A Project Report

on

Salary Prediction Using Machine Learning

Submitted in partial fulfillment of the

requirement for the award of the degree of

Bachelor of Technology in Computer Science and Engineering

GALGOTIAS UNIVERSITY

(Established under Galgotias University Uttar Pradesh Act No. 14 of 2011)

Under The Supervision of
Dr. Ravindra Kumar Chahar
Associate Professor
Department of Computer Science and Engineering

Submitted By

18SCSE1180064 – MOHIT AHUJA

18SCSE1180021 – HARSH PATHAK

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

GALGOTIAS UNIVERSITY, GREATER NOIDA, INDIA

DECEMBER - 2021

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING
GALGOTIAS UNIVERSITY, GREATER NOIDA

CANDIDATE'S DECLARATION

We hereby certify that the work which is being presented in the project, entitled **" Salary Prediction Using Machine Learning"** in partial fulfillment of the requirements for the award of the **BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE AND ENGINEERING** submitted in the **School of Computing Science and Engineering** of Galgotias University, Greater Noida, is an original work carried out during the period of **JULY-2021 to DECEMBER-2021**, under the supervision of **Dr. Ravindra Kumar Chahar, Associate Professor**, **Department of Computer Scienceand Engineering** of School of Computing Science and Engineering , Galgotias University, Greater Noida

The matter presented in the project has not been submitted by us for the award of any other degree of this or any other places.

18SCSE1180064 – MOHIT AHUJA
18SCSE1180021 – HARSH PATHAK

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Supervisor

(Dr. Ravindra Kumar Chahar, Associate Professor)

CERTIFICATE

The Final Project Viva-Voce examination of **18SCSE1180064 – MOHIT AHUJA, 18SCSE1180021 – HARSH PATHAK** has been held on _____ and their work is recommended for the award of **BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE AND ENGINEERING**.

Signature of Examiner(s) **Signature of Supervisor(s)**

Signature of Project Coordinator **Signature of Dean**

Date:

Place:

Abstract

Machine learning is a technology which allows a software program to became more accurate at pretending more accurate results without being explicitly programmed and also ML algorithms uses historic data to predicts the new outputs. Because of this ML gets a distinguish attention. Now a day's prediction engine has become so popular that they are generating accurate and affordable predictions just like a human, and being using industry to solve many of the problems. Predicting justified salary for employee is always being a challenging job for an employer. In this project, a salary prediction model is made with suitable algorithm using key features required to predict the salary of employee.

The main aim of the project is to predict the salary of graduates and make a suitable user-friendly graph. From this prediction the salary of an employee can be observed according to a particular field according to their qualifications. It helps to see the growth of any field.

In the project, we have used Linear Regression as an algorithm for prediction. Linear regression carries out a task that may predict the value of a dependent variable (y) on basis of an independent variable (x) that is given. Therefore, this kind of regression technique looks for a linear type of relationship between input x and output y. Apart from Linear Regression, other types of regression techniques are also used like the Decision Tree Regressor and Random Forest Regressor.

Since nothing in this universe can be termed as "perfect", thus a lot of features can be added to make the system more widely acceptable and more user friendly. This will not only help to predict salaries of other fields but also will be more user beneficial.

In the upcoming phase of our project we will be able to connect an even larger dataset to this model so that the training can be even better. This model should check for new data, once in a month, and incorporate them to expand the dataset and produce better results.

Table of Contents

List of Figures

ACRONYMS

I/P & O/P	Input and Output
ML	Machine Learning
ANN	Artificial Neural Network
LR	Linear Regression
DC	Decision Trees
CV	Computer Vision
NLP	Natural Language Processing

Terminology Used

- An Algorithm is a set of rules that a machine follows to achieve a particular goal. An algorithm can be considered as a recipe that defines the inputs, the output and all the steps needed to get from the inputs to the output.
- Machine Learning is a set of methods that allow computers to learn from data to make and improve predictions.
- A Machine Learning Model is the learned program that maps inputs to predictions. This can be a set of weights for a linear model or for a neural network.
- A Dataset is a table with the data from which the machine learns. The dataset contains the features and the target to predict. When used to induce a model, the dataset is called training data.
- The Prediction is what the machine learning model "guesses" what the target value should be based on the given features.

CHAPTER 1

INTRODUCTION

Now days, one of the major reasons an employee switches a company is the salary of the employee. Employees keep switching the company to get the expected salary. And it results in loss for the company and to overcome this loss we came with an idea what if the employee gets the desired/expected salary from the Company or Organization. In this Competitive world everyone has a higher expectation and goals.

But we cannot randomly provide everyone their expected salary there should be a system which should measure the ability of the Employee for the Expected salary. We cannot decide the exact salary but we can predict it by using certain data sets.

A prediction is an assumption about a future event. A prediction is sometimes, though not always, is based upon knowledge or experience. Future events are not necessarily certain, thus confirmed exact data about the future is in many cases are impossible, a prediction may be useful to help in preparing plans about probable developments. In this project, the salary of an employee of an organization is to be predicted on basis of past experience and the educational qualifications of the individual. Here the history of salary has been observed and then on basis of that salary of a person after a certain period of time it can be calculated automatically.

In order to gain useful insights into the job recruitment, we compare different strategies and machine learning models. The methodology different phases like: Data collection, Data cleaning, Manual feature engineering, Data set description, Automatic feature selection, Model selection, Model training and validation, Model comparison.

The process of learning begins with observations or data, such as examples, direct experience, or instruction, in order to look for patterns in data and make better decisions in the future based on the examples that we provide. The primary aim is to allow the computers learn automatically without human intervention or assistance and adjust actions accordingly.

Machine learning algorithms are broadly classified into three divisions, namely; Supervised learning, Unsupervised learning and Reinforcement learning.

- **Supervised learning:-** Supervised learning is a learning in which we teach or train the machine using data which is well labelled that means some data is already tagged with correct answer. After that, machine is provided with new set of examples so that supervised learning algorithm analyses the training data and produces a correct outcome from labelled data.

  Basically, they can apply what has been learned in the past to new data using labelled examples to predict future events. Starting from the analysis of a known training dataset, the learning algorithm produces an inferred function to make predictions about the output values. The system is able to provide targets for any new input after sufficient training. The learning algorithm can also compare its output with the correct, intended output and find errors in order to modify the model accordingly.

- **Unsupervised Learning:**- In contrast, unsupervised machine learning algorithms are used when the information used to train is neither classified nor labelled. Unsupervised learning studies how systems can infer a function to describe a hidden structure from unlabelled data. The system doesn't figure out the right output, but it explores the data and can draw inferences from datasets to describe hidden structures from unlabelled data. Unsupervised learning is the training of machine using information that is neither classified nor labelled and allowing the algorithm to act on that information without guidance. Here the task of machine is to group unsorted information according to similarities, patterns and differences without any prior training of data. Unlike, supervised learning, no teacher is provided that means no training will be given to the machine. Therefore, machine is restricted to find the hidden structure in unlabelled data by our-self.

- **Reinforcement learning:-** Reinforcement learning is an area of Machine Learning. Reinforcement. It is about taking suitable action to maximize reward in a particular

situation. It is employed by various software and machines to find the best possible behaviour or path it should take in a specific situation. Reinforcement learning differs from the supervised learning in a way that in supervised learning the training data has the answer key with it so the model is trained with the correct answer itself whereas in reinforcement learning, there is no answer but the reinforcement agent decides what to do to perform the given task. In the absence of training dataset, it is bound to learn from its experience.

Reinforcement machine learning algorithms is a learning method that interacts with its environment by producing actions and discovers errors or rewards. Trial and error search and delayed reward are the most relevant characteristics of reinforcement learning. This method allows machines and software agents to automatically determine the ideal behavior within a specific context in order to maximize its performance. Simple reward feedback is required for the agent to learn which action is best; this is known as the reinforcement signal.

The project uses various regression techniques for predicting the salary of the employees. The techniques are listed as follows.

1. **Linear Regression:** In Linear regression we are given a number of predictor variables and a continuous response variable, and we try to find a relationship between those variables that allows us to predict a continuous outcome.
2. For example, given X and Y, we fit a straight line that minimize the distance using methods to estimate the coefficients like Ordinary Least Squares and Gradient Descent between the sample points and the fitted line.
3. **Decision Tree Regressor:** Decision tree builds regression or classification models in the form of a tree structure. It breaks down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. The final result is a tree with decision nodes and leaf nodes. A decision node has two or more branches, each representing values for the attribute tested. Leaf node represents a decision on the numerical target. The topmost decision node in a tree which corresponds to the best predictor called root node. Decision trees can handle both categorical and numerical data.

4. **Random Forest Regressor:** Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time For regression tasks, the mean or average prediction of the individual trees is returned.

REQUIRED TOOLS

- Python: Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together.
- Jupyter Notebook: The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text. Uses include: data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much more.
- Anaconda Navigator: Anaconda Navigator is a desktop graphical user interface (GUI) included in Anaconda® distribution that allows you to launch applications and easily manage conda packages, environments, and channels without using command-line commands.

CHAPTER 2
LITERATURE SURVEY

1) Susmita Ray," A Quick Review of Machine Learning Algorithms," 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (Com-ITCon), India, 14th -16th Feb 2019 a brief review of various machine learning algorithms which are most frequently used to solve classification, regression and clustering problems. The advantages, disadvantages of these algorithms have been discussed along with comparison of different algorithms (wherever possible) in terms of performance, learning rate etc. Along with that, examples of practical applications of these algorithms have been discussed.

2) Sananda Dutta, Airiddha Halder, Kousik Dasgupta," Design of a novel Prediction Engine for predicting suitable salary for a job" 2018 Fourth International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN) - focused on the problem of predicting salary for job advertisements in which salary are not mentioned and also tried to help fresher to predict possible salary for different companies in different locations. The corner stone of this study is a dataset provided by ADZUNA. model is well capable to predict precise value.

3) Pornthep Khongchai, Pokpong Songmuang, "Improving Students' Motivation to Study using Salary Prediction System" - proposed prediction model using Decision tree technique with seven features. Moreover, the result of the system is not only a predicted salary, but also the 3-highest salary of the graduated students which share common attributes to the users. To test the system's efficiency, they set up an experiment by using 13,541 records of actual graduated student data. The total result in accuracy is 41.39%.

4) Phuwadol Viroonluecha, Thongchai Kaewkiriya," Salary Predictor System for Thailand Labour Workforce using Deep Learning" - used Deep learning techniques to construct a model which predicts the monthly salary of job seekers in Thailand solving a regression problem which is a numerical outcome is effective. We used five-month personal profile data from wellknown job search website for the analysis. As a result, Deep learning model has strong performance whether accuracy or process time by RMSE 0.774 x 104 and only 17 seconds for runtime.

MERITS OF THE PROPOSED SYSTEM

1. Easily identifies trends and patterns: Machine Learning Models can review large volumes of data and discover specific trends and patterns that would not be apparent to humans. For instance, for an e-commerce website like Amazon, it serves to understand the browsing behaviours and purchase histories of its users to help cater to the right products, deals, and reminders relevant to them. It uses the results to reveal relevant advertisements to them.

2. No human intervention needed (automation): With implementation of ML model, there is no need to have any eye on the project at every step of the way. Since, giving machines the ability to learn, lets them make predictions and also improve the algorithms on their own. A common example of this is anti-virus softwares; they learn to filter new threats as they are recognized. ML is also good at recognizing spam.

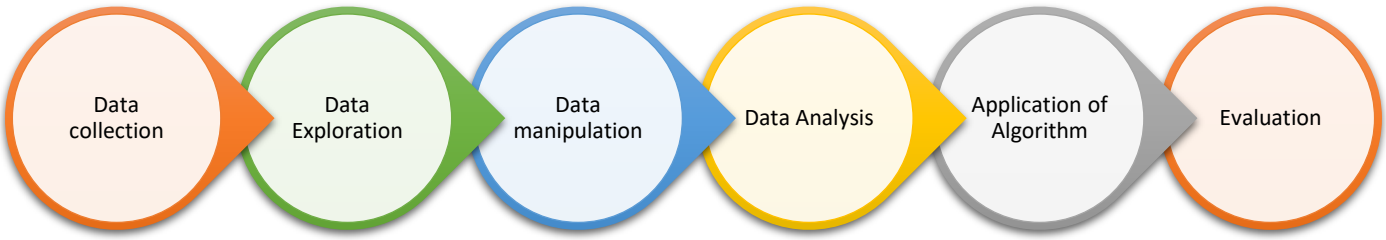
3. Continuous Improvement: As ML algorithms gain experience, they keep improving in accuracy and efficiency. This lets them make better decisions.

4. Handling multi-dimensional and multi-variety data: Machine Learning algorithms are good at handling data that are multi-dimensional and multi-variety, and they can do this in dynamic or uncertain environments.

5. Wide Applications: You could be an e-tailer or a healthcare provider and make ML work for you. Where it does apply, it holds the capability to help deliver a much more personal experience to customers while also targeting the right customers.

CHAPTER 3

ARCHITECTURAL FLOW OF THE PROPOSED SYSTEM



An Architectural Diagram or a pipeline is used to help automate machine learning workflows. They operate by enabling a sequence of data to be transformed and correlated together in a model that can be tested and evaluated to achieve an outcome, whether positive or negative.

The pipeline/ Diagram consists of several steps to train a model. Machine learning pipelines are iterative as every step is repeated to continuously improve the accuracy of the model and achieve a successful algorithm. To build better machine learning models, and get the most value from them, accessible, scalable and durable storage solutions are imperative, paving the way for on-premises object storage. The steps include:

- Data Collection: Collecting raw data from billions of datasets available.
- Data Exploration: Exploring the data & the features related and being familiar with the data-types.
- Data Manipulation: Includes Cleaning of data, treating missing, repetitive values that are present.
- Data Analysis: Analysing the data to increase efficiency while applying the best Algorithm & feature selection according to our preferences.
- Application of Algorithm: Applying the algorithm to the model.
- Evaluation: Using evaluation metrices to calculate the least error and following the above to make further changes.

CHAPTER 4

DESCRIPTION OF MODULES

For the Salary Prediction Model, we will be using the python library modules such as numpy, pandas, matplotlib & sklearn, though which we will import different functions for computing the model. Further, we will use Flask library which is one of the most important module as it connects our backend end code with the front end.

1. Pandas: Pandas is a fast, powerful, flexible and easy to use open source data analysis and manipulation tool, built on top of the python programming language. It is a high-level data manipulation tool It is built on the Numpy package and its key data structure is called the DataFrame. DataFrames allow you to store and manipulate tabular data in rows of observations and columns of variables.

2. Numpy: NumPy is a Python package that stands for 'Numerical Python'. It is the core library for scientific computing, which contains a powerful n-dimensional array object.
It is a powerful N-dimensional array object which is in the form of rows and columns. We can initialize NumPy arrays from nested Python lists and access it elements.

3. Matplotlib: Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. It is a multi-platform data visualization library built on NumPy arrays and designed to work with the broader SciPy stack. One of the greatest benefits of visualization is that it allows us visual access to huge amounts of data in easily digestible visuals. Matplotlib consists of several plots like line, bar, scatter, histogram etc.

4. Scikit learn: Scikit-learn is probably the most useful library for machine learning in Python. The sklearn library contains a lot of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction. sklearn is used to build machine learning models. It should not be used for reading the data, manipulating and summarizing it. Components of scikit-learn:

- Supervised learning algorithms
- Cross-validation
- Unsupervised learning algorithms
- Various toy datasets
- Feature extraction

5. Flask: Flask is a web application framework written in Python. Armin Ronacher, who leads an international group of Python enthusiasts named Pocco, develops it. Flask is based on Werkzeug WSGI toolkit and Jinja2 template engine. Flask is considered more Pythonic than the Django web framework because in common situations the equivalent Flask web application is more explicit. Flask is also easy to get started with as a beginner because there is little boilerplate code for getting a simple app up and running.

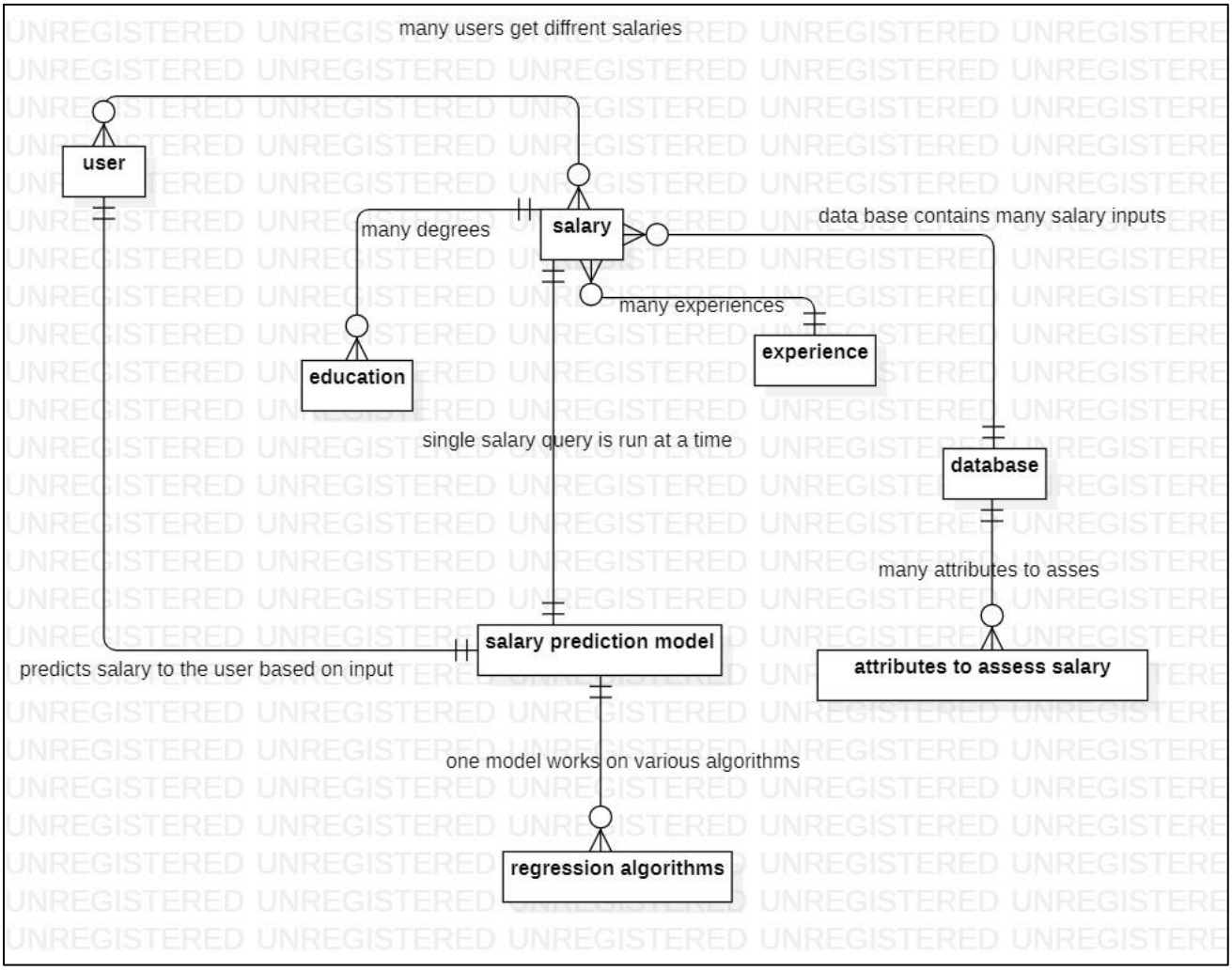
6. Html & CSS: Hypertext Markup Language (HTML) is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript. Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language such as HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.

7. Heroku: Heroku is a cloud platform as a service (PaaS) supporting several programming languages. One of the first cloud platforms, Heroku has been in development since June 2007. We will deploy our model on the Heroku platform.

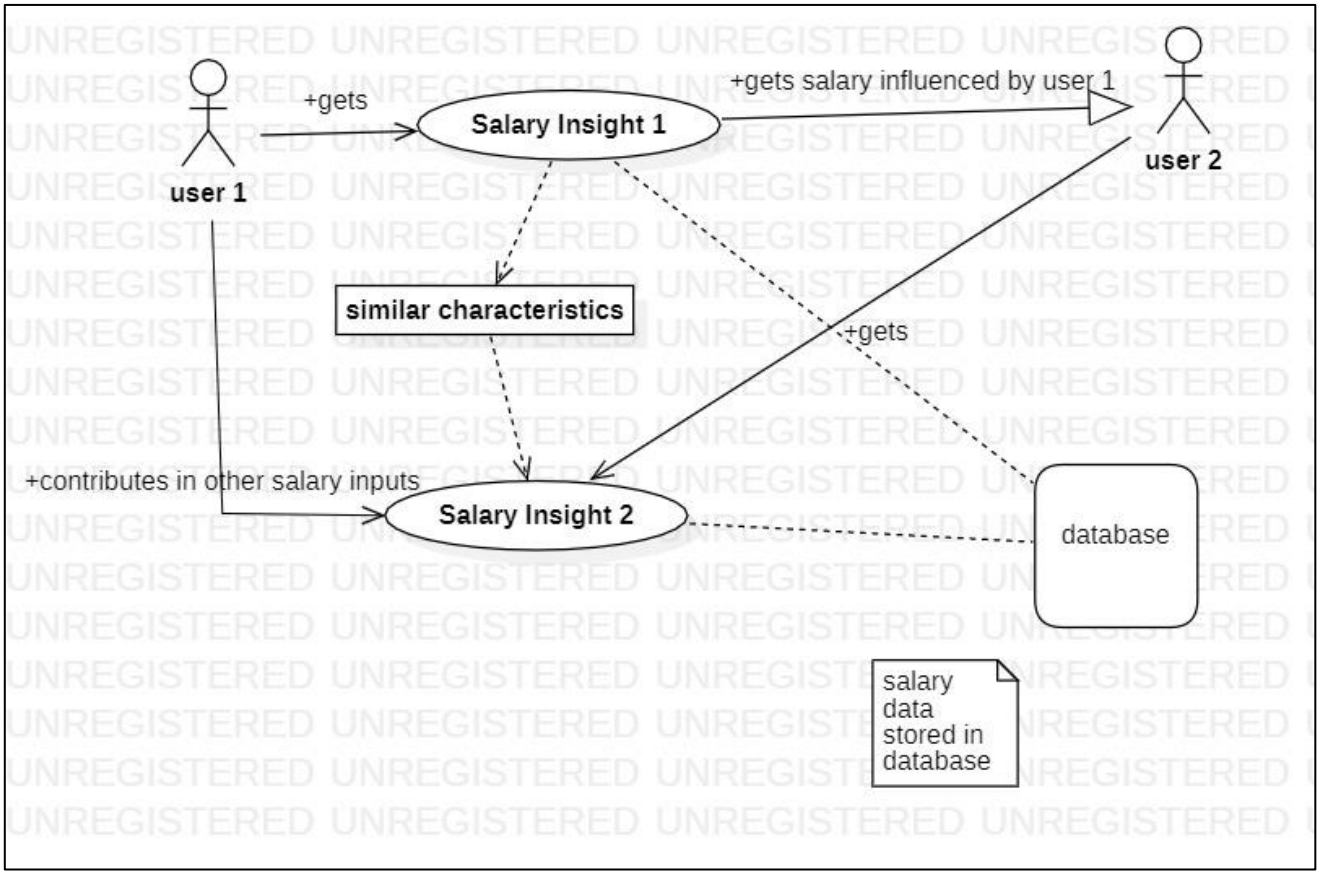
CHAPTER 5

UML DIAGRAMS

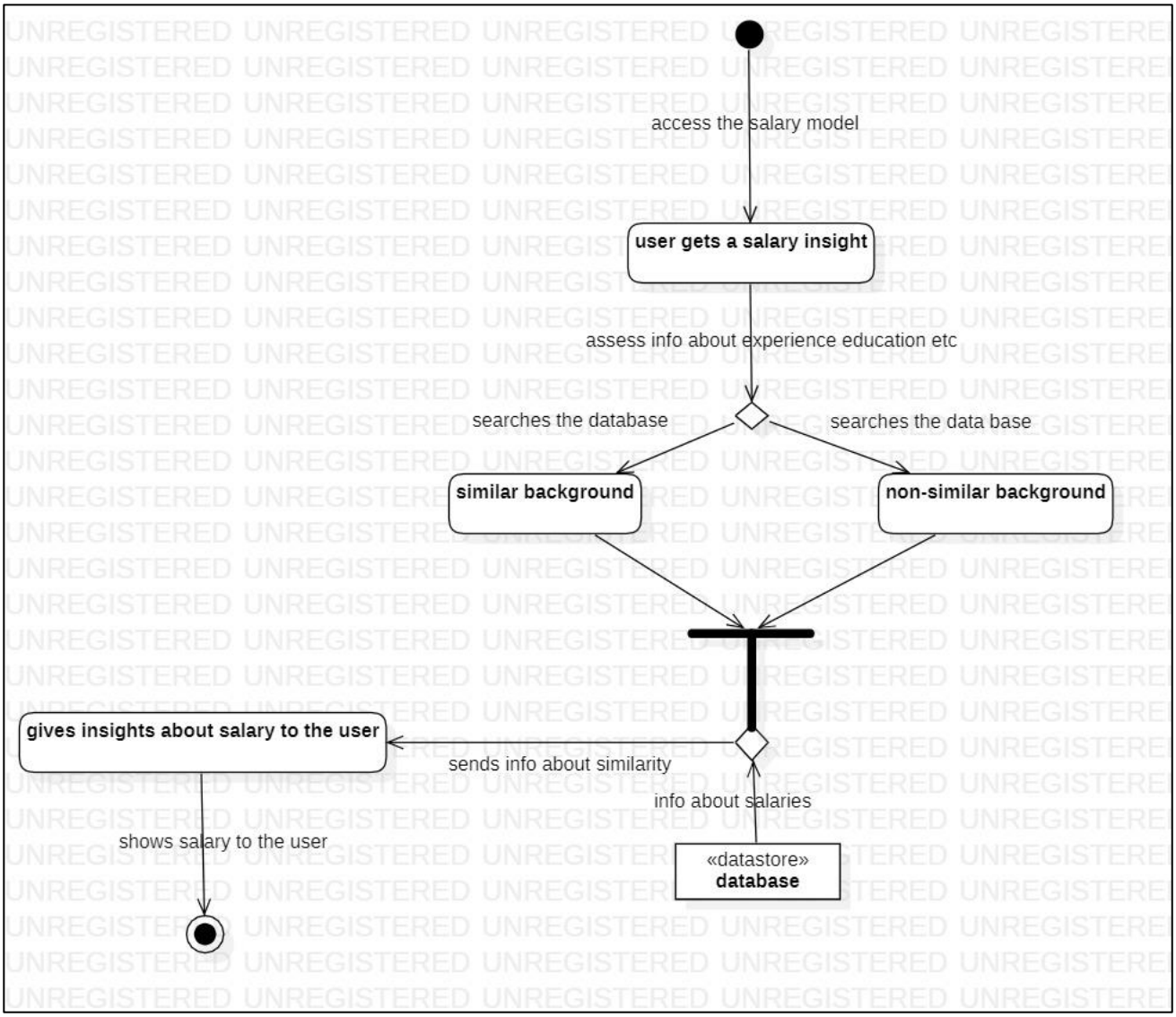
1. **ER Diagram:** An entity relationship diagram (ERD) shows the relationships of entity sets stored in a database. An entity in this context is an object, a component of data. An entity set is a collection of similar entities. These entities can have attributes that define its properties.



2. **Use Case Diagram:** A use case diagram is a graphical depiction of a user's possible interactions with a system. A use case diagram shows various use cases and different types of users the system has and will often be accompanied by other types of diagrams as well.



3. **Activity Diagram:** An activity diagram is a behavioural diagram i.e. it depicts the behaviour of a system. An activity diagram portrays the control flow from a start point to a finish point showing the various decision paths that exist while the activity is being executed.



CHAPTER 7

IMPLEMENTATION OF THE MODEL

1. **The Data**: Data collection is the first real step towards the real development of a machine learning model, collecting data. This is a critical step that will cascade in how good the model will be, the more and better data that we get, the better our model will perform. Our dataset named "survey_results_public" is a raw dataset. It means that a lot of pre-processing is required so that all it becomes useful for evaluation. Our dataset consists of 83439 rows and 48 features that will help us to predict the sales of the house and is fairly a big dataset.

2. **Loading the Data:** We load the dataset into our notebook using the pandas dataframe.

```
In [3]: import pandas as pd
        import matplotlib.pyplot as plt

        df = pd.read_csv("survey_results_public.csv")
        df.shape

Out[3]: (83439, 48)
```

3. **Data Pre-Processing:** Our Next step is to convert our data set into best possible format so that we can extract what all features are required to predict the price of the house. This is where all cleaning of our data takes place, be it treating the missing values, treating repetitive values, or addition of different features according to our needs. Once they are identified, there are several ways to deal with them:

   Eliminating the samples or features with missing values. (we risk to delete relevant information or too many samples).

   Imputing the missing values, with some pre-built estimators such as the Imputer class from scikit learn.

```
In [5]: df.head()
```

Out[5]:

	ResponseId	MainBranch	Employment	Country	US_State	UK_Country	EdLevel	Age1stCode	LearnCode	YearsCode	...	Age	Gender	Trans	
0	1	I am a developer by profession	Independent contractor, freelancer, or self-em...	Slovakia	NaN	NaN	Secondary school (e.g. American high school, G...	18 - 24 years	Coding Bootcamp;Other online resources (ex: vi...	NaN	...	25-34 years old	Man	No	H
1	2	I am a student who is learning to code	Student, full-time	Netherlands	NaN	NaN	Bachelor's degree (B.A., B.S., B.Eng., etc.)	11 - 17 years	Other online resources (ex: videos, blogs, etc...	7	...	18-24 years old	Man	No	H
2	3	I am not primarily a developer, but I write co...	Student, full-time	Russian Federation	NaN	NaN	Bachelor's degree (B.A., B.S., B.Eng., etc.)	11 - 17 years	Other online resources (ex: videos, blogs, etc...	NaN	...	18-24 years old	Man	No	F
3	4	I am a developer by profession	Employed full-time	Austria	NaN	NaN	Master's degree (M.A., M.S., M.Eng., MBA, etc.)	11 - 17 years	NaN	NaN	...	35-44 years old	Man	No	H

```
In [4]: df.dtypes

Out[4]: ResponseId                    int64
        MainBranch                   object
        Employment                   object
        Country                      object
        US_State                     object
        UK_Country                   object
        EdLevel                      object
        Age1stCode                   object
        LearnCode                    object
        YearsCode                    object
        YearsCodePro                 object
        DevType                      object
        OrgSize                      object
        Currency                     object
        CompTotal                   float64
        CompFreq                     object
        LanguageHaveWorkedWith       object
        LanguageWantToWorkWith       object
        DatabaseHaveWorkedWith       object
        DatabaseWantToWorkWith       object
        PlatformHaveWorkedWith       object
        PlatformWantToWorkWith       object
        WebframeHaveWorkedWith       object
        WebframeWantToWorkWith       object
        MiscTechHaveWorkedWith       object
        MiscTechWantToWorkWith       object
```

4. **Data Exploration:** Further, we explore our data as much as possible to know the features very well. We get to know the count of each features, their mean values, standard deviation, min and max value etc.

```
In [6]: df.describe()
```

Out[6]:

	ResponseId	CompTotal	ConvertedCompYearly
count	83439.000000	4.718300e+04	4.684400e+04
mean	41720.000000	2.119407e+69	1.184262e+05
std	24086.908893	4.603702e+71	5.272944e+05
min	1.000000	0.000000e+00	1.000000e+00
25%	20860.500000	1.600000e+04	2.702500e+04
50%	41720.000000	6.700000e+04	5.621100e+04
75%	62579.500000	1.400000e+05	1.000000e+05
max	83439.000000	1.000000e+74	4.524131e+07

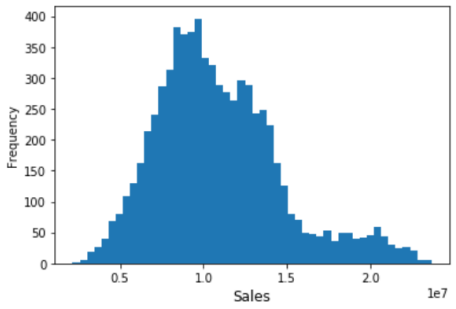
```
In [8]: df.isnull().sum()

Out[8]: ResponseId                    0
        MainBranch                    0
        Employment                  116
        Country                       0
        US_State                  68519
        UK_Country                79021
        EdLevel                     313
        Age1stCode                  196
        LearnCode                   476
        YearsCode                  1798
        YearsCodePro              22223
        DevType                   16955
        OrgSize                   22713
        Currency                  22359
        CompTotal                 36256
        CompFreq                  31289
```

```
In [59]: df.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 34756 entries, 7 to 64154
Data columns (total 5 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Country        34756 non-null  object
 1   EdLevel        34188 non-null  object
 2   YearsCodePro   34621 non-null  object
 3   Employment     34717 non-null  object
 4   Salary         34756 non-null  float64
dtypes: float64(1), object(4)
memory usage: 1.6+ MB

In [62]: df['Country'].value_counts()

Out[62]: United States                       7569
         India                               2425
         United Kingdom                      2287
         Germany                             1903
         Canada                              1178
                                             ...
         San Marino                             1
         Saint Vincent and the Grenadines       1
         Namibia                                1
         Benin                                  1
         Monaco                                 1
```

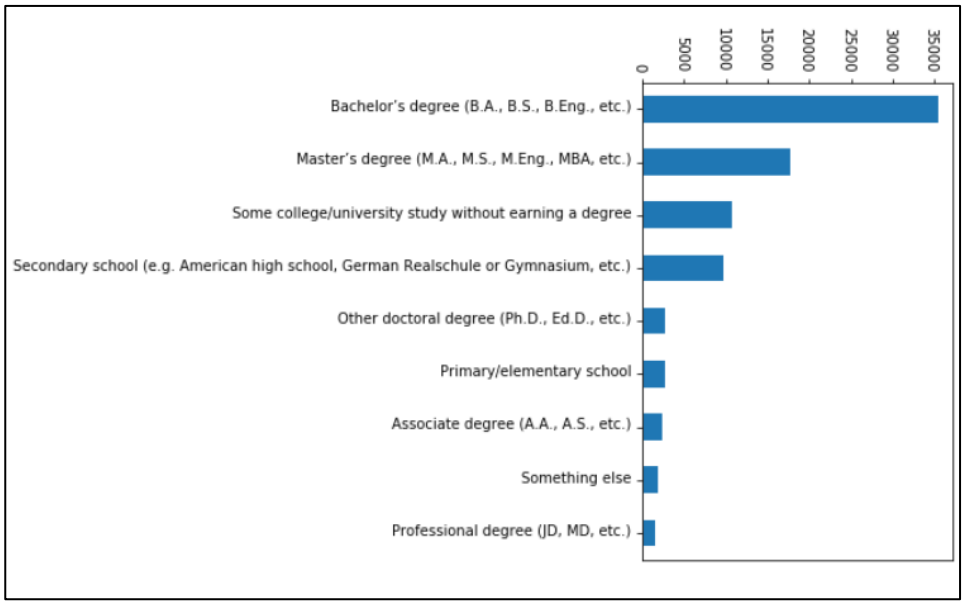
5. **Univariate Analysis:** Uni means one and variate means variable, so in univariate analysis, there is only one dependable variable. The objective of univariate analysis is to derive the data, define and summarize it, and analyze the pattern present in it. In a dataset, it explores each variable separately.

Histogram

```
In [6]: ## target variable
        df['SALES_PRICE'].plot.hist(bins = 50)
        plt.xlabel('Sales', fontsize=12)

Out[6]: Text(0.5, 0, 'Sales')
```



- The distribution of the target variable is slightly right skewed.
- We can see a small number of houses with a very high price.



Bar Plot for Education Level

6. **Data Manipulation:** Data Manipulation plays a very crucial role in the machine learning pipeline, as all the cleaning of the data takes place in this step. The process includes finding and treating missing values in the datset and then imputing them with different techniques like mean, mode, median, average or even droping the column(if irrelevant). Also, outliers are treated in this very step as deviate the plots from their actual positions.

```
In [58]: df = df[df["Salary"].notnull()]
         df.head()
```

Out[58]:

	Country	EdLevel	YearsCodePro	Employment	Salary
7	United States	Bachelor's degree (B.A., B.S., B.Eng., etc.)	13	Employed full-time	116000.0
9	United Kingdom	Master's degree (M.A., M.S., M.Eng., MBA, etc.)	4	Employed full-time	32315.0
10	United Kingdom	Bachelor's degree (B.A., B.S., B.Eng., etc.)	2	Employed full-time	40070.0
11	Spain	Some college/university study without earning ...	7	Employed full-time	14268.0
12	Netherlands	Secondary school (e.g. American high school, G...	20	Employed full-time	38916.0

```
In [60]: df = df.dropna()
         df.isnull().sum()
```

```
Out[60]: Country         0
         EdLevel         0
         YearsCodePro    0
         Employment      0
         Salary          0
         dtype: int64
```

```
In [61]: df = df[df["Employment"] == "Employed full-time"]
         df = df.drop("Employment", axis=1)
         df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 30019 entries, 7 to 64154
Data columns (total 4 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   Country       30019 non-null  object 
 1   EdLevel       30019 non-null  object 
 2   YearsCodePro  30019 non-null  object 
 3   Salary        30019 non-null  float64
dtypes: float64(1), object(3)
memory usage: 1.1+ MB
```

```
In [63]: def shorten_categories(categories, cutoff):
             categorical_map = {}
             for i in range(len(categories)):
                 if categories.values[i] >= cutoff:
                     categorical_map[categories.index[i]] = categories.index[i]
                 else:
                     categorical_map[categories.index[i]] = 'Other'
             return categorical_map
```

```
In [64]: country_map = shorten_categories(df.Country.value_counts(), 400)
         df['Country'] = df['Country'].map(country_map)
         df.Country.value_counts()
```

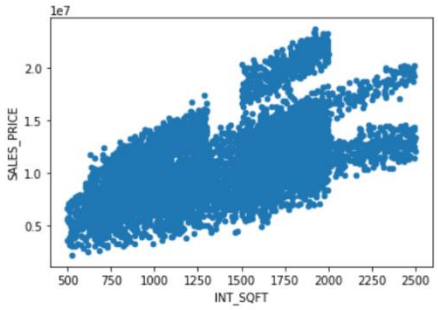
```
Out[64]: Other                 8549
         United States         7569
         India                 2425
         United Kingdom        2287
         Germany               1903
         Canada                1178
         Brazil                 991
         France                 972
         Spain                  670
         Australia              659
         Netherlands            654
         Poland                 566
         Italy                  560
         Russian Federation     522
         Sweden                 514
         Name: Country, dtype: int64
```

7. **Bivariate Analysis:** As the name suggests, bivariate analysis is the analysis of 2 features taken together. It is one of the simplest forms of statistical analysis, used to find out if there is a relationship between two sets of values. It usually involves the variables X and Y. Again, we randomly pick up any two features, one pair at a time and analyse it using histograms, bar graphs, plots etc.

1. Interior area and sales price (target)

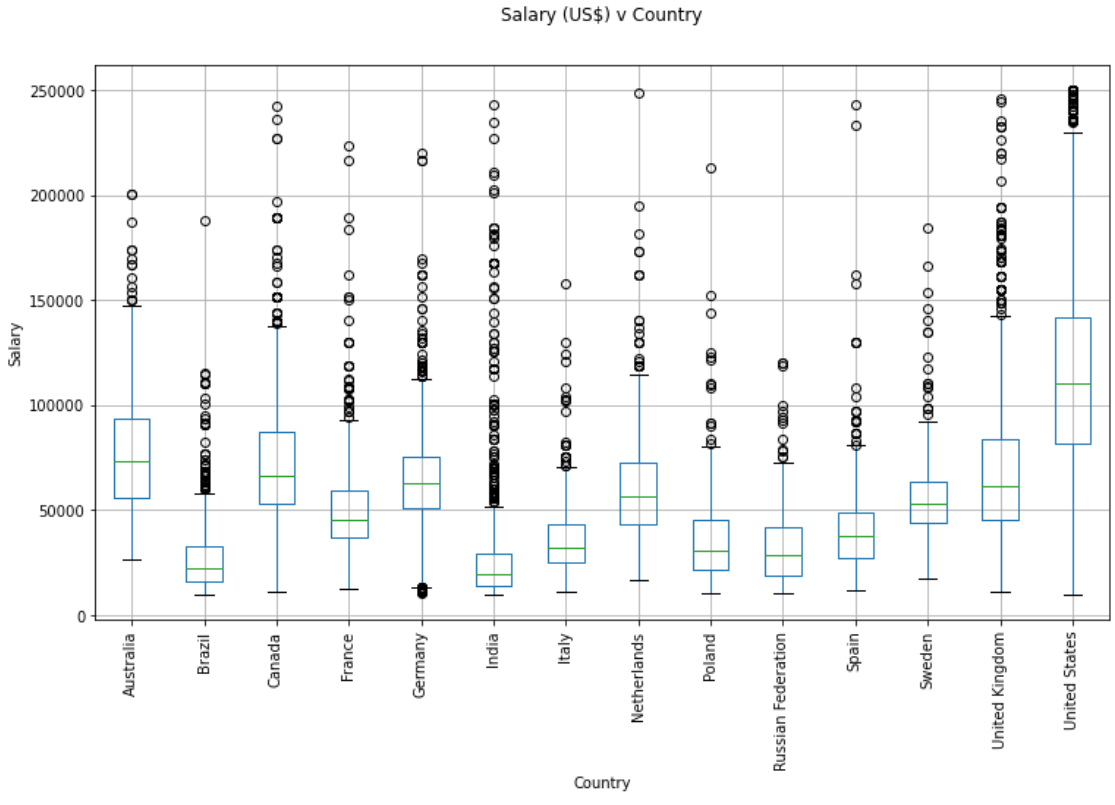
In [46]:
```
# interior area and sales price (target)

df.plot.scatter('INT_SQFT','SALES_PRICE')
```

Out[46]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x28191ad2d08>
```



- A very clear linear relationship can be seen between the interior area and sales price
- These variables have a positive correlation

In [67]:
```
fig, ax = plt.subplots(1,1, figsize=(12, 7))
df.boxplot('Salary', 'Country', ax=ax)
plt.suptitle('Salary (US$) v Country')
plt.title('')
plt.ylabel('Salary')
plt.xticks(rotation=90)
plt.show()
```



8. **Feature Scaling:** This is a crucial step in the preprocessing phase as the majority of machine learning algorithms perform much better when dealing with features that are on the same scale. The most common techniques are:

- Normalization: it refers to rescaling the features to a range of [0,1], which is a special case of min-max scaling. To normalize our data we'll simply need to apply the min-max scaling method to each feature column.

$$X_{changed} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

- Standardization: it consists in centering the feature columns at mean 0 with standard deviation 1 so that the feature columns have the same parameters as a standard normal distribution (zero mean and unit variance). This makes much more easier for the learning algorithms to learn the weights of the parameters. In addition, it keeps useful information about outliers and makes the algorithms less sensitive to them.

$$z = \frac{x - \mu}{\sigma}$$

μ = Mean

σ = Standard Deviation

9. **Implementing the Model:** Here comes the part where the actual machine learning algorithms are being implemented. As stated above, we are using Linear Regression Machine Learning Algorithm to predict the house price under Chennai house price Prediction model.

10. **Segregating Dependant and Independent Variables**: Independent variables (also referred to as Features) are the input for a process that is being analyzes. Dependent variables are the output of the process. For example:

y=f(x) Where,

x= independent variable

y= dependent variable

This means any changes in x will cause a change in the value of y. The change can be negative or positive. In Our Model, we have "SALARY" as our target/ dependent variable and all other features are considered as independent variables.

Dependent & Independent Variables

```
In [75]: X = df.drop("Salary", axis=1)
         y = df["Salary"]
```

11. **Splitting the Data Set into Train and Test Dataset**: We will split our data in three parts: training, testing and validating sets. We train our model with training data, evaluate it on validation data and finally, once it is ready to use, test it one last time on test data. The ultimate goal is that the model can generalize well on unseen data, in other words, predict accurate results from new data, based on its internal parameters adjusted while it was trained and validated.

In our Model, we have divided our dataset into a 70:30 ration, i.e., the training data consists of 70% of the dataset while the testing data consists of the remaining 30% of the dataset. To split the data we use train_test_split function provided by scikit-learn library.

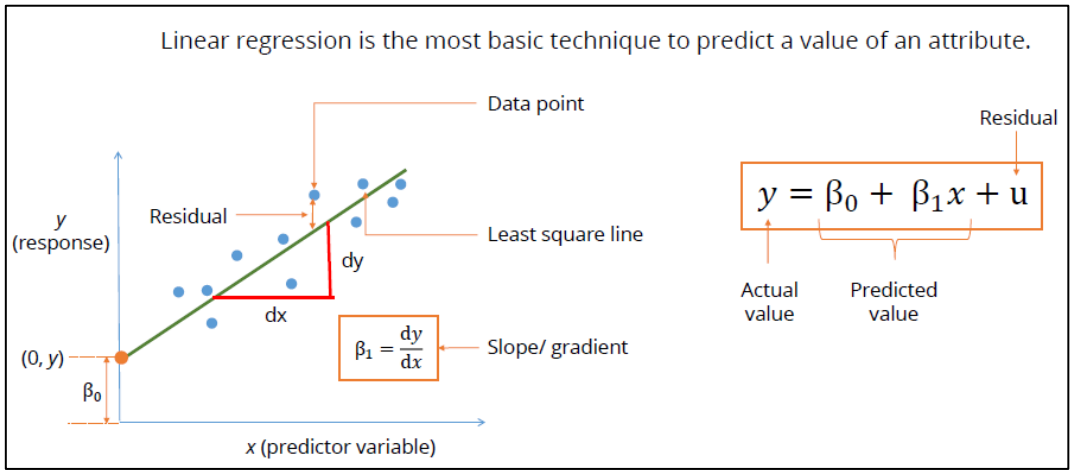
Train-Test Split

```
In [40]: from sklearn.model_selection import train_test_split
         train_x, test_x, train_y, test_y = train_test_split(X, y, test_size = 0.3, random_state = 2)

         train_x.shape, test_x.shape, train_y.shape, test_y.shape

Out[40]: ((18361, 3), (7869, 3), (18361,), (7869,))
```

12. **Implementing Linear Regression**:

    i) Learning Phase: In Linear regression we are given a number of predictor variables and a continuous response variable, and we try to find a relationship between those variables that allows us to predict a continuous outcome.

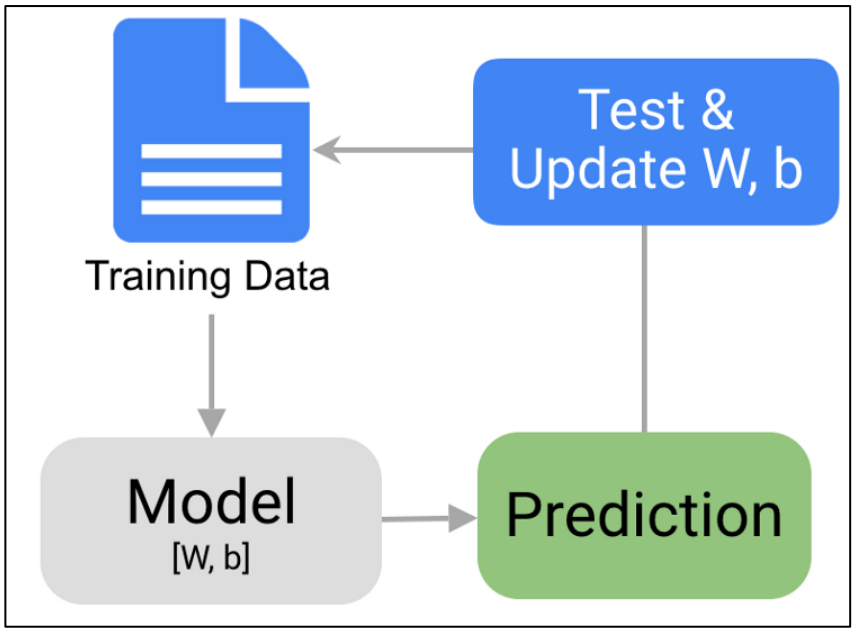
For example, given X and Y, we fit a straight line that minimize the distance using methods to estimate the coefficients like Ordinary Least Squares and Gradient Descent between the sample points and the fitted line. We'll use the intercept and slope learned, that form the fitted line, to predict the outcome of new data.



The formula for the straight line is y = B0 + B1x +u. Where x is the input, B1 is the slope, B0 the y-intercept, u the residual and y is the value of the line at the postion x.

The values available for being trained are B0 and B1, which are the values that affect the position of the line, since the only other variables are x (the input and y, the output (the residual is not considered). These values (B0 and B1) are the "weights" of the predicting funtion.

These weights and other, called biases, are the parameters that will be arranged together as matrixes.



The process is repeated, one iteration (or step) at a time. In each iteration the initial random line moves closer to the ideal and more accurate one.

Model Implementation using Linear Regression

```
In [44]: from sklearn.linear_model import LinearRegression

In [45]: linear_reg = LinearRegression()
         linear_reg.fit(X, y.values)

Out[45]: LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
```

- Optimization is the process of adjusting a model to get the best performance possible on training data (the learning process).
- Generalization is how well the model performs on unseen data. The goal is to obtain the best generalization ability.

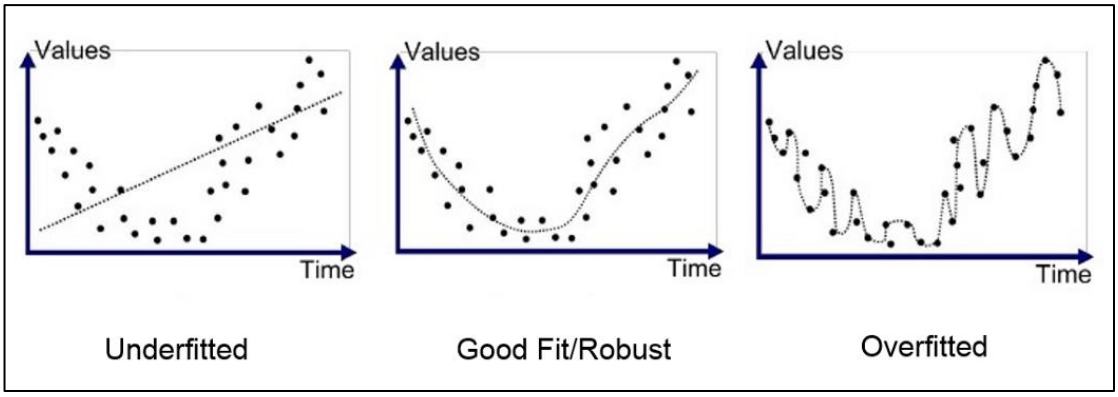
At the beginning of training, those two issues are correlated, the lower the loss on training data, the lower the loss on test data. This happens while the model is still underfitted: there is still learning to be done, it hasn't been modelled yet all the relevant parameters of the model.

There are two ways to avoid this overfitting, getting more data and regularization.

- Getting more data is usually the best solution, a model trained on more data will naturally generalize better.
- Regularization is done when the latter is not possible, it is the process of modulating the quantity of information that the model can store or to add constraints on what information it is allowed to keep. If the model can only memorize a small number of patterns, the optimization will make it to focus on the most relevant ones, improving the chance of generalizing well.

Regularization is done mainly by the following techniques:

- Reducing the model's size: Reducing the number of learnable parameters in the model, and with them its learning capacity.
- Adding weight regularization: L1 Regularization & L2 Regularization.



13. **Implementing Decision Tree:** Decision Tree is a decision-making tool that uses a flowchart-like tree structure or is a model of decisions and all of their possible results, including outcomes, input costs, and utility. Decision-tree algorithm falls under the category of supervised learning algorithms. It works for both continuous as well as categorical output variables.

Model Implementation using Decision Tree

```
In [49]: from sklearn.tree import DecisionTreeRegressor
         dec_tree_reg = DecisionTreeRegressor(random_state=0)
         dec_tree_reg.fit(X, y.values)

Out[49]: DecisionTreeRegressor(ccp_alpha=0.0, criterion='mse', max_depth=None,
                               max_features=None, max_leaf_nodes=None,
                               min_impurity_decrease=0.0, min_impurity_split=None,
                               min_samples_leaf=1, min_samples_split=2,
                               min_weight_fraction_leaf=0.0, presort='deprecated',
                               random_state=0, splitter='best')

In [51]: y_pred = dec_tree_reg.predict(X)
```

14. **Evaluation of the Model:** The final Step of the model is evaluating it using appropriate evaluation matrices. We have evaluated our model using score() and R2-sqaure metrics as it suits our model perfectly.

Model Evaluation in Linear Regression

```
In [47]: from sklearn.metrics import mean_squared_error, mean_absolute_error
         import numpy as np
         error = np.sqrt(mean_squared_error(y, y_pred))

In [48]: error

Out[48]: 41838.096750079996
```

Model Evaluation in Decision Tree

```
In [52]: error = np.sqrt(mean_squared_error(y, y_pred))
         print("${:,.02f}".format(error))

         $31,267.18
```

CHAPTER 8

CONCLUSION

In today's real world, it has become tough to store such huge data and extract them for one's own requirement. Also, the extracted data should be useful. The system makes optimal use of the Linear Regression Algorithm. The system makes use of such data in the most efficient way. The linear regression algorithm helps to fulfil customers by increasing the accuracy of estate choice and reducing the risk of investing in an estate.

Our Model Predicted An Accuracy score of 95.68% on the training dataset while it predicted an Accuracy score of 95.33% on the testing dataset. Since there is a very minute difference between the training and testing scores, we can say that our model has performed extremely well on the given dataset, that with such a high % score. It is illustrated that the approach contributes positively according to the evaluation.

FUTURE WORKS

Since nothing in this universe can be termed as "perfect", thus a lot of features can be added to make the system more widely acceptable and more user friendly. This will not only help to predict rates of other aras in the city but also will be more user beneficial.

In the upcoming phase of our project we will be able to connect an even larger dataset to this model so that the training can be even better. This model should check for new data, once in a month, and incorporate them to expand the dataset and produce better results

We can try out other dimensionality reduction techniques like Uni-variate Feature Selection and Recursive feature elimination in the initial stages.

Another major future scope that can be added is providing the model with estate database of more cities which will provide the user to explore more graduates and reach an accurate decision. More factors like training period that affect the job salary of a graduate shall be added. In-depth details of every individual will be added to provide ample details of a desired estate. This will help the system to run on a larger level.

REFERENCES

1. https://expertsystem.com/machine-learning-definition/#:~:text=Machine%20learning%20is%20an%20application,use%20it%20learn%20for%20themselves.
2. Fisher, R.A. (1922). "The goodness of fit of regression formulae, and the distribution of regression coefficients". Journal of the Royal Statistical Society. 85 (4): 597–612.
3. Schmidhuber, J. (2015). "Deep Learning in Neural Networks: An Overview". Neural Networks. 61: 85–117.
4. Yan, Xin (2009), Linear Regression Analysis: Theory and Computing, World Scientific, pp. 1–2.
5. Vapnik, V. N. The Nature of Statistical Learning Theory (2nd Ed.), Springer Verlag, 2000.
6. https://www.python.org/doc/essays/blurb/#:~:text=Python%20is%20an%20interpreted%2C%20object,programming%20language%20with%20dynamic%20semantics.&text=Python's%20simple%2C%20easy%20to%20learn,program%20modularity%20and%20code%20reuse.
7. https://medium.com/app-affairs/9-applications-of-machine-learning-from-day-to-day-life-112a47a429d0
8. https://www.kaggle.com/c/house-prices-advanced-regression-techniques/data
9. http://ijcsma.com/publications/march2019/V7I302.pdf
10. https://towardsdatascience.com/predicting-house-prices-with-linear-regression-machine-learning-from-scratch-part-ii-47a0238aeac1
11. https://flask.palletsprojects.com/en/1.1.x/