SIX WEEK SUMMER TRAINING

REPORT

On

HR-CHURN prediction Project

Submitted by

Mohit Kumar Mahato

Registration No. 11913514

Program Name: Computer Science Engineering

Under the Guidance of

Swapnil Desai

School of Computer Science & Engineering

Lovely Professional University, Phagwara

(June – July 2021)

DECLARATION

I hereby declare that I have completed my six weeks summer training at Board Infinity from 1st June 2021 to 9th July 2021 under the guidance of Swapnil Desai . I have declared that I have worked with full dedication during these six weeks of training and my learning outcomes fulfil the requirements of training for the award of degree of Bachelor of Technology (Computer Science Engineering) , Lovely Professional University, Phagwara.

Mohit Kumar Mahato

Name of Student: Mohit Kumar Mahato

Registration Number: 11913514

Date: 19/07/2021

Acknowledgement

I am over helmed in all humbleness and gratefulness to acknowledge my depth to all those who have helped me to put these ideas, well above the level of simplicity and into something concrete.
I would like to express my special thanks of gratitude to my teacher Swapnil Desai as well as Board Infinity who gave me the golden opportunity to do this wonderful project on the topic HR-CHURN-PREDICTION, which also helped me in doing a lot of Research and i came to know about so many new things. I am thankful to them.

Any attempt at any level can ‘t be satisfactorily completed without the support and guidance of MY parents and friends.
I would like to thank my parents who helped me a lot in gathering different information, collecting data and guiding me from time to time in making this project, despite of their busy schedules, they gave me different ideas in making this project unique.

Summer training certificate

CERTIFICATE OF COMPLETION

THIS CERTIFICATE IS AWARDED TO

Mohit Kumar Mahato

for successfully completing
Machine Learning Course

09th Jul, 2021		BI31ML35479075
ISSUED DATE	CEO, Board Infinity Sumesh Nair	CERTIFICATE NO.

BOARD

Table of content:-

Chapter 1 - Introduction to the project Undertaken:

Profile Of the Problem:

Case Study:

There is an ever increase in focus of effective requirement. An organization invest a lot of time and resources in search of potential candidates. The investment become loses is the selected candidate do not join organization in the end.

Challenges:

- ➢ Recruiter need to understand the chances of candidate of joining the organization.
- ➢ There are numerous factors for which the candidate can backout of the job.
- ➢ Confidential data cannot be obtained.

Research

The variables collected were as follows:

1. Candidate reference number
   Unique number to identify the candidate

2. DOJ extended
   Binary variable identifying whether candidate asked for date of joining extension (Yes/No)

3. Duration of accept offer
   Number of days taken by the candidate to accept the offer (continuous variable)

4. Notice period
   Notice period to be served in the parting company before candidate can join this company (continuous variable)

5. Offered band

   Band offered to the candidate based on experience and performance in interview rounds (categorical variable labelled C0/C1/C2)

6. Percentage hike Expected

   Percentage hike expected by the candidate (continuous variable)

7. Percentage hike offered

   Percentage hike offered by the company (continuous variable)

8. Percentage Difference

   Difference of hike offered and hike expected is considered

9. Joining Bonus

   Binary variable indicating if joining bonus was given or not (Yes/No)

10. Gender

    Gender of the candidate (Male/Female)

11. Candidate source

    Source from which resume of the candidate was obtained (categorical variables with categories: Employee referral/Agency/Direct)

12. Year of experience

    Relevant years of experience of the candidate for the position offered (continuous variable)

13. LOB

    Line of business for which offer was rolled out (categorical variable)

14. DOB

    Date of birth of the candidate

15. Joining location

    Company location for which offer was rolled out for candidate to join (categorical variable)

16. Candidate relocation status

    Binary variable indicating whether candidate has to relocate from one city to another city for joining (Yes/No)

17. HR status

    Final joining status of candidate (Joined/Not Joined)

Technologies Learnt

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. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms and can be freely distributed.

Areas where python is used:

1. Artificial Intelligence and Machine Learning
2. Web Development
3. Game Development
4. Software Development
5. Scientific and Numeric Application

Data Wrangling:

Data disputes include processing data in a variety of ways such as merging, grouping, merging etc. for the purpose of analysis or to make them ready for use with another set of data. Python has built-in features to use these conflicting methods across various data sets to achieve analytical purpose.

The data wrangling in python interacts with the function below:

1. Data exploration
2. Dealing with missing values
3. Reshaping data
4. Filtering data

Descriptive Statistics:

Descriptive statistics are short descriptive coefficients that summarize a set of data provided, either for public representation or a sample of people. Descriptive statistics were divided into measures of inclination and variance measures (spreads). Moderate inclination methods include mean, medium, and mode, while variability measures include standard deviation, variability, quantity and maximum variability, kurtosis, and skewness.

Artificial Intelligence:

Artificial intelligence (AI) is a wide-ranging branch of computer science concerned with building smart machines capable of performing tasks that typically require human intelligence. AI is an interdisciplinary science with multiple approaches, but advancements in machine learning and deep learning are creating a paradigm shift in virtually every sector of the tech industry.

Some Applications where AI is used:

1. Google Maps
2. Face Detection and Recognition
3. Chatbots
4. Search and Recommendation System
5. Natural Language Processing
6. Social Intelligence
7. General Intelligence

Natural Language Processing:

Natural language processing (NLP) refers to the computer science department - and in particular, the branch of artificial intelligence or AI - which is responsible for empowering computers to understand text and spoken words in the same way as humans.

The NLP incorporates computational linguistics — a model based on the principles of human language — and mathematics, learning equipment, and in-depth learning models. Together, these technologies enable computers to process human language in the form of text or voice data and 'understand' its full meaning, completed for the purpose of a platform or author.

NLP runs computer programs that translate text from one language to another, respond to spoken commands, and summarize large volumes of text quickly - even in real time. There is a great opportunity for you to interact with NLP in the form of GPS voice applications, digital assistants, text-to-speech software, customer service conversations, and other consumer convenience. But NLP also plays a growing role in business solutions that help simplify business operations, increase employee productivity, and simplify critical business processes.

Several NLP activities reduce human text and voice data in ways that help the computer have an idea of what it involves. Some of these activities include the following:

- Speech recognition, also called speech-to-text, is the function of faithfully converting voice data into text data. Speech recognition is required for any app that follows voice commands or answers spoken questions. What makes speech recognition particularly challenging is the way people speak - fast, loose words together, with different accents and pronunciation, in different ways of speaking, and in the use of incorrect grammar.

- Part of the marking of a speech, also called grammar marking, is the process of determining the part of speech of a particular word or piece of text based on its use and context. Part of the expression refers to 'doing' as an action from 'I can make a paper airplane,' and as a noun in 'What car building do you have?
- Noun distinction is the choice of the meaning of a word that has multiple meanings in the process of semantic analysis that determines the word that makes the most sense in a given

context. For example, the conceptual variation of a word helps to separate the meaning of the verb 'do' 'to make a distance' (gain) compared to 'make a bet' (place).

- A collaborative reference solution is a function of determining when and where two words refer to the same business and when. The most common example is to define a person or thing referred to in a particular pronoun (e.g., 'She' = 'Mary'), but may also involve identifying a metaphor or expression in a text (eg 'Bear' is not an animal but a furry adult).

- Neurological analysis attempts to exclude subjective qualities — attitudes, emotions, sarcasm, confusion, suspicion — in the text.

- Generation of indigenous language is sometimes described as contrary to the familiarity of speech or textual speaking; it is the task of putting organized information into human languages.

Machine Learning:

Machine learning (ML) is a type of artificial intelligence (AI) that allows software applications to become more accurate at predicting outcomes without being explicitly programmed to do so. Machine learning algorithms use historical data as input to predict new output values.

There are 4 main Types of Machine Learning

- Supervised Learning:

  Supervised learning is one of the most basic types of machine learning. In this type, the machine learning algorithm is trained on labeled data. Even though the data needs to be labeled accurately for this method to work, supervised learning is extremely powerful when used in the right circumstances.

  - ✓ Classification:

    It is a Targeted Reading activity where the output has defined labels (exact value). Example above Figure A, Output - Purchase defines labels namely 0 or 1; 1 means the customer will buy and 0 means the customer will not buy. The purpose here is to predict the divisive values of a particular class and to evaluate them on the basis of accuracy. It can be in binary categories or in multiple categories. According to the binary category, the model predicts either 0 or 1; yes or no but in the case of multiple class divisions, the model predicts more than one category.

  - ✓ Regression:

    It is a Targeted Reading activity where the output is a continuous value. Example above Figure B, Output - Wind speed does not have any different value but continues at a certain distance. The goal here is to predict the closest value to the actual output as our model can then test is performed by calculating the error value. A bit of an error grows with the accuracy of our regression model.

❖ Unsupervised Learning:

Unsupervised machine learning holds the advantage of being able to work with unlabeled data. This means that human labor is not required to make the dataset machine-readable, allowing much larger datasets to be worked on by the program. In supervised learning, the labels allow the algorithm to find the exact nature of the relationship between any two data points. However, unsupervised learning does not have labels to work off of, resulting in the creation of hidden structures.

❖ Reinforcement Learning:

Reinforcement learning directly takes inspiration from how human beings learn from data in their lives. It features an algorithm that improves upon itself and learns from new situations using a trial-and-error method. Favorable outputs are encouraged, or 'reinforced', and non-favorable outputs are discouraged or 'punished'. Based on the psychological concept of conditioning,

Deep Learning:

Deep learning is a lower set of machine learning, which is a neural network with three or more layers. These neural networks try to mimic the way the human brain works - even though they do not match their capabilities - allowing it to "learn" from big data. While a single-layer neural network can still make similar predictions, additional hidden layers can help increase and improve accuracy.

In-depth learning drives many intelligent applications (AI) and services that improve automation, perform analytical and physical tasks without human intervention. In-depth learning technologies are behind everyday products and services (such as digital assistants, voice-enabled remotes, and credit card fraud) and emerging technologies (such as self-driving cars).

Deep neural networks consist of multiple layers of interconnected nodes, each structure on top of the previous layer of refinement and increasing prediction or classification. This continuous network integration is called forward distribution. The input and output layers of the deep neural network are called the visible layers. The input layer where the in-depth learning model enters the data for processing, and the output layer is where the final prediction or division is made.

There are different types of neural networks to address specific problems or datasets. For example,

- Convolutional neural networks (CNNs), widely used in computer viewing and image editing applications, can detect features and patterns within an image, enabling functions, such as object detection or recognition. In 2015, CNN challenged a person for the first time.
- Repetitive network neural network (RNNs) is commonly used in native language applications and speech recognition as it captures sequential data or times.

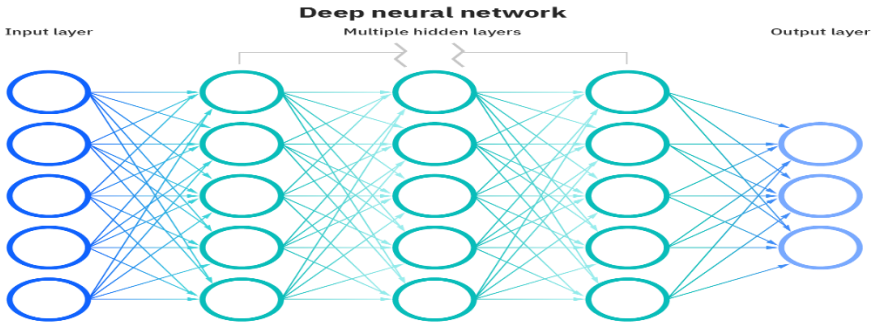
Neural Network:

Neural networks, also known as artificial neural networks (ANNs) or simulated neural networks (SNNs), are a subset of machine learning and are at the heart of deep learning algorithms. Their name and structure are inspired by the human brain, mimicking the way that biological neurons signal to one another.

Artificial neural networks (ANNs) are comprised of a node layer, containing an input layer, one or more hidden layers, and an output layer. Each node, or artificial neuron, connects to another and has an associated weight and threshold. If the output of any individual node is above the specified threshold value, that node is activated, sending data to the next layer of the network. Otherwise, no data is passed along to the next layer of the network.

Neural networks are widely used, for applications for financial performance, business planning, trading, business statistics and product storage. Neural networks have also gained widespread acceptance in business processes such as forecasting and marketing solutions, fraud detection and risk assessment.

The neural network analyzes price data and explores opportunities to make trading decisions based on data analysis. Networks can distinguish offline dependencies and patterns other methods of technical analysis cannot



Reason For Choosing This Technology:

Machine learning model predictions allow businesses to make more accurate estimates of the potential consequences of a historical data-based query, which can be about all sorts of things - customer dismissal opportunities, potentially fraudulent activity, and more. This gives the business information that results in a significant business value. For example, if a model predicts that a customer may emerge, an entity may identify them with a specific contact and access that will prevent the loss of that customer.

Python provides short and readable code. While complex algorithms and the flow of various functions work after machine learning and AI, Python simplicity allows developers to write reliable programs. The developers found that they put all their efforts into solving the ML problem instead of focusing on the nuances of language technology.

Additionally, Python attracts as many developers as possible. Python code is understandable to humans, making it easy to build machine learning models.

Many programmers claim that Python is more accurate than other programming languages. Some point to multiple frameworks, libraries, and extensions that facilitate the implementation of different functions. It is generally accepted that Python is ideal for collaborative use when more engineers are involved. Since Python is a universal target language, it can create a complex set of learning activities and allow you to create prototypes quickly allowing you to test your product for machine learning purposes.

Chapter 2 - Problem Analysis:-

Product definition:

We are provided Datasheet which consists data of 8998 candidates which includes

Candidate reference Number , Date of Joining extension , Duration of accept the offer , Notice Period , Offered band , Percentage hike expected , Percentage hike offered , Percentage difference , Joining bonus , Gender , Candidate source , Year of Experience , Line of Business , Cate of Birth , Joining Location , Candidate Relocation status HR status

I have to train my the Machine using this data so that when it can predict the Chance of upcoming candidate HR status whether they will be joining the organization or not .

So basically we have to use Supervised learning where our model will be trained using the data sheet Input variables except Hr status and output variable Hr status which is Final joining status of the candidate.

Here Independent variables are all the variables except HR status and dependent Variable is Hr status which depends on the other factors .

Software Requirement Analysis :-

 # Python :

Python is easy while the topics of linear algebra or calculus can be so perplexing, they require the maximum amount of effort. Python can be executed rapidly which allows ML engineers to approve an idea immediately.

Python is already very well-known and thus, it has many various libraries and frameworks that can be utilized by engineers. These libraries and frameworks are truly valuable in saving time which makes Python significantly more well-known.

Many cross-language tasks can be performed effectively on Python due to its portable and extensible nature.

 # NumPy Package Python :

NumPy is the fundamental package for scientific computing in Python. It is a Python library that provides a multidimensional array object, various derived objects (such as masked arrays and matrices), and an assortment of routines for fast operations on arrays, including mathematical, logical, shape manipulation, sorting, selecting, I/O, discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation and much more. It will be used in the project for various calculation.

 # Pandas Package Python :

It will be used for Tabular data with heterogeneously typed columns, as in an SQL table or Excel spreadsheet or csv file.

Pandas is a Python package providing fast, flexible, and expressive data structures designed to make working with “relational” or “labelled” data both easy and intuitive. It aims to be the fundamental high-level building block for doing practical, real-world data analysis in Python.

#Matplotlib Package Python :

matplotlib is a collection of functions that make matplotlib work like MATLAB. Each pyplot function makes some change to a figure: e.g., creates a figure, creates a plotting area in a figure, plots some lines in a plotting area, decorates the plot with labels, etc. It will be used to analysis data using graph .

Seaborn: statistical data visualization Python Package :

Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.

Seaborn helps you explore and understand the data. Its plotting functions operate on data frames and arrays containing whole datasets and internally perform the necessary semantic mapping and statistical aggregation to produce informative plots. Its dataset-oriented, declarative API lets us focus on what the different elements of your plots mean, rather than on the details of how to draw them.

#Scikit-learn (sklearn) python Machine Learning Package:

Scikit-learn (Sklearn) is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modelling including classification, regression, clustering and dimensionality reduction via a consistence interface in Python. This library, which is largely written in Python, is built upon NumPy, SciPy and Matplotlib

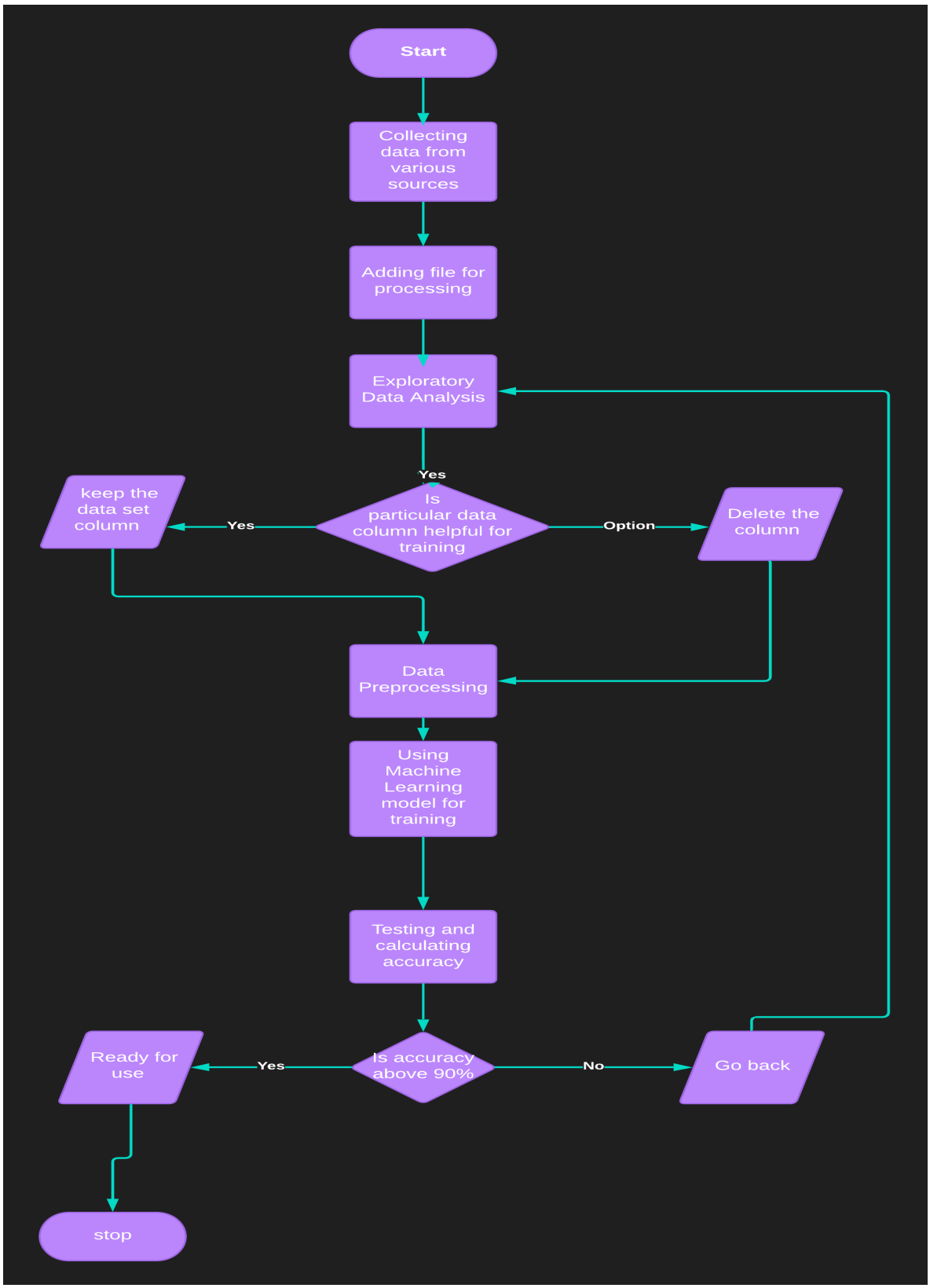
#Random forest classifier:

A random forest is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting. The sub-sample size is controlled with the max_samples parameter if bootstrap=True (default), otherwise the whole dataset is used to build each tree.

It is perhaps the most popular and widely used machine learning algorithm given its good or excellent performance across a wide range of classification and regression predictive modelling problems. It is also easy to use given that it has few key hyperparameters and sensible heuristics for configuring these hyperparameters.

Design

Flowchart :



Chapter 3 - Implementation:

Importing the data set:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns %matplotlib inline
```

The Pandas dataframe.info () function is used to obtain a short summary of the data name. It really helps when doing data analysis. To get a quick overview of the database we use the dataframe.info function.

```
hire_df=pd.read_csv("HR_Data.csv")


hire_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8998 entries, 0 to 8997
Data columns (total 18 columns):
 #   Column                       Non-Null Count  Dtype
---  ------                       --------------  -----
 0   SLNO                         8998 non-null   int64
 1   Candidate.Ref                8998 non-null   int64
 2   DOJ.Extended                 8998 non-null   object
 3   Duration.to.accept.offer     8998 non-null   int64
 4   Notice.period                8998 non-null   int64
 5   Offered.band                 8998 non-null   object
 6   Pecent.hike.expected.in.CTC  8998 non-null   float64
 7   Percent.hike.offered.in.CTC  8998 non-null   float64
 8   Percent.difference.CTC       8998 non-null   float64
 9   Joining.Bonus                8998 non-null   object
 10  Candidate.relocate.actual    8998 non-null   object
 11  Gender                       8998 non-null   object
 12  Candidate.Source             8998 non-null   object
 13  Rex.in.Yrs                   8998 non-null   int64
 14  LOB                          8998 non-null   object
 15  Location                     8998 non-null   object
 16  Age                          8998 non-null   int64
 17  Status                       8998 non-null   object
dtypes: float64(3), int64(6), object(9)
memory usage: 1.2+ MB
```

- The data review took the help of the ".head ()" library function of the pandas library which returns the first five data sets. Similarly ".tail ()" returns the last five views of the data.

```
hire_df.head()
```

	SLNO	Candidate.Ref	DOJ.Extended	Duration.to.accept.offer	Notice.period	Offered.band	Pecent.hike.expected.in.CTC	Percent.hike.offered.in.CTC	Perc
0	1	2110407	Yes	14	30	E2	-20.79	13.16	42.86
1	2	2112635	No	18	30	E2	50.00	320.00	180.(
2	3	2112838	No	3	45	E2	42.84	42.84	0.00
3	4	2115021	No	26	30	E2	42.84	42.84	0.00
4	5	2115125	Yes	1	120	E2	42.59	42.59	0.00

Exploratory Data Analysis:

Data Evaluation Analysis refers to the critical process of conducting preliminary data analysis to detect patterns, inaccuracies, assumptions testing and hypothetical tests with the help of summary statistics and graphical presentations.

```
hire_df.columns
```

```
Index(['SLNO', 'Candidate.Ref', 'DOJ.Extended', 'Duration.to.accept.offer',
       'Notice.period', 'Offered.band', 'Pecent.hike.expected.in.CTC',
       'Percent.hike.offered.in.CTC', 'Percent.difference.CTC',
       'Joining.Bonus', 'Candidate.relocate.actual', 'Gender',
       'Candidate.Source', 'Rex.in.Yrs', 'LOB', 'Location', 'Age', 'Status'],
      dtype='object')
```

checking binary data column

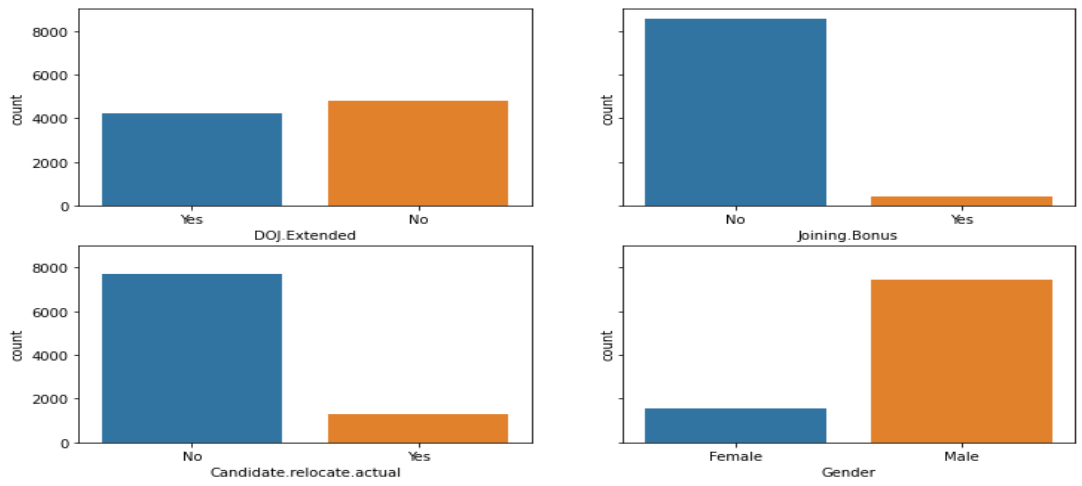
```
columns = hire_df.columns
binary_cols = []

for col in columns:
    if hire_df[col].value_counts().shape[0] == 2:
        binary_cols.append(col)
```

```
binary_cols
```

```
['DOJ.Extended',
 'Joining.Bonus',
 'Candidate.relocate.actual',
 'Gender',
 'Status']
```

```
#class distribution of binary features.

fig, axes = plt.subplots(2, 2, figsize=(12, 7), sharey=True)

sns.countplot("DOJ.Extended", data=hire_df, ax=axes[0,0])
sns.countplot("Joining.Bonus", data=hire_df, ax=axes[0,1])
sns.countplot("Candidate.relocate.actual", data=hire_df, ax=axes[1,0
])



sns.countplot("Gender", data=hire_df, ax=axes[1,1])
```

```
<AxesSubplot:xlabel='Gender', ylabel='count'>
```




```
#checking the relation of joining bonus , candidate relocate and gender with
joining status #first we change the values of status from 'yes'/'no' to 0/1
```

```
num = {'Joined':1, 'Not Joined':0}
hire_df.Status.replace(num, inplace=True)
hire_df["Status"]=pd.to_numeric(hire_df["Status"],
downcast="integer")
```

```
hire_df[['Gender','Status']].groupby(['Gender']).mean()
```

	Status
Gender	
Female	0.823985
Male	0.810796

```
hire_df[['Joining.Bonus','Status']].groupby(['Joining.Bonus']).mean()
```

	Status
Joining.Bonus	
No	0.813425
Yes	0.805755

```
hire_df[['Candidate.relocate.actual','Status']].groupby(['Candidate.
relocate.actual']).mean()
```

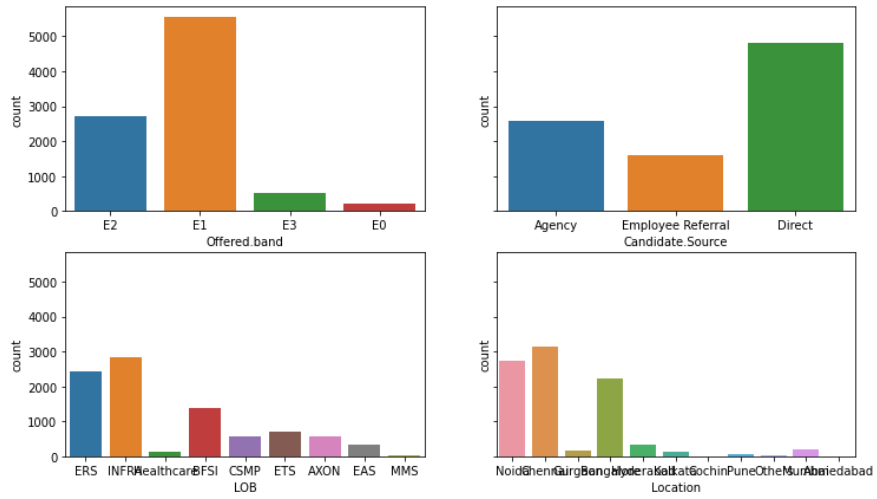
Candidate.relocate.actual	
No	0.781785
Yes	1.000000

Except for Candidate.relocate.actual no other feature has significant effect on Joining status so we will only include Candidate.relocate.actual feature in model of all the binary data features

Checking the effect of other features on the target

```
fig, axes = plt.subplots(2, 2, figsize=(12, 7), sharey=True)

sns.countplot("Offered.band", data=hire_df, ax=axes[0,0])
sns.countplot("Candidate.Source", data=hire_df, ax=axes[0,1])
sns.countplot("LOB", data=hire_df, ax=axes[1,0])
sns.countplot("Location", data=hire_df, ax=axes[1,1])
```

```
<AxesSubplot:xlabel='Location', ylabel='count'>
```




```
hire_df[['Offered.band','Status']].groupby(['Offered.band']).mean()
```

	Status
Offered.band	
E0	0.763033
E1	0.813106
E2	0.809735
E3	0.851485

```
hire_df[['Candidate.Source','Status']].groupby(['Candidate.Source'])
.mean()
```

	Status
Candidate.Source	
Agency	0.758221
Direct	0.820112
Employee Referral	0.880124

```
hire_df[['LOB','Status']].groupby(['LOB']).mean()
```

	Status
LOB	
AXON	0.774648
BFSI	0.758596
CSMP	0.815199
EAS	0.734104
ERS	0.781211
ETS	0.831169
Healthcare	0.822581
INFRA	0.877895
MMS	1.000000

```
hire_df.LOB.value_counts()
```

```
: INFRA         2850
  ERS           2427
  BFSI          1396
  ETS            693
  CSMP           579
  AXON           568
  EAS            346
  Healthcare     124
  MMS             15
  Name: LOB, dtype: int64
```

```
hire_df[['Location','Status']].groupby(['Location']).mean()
```

	Status
Location	
Ahmedabad	0.833333
Bangalore	0.781264
Chennai	0.789273
Cochin	0.875000
Gurgaon	0.808219
Hyderabad	0.780059
Kolkata	0.775194
Mumbai	0.893401
Noida	0.866202
Others	1.000000
Pune	0.791667

I won't use location for training the model

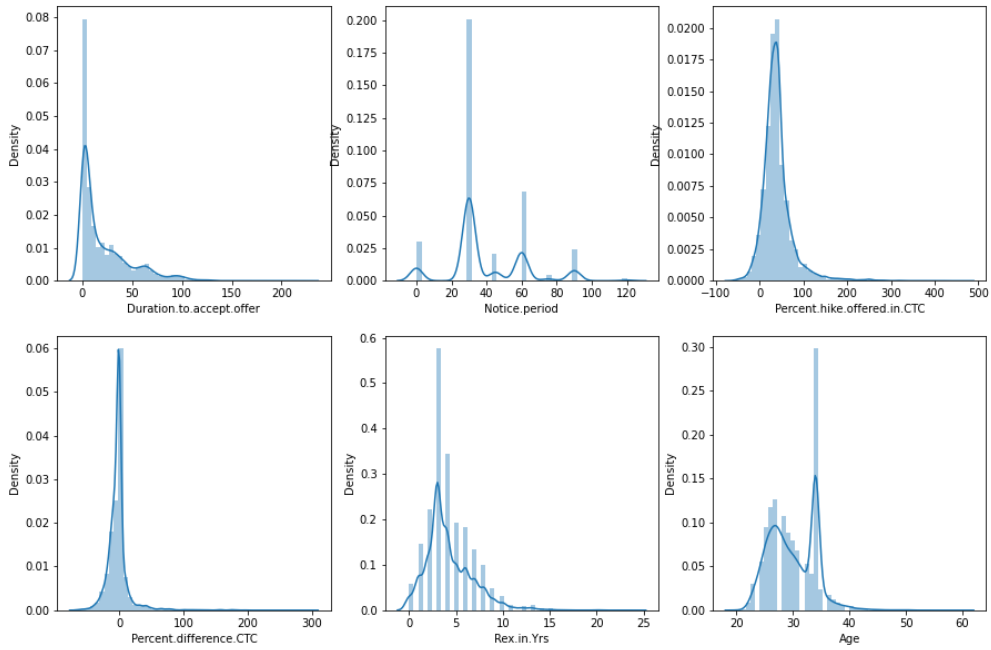
Checking continuous variables:

Continuous variance can take an unlimited number of values between the lowest and highest rated points. Continuous variations include things like speed and distance. Further details are highly desirable for unlimited statistics; however, they are often of little help in data mining and are often embedded in separate data or sets

The important continuous variables to check are :- Duration.to.accept.offer , Notice.period , Percent.hike.offered.in.CTC , Percent.difference.CTC , Rex.in.Yrs and Age .

```
fig, axes = plt.subplots(2,3, figsize=(15, 10))

sns.distplot(hire_df["Duration.to.accept.offer"], ax=axes[0,0])
sns.distplot(hire_df["Notice.period"], ax=axes[0,1])
sns.distplot(hire_df["Percent.hike.offered.in.CTC"], ax=axes[0,2])
sns.distplot(hire_df["Percent.difference.CTC"], ax=axes[1,0])
sns.distplot(hire_df["Rex.in.Yrs"], ax=axes[1,1])
sns.distplot(hire_df["Age"], ax=axes[1,2])
```

```
<AxesSubplot:xlabel='Age', ylabel='Density'>
```




```
hire_df[['Percent.hike.offered.in.CTC','Percent.difference.CTC','Status']].groupby('Status').mean()
```

	Percent.hike.offered.in.CTC	Percent.difference.CTC
Status		
0	38.588460	-2.929298
1	41.147158	-1.263402

The Percent.difference.CTC is important to consider for the model and we can leave the Percent.hike.offered.in.CTC

```
hire_df[['Duration.to.accept.offer','Notice.period',"Rex.in.Yrs","Age",'Status']].groupby('Status').mean()
```

	Duration.to.accept.offer	Notice.period	Rex.in.Yrs	Age
Status				
0	24.956599	48.192628	4.439358	29.517836
1	20.617687	37.233461	4.193002	30.004647

and also the Duration.to.accept.offer and Notice.period columns seems to be effecting the status column data so we will include these two featues too

Dropping irrelevent features of the dataframe

After analizing all the different types of variables i have decided to drop the following columns :- 'SLNO', 'Candidate.Ref', 'DOJ.Extended','Joining.Bonus','Gender','Rex.in.Yrs', 'Location', 'Age'

```
hire_df.drop(['SLNO', 'Candidate.Ref', 'DOJ.Extended', 'Joining.Bonus','Gender', 'Rex.in.Yrs', 'Location', 'Age'], axis=1, inplace=True)
```

```
hire_df.head()
```

	Duration.to.accept.offer	Notice.period	Offered.band	Pecent.hike.expected.in.CTC	Percent.hike.offered.in.CTC	Percent.difference.CTC	Candidate.relocate
0	14	30	E2	-20.79	13.16	42.86	No
1	18	30	E2	50.00	320.00	180.00	No
2	3	45	E2	42.84	42.84	0.00	No
3	26	30	E2	42.84	42.84	0.00	No
4	1	120	E2	42.59	42.59	0.00	Yes

Data Pre-processing:

In any machine learning process, data processing is the process by which data is converted, or coded, to bring it to a point where the machine is now easier to process. In other words, data features can now be easily translated by algorithm.

```
from sklearn.preprocessing import LabelEncoder, OneHotEncoder
from sklearn.preprocessing import MinMaxScaler
```

```
cat_feat=['Candidate.relocate.actual','Offered.band','Candidate.Source','LOB']
X = pd.get_dummies(hire_df, columns=cat_feat, drop_first=True)
```

```
sc = MinMaxScaler()
a = sc.fit_transform(hire_df[['Duration.to.accept.offer']])
b = sc.fit_transform(hire_df[['Notice.period']])
```

```
c = sc.fit_transform(hire_df[['Pecent.hike.expected.in.CTC']])
d = sc.fit_transform(hire_df[['Percent.hike.offered.in.CTC']])
e = sc.fit_transform(hire_df[['Percent.difference.CTC']])
```

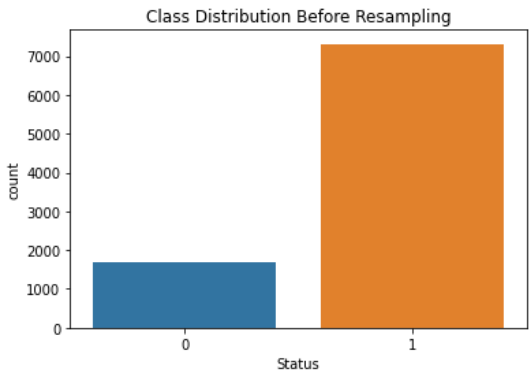
```
X["Duration.to.accept.offer"]=a
X["Notice.period"]=b
X["Pecent.hike.expected.in.CTC"]=c
X["Percent.hike.offered.in.CTC"]=d
X["Percent.difference.CTC"]=e
```

Sampling:

Data sampling is a mathematical analysis process used to select, manipulate and analyze a representative subset of data points to identify patterns and trends in a large data set being tested. It empowers data scientists, speculators and other data analysts to work with a small, manageable amount of data on statistical data to create and use analytics models at a faster rate, while more accurate results are being released.

```
sns.countplot('Status', data=hire_df).set_title('Class Distribution Before Resampling')
```

```
: Text(0.5, 1.0, 'Class Distribution Before Resampling')
```



```
X_no = X[X.Status == 0]
X_yes = X[X.Status == 1]
```

```
print(len(X_no),len(X_yes))
```

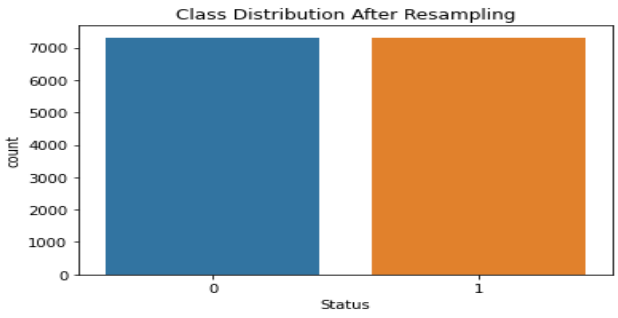
```
1682 7316
```

```
X_no_upsampled = X_no.sample(n=len(X_yes), replace=True, random_state=42)
print(len(X_no_upsampled))
```

```
7316
```

```
X_upsampled = X_yes.append(X_no_upsampled).reset_index(drop=True)
```

```
sns.countplot('Status', data=X_upsampled).set_title('Class Distribution After Resampling')
```



Machine Learning Model:

```
from sklearn.model_selection import train_test_split
```

```
X = X_upsampled.drop(['Status'], axis=1) #features (independent variables)
y = X_upsampled['Status'] #target (dependent variable)
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state=42)
```

Using Random Forest:

Random Forest is a machine learning method used to solve retreat and separation problems. It uses learning together, which is a multi-faceted approach to providing solutions to complex problems.

The random forest algorithm contains a lot of decision trees. The 'forest' created by the random forest algorithm is trained by combining or combining bootstrap. Bagging is an integrated meta-algorithm that improves the accuracy of machine learning algorithms.

The (random forest) algorithm determines the outcome based on the prediction of the decision trees. It predicts by taking or measure the product from different trees. Increasing the number of trees increases the accuracy of the result.

The random forest eliminates the limitations of the decision tree algorithm. It reduces database overload and increases accuracy. It generates predictions without the need for multiple configurations in packages (like scikit-learn).

```
from sklearn.ensemble import RandomForestClassifier from sklearn.metrics import accuracy_score from sklearn.metrics import confusion_matrix
```

```
clf_forest = RandomForestClassifier(n_estimators=150, max_depth=20)
```

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

```
RandomForestClassifier(bootstrap=True, class_weight=None, criterion='gini',
                       max_depth=20, max_features='auto', 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, n_estimators=150,
                       n_jobs=None, oob_score=False, random_state=None,
                       verbose=0, warm_start=False)
```

```
pred = clf_forest.predict(X_train)
```

```
accuracy_score(y_train, pred)
```

```
0.9892353695002136
```

Confusion Matrix:

Confusion Matrix is the N x N matrix used to evaluate the performance of a division model, in which N is the number of target classes. The matrix compares actual identification values with those predicted by machine learning model. This gives us a complete idea of how well our classification model works and what types of errors it makes

```
confusion_matrix(y_train, pred)
```

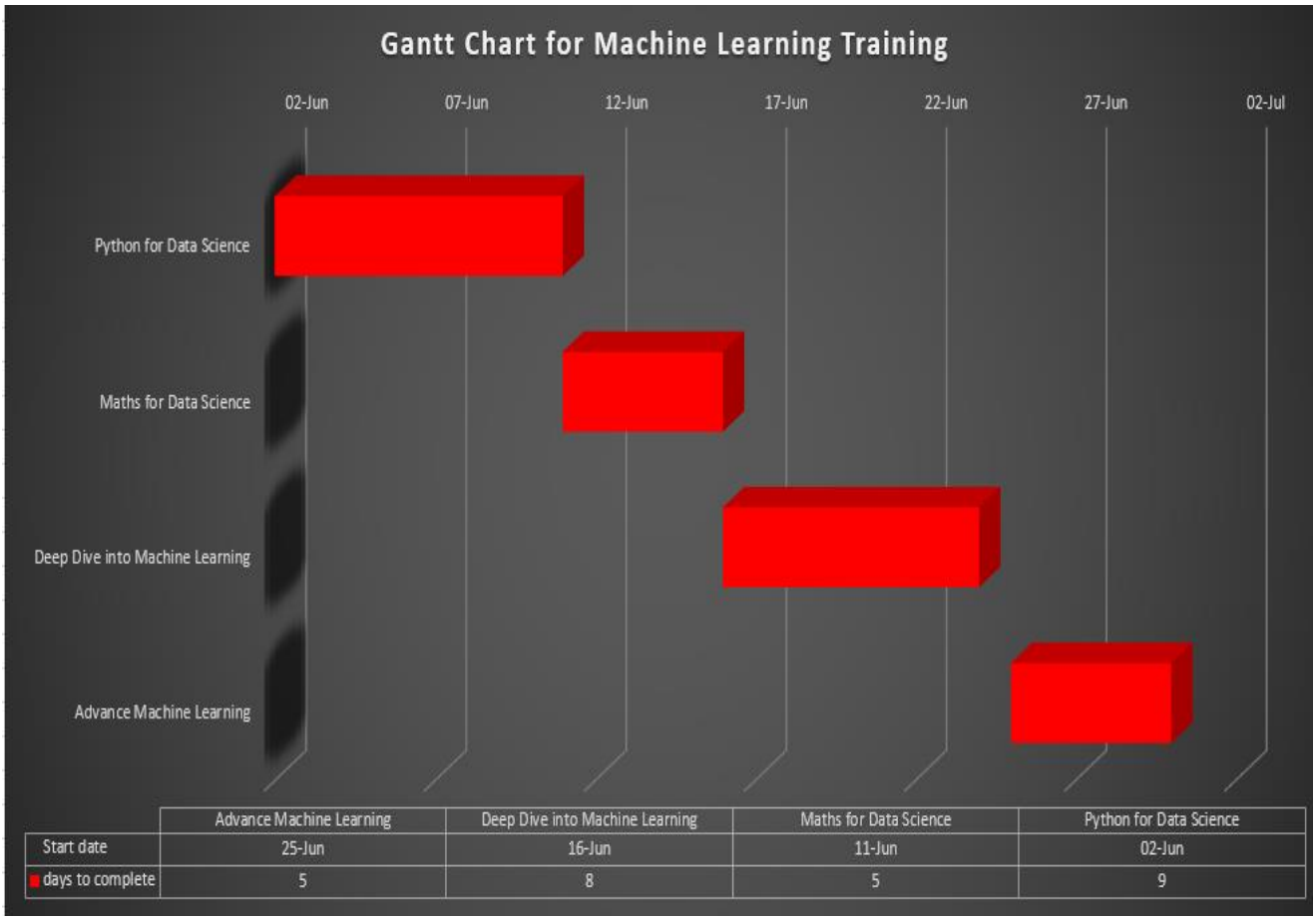
```
array([[5841,    6],
       [ 120, 5738]], dtype=int64)
```

```
pred_test = clf_forest.predict(X_test)
```

```
accuracy_score(y_test, pred_test)
```

```
0.9169798428425009
```

Gantt Chart for Machine Learning Training :



	Advance Machine Learning	Deep Dive into Machine Learning	Maths for Data Science	Python for Data Science
Start date	25-Jun	16-Jun	11-Jun	02-Jun
days to complete	5	8	5	9

Final chapter - Learning outcomes:-

During the training I learned about -

- ✓ Data wrangling in Python using NumPy and Pandas.
- ✓ Linear Regression
- ✓ Logistic Regression
- ✓ Decision Trees
- ✓ Random Forests
- ✓ Visualization using Matplotlib.
- ✓ NLP and Text Mining
- ✓ Deep Learning
- ✓ Neural networks
- ✓ cleaning the data to make it suitable for training the model.
- ✓ visualizing the different variables of the data sets using Python
- ✓ removing overfitting and underfitting issues
- ✓ Model analysis

References:

- https://matplotlib.org/
- https://pandas.pydata.org/
- https://numpy.org/
- https://www.geeksforgeeks.org/
- https://www.boardinfinity.com/
- https://www.kaggle.com/
- https://www.section.io/
- https://www.wikipedia.org/
- https://machinelearningmastery.com/