with value of 55% & 53% .
- Rest of the other variables doesn’t seem to have a strong correlation. They have minimal correlation with each other.

- **Pairplot**
- After this we created a pairplot.
- **Pairplot** function allows the users to create an axis grid via which each numerical variable stored in data is shared across the X- and Y-axis in the structure of columns and rows. We can create the Scatter plots in order to display the pairwise relationships in addition to the distribution plot displaying the data distribution in the column diagonally.
- It is as follows :

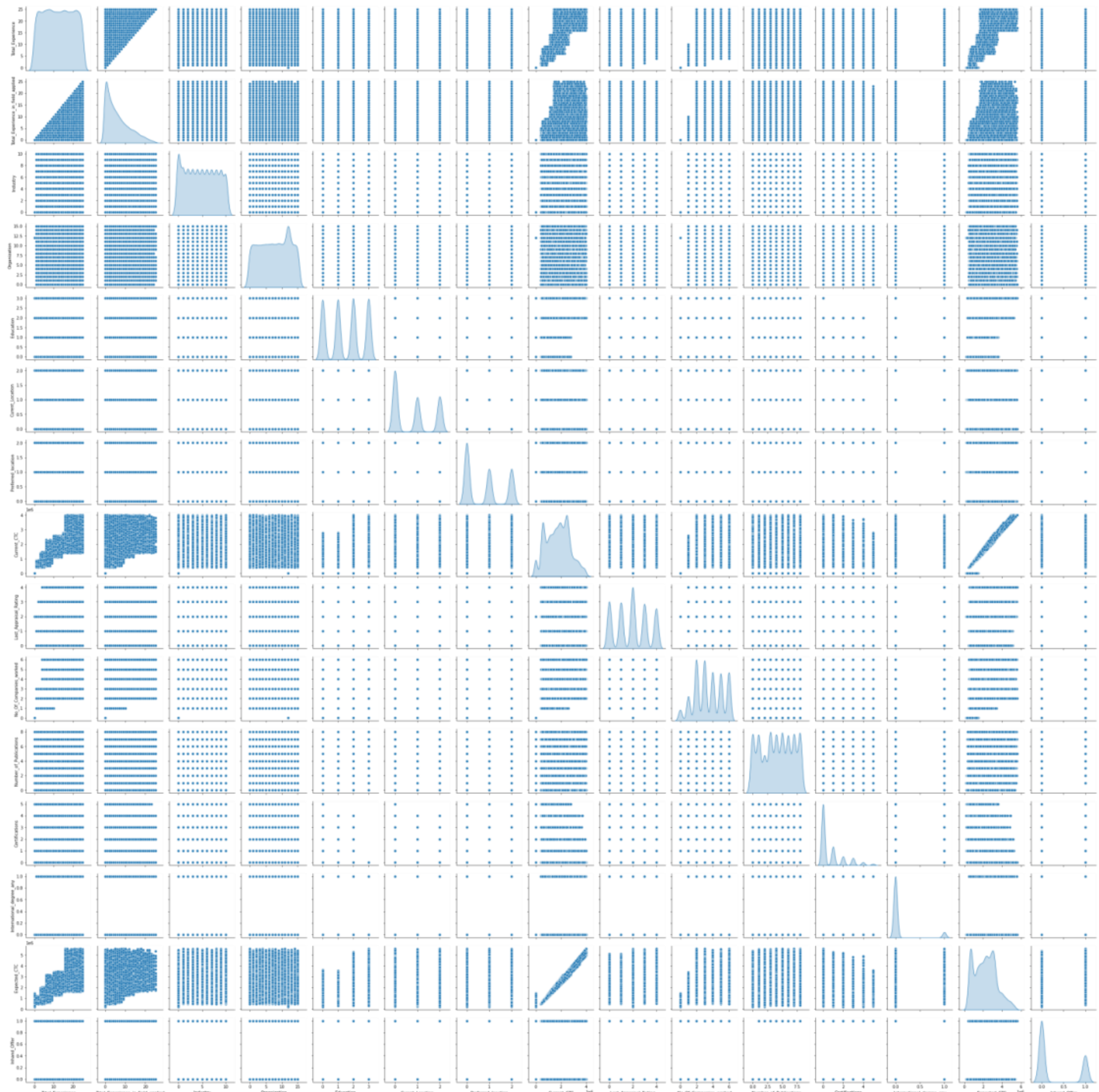

**Figure 21 : Pairplot**

# Sections 3 : Model Development

- After EDA comes the model building. In this phase , we approached with the problem statement and objective that we need to build a **Multi Linear Regression Model** to find the solution.
- Then we proceeded to build MLR model. With the help of statsmodel library, we builded a base model and find out that it contained multi-colinearity using VIF which made the model overfitted. Refer to VIF in the Appendix about it was used to find multi colinearity.
- So, we used the hyper parameter tuning and builded multiple models till the time all the multi-colinearity was nowhere to be found in the model.

OLS Regression Results

| | | | |
|---|---|---|---|
| Dep. Variable: | Expected_CTC | R-squared: | 0.987 |
| Model: | OLS | Adj. R-squared: | 0.987 |
| Method: | Least Squares | F-statistic: | 1.216e+05 |
| Date: | Sun, 11 Dec 2022 | Prob (F-statistic): | 0.00 |
| Time: | 18:02:55 | Log-Likelihood: | -2.8998e+05 |
| No. Observations: | 21944 | AIC: | 5.800e+05 |
| Df Residuals: | 21929 | BIC: | 5.801e+05 |
| Df Model: | 14 | | |
| Covariance Type: | nonrobust | | |

| | coef | std err | t | P>\|t\| | [0.025 | 0.975] |
|---|---|---|---|---|---|---|
| Intercept | -1.242e+05 | 4166.603 | -29.818 | 0.000 | -1.32e+05 | -1.16e+05 |
| Total_Experience | 79.6060 | 272.488 | 0.292 | 0.770 | -454.491 | 613.703 |
| Total_Experience_in_field_applied | -32.7446 | 204.342 | -0.160 | 0.873 | -433.269 | 367.780 |
| Industry | -7237.4487 | 281.327 | -25.726 | 0.000 | -7788.869 | -6686.028 |
| Organization | 3087.0167 | 195.558 | 15.786 | 0.000 | 2703.710 | 3470.324 |
| Education | 7.915e+04 | 1021.596 | 77.479 | 0.000 | 7.72e+04 | 8.12e+04 |
| Curent_Location | 1919.8534 | 1073.752 | 1.788 | 0.074 | -184.778 | 4024.484 |
| Preferred_location | 352.6345 | 1075.825 | 0.328 | 0.743 | -1756.061 | 2461.330 |
| Current_CTC | 1.2286 | 0.002 | 576.520 | 0.000 | 1.224 | 1.233 |
| Last_Appraisal_Rating | 7.479e+04 | 701.899 | 106.560 | 0.000 | 7.34e+04 | 7.62e+04 |
| No_Of_Companies_worked | -1.906e+04 | 584.104 | -32.624 | 0.000 | -2.02e+04 | -1.79e+04 |
| Number_of_Publications | 3040.5282 | 357.278 | 8.510 | 0.000 | 2340.238 | 3740.818 |
| Certifications | 2767.3163 | 1473.604 | 1.878 | 0.060 | -121.054 | 5655.687 |
| International_degree_any | -1.16e+04 | 3292.626 | -3.524 | 0.000 | -1.81e+04 | -5151.016 |
| Inhand_Offer | 2.05e+04 | 2159.064 | 9.496 | 0.000 | 1.63e+04 | 2.47e+04 |

| | | | |
|---|---|---|---|
| Omnibus: | 11935.057 | Durbin-Watson: | 1.997 |
| Prob(Omnibus): | 0.000 | Jarque-Bera (JB): | 153747.662 |
| Skew: | 2.343 | Prob(JB): | 0.00 |
| Kurtosis: | 15.091 | Cond. No. | 9.57e+06 |

**Figure 22 : Base Model**

- After all the efforts , we finally builded the model which was perfect in all its way and its R squared value = 0.987 & Adj. R squared value = 0.987

OLS Regression Results

| | | | |
|---|---|---|---|
| Dep. Variable: | Expected_CTC | R-squared: | 0.987 |
| Model: | OLS | Adj. R-squared: | 0.987 |
| Method: | Least Squares | F-statistic: | 1.892e+05 |
| Date: | Sun, 11 Dec 2022 | Prob (F-statistic): | 0.00 |
| Time: | 18:02:58 | Log-Likelihood: | -2.8998e+05 |
| No. Observations: | 21944 | AIC: | 5.800e+05 |
| Df Residuals: | 21934 | BIC: | 5.801e+05 |
| Df Model: | 9 | | |
| Covariance Type: | nonrobust | | |

| | coef | std err | t | P>\|t\| | [0.025 | 0.975] |
|---|---|---|---|---|---|---|
| Intercept | -1.201e+05 | 3777.338 | -31.792 | 0.000 | -1.27e+05 | -1.13e+05 |
| Industry | -7227.1469 | 281.269 | -25.695 | 0.000 | -7778.454 | -6675.840 |
| Organization | 3092.5457 | 195.549 | 15.815 | 0.000 | 2709.256 | 3475.836 |
| Education | 7.832e+04 | 870.851 | 89.941 | 0.000 | 7.66e+04 | 8e+04 |
| Current_CTC | 1.2291 | 0.001 | 1113.463 | 0.000 | 1.227 | 1.231 |
| Last_Appraisal_Rating | 7.48e+04 | 701.874 | 106.567 | 0.000 | 7.34e+04 | 7.62e+04 |
| No_Of_Companies_worked | -1.903e+04 | 580.300 | -32.800 | 0.000 | -2.02e+04 | -1.79e+04 |
| Number_of_Publications | 3039.9551 | 357.238 | 8.510 | 0.000 | 2339.743 | 3740.167 |
| International_degree_any | -1.148e+04 | 3290.858 | -3.487 | 0.000 | -1.79e+04 | -5025.612 |
| Inhand_Offer | 2.048e+04 | 2158.925 | 9.487 | 0.000 | 1.62e+04 | 2.47e+04 |

| | | | |
|---|---|---|---|
| Omnibus: | 11923.901 | Durbin-Watson: | 1.996 |
| Prob(Omnibus): | 0.000 | Jarque-Bera (JB): | 153498.521 |
| Skew: | 2.341 | Prob(JB): | 0.00 |
| Kurtosis: | 15.082 | Cond. No. | 8.61e+06 |

**Figure 23 : Final Model**

- After building the best model, we then checked the validation of model by trying to run the model on test data.
- As a result, the model run successfully . Thereby, confirming that the model build is very good.
- In addition to it, we plotted scatter plot & Distplot for the best model and giving the graphical representation of the best model build.

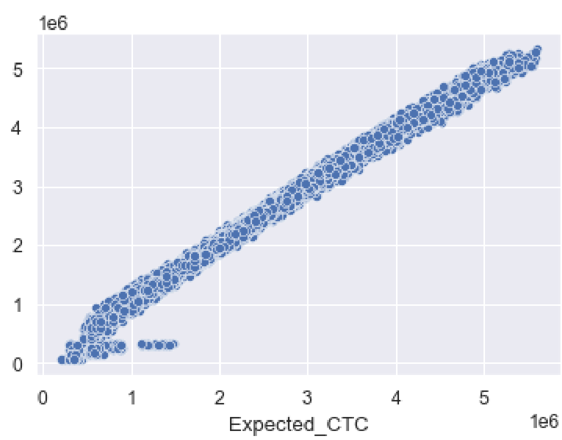


**Figure 24 : Scatter Plot - Train Data Best Model MLR**

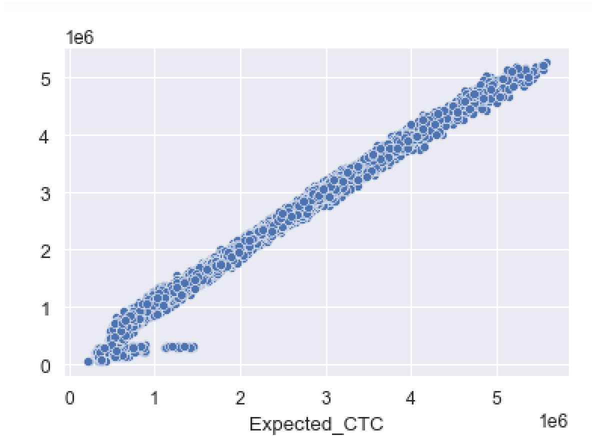


**Figure 25 : Scatter Plot - Test Data Best Model MLR**

**Interpretations :**

- This shows a linear relationship as the **predicted** and **actuals** values were very close to each other. Hence the R2 is also high.
- We inferred that both scatter plots for train and test are quite similar.
- It means that the model we build using sklearn library was a very good model. It fits well.

- Afterwards, we also did density plot for fitted values to check the predicted vs actual values. If the majority of plot overlaps then the model is very good fit.

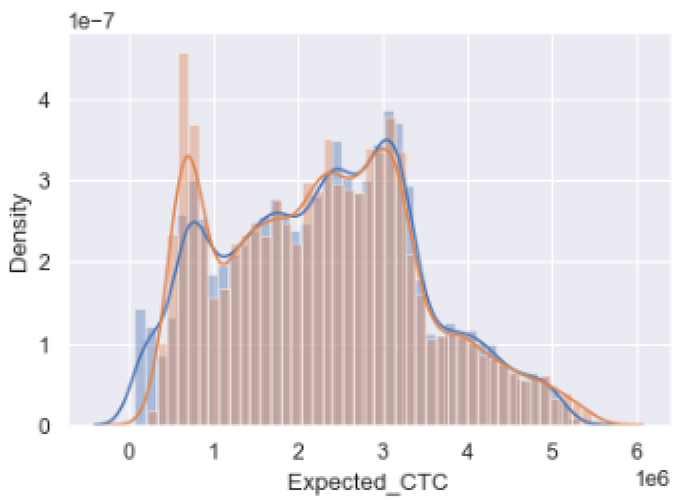


**Figure 26 : Density Plot - Fitted(predicted) vs Actual Values**

**Interpretations :**

- From the density plot, we inferred that the predicted values and actual values were predicting very much likely or in other words we can say they are very much similar to each other.
- It means that our linear regression model was very good.

**NOTE** – Refer to Appendix for MLR Best Model

**Conclusion:**

- Hence **Model 6** is the **Best Model** said to be a very good linear regression model as its **prediction value** is very close to the **actual value**.

## Sections 4 : Final Recommendation

- As per the model development and validation, we inferred some useful business insights with the help of which we are giving out point of business recommendations.
- These are as follows :
  - Current CTC is an important factor in determining the Expected CTC of an employee. So, the company should always keep in tab with the Current CTC of the candidate who is applying keeping in mind the role & department he/she is applying for.
  - Location is another important factor for the determination of Expected CTC. Nowadays, many employees prefer to work at their own preferred location because of many reasons including family, transportation etc. So, the company should come up with a strategy to attract the candidates who are well qualified so that they could neglect there preferred location and become willing to change their location. Strategies such as increased salary but would be incentives only(based on performance in the company). This kind of strategy is quite useful to attract candidates.
  - Employees who work earlier for big organization tends to become more successful , so company should also consider these types of candidates as their target to go for as they bring valuable knowledge and strategies that can help company grow to bigger heights.

## Bibliography

- Great Learning Notes & Videos
- Google
- KAGGLE

# Appendix

## 1. Data dictionary

- The full forms of the variables are present in this.

| IDX | Index |
|---|---|
| Applicant_ID | Application ID |
| Total_Experience | Total industry experience |
| Total_Experience_in_field_applied | Total experience in the field applied for (past work experience that is relevant to the job) |
| Department | Department name of current company |
| Role | Role in the current company |
| Industry | Industry name of current field |
| Organization | Organization name |
| Designation | Designation in current company |
| Education | Education |
| Graduation_Specialization | Specialization subject in graduation |
| University_Grad | University or college in Graduation |
| Passing_Year_Of_Graduation | Year of passing Graduation |
| PG_Specialization | Specialization subject in Post-Graduation |
| University_PG | University or college in Post-Graduation |
| Passing_Year_Of_PG | Year of passing Post Graduation |
| PHD_Specialization | Specialization subject in Post-Graduation |
| University_PHD | University or college in Post Doctorate |
| Passing_Year_Of_PHD | Year of passing PHD |
| Curent_Location | Curent Location |
| Preferred_location | Preferred location to work in the company applied |
| Current_CTC | Current CTC |
| Inhand_Offer | Holding any offer in hand (Y: Yes, N:No) |
| Last_Appraisal_Rating | Last Appraisal Rating in current company |
| No_Of_Companies_worked | No. of companies worked till date |
| Number_of_Publications | Number of papers published |
| Certifications | Number of relevant certifications completed |
| International_degree_any | Hold any international degree (1: Yes, 0: No) |
| Expected_CTC | Expected CTC (Final CTC offered by Delta Ltd.) |

**Figure 27 : Data Dictionary**

2. **Check For Null Values**

We used info function and it showed the data type, data shape and null values present in the data set.

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 25000 entries, 0 to 24999
Data columns (total 29 columns):
 #   Column                          Non-Null Count  Dtype  
---  ------                          --------------  -----  
 0   IDX                             25000 non-null  int64  
 1   Applicant_ID                    25000 non-null  int64  
 2   Total_Experience                25000 non-null  int64  
 3   Total_Experience_in_field_applied  25000 non-null  int64  
 4   Department                      22222 non-null  object 
 5   Role                            24037 non-null  object 
 6   Industry                        24092 non-null  object 
 7   Organization                    24092 non-null  object 
 8   Designation                     21871 non-null  object 
 9   Education                       25000 non-null  object 
 10  Graduation_Specialization       18820 non-null  object 
 11  University_Grad                 18820 non-null  object 
 12  Passing_Year_Of_Graduation      18820 non-null  float64
 13  PG_Specialization               17308 non-null  object 
 14  University_PG                   17308 non-null  object 
 15  Passing_Year_Of_PG              17308 non-null  float64
 16  PHD_Specialization              13119 non-null  object 
 17  University_PHD                  13119 non-null  object 
 18  Passing_Year_Of_PHD             13119 non-null  float64
 19  Curent_Location                 25000 non-null  object 
 20  Preferred_location              25000 non-null  object 
 21  Current_CTC                     25000 non-null  int64  
 22  Inhand_Offer                    25000 non-null  object 
 23  Last_Appraisal_Rating           24092 non-null  object 
 24  No_Of_Companies_worked          25000 non-null  int64  
 25  Number_of_Publications          25000 non-null  int64  
 26  Certifications                  25000 non-null  int64  
 27  International_degree_any        25000 non-null  int64  
 28  Expected_CTC                    25000 non-null  int64  
dtypes: float64(3), int64(10), object(16)
memory usage: 5.5+ MB
```

**Figure 28 : Null Values Check**

Interpretations :

- There are total 25000 entries in the dataset.
- In the dataset, there are 29 columns.
- Many columns have null values present in them. They are as follows :

a. Department
b. Role
c. Industry
d. Organization
e. Designation
f. Graduation_Specialization
g. University_Grad
h. Passing_Year_Of_Graduation
i.PG_Specialization
j. University_PG
K. Passing_Year_Of_PG
L. PHD_Specialization
M. University_PHD
N. Passing_Year_Of_PHD
O. Last_Appraisal_Rating

**Figure 29 : Null Values List**

## 3. Check for Anomalies/ Bad Data

- Below is the code for checking the anomalies for categorical data :

### Checking For Anomilies/Bad Data

In [12]:
```
for variable in cat:
    print(variable,":", sum(df[variable] == '?'))
```

```
Department : 0
Role : 0
Industry : 0
Organization : 0
Designation : 0
Education : 0
Graduation_Specialization : 0
University_Grad : 0
PG_Specialization : 0
University_PG : 0
PHD_Specialization : 0
University_PHD : 0
Curent_Location : 0
Preferred_location : 0
Inhand_Offer : 0
Last_Appraisal_Rating : 0
```

**Figure 30 : Anomalies Check Categorical Data 1**

```
In [13]: for variable in cat:
             print(variable,":", sum(df[variable] == '$'))
```

```
Department : 0
Role : 0
Industry : 0
Organization : 0
Designation : 0
Education : 0
Graduation_Specialization : 0
University_Grad : 0
PG_Specialization : 0
University_PG : 0
PHD_Specialization : 0
University_PHD : 0
Curent_Location : 0
Preferred_location : 0
Inhand_Offer : 0
Last_Appraisal_Rating : 0
```

**Figure 31 : Anomalies Check Categorical Data 2**

- Below is the code for checking anomalies in numerical data :

```
for variable in num:
    print(variable,":", sum(df[variable] == '?'))
```

```
IDX : 0
Applicant_ID : 0
Total_Experience : 0
Total_Experience_in_field_applied : 0
Passing_Year_Of_Graduation : 0
Passing_Year_Of_PG : 0
Passing_Year_Of_PHD : 0
Current_CTC : 0
No_Of_Companies_worked : 0
Number_of_Publications : 0
Certifications : 0
International_degree_any : 0
Expected_CTC : 0
```

**Figure 32 : Anomalies Check Numerical Data 1**

```python
for variable in num:
    print(variable,":", sum(df[variable] == '$'))
```

```
IDX : 0
Applicant_ID : 0
Total_Experience : 0
Total_Experience_in_field_applied : 0
Passing_Year_Of_Graduation : 0
Passing_Year_Of_PG : 0
Passing_Year_Of_PHD : 0
Current_CTC : 0
No_Of_Companies_worked : 0
Number_of_Publications : 0
Certifications : 0
International_degree_any : 0
Expected_CTC : 0
```

**Figure 33 : Anomalies Check Numerical Data 2**

## 4. Encoding

- We did encoding on necessary variables. They are as follows :

### Label Encoding

#### Transforming Data

In order to proceed to linear regression, all the columns must be in numerical format, thus all the necessary categorical data ('**Education**' in this case)must be changed into numerical data type.

```python
data['Education'].unique()
```

```
array(['PG', 'Doctorate', 'Grad', 'Under Grad'], dtype=object)
```

```python
data["Education"]=data["Education"].replace({"Under Grad":0,"Grad":1,"PG":2,"Doctorate":3})
```

```python
data['Education'].unique()
```

```
array([2, 3, 1, 0], dtype=int64)
```

**Interpretation :**

- The '**Education**' variable had been transformed into numerical(int) successfully and also assigned the number according to the priority wise.

**Figure 34 : Appendix Encoding 1**

### Ordinal Encoding

```python
data['Last_Appraisal_Rating'].unique()
```

```
array(['B', 'Key_Performer', 'C', 'A', 'D'], dtype=object)
```

```python
data["Last_Appraisal_Rating"]=data["Last_Appraisal_Rating"].replace({"D":0,"C":1,"B":2,"A":3,"Key_Performer":4})
```

```python
data['Last_Appraisal_Rating'].unique()
```

```
array([2, 4, 1, 3, 0], dtype=int64)
```

**Interpretation :**

- We had successfully done ordinal encoding.

**Figure 35 : Appendix Encoding 2**

### Label Encoding for 'Inhand_Offer'

```
# Importing LabelEncoder from Sklearn
# library from preprocessing Module.
from sklearn.preprocessing import LabelEncoder

# Creating a instance of label Encoder.
le = LabelEncoder()

# Using .fit_transform function to fit label
# encoder and return encoded label
label = le.fit_transform(data['Inhand_Offer'])

# printing label
label
```

```
array([0, 1, 1, ..., 0, 1, 0])
```

**Figure 36 : Appendix Encoding 3**

```
# removing the column 'Inhand_Offer' from data as it is of no use now.
data.drop("Inhand_Offer", axis=1, inplace=True)

# Appending the array to our dataFrame with column name 'Inhand_Offer'
data["Inhand_Offer"] = label

# printing Dataframe
data

#For 'Inhand_Offer', after Label Encoding 'N' represents '0' & 'Y' represents '1'.
```

| n | Current_CTC | Last_Appraisal_Rating | No_Of_Companies_worked | Number_of_Publications | Certifications | International_degree_any | Expected_CTC | Inhand_Offer |
|---|---|---|---|---|---|---|---|---|
| .1 | 0 | 2 | 0 | 0 | 0 | 0 | 384551 | 0 |
| .3 | 2702664 | 4 | 2 | 4 | 0 | 0 | 3783729 | 1 |
| .2 | 2236661 | 4 | 5 | 3 | 0 | 0 | 3131325 | 1 |
| .1 | 2100510 | 1 | 5 | 3 | 0 | 0 | 2608833 | 0 |
| .1 | 1931644 | 1 | 2 | 3 | 0 | 0 | 2221390 | 0 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... |
| .1 | 3410899 | 2 | 3 | 6 | 0 | 0 | 4434168 | 0 |
| .3 | 1350793 | 2 | 6 | 7 | 0 | 0 | 1756030 | 1 |
| .1 | 1681796 | 1 | 4 | 5 | 2 | 0 | 1934065 | 0 |
| .1 | 3311090 | 2 | 3 | 1 | 1 | 0 | 4370638 | 1 |
| .3 | 935897 | 3 | 2 | 6 | 0 | 0 | 1216666 | 0 |

**Figure 37 : Appendix Encoding 4**

**Transforming Data**

```
data["Industry"]=data["Industry"].replace({"Training":0,"IT":1,"Insurance":2,"BFSI":3,"Automobile":4,"Analytics":5,"Retail":6,"T

data["Organization"]=data["Organization"].replace({"A":0,"B":1,"C":2,"D":3,"E":4,"F":5,"G":6,"H":7,"I":8,"J":9,"K":10,"L":11,"M"

data["Curent_Location"]=data["Curent_Location"].replace({"Tier_1":0,"Tier_2":1,"Tier_3":2})

data["Preferred_location"]=data["Preferred_location"].replace({"Tier_1":0,"Tier_2":1,"Tier_3":2})
```

**Figure 38 : Appendix Encoding 5**

## 5. Outliers

- Below is the code present for outliers treatment :

All the **Outliers** present in all three variables are as follows :

```
Q1_Certifications = np.percentile(data['Certifications'], 25, interpolation = 'midpoint')
Q2_Certifications = np.percentile(data['Certifications'], 50, interpolation = 'midpoint')
Q3_Certifications = np.percentile(data['Certifications'], 75, interpolation = 'midpoint')
IQR_Certifications= Q3_Certifications - Q1_Certifications
print('Interquartile range is', IQR_Certifications)
low_lim_Certifications = Q1_Certifications - 1.5 * IQR_Certifications
up_lim_Certifications = Q3_Certifications + 1.5 * IQR_Certifications
print('low_limit is', low_lim_Certifications)
print('up_limit is', up_lim_Certifications)
outlier_Certifications =[]
for x in data['Certifications']:
    if ((x> up_lim_Certifications) or (x<low_lim_Certifications)):
        outlier_Certifications.append(x)
print(' outlier in the dataset is', outlier_Certifications)
```

```
Interquartile range is 1.0
low_limit is -1.5
up_limit is 2.5
 outlier in the dataset is [5, 4, 3, 3, 3, 4, 3, 3, 3, 4, 3, 3, 3, 4, 3, 3, 4, 5, 4, 3, 3, 4, 3, 3, 5, 3, 3, 3, 4, 4, 4, 4, 3,
3, 3, 5, 3, 5, 3, 4, 3, 3, 4, 3, 4, 4, 3, 3, 4, 3, 3, 3, 3, 3, 3, 3, 3, 3, 4, 3, 3, 5, 3, 3, 4, 3, 5, 3, 3, 3, 3, 3, 5, 4, 3,
3, 3, 3, 3, 3, 5, 3, 4, 3, 3, 3, 5, 3, 3, 4, 4, 4, 3, 3, 3, 3, 4, 3, 4, 5, 3, 5, 3, 3, 3, 3, 4, 3, 3, 3, 4, 3, 3, 3, 3, 3, 3,
5, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 4, 3, 3, 4, 3, 5, 3, 3, 3, 4, 3, 4, 3, 3, 3, 3, 5, 5, 3, 3, 4, 3, 3, 4, 3, 3, 3, 3, 5,
3, 4, 5, 3, 3, 3, 3, 5, 3, 4, 3, 4, 3, 3, 3, 4, 3, 3, 3, 3, 4, 3, 3, 3, 4, 3, 3, 5, 3, 3, 3, 5, 3, 4, 3, 3, 5, 5, 5, 3, 3, 3,
3, 3, 3, 4, 3, 3, 3, 3, 3, 3, 3, 3, 4, 5, 3, 5, 3, 3, 3, 5, 3, 3, 5, 3, 4, 4, 5, 3, 4, 4, 4, 3, 3, 3, 5, 3, 3, 3, 3, 4, 3, 3,
4, 4, 3, 3, 3, 3, 3, 3, 3, 5, 3, 3, 3, 3, 3, 3, 3, 5, 4, 4, 4, 3, 3, 4, 3, 3, 3, 5, 3, 3, 5, 4, 3, 3, 3, 3, 3, 4, 3, 4, 5, 3,
```

**Figure 39 : Outlier Treatment 1**

```
data['Certifications'].value_counts()
```

```
0    15215
1     4644
2     2198
3     1818
4      777
5      348
Name: Certifications, dtype: int64
```

```
(data['Certifications'] > 2.5).value_counts()
```

```
False    22057
True      2943
Name: Certifications, dtype: int64
```

```
outlier_filter_Certifications = data['Certifications'] < 2.5
data = data[outlier_filter_Certifications]
```

```
(data['Certifications'] > 2.5).value_counts()
```

```
False    22057
Name: Certifications, dtype: int64
```

**Outlier** from **"Certifications"** Removed Successfully.

**Figure 40 : Outlier Treatment 2**

```
Q1_Total_Experience_in_field_applied = np.percentile(data['Total_Experience_in_field_applied'], 25, interpolation = 'midpoint')
Q2_Total_Experience_in_field_applied = np.percentile(data['Total_Experience_in_field_applied'], 50, interpolation = 'midpoint')
Q3_Total_Experience_in_field_applied = np.percentile(data['Total_Experience_in_field_applied'], 75, interpolation = 'midpoint')
IQR_Total_Experience_in_field_applied = Q3_Total_Experience_in_field_applied - Q1_Total_Experience_in_field_applied
print('Interquartile range is', IQR_Total_Experience_in_field_applied)
low_lim_Total_Experience_in_field_applied = Q1_Total_Experience_in_field_applied - 1.5 * IQR_Total_Experience_in_field_applied
up_lim_Total_Experience_in_field_applied = Q3_Total_Experience_in_field_applied + 1.5 * IQR_Total_Experience_in_field_applied
print('low_limit is', low_lim_Total_Experience_in_field_applied)
print('up_limit is', up_lim_Total_Experience_in_field_applied)
outlier =[]
for y in data['Total_Experience_in_field_applied']:
    if ((y > up_lim_Total_Experience_in_field_applied) or (y < low_lim_Total_Experience_in_field_applied)):
        outlier.append(y)
print(' outlier in the dataset is', outlier)
```

```
Interquartile range is 9.0
low_limit is -12.5
up_limit is 23.5
 outlier in the dataset is [25, 25, 25, 24, 24, 25, 24, 24, 24, 24, 24, 24, 25, 24, 24, 25, 24, 24, 24, 25, 25, 25, 24, 25, 24,
25, 24, 24, 24, 25, 25, 24, 24, 24, 25, 25, 24, 24, 25, 24, 24, 24, 24, 25, 24, 24, 25, 24, 24, 25, 24, 24, 24, 24, 25, 24, 24,
24, 24, 25, 24, 24, 24, 24, 25, 24, 25, 24, 24, 25, 24, 24, 24, 25, 24, 24, 24, 24, 24, 24, 25, 24, 24, 24, 24, 25, 24, 24, 25,
24, 24, 24, 24, 25, 24, 25, 24, 24, 25, 24, 24, 25, 24, 24, 24, 25, 25, 24, 24, 24, 24, 24, 25]
```

Activate Wi

**Figure 41 : Outlier Treatment 3**

```
(data['Total_Experience_in_field_applied'] > 23.5).value_counts()
```

```
False    21944
True       113
Name: Total_Experience_in_field_applied, dtype: int64
```

```
outlier_filter_Total_Experience_in_field_applied = data['Total_Experience_in_field_applied'] < 23.5
data = data[outlier_filter_Total_Experience_in_field_applied]
```

```
(data['Total_Experience_in_field_applied'] > 23.5).value_counts()
```

```
False    21944
Name: Total_Experience_in_field_applied, dtype: int64
```

**Outlier** from **"Total_Experience_in_field_applied"** Removed Successfully.

**Figure 42 : Outlier Treatment 4**

- After removing all outliers, then we plotted the graph of boxplot. It is as follows :

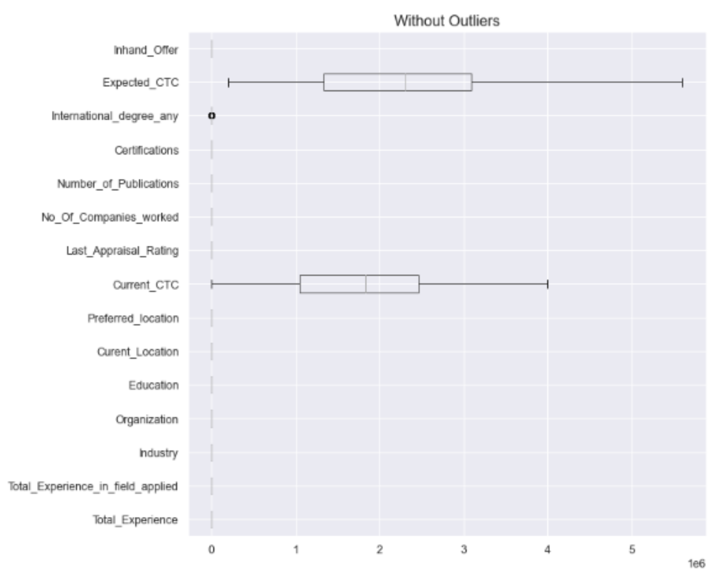


**Figure 43 : Appendix After Outlier Treatment**

## 6. Encoding & Grouping

- This is how encoding and grouping was done :

### Ordinal Encoding

```
data['Last_Appraisal_Rating'].unique()
```

```
array(['B', 'Key_Performer', 'C', 'A', 'D'], dtype=object)
```

```
data["Last_Appraisal_Rating"]=data["Last_Appraisal_Rating"].replace({"D":0,"C":1,"B":2,"A":3,"Key_Performer":4})
```

```
data['Last_Appraisal_Rating'].unique()
```

```
array([2, 4, 1, 3, 0], dtype=int64)
```

**Interpretation :**

- We had successfully done ordinal encoding.

**Figure 44 : Ordinal Encoding 1**

**Transforming Data**

```
data["Industry"]=data["Industry"].replace({"Training":0,"IT":1,"Insurance":2,"BFSI":3,"Automobile":4,"Analytics":5,"Retail":6,"Te

data["Organization"]=data["Organization"].replace({"A":0,"B":1,"C":2,"D":3,"E":4,"F":5,"G":6,"H":7,"I":8,"J":9,"K":10,"L":11,"M":

data["Curent_Location"]=data["Curent_Location"].replace({"Tier_1":0,"Tier_2":1,"Tier_3":2})

data["Preferred_location"]=data["Preferred_location"].replace({"Tier_1":0,"Tier_2":1,"Tier_3":2})
```

```
data.drop(['Applicant_ID'],axis=1,inplace= True)
```

**Figure 45 : Ordinal Encoding 2**

### Label Encoding

**Transforming Data**

In order to proceed to linear regression, all the columns must be in numerical format, thus all the necessary categorical data ('**Education**' in this case)must be changed into numerical data type.

```
data['Education'].unique()
```

```
array(['PG', 'Doctorate', 'Grad', 'Under Grad'], dtype=object)
```

```
data["Education"]=data["Education"].replace({"Under Grad":0,"Grad":1,"PG":2,"Doctorate":3})
```

```
data['Education'].unique()
```

```
array([2, 3, 1, 0], dtype=int64)
```

**Interpretation :**

- The '**Education**' variable had been transformed into numerical(int) successfully and also assigned the number according to the priority wise.

**Figure 46 : Label Encoding 1**

## Label Encoding for 'Inhand_Offer'

```python
# Importing LabelEncoder from Sklearn
# library from preprocessing Module.
from sklearn.preprocessing import LabelEncoder

# Creating a instance of label Encoder.
le = LabelEncoder()

# Using .fit_transform function to fit label
# encoder and return encoded label
label = le.fit_transform(data['Inhand_Offer'])

# printing label
label
```

```
array([0, 1, 1, ..., 0, 1, 0])
```

**Figure 47 : Label Encoding 2**

```python
# removing the column 'Inhand_Offer' from data as it is of no use now.
data.drop("Inhand_Offer", axis=1, inplace=True)

# Appending the array to our dataFrame with column name 'Inhand_Offer'
data["Inhand_Offer"] = label

# printing Dataframe
data

#For 'Inhand_Offer', after Label Encoding 'N' represents '0' & 'Y' represents '1'.
```

| | Applicant_ID | Total_Experience | Total_Experience_in_field_applied | Industry | Organization | Education | Curent_Location | Preferred_location | Current_CTC |
|---|---|---|---|---|---|---|---|---|---|
| **0** | 22753 | 0 | 0 | NaN | NaN | 2 | Tier_3 | Tier_1 | 0 |
| **1** | 51087 | 23 | 14 | Analytics | H | 3 | Tier_1 | Tier_3 | 2702664 |
| **2** | 38413 | 21 | 12 | Training | J | 3 | Tier_1 | Tier_2 | 2236661 |
| **3** | 11501 | 15 | 8 | Aviation | F | 3 | Tier_2 | Tier_1 | 2100510 |
| **4** | 58941 | 10 | 5 | Insurance | E | 1 | Tier_1 | Tier_1 | 1931644 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **24995** | 25550 | 18 | 13 | Automobile | I | 2 | Tier_2 | Tier_1 | 3410899 |
| **24996** | 53442 | 12 | 8 | Analytics | B | 0 | Tier_1 | Tier_3 | 1350793 |
| **24997** | 15777 | 22 | 8 | Insurance | D | 0 | Tier_1 | Tier_1 | 1681796 |
| **24998** | 57616 | 25 | 8 | BFSI | D | 2 | Tier_1 | Tier_1 | 3311090 |
| **24999** | 20788 | 8 | 0 | Automobile | P | 1 | Tier_2 | Tier_3 | 935897 |

25000 rows × 16 columns



**Figure 48 : Label Encoding 3**

7. **Imputation**

- We imputed using central tendency mode to impute null values .

**Imputation Of Values**

**Industry**

```
]: data['Industry'] = data['Industry'].fillna(data['Industry'].mode()[0])
```

**Organization**

```
]: data['Organization'] = data['Organization'].fillna(data['Organization'
```

```
]: data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 25000 entries, 0 to 24999
Data columns (total 16 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   Applicant_ID                    25000 non-null  int64
 1   Total_Experience                25000 non-null  int64
 2   Total_Experience_in_field_applied  25000 non-null  int64
 3   Industry                        25000 non-null  category
 4   Organization                    25000 non-null  category
 5   Education                       25000 non-null  int64
 6   Curent_Location                 25000 non-null  object
 7   Preferred_location              25000 non-null  object
 8   Current_CTC                     25000 non-null  int64
 9   Last_Appraisal_Rating           25000 non-null  int64
 10  No_Of_Companies_worked          25000 non-null  int64
 11  Number_of_Publications          25000 non-null  int64
 12  Certifications                  25000 non-null  int64
 13  International_degree_any        25000 non-null  int64
 14  Expected_CTC                    25000 non-null  int64
 15  Inhand_Offer                    25000 non-null  int32
dtypes: category(2), int32(1), int64(11), object(2)
```

**Figure 49 : Appendix Imputation**

## 8. Model Development

- Multi colinearity was checked using VIF and then removed accordingly.

Now, let us check and treat the multicollinearity problem if it is present.

Now, we will calculate the Variance Inflation Factor (VIF). We will calculate the Variance Inflation Factor by an user defined function.

VIF regresses the dependent variables amongst themselves and then calculates the VIF values based on the $R2$ of each such regression.

```
: def vif_cal(input_data):
      x_vars=input_data
      xvar_names=input_data.columns
      for i in range(0,xvar_names.shape[0]):
          y=x_vars[xvar_names[i]]
          x=x_vars[xvar_names.drop(xvar_names[i])]
          rsq=SM.ols(formula="y~x", data=x_vars).fit().rsquared
          vif=round(1/(1-rsq),2)
          print (xvar_names[i], " VIF = " , vif)
```

**Figure 50 : VIF - Multi colinearity Check 1**

```
vif_cal(input_data=data[['Total_Experience', 'Total_Experience_in_field_applied', 'Industry', 'Organization', 'Education',
        'Curent_Location', 'Preferred_location', 'Current_CTC', 'Last_Appraisal_Rating','No_Of_Companies_worked',
        'Number_of_Publications','Certifications','International_degree_any','Inhand_Offer']])
```

```
Total_Experience  VIF =  5.14
Total_Experience_in_field_applied  VIF =  1.7
Industry  VIF =  1.03
Organization  VIF =  1.01
Education  VIF =  1.52
Curent_Location  VIF =  1.0
Preferred_location  VIF =  1.0
Current_CTC  VIF =  4.92
Last_Appraisal_Rating  VIF =  1.13
No_Of_Companies_worked  VIF =  1.22
Number_of_Publications  VIF =  1.09
Certifications  VIF =  1.19
International_degree_any  VIF =  1.01
Inhand_Offer  VIF =  1.22
```

**Interpretations :**

- From the **Base Model** , we inferred that many variables have p_value higher than the significance vlaue(0.05).
- The variables that have higher p_value than significance value are **'Total_Experience'** , **'Total_Experience_in_field_applied'** , **'Curent_Location'** , **'Preferred_location'** , **'Certifications'** .
- Out of all these, **'Total_Experience_in_field_applied'** have maximum p_value, thus it is the most insignificant and shall be removed and a new model shall be build.

**Figure 51 : VIF Multi Colinearity Check 2**

- Scatter Plot of Residual of the **6th Model MLR** also called as **Best Model**

```
#Linear Relationship b/w Dependent and Independent Varaibles

sns.scatterplot(model_MLR_6.resid,model_MLR_6.fittedvalues)
```

<AxesSubplot:>

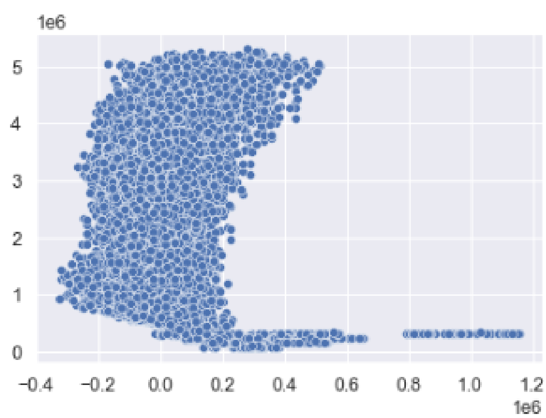


**Figure 52 : Appendix Residual Plot - Best Model MLR**