EMPLOYEES SALARY PREDICTION
CAPSTONE PROJECT- FINAL REPORT

Submitted towards partial fulfilment of the criteria
for award of PGP-DSBA by GLIM

SUBMITTED BY
Arun Sivaji
PGP-DSBA Bangalore 2021-22

Project Mentor & Research Supervisor:
Mr. Rishabh Pandey

Great Lakes Institute of Management
www.greatlakes.edu.in/Bangalore

Acknowledgements

We wish to place on record our deep appreciation for the guidance and help provided to us by our Mentor Mr. Rishabh Pandey, Bangalore.

Mr. Rishabh Pandey helps me narrow down on the choice of the Project as well as the scope and focus area of the Project. He gave us valuable feedback at every stage to enhance the process and the outputs.

We would also like to place on record our appreciation for the guidance provided by Mr.Nimesh Mafartia for giving us valuable feedback and being a source of inspiration in helping us to work on this project.

We certify that the work done by us for conceptualizing and completing this project is original and authentic.

Date: October 09, 2022

Place: Bangalore

Certificate of Completion

I hereby certify that the project titled “Employee Salary Prediction” was undertaken and completed under my supervision by Arun Sivaji student of the Postgraduate Program in Data Science & Business Analytics (PGP DSBA-October:2021).

Date: October 09, 2022

Place: Bangalore

Mr. Rishabh Pandey

Mentor

TABLE OF CONTENTS

CHAPTER 1. INTRODUCTION

Business Problem: To ensure there is no discrimination between employees, it is imperative for the Human Resources department of Delta Ltd. to maintain a salary range for each employee with similar profiles Apart from the existing salary, there is a considerable number of factors regarding an employee's experience and other abilities to which they get evaluated in interviews. Given the data related to individuals who applied in Delta Ltd, models can be built that can automatically determine salary which should be offered if the prospective candidate is selected in the company. This model seeks to minimize human judgment with regard to salary to be offered.

Goal & Objective: The objective of this exercise is to build a model, using historical data that will determine an employee's salary to be offered, such that manual judgments on selection are minimized. It is intended to have a robust approach and eliminate any discrimination in salary among similar employee profiles.

File: Data.csv

Target variable: Expected_CTC

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.)

1.1 OBJECTIVE AND SCOPE OF THE PROJECT:

1.0.1 Objective: The Primary objectives of the study are:

- Study the Employee Data to identify patterns of Expected salary w.r.t to various monitored parameters.
- Identify the Metrological factors that correlate with the other variables w.r.t Expected salary.
- Explore the possibility of developing a Predictive Model for predicting the level for key Expected Salary.

1.0.2 Scope:

- The scope of the study covers to analyse Expected salary to employees.
- The study covers to automate the Employees salary predicts.
- Models can be built that can automatically determine salary which should be offered if the prospective candidate is selected in the company. This model seeks to minimize human judgment with regard to salary to be offered

1.2 TOOLS & TECHNIQUES:

We have used the following Analytical techniques/Methodology for analysing the Data

1. Summary Statistics for each variable

2. Identification of frequency of standard violation for each of the factors

3. Using Graphs and Box Plots to visually represent them

4. Identification of significant Metrological factors through correlation and regression methodology 5. Using Multiple Linear Regression & Neural Network for Model Development

6. Tools used: Python, Tableau & Excel

7. Techniques: Box Plot, Histogram, Bar Chart, Line Chart, Infographics, Visual Clues, Correlation Matrix, Multiple Linear Regression, Artificial Neural Network

8. We have used Python Programming environment for our analysis and Tableau for data visualization

1.3 ANALYTICAL APPROACH:

The Analytical Approach will involve the following (not necessarily in the order) activities:

- Data extraction from Primary Data source as well as secondary data sources
- Data quality check
- Data cleaning and data preparation

- Study each of the variables by exploring the data
- Study the variables for its relevance for the study
- Identifying Y variable(s).
- Performing Univariate analysis for all variables
- Division of data into train and test
- Model Development
- Final Model
- Model Validation & Model Validation on Test
- Intervention Strategies and recommendations

We plan to use the following Seven Step Analytical Approach to the Project:

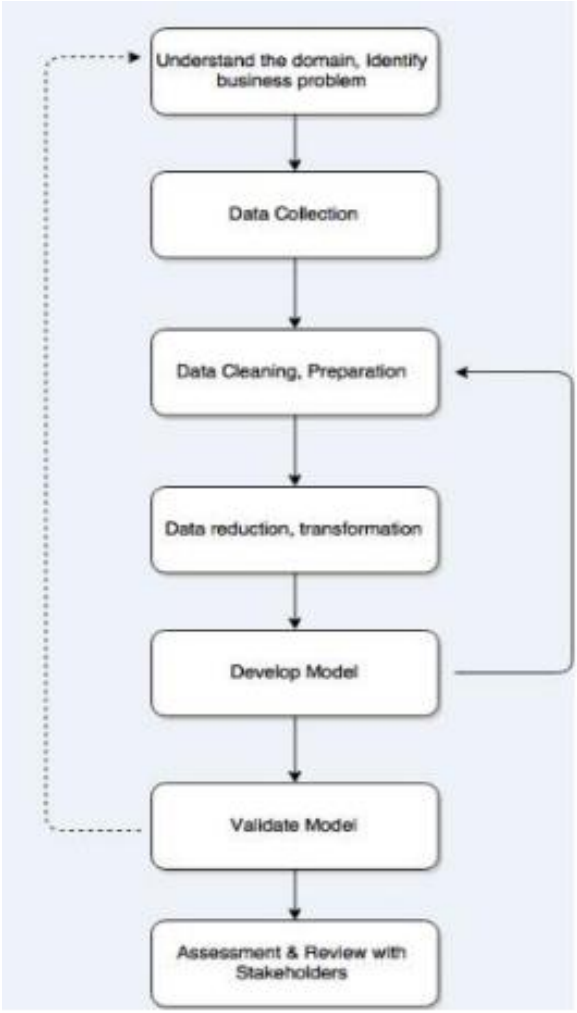


Figure 1: High Level Process Flow

1.4 Limitations:-

There are few limitations that this study has w.r.t data and the methodology that can be used.

- Due to time and cost constraints we could not deploy a primary source for data collection. We were not in a position to deploy primary employee data collection by deploying near ground level monitoring system that are typically used in advanced countries for such employee salary studies.

- Due to a very short window of 15 days for the Odd-Even Campaign, we had to live with a very small data size rendering the data unusable for any kind of rigorous statistical analysis.

- Since the Analysis & Models were built specifically for a particular company, the insights and the Models cannot be used for other locations in specific company.

- Since the Models were built on rather small data size (about for one company), the models need to be strengthened with at least another company or two data. Till such time the Models are likely to work in a larger range of values.

1.5 Data Collection:

Primary data is collected from the first-hand experience and is not used in the past. The data gathered by primary data collection methods are specific to the research's motive and highly accurate. Primary data collection methods can be divided into two categories: quantitative methods and qualitative methods.
The concept of data collection isn't a new one, as we'll see later, but the world has changed. There is far more data available today, and it exists in forms that were unheard of a century ago. The data collection process has had to change and grow with the times, keeping pace with technology

1.6 DATA TABLE – LIST OF VARIABLES

List of Variables and their types		
Variable	**Variable type**	**Data type**
Applicant_ID	Numerical	int64
Total_Experience	Numerical	int64
Total_Experience_in_field_applied	Numerical	int64
Department	Categorical	Object
Role	Categorical	Object
Industry	Categorical	Object
Organization	Categorical	Object
Designation	Categorical	Object
Education	Categorical	Object
Graduation_Specialization	Categorical	Object
University_Grad	Categorical	Object
Passing_Year_Of_Graduation	Categorical	Object
PG_Specialization	Categorical	Object
University_PG	Categorical	Object
Passing_Year_Of_PG	Categorical	Object
PHD_Specialization	Categorical	Object
University_PHD	Categorical	Object
Passing_Year_Of_PHD	Categorical	float64
Curent_Location	Categorical	Object
Preferred_location	Categorical	Object
Current_CTC	Numerical	int64
Inhand_Offer	Categorical	Object
Last_Appraisal_Rating	Categorical	Object
No_Of_Companies_worked	Categorical	int64
Number_of_Publications	Categorical	int64
Certifications	Categorical	int64
International_degree_any	Categorical	int64
Expected_CTC	Numerical	int64

1.7 Visual inspection of data (rows, columns, descriptive details)
Below image shows that data consist of 29 variables.
No of rows is 25000
No of columns is 29.
There is no any duplicate data's found.

	IDX	Applicant_ID	Total_Experience	Total_Experience_in_field_applied	Department	Role	Industry	Organization	Designation	Education	...
0	1	22753	0	0	NaN	NaN	NaN	NaN	NaN	PG	...
1	2	51087	23	14	HR	Consultant	Analytics	H	HR	Doctorate	...
2	3	38413	21	12	Top Management	Consultant	Training	J	NaN	Doctorate	...
3	4	11501	15	8	Banking	Financial Analyst	Aviation	F	HR	Doctorate	...
4	5	58941	10	5	Sales	Project Manager	Insurance	E	Medical Officer	Grad	...

5 rows × 29 columns

Figure 2: Dataset

Descriptive Details:-

	count	mean	std	min	25%	50%	75%	max
IDX	25000.0	1.250050e+04	7.217023e+03	1.0	6250.75	12500.5	18750.25	25000.0
Applicant_ID	25000.0	3.499324e+04	1.439027e+04	10000.0	22563.75	34974.5	47419.00	60000.0
Total_Experience	25000.0	1.249308e+01	7.471398e+00	0.0	6.00	12.0	19.00	25.0
Total_Experience_in_field_applied	25000.0	6.254340e+00	5.807706e+00	0.0	1.00	5.0	10.00	23.5
Current_CTC	25000.0	1.760945e+06	9.202125e+05	0.0	1027311.50	1802567.5	2443883.25	3999693.0
No_Of_Companies_worked	25000.0	3.482040e+00	1.690335e+00	0.0	2.00	3.0	5.00	6.0
Number_of_Publications	25000.0	4.089040e+00	2.606612e+00	0.0	2.00	4.0	6.00	8.0
Certifications	25000.0	7.736800e-01	1.199449e+00	0.0	0.00	0.0	1.00	5.0
International_degree_any	25000.0	8.172000e-02	2.739431e-01	0.0	0.00	0.0	0.00	1.0
Expected_CTC	25000.0	2.250155e+06	1.160480e+06	203744.0	1306277.50	2252136.5	3051353.75	5599570.0

Figure 3: Descriptive dataset

1. Numeric fields there is no any missing values.
2. Current salary and Expected salary variables are in continuous variables, So here we h ave to regression Algorithms.
3. Our target variables ha continuous so we have to do log transformation.
4. Independent variables Current _Salary also Log transformation need to be done.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 25000 entries, 0 to 24999
Data columns (total 22 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  PG_Specialization                   17308 non-null  object
 12  Curent_Location                     25000 non-null  object
 13  Preferred_location                  25000 non-null  object
 14  Current_CTC                         25000 non-null  int64
 15  Inhand_Offer                        25000 non-null  object
 16  Last_Appraisal_Rating               24092 non-null  object
 17  No_Of_Companies_worked              25000 non-null  int64
 18  Number_of_Publications              25000 non-null  int64
 19  Certifications                      25000 non-null  int64
 20  International_degree_any            25000 non-null  int64
 21  Expected_CTC                        25000 non-null  int64
dtypes: int64(10), object(12)
memory usage: 4.2+ MB
```

Figure 4: Dataset info

2. EDA and Business Implication

2.1 Uni-variate analysis to understand relationship b/w variables

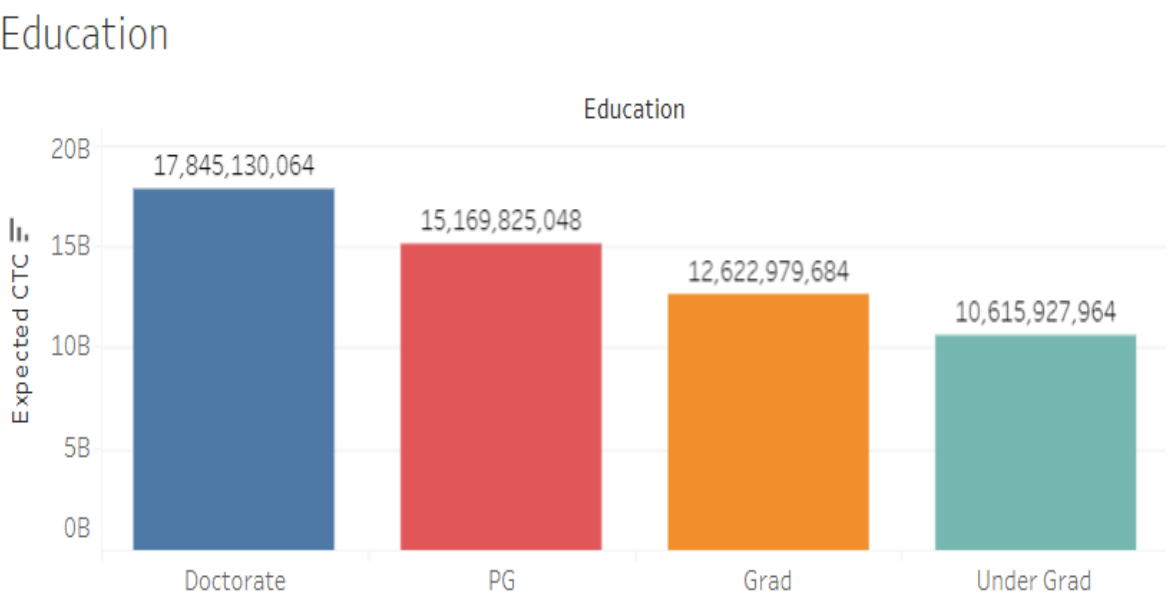


Figure: 5 Education vs Expected Salary

Doctorate professionals are expecting bit higher than the PG graduates. Under graduates salary expectation is low when compared to other professionals.

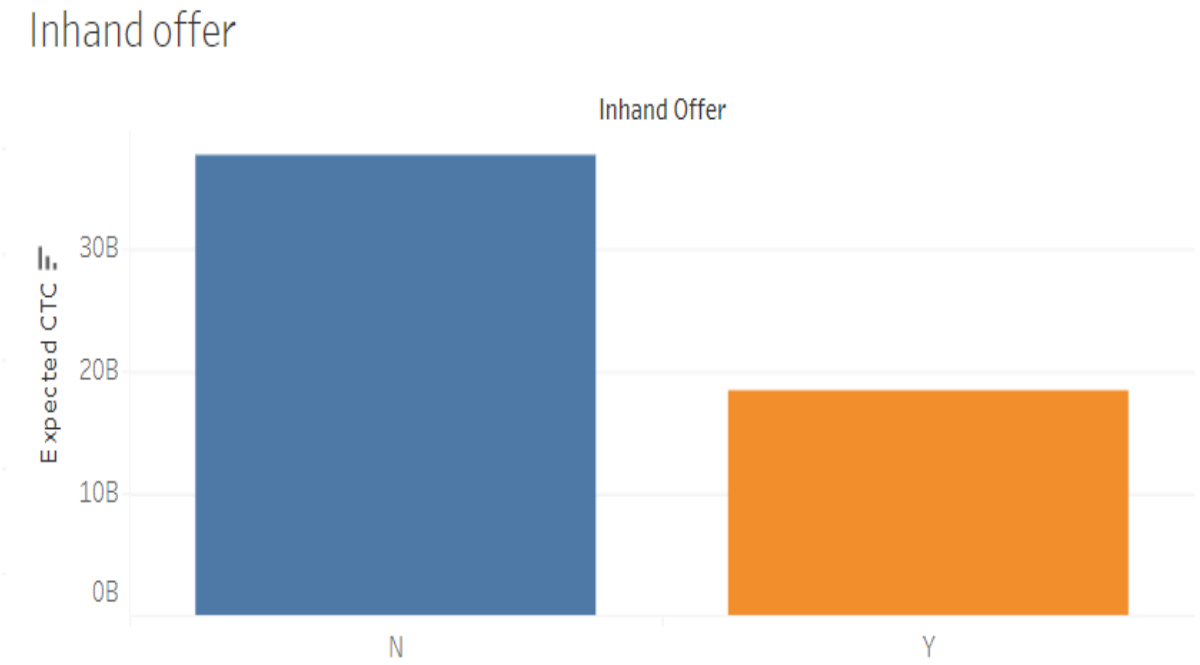


Figure 6: In hand offer vs Expected Salary

Most of the candidates not carrying In hand offer.

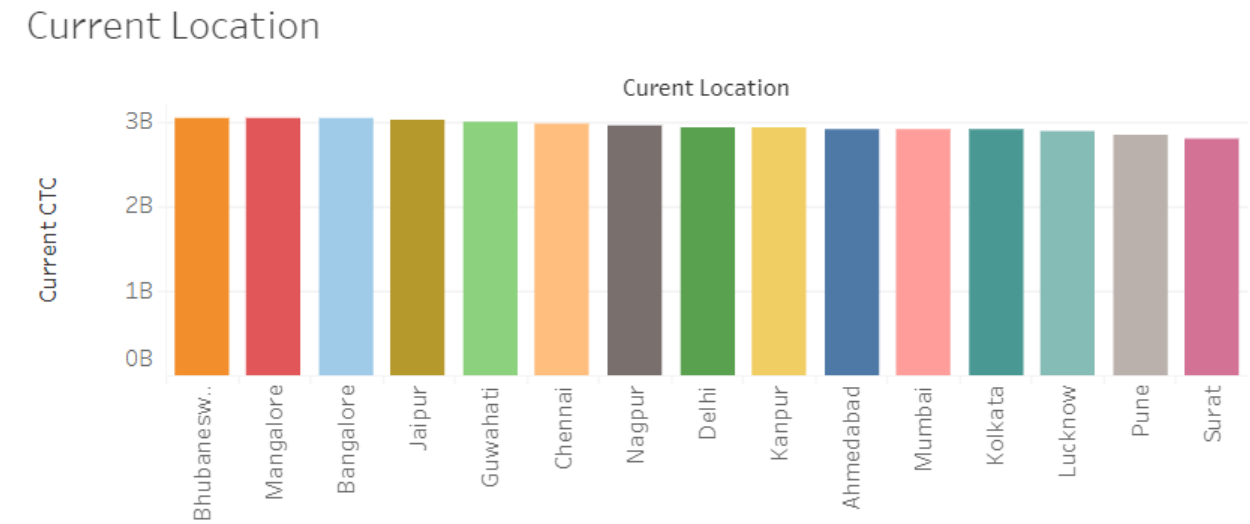


Figure 7: Current Location vs Current Salary

Bhubaneshwar and Mangalore candidates are getting higher salary than other location candidates.
Candidates from Surat and Pune are getting low salary when compared with other Location employees.

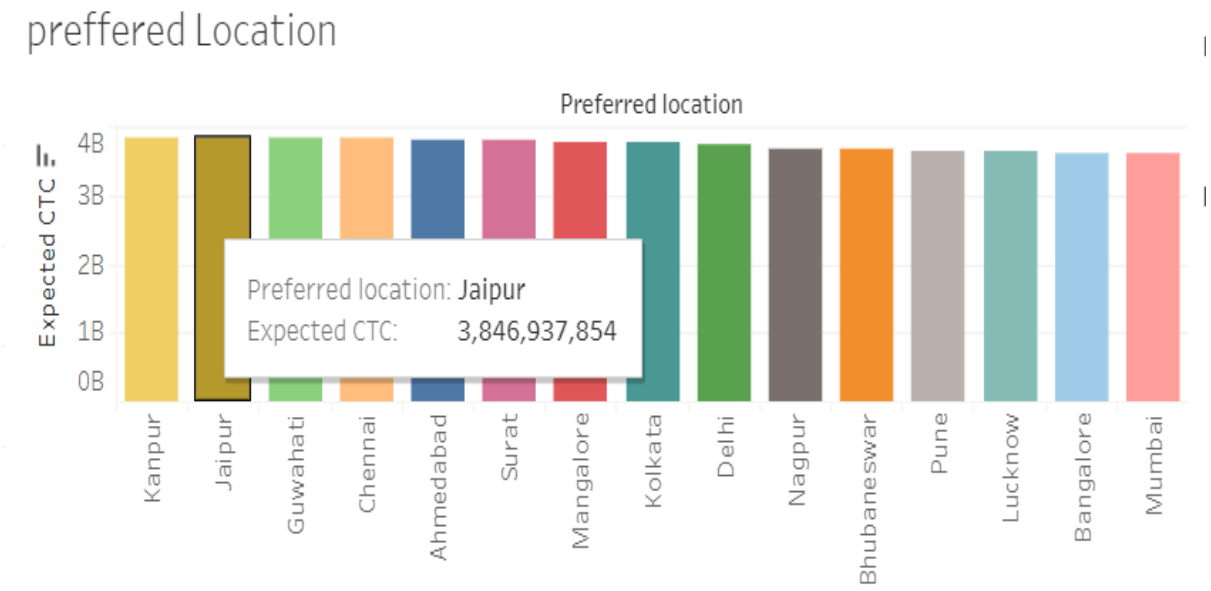


Figure 8: preferred location vs Expected Salary

Most of the candidates are preferred to work in Kanpur and Jaipur at the same time their salary expectations are also high.

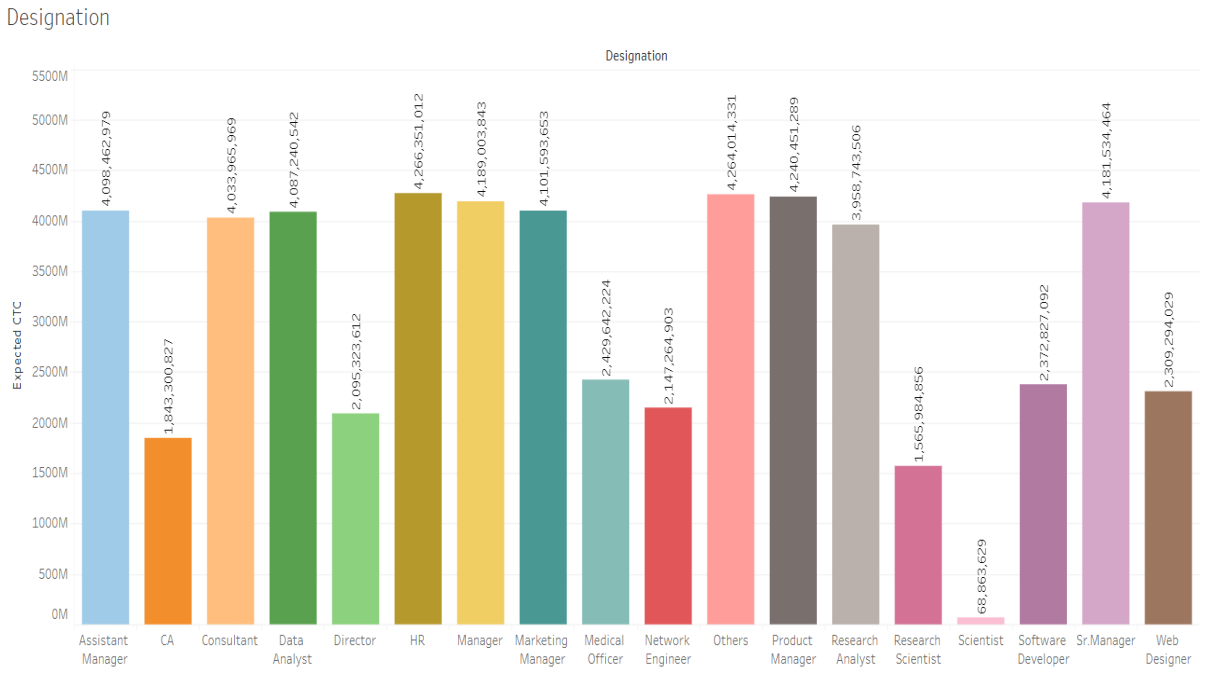


Figure 9: Designation vs Expected Salary

HR, Consultant, Data Analyst, Assistant Manages, Marketing Manager and Product Manager Professional are expecting higher in salary but same time Managers salary expectation is higher than the Sr, Managers.
Scientist's salary expectations is low when compared with other professionals.

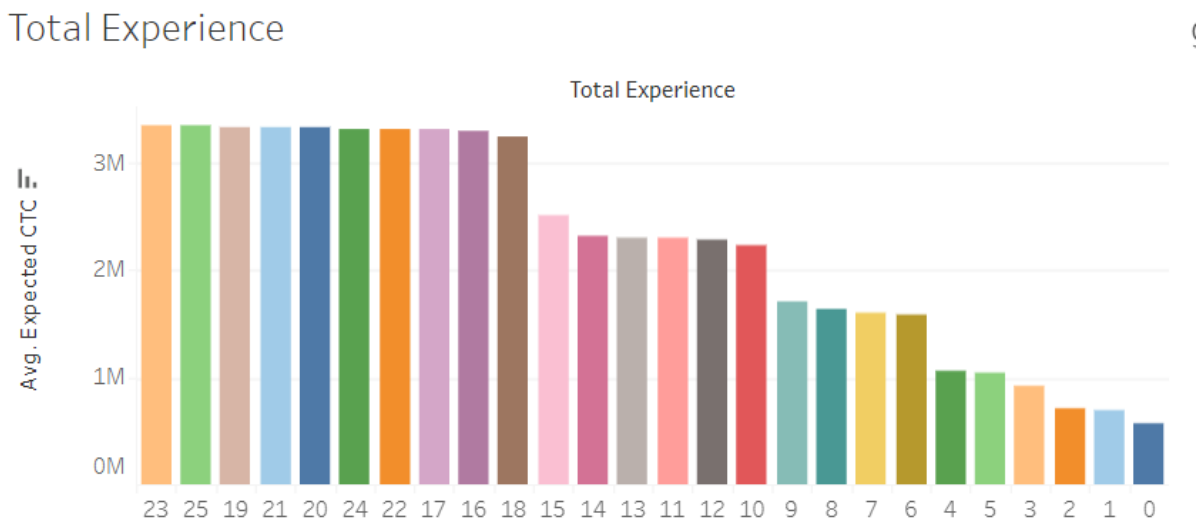


Figure 10: Total Experience vs Expected Salary

Candidates with 18 years of total experience are expecting higher salary.

Figure 11: Total Experience in the field vs Expected Salary

Candidates with 16 years of total experience in the field are expecting higher salary

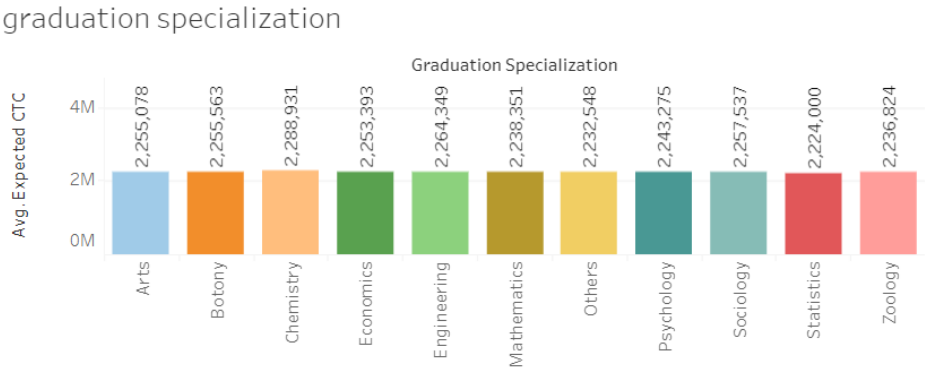


Figure 12: Graduation Specialization vs Expected Salary

2.2 Bivariate analysis (relationship between different variables, correlations)

Heat Map:-
A heat map contains values representing various shades of the same colour for each value to be plotted. Usually the darker shades of the chart represent higher values than the lighter shade. For a very different value a completely different colour can also be used.

Heat map (or heat map) is a data visualization technique that shows magnitude of a phenomenon as colour in two dimensions. The variation in colour may be by hue or intensity, giving obvious visual cues to the reader about how the phenomenon is clustered or varies over space.

Heat map represents these coefficients to visualize the strength of correlation among variables. It helps find features that are best for Machine Learning model building. The heat map transforms the correlation matrix into colour coding.

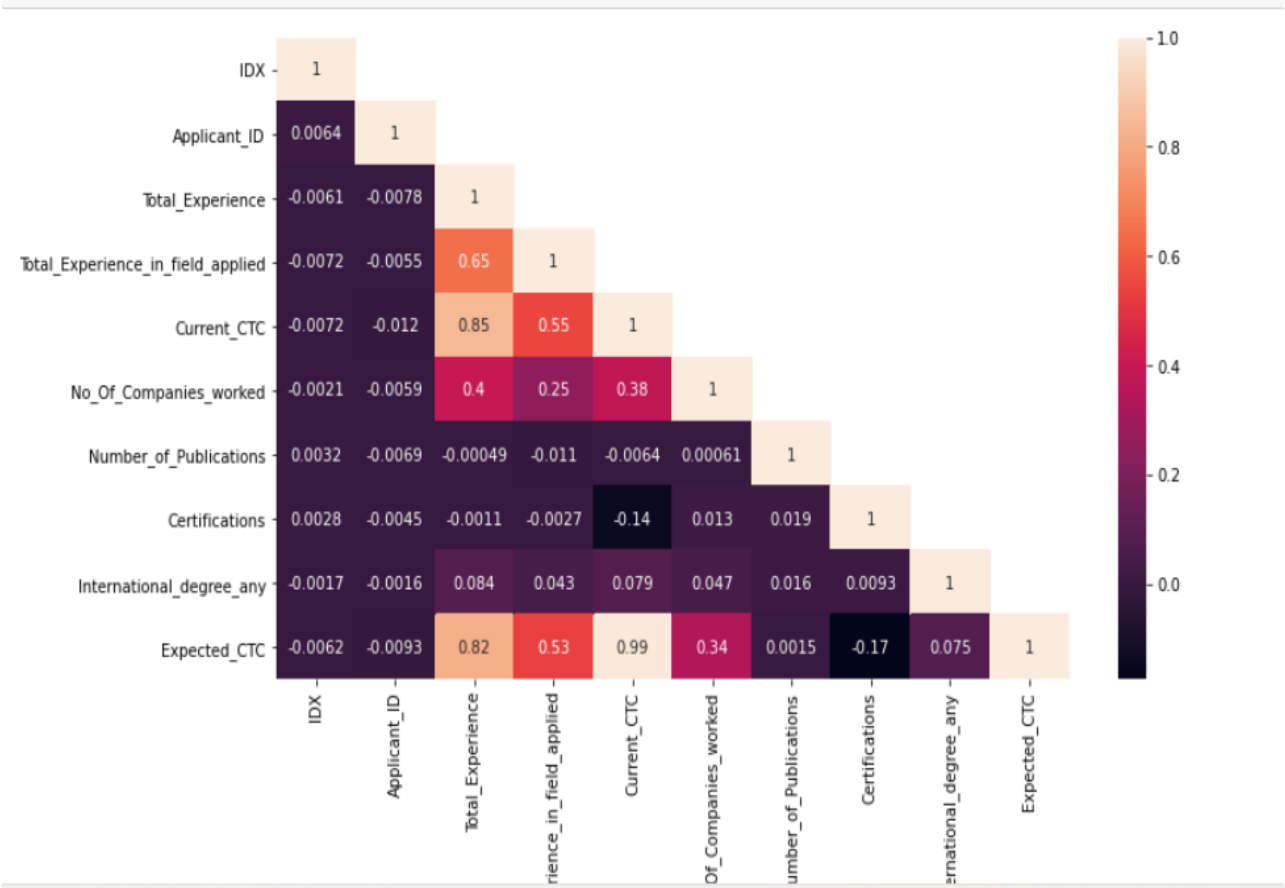


Figure 13: Heat map

Heat show co relation between the variables. Highly co-related values are shown in light colour and low co-related values are in dark colour.
Co-relation can be negative or positive. Positive co-relation which means its plays major role in target variables. If this variable increases target variable also increase. Here positively co-related values are **Current_CTC and Total_year_of_experience**. These variables are highly co-related with target variable.

PAIRPLOT:-

Pair plot is used to understand the best set of features to explain a relationship between two variables or to form the most separated clusters. It also helps to form some simple classification models by drawing some simple lines or make linear separation in our data-set.

A pairs plot is a matrix of scatterplots that lets you understand the pairwise relationship between different variables in a dataset. The easiest way to create a pairs plot in Python is to use the seaborn.

How do you interpret a pairwise scatter plot?

A scatter plot matrix shows all pairwise scatter plots for many variables. If the variables tend to increase and decrease together, the association is positive. If one variable tends to increase as the other decreases, the association is negative. If there is no pattern, the association is zero

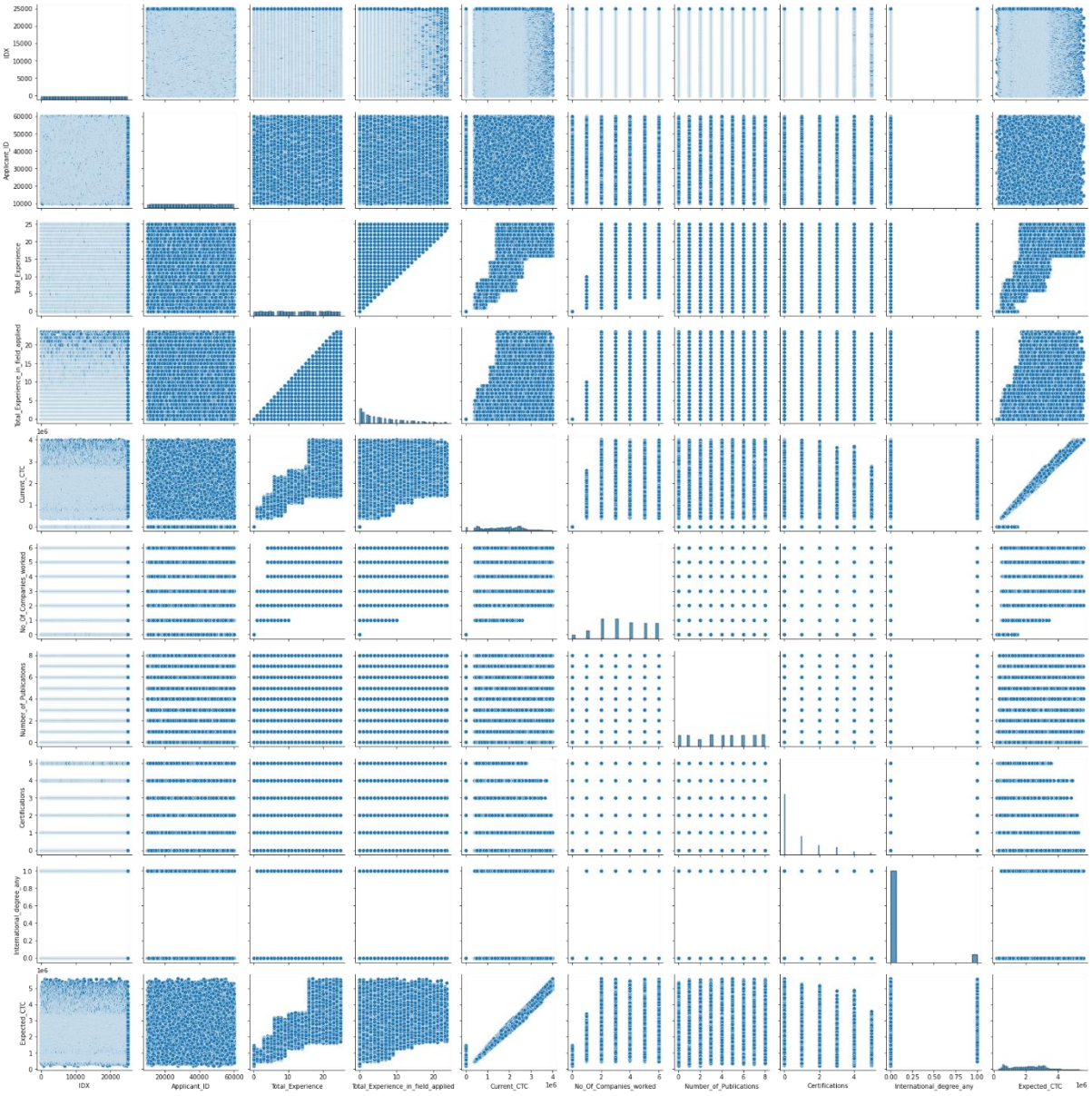


Figure 14: PairPlot

Removal of unwanted variables

```
PHD_Specialization           11881
University_PHD               11881
Passing_Year_Of_PHD          11881
PG_Specialization             7692
University_PG                 7692
Passing_Year_Of_PG            7692
Graduation_Specialization     6180
University_Grad               6180
Passing_Year_Of_Graduation    6180
Designation                   3129
Department                    2778
Role                           963
Industry                       908
Organization                   908
Last_Appraisal_Rating          908
dtype: int64
```

PHD_Specialization 11881

University_PHD 11881

Passing_Year_Of_PHD 11881

University_PG 7692

Passing_Year_Of_PG 7692

University_Grad 6180

Passing_Year_Of_Graduation 6180

All these features have high percentage of missing values and hence these features would be dropped from analysis. We are dropping them from our dataset to make sure that other valid observations do not get eliminated when we remove or impute the 'na' values.

3. Data Cleaning and Pre-processing

3.1. DATA PREPARATION

3.1.1. Variables Transformation

• For building the Multiple Linear Regression Model, all the variables were transformed using logarithm function.

3.1.2. Missing values and Outliers

• Missing value treatment was used by replacing with Simple Imputer.
• Expected_Salary, Current_Salary when Outliers were present, the record was imputed from outliers.
• Categorical fields are converted into numeric field by using replace () function.
Missing values are treated with Mode.

IDX	0
Applicant_ID	0
Total_Experience	0
Total_Experience_in_field_applied	0
Department	2778
Role	963
Industry	908
Organization	908
Designation	3129
Education	0
Graduation_Specialization	6180
PG_Specialization	7692
Curent_Location	0
Preferred_location	0
Current_CTC	0
Inhand_Offer	0
Last_Appraisal_Rating	908
No_Of_Companies_worked	0
Number_of_Publications	0
Certifications	0
International_degree_any	0
Expected_CTC	0

dtype: int64

IDX	0
Applicant_ID	0
Total_Experience	0
Total_Experience_in_field_applied	0
Department	0
Role	0
Industry	0
Organization	0
Designation	0
Education	0
Graduation_Specialization	0
PG_Specialization	0
Curent_Location	0
Preferred_location	0
Current_CTC	0
Inhand_Offer	0
Last_Appraisal_Rating	0
No_Of_Companies_worked	0
Number_of_Publications	0
Certifications	0
International_degree_any	0
Expected_CTC	0

dtype: int64

BOX PLOT:-

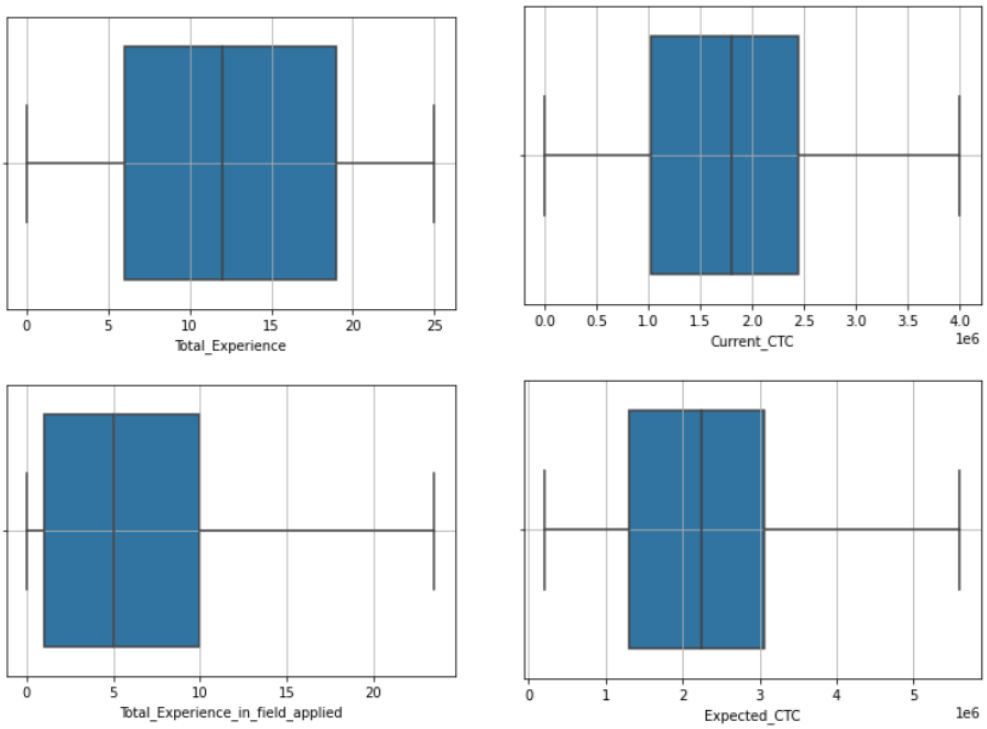


Figure 15: BoxPlot

Box plot shows that there is no any outliers present in numeric fields of dataset.

HISTOGRAM:-

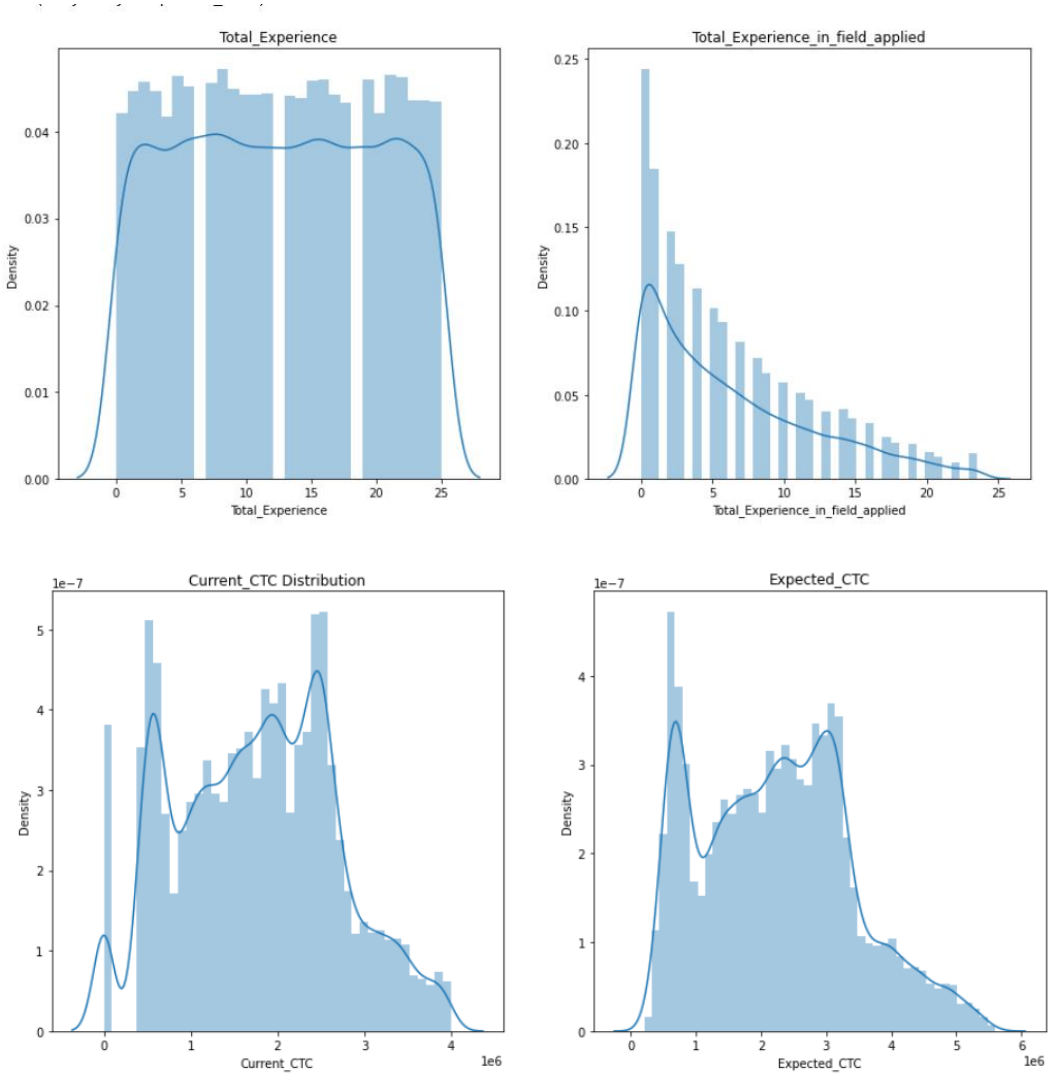


Figure 15: Histogram

Histogram shows that Numeric variables are not normally distributed.
For building the Multiple Linear Regression Model, all the variables were transformed using logarithm function.

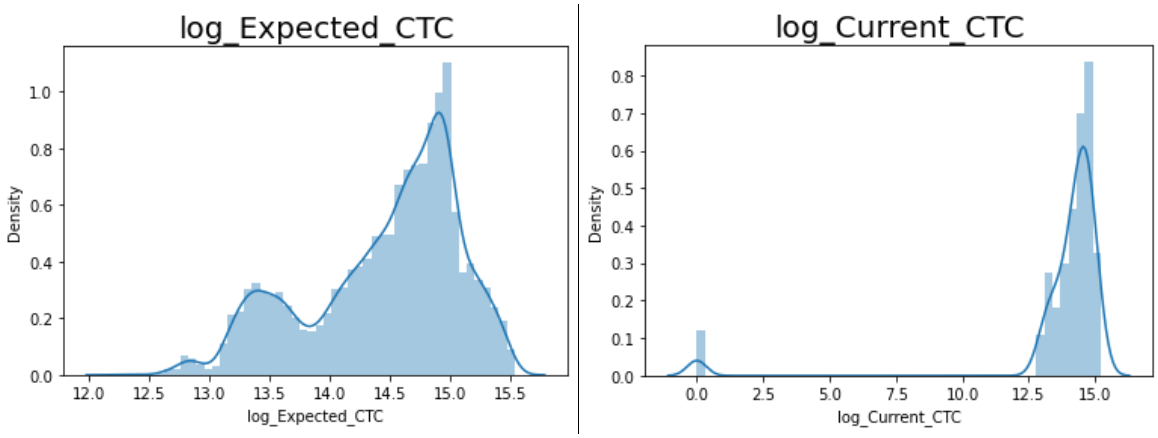


Figure 16: Histogram- After Log Transformation

4. Model building

Introduction about Supervised Machine Learning:-

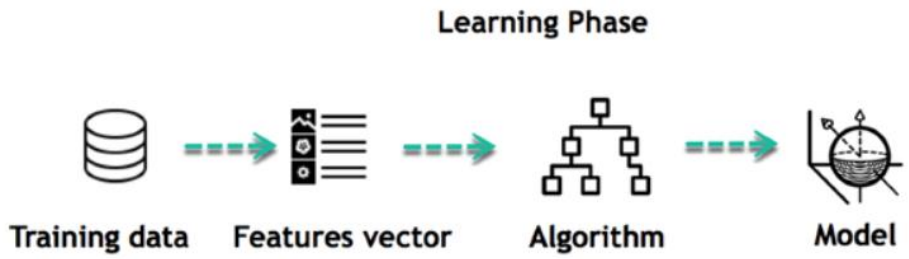
Supervised learning is a form of machine learning where an algorithm learns from examples of data. We progressively paint a picture of how supervised learning automatically generates a model that can make predictions about the real world. We also touch on how these models are tested, and difficulties that can arise in training them.

For example, you want to train a machine to help you predict how long it will take you to drive home from your workplace.

Here, you start by creating a set of labelled data. This data includes:

- Weather conditions
- Time of the day
- Holidays

All these details are your inputs in this supervised learning example. The output is the amount of time it took to drive back home on that specific day.



Working of Supervised Machine Learning

Types of Supervised Machine Learning Algorithms

Following are the types of Supervised Machine Learning algorithms:

Regression:

Regression technique predicts a single output value using training data.

Example: You can use regression to predict the house price from training data. The input variables will be locality, size of a house, etc.

Strengths: Outputs always have a probabilistic interpretation, and the algorithm can be regularized to avoid overfitting.

Weaknesses: Logistic regression may underperform when there are multiple or non-linear decision boundaries. This method is not flexible, so it does not capture more complex relationships.

Challenges in Supervised machine learning

Here, are challenges faced in supervised machine learning:

- Irrelevant input feature present training data could give inaccurate results
- Data preparation and pre-processing is always a challenge.
- Accuracy suffers when impossible, unlikely, and incomplete values have been inputted as training data

Advantages of Supervised Learning

Here are the advantages of Supervised Machine learning:

- Supervised learning in Machine Learning allows you to collect data or produce a data output from the previous experience
- Helps you to optimize performance criteria using experience
- Supervised machine learning helps you to solve various types of real-world computation problems.

Disadvantages of Supervised Learning

Below are the disadvantages of Supervised Machine learning:

- Decision boundary might be over trained if your training set which doesn't have examples that you want to have in a class
- You need to select lots of good examples from each class while you are training the classifier.
- Classifying can be a real challenge.
- Training for supervised learning needs a lot of computation time.

Best practices for Supervised Learning

- Before doing anything else, you need to decide what kind of data is to be used as a training set
- You need to decide the structure of the learned function and learning algorithm.
- Gathered corresponding outputs either from human experts or from measurements

Summary

- In Supervised learning algorithms, you train the machine using data which is well "labelled."
- You want to train a machine which helps you predict how long it will take you to drive home from your workplace is an example of Supervised learning.
- Regression and Classification are two dimensions of a Supervised Machine Learning algorithm.
- Supervised learning is a simpler method while unsupervised learning is a complex method.
- The biggest challenge in supervised learning is that irrelevant input feature present training data could give inaccurate results.
- The main advantage of supervised learning is that it allows you to collect data or produce a data output from the previous experience.
- The drawback of this model is that decision boundary might be overstrained if your training set doesn't have examples that you want to have in a class.

Build various models (You can choose to build models for either or all of descriptive, predictive or prescriptive purposes).

List models trained.

1. Linear Regression
2. Random Forest
3. Decision Tree
4. Support Vector Regression
5. AdaBoost Regressor
6. Gradient boost Regressor

4.2 Linear Regression:-

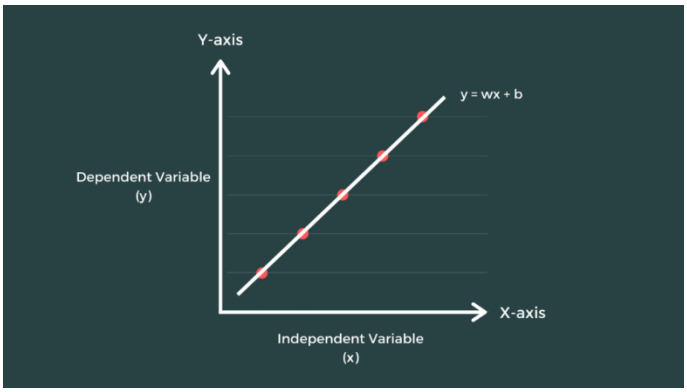
Simple Linear Regression is a linear regression algorithm used in datasets containing a single dependent variable and a single independent variable. It is also sometimes referred to as linear regression with single variable.

The relationship between the two variables is obtained by multiplying the independent variable by a constant value (known as weight) and then adding a constant (known as bias) to the product. Mathematically, the formula for linear regression can be represented as,

$$y=wx+b$$

Where,
y is the dependent variable,
x is the independent variable,
w is the weight,
b is the bias or the intercept, and
n is a positive integer.

If you are familiar with mathematics, the equation for linear regression is the same as the equation of a straight line, i.e., where represents the slope of the straight line and represents the -intercept of the line. So, a simple linear regression model is nothing but a straight line in two-dimensions**.**



The goal of simple linear regression model in Machine Learning is to find the value of weight and bias, , that can best predict the value of for a given value of . This is also known as **'fitting the model to the available data'**.

Train-Test Split data:-

Here we divided data into Train data 70% and Test data 30% and Random state 42.

X_train, X_test, y_train, y_test = train_test_split(X, y, test size=0.30, random state=42)

Co-efficient of variables:-

The coefficient for Total_Experience is 0.06017035168714096
The coefficient for Total_Experience_in_field_applied is 0.00012789686330569287
The coefficient for Department is 0.000853019807344984
The coefficient for Role is 0.0005190298029514842
The coefficient for Industry is 0.0001237596941968856
The coefficient for Organization is 1.87251567038362e-05
The coefficient for Designation is -0.0003763017414903215
The coefficient for Education is 0.10818187159331252
The coefficient for Graduation Specialization is 3.582855720610099e-05
The coefficient for PG_Specialization is -0.0001002158070836205
The coefficient for Curent_Location is 0.0008855589460580327
The coefficient for Preferred_location is -0.0005706598512819879
The coefficient for Inhand_Offer is -0.014694209718333448
The coefficient for Last_Appraisal_Rating is -0.029389293345265865
The coefficient for No_Of_Companies_worked is 0.012342562874181139
The coefficient for Number_of_Publications is 0.002556037010447743
The coefficient for Certifications is -0.042840838008713286
The coefficient for International_degree_any is 0.025993907844591352
The coefficient for log_Current_CTC is 0.04556058698024445

T**he intercept for our model:-**

Intercept	12.952141
Total_Experience	0.060170
Total_Experience_in_field_applied	0.000128
log_Current_CTC	0.045560
Education	0.108182
Graduation_Specialization	0.000036
Designation	-0.000376
Department	0.000853
Role	0.000519
Industry	0.000124
PG_Specialization	-0.000101
Curent_Location	0.000885
Preferred_location	-0.000571
Inhand_Offer	-0.014694
Last_Appraisal_Rating	-0.029390
No_Of_Companies_worked	0.012343
Number_of_Publications	0.002556
Certifications	-0.042841
International_degree_any	0.025992

OLS Regression Results

Dep. Variable:	log_Expected_CTC	R-squared:	0.795
Model:	OLS	Adj. R-squared:	0.795
Method:	Least Squares	F-statistic:	3765.
Date:	Sun, 18 Sep 2022	Prob (F-statistic):	0.00
Time:	18:11:40	Log-Likelihood:	-2961.8
No. Observations:	17500	AIC:	5962.
Df Residuals:	17481	BIC:	6109.
Df Model:	18		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	12.9521	0.019	675.572	0.000	12.915	12.990
Total_Experience	0.0602	0.000	142.108	0.000	0.059	0.061
Total_Experience_in_field_applied	0.0001	0.000	0.262	0.794	-0.001	0.001
log_Current_CTC	0.0456	0.001	47.680	0.000	0.044	0.047
Education	0.1082	0.002	51.694	0.000	0.104	0.112
Graduation_Specialization	3.588e-05	0.001	0.050	0.960	-0.001	0.001
Designation	-0.0004	0.000	-0.775	0.438	-0.001	0.001
Department	0.0009	0.001	1.288	0.198	-0.000	0.002
Role	0.0005	0.000	1.055	0.292	-0.000	0.001
Industry	0.0001	0.001	0.179	0.858	-0.001	0.001
PG_Specialization	-0.0001	0.001	-0.138	0.891	-0.002	0.001
Curent_Location	0.0009	0.001	1.762	0.078	-9.98e-05	0.002
Preferred_location	-0.0006	0.001	-1.137	0.256	-0.002	0.000
Inhand_Offer	-0.0147	0.005	-2.817	0.005	-0.025	-0.004
Last_Appraisal_Rating	-0.0294	0.002	-18.209	0.000	-0.033	-0.026
No_Of_Companies_worked	0.0123	0.001	8.328	0.000	0.009	0.015
Number_of_Publications	0.0026	0.001	2.953	0.003	0.001	0.004
Certifications	-0.0428	0.002	-21.901	0.000	-0.047	-0.039
International_degree_any	0.0260	0.009	3.013	0.003	0.009	0.043

Omnibus:	67.504	Durbin-Watson:	1.995
Prob(Omnibus):	0.000	Jarque-Bera (JB):	66.943
Skew:	-0.140	Prob(JB):	2.91e-15
Kurtosis:	2.885	Cond. No.	294.

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

Graph actual vs predicted.

X-axis: Actual and Y-axis: Predicted.

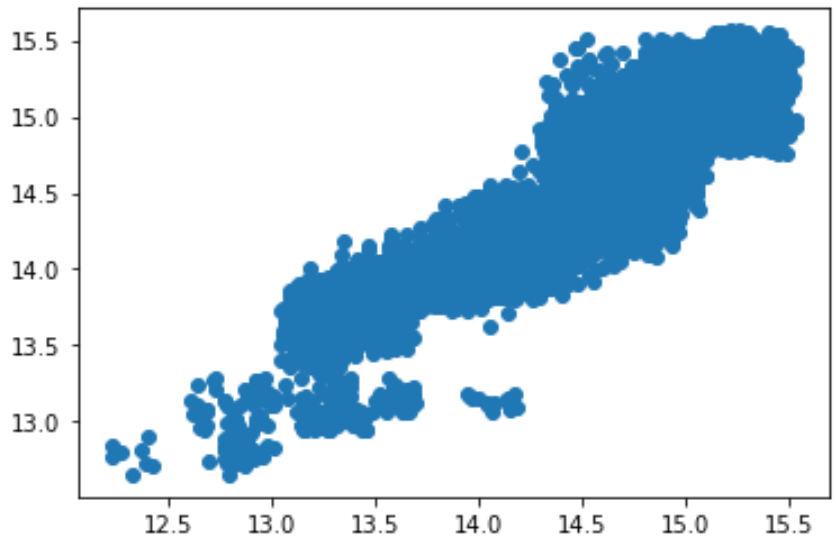


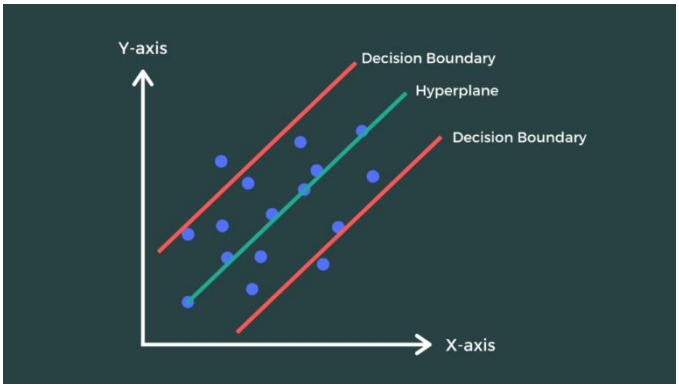
Figure 17: actual vs predicted.

Model	RMSE	R2(R.Square)
Linear Regression	0.283	0.794

4.3 Support Vector Regression:-

Support Vector Regression (SVR) is a supervised learning model that can be used to perform both linear and nonlinear regressions. In the previous lessons, we learned that the goal of applying linear regression is to minimize the error between the prediction and data. However, the goal of applying Support Vector Regression to a data set is to make sure that the errors do not exceed the threshold. In SVR, we fit as many instances as possible between the lines while limiting the margin violation. An SVR model uses the following hyper parameters in its model that determine the performance of the model.

- **Kernel**: The function used to map lower-dimensional data into higher dimensional data.
- **Hyper Plane**: The separation line between the data classes. For a Support Vector Regression problem, a hyperplane is a line that will help us predict the continuous value or target value.
- **Decision Boundary line**: The boundary lines are essentially the decision boundaries of the hyperplane. The support vectors can be on the Boundary lines or outside it. The best fit line is determined on the basis of the hyperplane having the maximum number of points inside its boundary line.
- **Support Vectors** are the data points that are closest to the decision boundary. The distance of the points is minimum or least.



Train-Test Split data:-

Here we divided data into Train data 70% and Test data 30% and Random state 42.

X_train, X_test, y_train, y_test = train_test_split(X, y, test size=0.30, random state=42)

Fitting the model:-

```
model = SVR(kernel = 'rbf')
# Fitting the SVR model to the data
model.fit(X_train, y_train)
```

```
SVR()
```

Let us now calculate the loss between the actual target values in the testing set and the values predicted by the model with the use of a cost function called the Root Mean Square Error (RMSE).

$$RMSE = \sqrt{(\frac{1}{n})\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

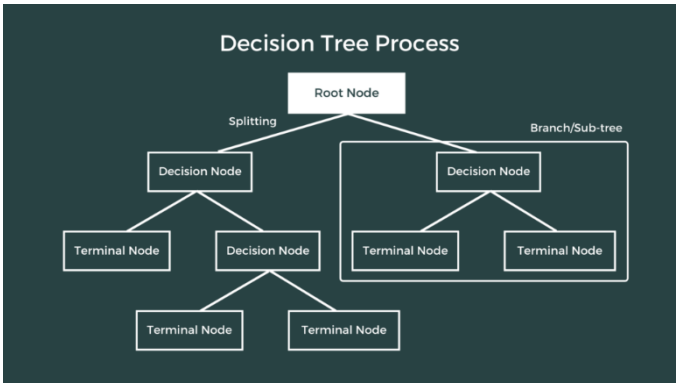
where,
y_i is the actual target value,
$\hat{y}_i$ is the predicted target value, and
n is the total number of data points.

The RMSE of a model determines the absolute fit of the model to the data. In other words, it indicates how close the actual data points are to the model's predicted values. A low value of RMSE indicates a better fit and is a good measure for determining the accuracy of the model's predictions.

Model	RMSE	R2(R.Square)
Support Vector Regression	0.102	0.976

4.4 Decision Tree Regressor:-

A decision tree is one of the most frequently used Machine Learning algorithms for solving **regression** as well as **classification** problems. As the name suggests, the algorithm uses a tree-like model of decisions to either predict the target value (regression) or predict the target class (classification). Before diving into how decision trees work, first, let us be familiar with the basic terminologies of a decision tree:



- **Root Node:** This represents the topmost node of the tree that represents the whole data points.
- **Splitting:** It refers to dividing a node into two or more sub-nodes.
- **Decision Node:** They are the nodes that are further split into sub-nodes, i.e., this node that is split is called a decision node.
- **Leaf / Terminal Node:** Nodes that do not split are called Leaf or Terminal nodes. These nodes are often the final result of the tree.
- **Branch / Sub-Tree:** A subsection of the entire tree is called branch or sub-tree.
- **Parent and Child Node:** A node, which is divided into sub-nodes is called a parent node of sub-nodes whereas sub-nodes are the child of the parent node. In the figure above, the decision node is the parent of the terminal nodes (child).
- **Pruning:** Removing sub-nodes of a decision node is called pruning. Pruning is often done in decision trees to prevent overfitting.

How Decision Tree Works:-

The process of splitting starts at the root node and is followed by a branched tree that finally leads to a leaf node (terminal node) that contains the prediction or the final outcome of the algorithm. Construction of decision trees usually works top-down, by choosing a variable at each step that best splits the set of items. Each sub-tree of the decision tree model can be represented as a binary tree where a decision node splits into two nodes based on the conditions.

Decision trees where the target variable or the terminal node can take continuous values (typically real numbers) are called **regression trees** which will be discussed in this lesson. If the target variable can take a discrete set of values these trees are called **classification trees.**

Train-Test Split data:-

Here we divided data into Train data 70% and Test data 30% and Random state 42.

X_train, X_test, y_train, y_test = train_test_split(X, y, test size=0.30, random state=42)

Fitting the model:-

```
from sklearn.tree import DecisionTreeRegressor

# create a regressor object
regressor = DecisionTreeRegressor(random_state = 42)

# fit the regressor with X and Y data
regressor.fit(X_train, y_train)
```

```
DecisionTreeRegressor(random_state=42)
```

Graph actual vs predicted.

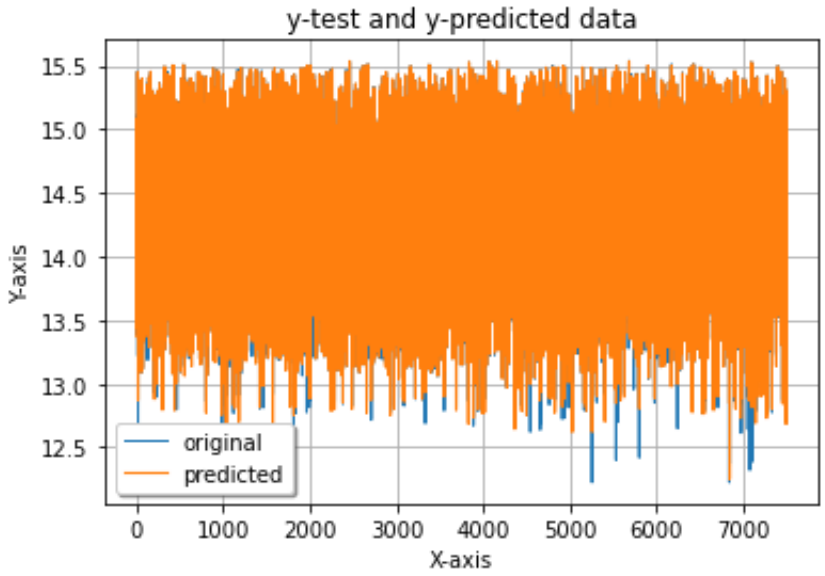


Figure 18: actual vs predicted.

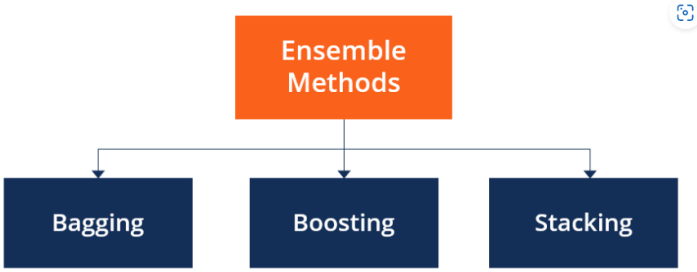
Model	RMSE	R2(R.Square)
Decision Tree Regressor	0.040	1.000

Model Tuning and business implication:-

Tuning is the process of maximizing a model's performance without overfitting or creating too high of a variance.

4.5 Ensemble modelling, wherever applicable:-

Ensemble methods are techniques that aim at improving the accuracy of results in models by combining multiple models instead of using a single model. The combined models increase the accuracy of the results significantly. This has boosted the popularity of ensemble methods in machine learning.



- Ensemble methods aim at improving predictability in models by combining several models to make one very reliable model.
- The most popular ensemble methods are boosting, bagging, and stacking.
- Ensemble methods are ideal for regression and classification, where they reduce bias and variance to boost the accuracy of models.

Bagging:-

Bagging, the short form for bootstrap aggregating, is mainly applied in classification and regression. It increases the accuracy of models through decision trees, which reduces variance to a large extent. The reduction of variance increases accuracy, eliminating overfitting, which is a challenge to many predictive models.

Boosting

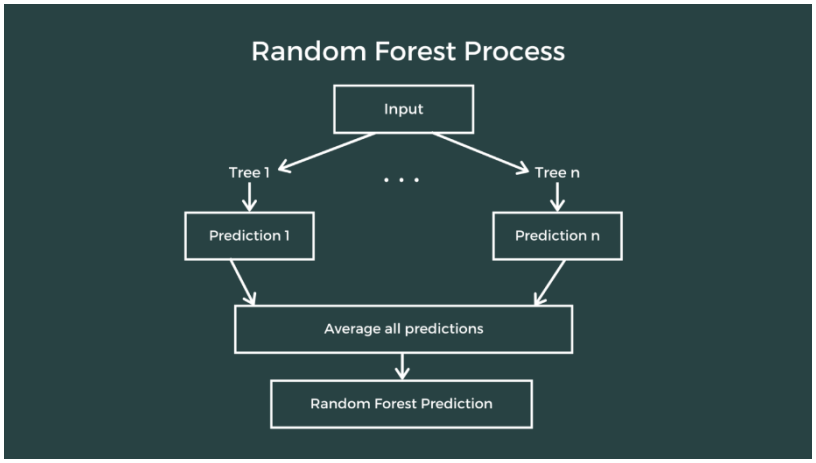
Boosting is an ensemble technique that learns from previous predictor mistakes to make better predictions in the future. The technique combines several weak base learners to form one strong learner, thus significantly improving the predictability of models. Boosting works by arranging weak learners in a sequence, such that weak learners learn from the next learner in the sequence to create better predictive models.

Boosting takes many forms, including gradient boosting, Adaptive Boosting (AdaBoost), and XGBoost (Extreme Gradient Boosting). AdaBoost uses weak learners in the form of decision trees, which mostly include one split that is popularly known as decision stumps. AdaBoost's main decision stump comprises observations carrying similar weights.

Random Forest Regressor:-

Random Forest Regression algorithms are a class of Machine Learning algorithms that use the combination of multiple random decision trees each trained on a subset of data. The use of multiple trees gives stability to the algorithm and reduces variance. The random forest regression algorithm is a commonly used model due to its ability to work well for large and most kinds of data.

Random Forest Regression algorithms are a class of Machine Learning algorithms that use the combination of multiple random decision trees each trained on a subset of data. The use of multiple trees gives stability to the algorithm and reduces variance. The random forest regression algorithm is a commonly used model due to its ability to work well for large and most kinds of data.



Train-Test Split data:-

Here we divided data into Train data 70% and Test data 30% and Random state 42.

X_train, X_test, y_train, y_test = train_test_split(X, y, test size=0.30, random state=42)

Fitting the model:-

```
regressor = RandomForestRegressor(n_estimators = 100, random_state = 42)

# fit the regressor with x and y data
regressor.fit(X_train, y_train)
```

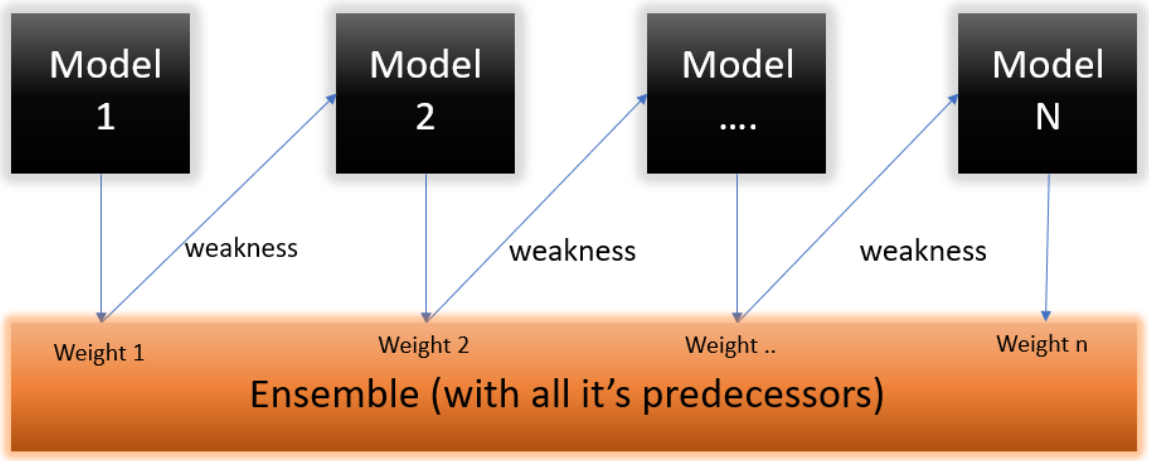
```
RandomForestRegressor(random_state=42)
```

Model Evaluation:-

Model	RMSE	R2(R.Square)
Random Forest Regressor	0.033	0.999

AdaBoost Regressor:-

AdaBoost also called Adaptive Boosting is a technique in Machine Learning used as an Ensemble Method. The most common algorithm used with AdaBoost is decision trees with one level that means with Decision trees with only 1 split. These trees are also called **Decision Stumps.**



Train-Test Split data:-

Here we divided data into Train data 70% and Test data 30% and Random state 42.

X_train, X_test, y_train, y_test = train_test_split(X, y, test size=0.30, random state=42)

Fitting the model:-

```
ada_reg = AdaBoostRegressor(n_estimators=100)
print(ada_reg)
```

```
AdaBoostRegressor(n_estimators=100)
```

```
ada_reg.fit(X_train, y_train)
```

```
AdaBoostRegressor(n_estimators=100)
```

Cross-Validation:-

```
scores = cross_val_score(ada_reg, X_train,y_train,cv=5)
print("Mean cross-validataion score: %.2f" % scores.mean())
```

```
Mean cross-validataion score: 0.97
```

Why should we use k-fold cross-validation?

K-Fold Cross-Validation:

It ensures that the score of our model does not depend on the way we select our train and test subsets. In this approach, we divide the data set into k number of subsets and the holdout method is repeated k number of times

```
kfold = KFold(n_splits=10, shuffle=True)
kf_cv_scores = cross_val_score(ada_reg, X_train, y_train, cv=kfold )
print("K-fold CV average score: %.2f" % kf_cv_scores.mean())
```

K-fold CV average score: 0.96

Graph actual vs predicted.

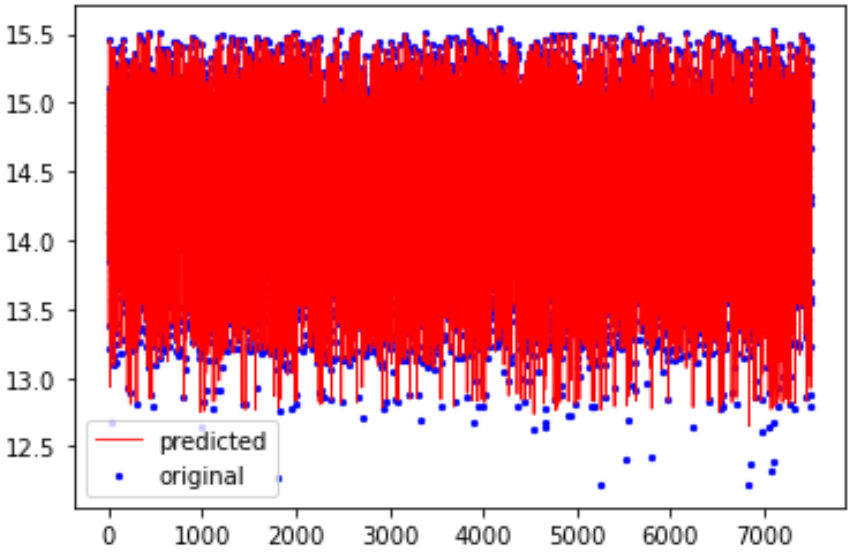


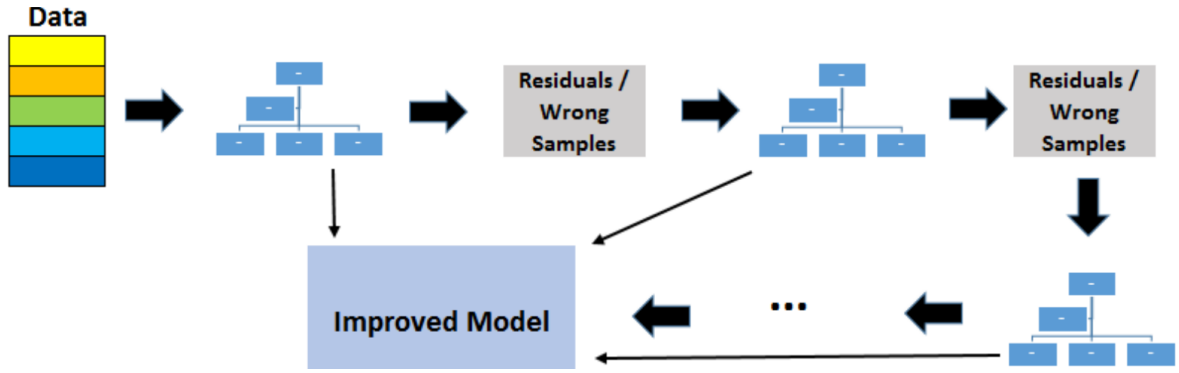
Figure 19: actual vs predicted.

Model Evaluation:-

Model	RMSE	R2(R.Square)
AdaBoost Regressor	0.120	0.963

Gradient Boosting Regressor:-

Gradient boosting works by building simpler (weak) prediction models sequentially where each model tries to predict the error left over by the previous model. Because of this, the algorithm tends to over fit rather quick. But, what is a weak learning model? A model that does slightly better than random predictions



Train-Test Split data:-

Here we divided data into Train data 70% and Test data 30% and Random state 42.

X_train, X_test, y_train, y_test = train_test_split(X, y, test size=0.30, random state=42)

Fitting the model:-

```
model = GradientBoostingRegressor()
```

```
model.fit(X_train, y_train)
```

```
GradientBoostingRegressor()
```

Model Evaluation:-

Model	RMSE	R2(R.Square)
Gradient Boosting Regressor	0.04	0.994

5. Model validation

Test your predictive model against the test set using various appropriate performance metrics.

A regression problem is about predicting a quantity. A simple example of a regression problem is prediction of the selling price of a real estate property based on its attributes (location, square meters available, condition, etc.).

To evaluate how good your regression model is, you can use the following metrics:

- **R-squared:** indicate how many variables compared to the total variables the model predicted. R-squared does not take into consideration any biases that might be present in the data. Therefore, a good model might have a low R-squared value, or a model that does not fit the data might have a high R-squared value.
- **Average error**: the numerical difference between the predicted value and the actual value.
- **Mean Square Error (MSE):** good to use if you have a lot of outliers in the data.
- **Median error**: the average of all difference between the predicted and the actual values.
- **Average absolute error**: similar to the average error, only you use the absolute value of the difference to balance out the outliers in the data.
- **Median absolute error:** represents the average of the absolute differences between prediction and actual observation. All individual
- l differences have equal weight, and big outliers can therefore affect the final evaluation of the model.

Model	RMSE	R2(R.Square)
Linear Regression	0.283	0.794
Support Vector Regression	0.102	0.976
Decision Tree Regressor	0.040	1.000
AdaBoost Regressor	0.120	0.963
Gradient Boosting Regressor	0.04	0.994

6. Final interpretation / recommendation:-

Decision Tree Regressor model we can see that the RMSE value for this model is lesser than the previous model. We can also see that the R^2 value for this model is higher than the previous model. This tells us that this model fits the data worse than the previous model.

- Both RMSE and R^2 quantify how well a regression model fits a dataset.
- The RMSE tells us how well a regression model can predict the value of the response variable in absolute terms while R^2 tells us how well a model can predict the value of the response variable in percentage terms.
- It's useful to calculate both the RMSE and R^2 for a given model because each metric gives us useful information.

In this case Decision tree repressor model is the best model to fit for this dataset.

Gradient Boosting Regressor model we can see that the RMSE value for this model is lesser than the previous model. We can also see that the R^2 value for this model is higher than the previous model. This tells us that this model fits the data worse than the AdaBoost Regressor model.

Gradient Boosting Regressor model is the best optimum model.

Recommendations:-

- 1. Total years of experience has major co-relation with salary, So Expected salary calculated based on Total years of experience.
- Most of the candidates not carrying In hand offer.
- Scientist's salary expectations is low when compared with other professionals.
- Current_CTC and Total years of experience is the parameter which exclaims most of the variation in expected alary, there is very less tuning needed.
- Candidates preferred Bhubaneshwar and Mangalore when compared with other metro cities.
- HR, Consultant, Data Analyst, Assistant Manages, Marketing Manager and Product Manager Professional are expecting higher in salary but same time Managers salary expectation is higher than the Sr, Managers.

----------END OF THE REPORT----------