Date	17 November 2022
Team ID	PNT2022TMID06047
Project Name	Project - Corporate Employee Attrition Analytics
Team Members	Team Lead – Sriman Balaji. R Member 1 – Srishuwetha. A. M Member 2 – Sandeepkumar. S Member 3 – Suraj Kumar. R

1. INTRODUCTION

Corporate companies and Industries are the two main parts that support to the progress and growth of the Country. Workforce or manpower are very important constituent of an organization. The performance and the growth of the company depend on the sustainability of the employees. Attrition and Retension are two main opposite phenomenon which serves different purposes but the crux of the connection is that one makes way for the other. World markets are becoming competitive with time which has changed the culture of the workspace. The presence of work force , emergent imbalance in the supply and the demand of qualified personnel and increased emphasis on work life balance have created challenges for the company's HR and manager to identify the right person to hire for right position. Attrition and retentions are the two faces that represents the way to identify the business employments trends, overall business growth, motivation and growth. It is observed as in global competitive organization are investing considerable amount of intrest , time and money on the employee

attrition because losing a valid employee negative impact in the form of knowledge value , apprehensive colleagues, lost capital, loss of good name of the organization and leads to the failure of the company or organization.

1.1 PROJECT OVERVIEW

The key to success to the organization is attracting and retaining top talented. As an HR it is one of the key tasks to determine which factor keeps the employee in the company and which prompts others to leave. Given in the data is a set of data points on the employees who are either currently working within or have resigned the company. The objective is to identify and improve these factors to prevent hte loss of good people in the organization.

1.2 PURPOSE

- To analyze the factors that causes the employee attrition through predictive analysis and to give suggestions by modelling techniques to reduce the cause of retention.
- Visualization Charts are prepared to highlight the insights for the given dataset

- Creating dashboard for the HR and managers for understanding the reasons for attrition and to take necessary measures in the organization.

2. LITERATURE SURVEY

Employee attrition is referred as reduction in number of employees in an organization. For Corporate industry, employee attrition has become a known challenge since last two decades. Employees leave the organization for various reasons. A few reasons are, demand of high salary, change in technology or role, professional challenges etc. High attrition leads to expense over multiple attributes and functions in the company. Recruitment, Training and Development costs increases overall cost on the employees.

2.1 EXISTING PROBLEM

In recent years, the employer and employee both have lost belief in each other. The former feels that employee can leave the organization anytime and the latter apprehends that he or she can be expelled anytime by the former one. Whosoever is responsible, irrespective of this; loss of workforce is inevitable. This loss of workforce for any reason is called attrition. Irrespective of the kind

of industry or the structure of the organization, attrition is a common problem in every organization which not only hampers production but also results in heavy long-run costs and loss of goodwill to the organization. Therefore, there arises a need to delve into this multi-dimensional problem and come out with feasible solutions

2.2 REFERENCES

Attrition Defined Attrition, in Human Resource Management (HRM), refers to the situation of employees leaving the company. It is measured with a measuring unit called attrition rate, which calculates the number of employees leaving the company (either resigning voluntarily or involuntarily laid off by the company). (www.mbaskool.com/business-concepts/human-resources, 2013).

Employee attrition & retention is manifestation of employee movement in an organization, which is deliberated by researcher in HR. They are two sides of same coin. Employee attrition & retention may be result of the negative or positive influence of the various factors (Zhang, 2005).

According to Cascio and W.Bourdeau (2008) voluntary attrition happens when an employee resigns an organization to grab another career opportunity, he may relocate with his family to different place or simply leave the organization for his

personal reasons. Retirement is one biggest form of voluntary turnover, i.e leaving a job at his own will. Voluntary turnover is a serious issue for modern organizations these days because experienced and intellectual capital is increasingly critical for sustained competitiveness (Boudreau & Ramstad, 2007; Lepak & Snell, 2002).

Corey Harris (Walden University 2018) researched on “Employee Retention Strategies in the Information Technology Industry” and mentioned that “Productivity declines when employees voluntarily leave an organization” Dr. Shikha N. Khera1 , Ms. Karishma Gulati2 (Delhi 2012), concluded in their study on “Human Resource Information System and its impact on Human Resource Planning: A perceptual analysis of Information Technology companies” that Being an information system of human resources, it can store voluminous data about the employees, that not only helps in identifying the occupied and unoccupied positions but also whether the person at particular position is fit for the job or not.

Hardik P. K. (2016) , researched on “a study on employee attrition: with special reference to Kerala IT Industry”. His research examined the relationship between organizational factors and attrition of IT professional’s The result can conclude that the organizational factors played significant role in predicting the variance in turnover intention (attrition) of Kerala IT professionals. Therefore,

the HR mangers in IT organizations may take into consideration the problems with organizational factors of their workers to reduce the turnover intention of the skilled employees.

Bodjrenou Kossivi, Ming Xu, Bomboma Kalgora (May 2016) published “Study on Determining Factors of Employee Retention”. The study concluded: Employees are the most valuable assets of an organization. Their significance to organizations calls for not only the need to attract the best talents Mukt Shabd Journal Volume IX, Issue VII, JULY/2020 ISSN NO : 2347-3150 Page No : 2752 but also the necessity to retain them for a long term. Broad factors are development opportunities, compensation, work-life balance, management/leadership, work environment, social support, autonomy, training, and development.

Brijesh Kishore Goswami, Sushmita Jha (April 2012) in their study on “Attrition Issues and Retention Challenges of Employees” have stated that, Organizations planning should be giving close attention to why attrition is occurring in the pre-set. To ignore why people are leaving the organization is to ignore the organization’s greatest asset – its people. People are needed to accomplish the task, but people are more than just tasks they perform. They are dreams, hopes, ambitions, creativity, and innovation.

To recognize and cultivate these valuable assets is one of the surest ways to build an organization that leads rather than follows in domestic and global

markets. Thus, Organizations should create an environment that fosters ample growth opportunities, appreciation for the work accomplished and a friendly cooperative atmosphere that makes an employee feel connected in every respect to the organization.

Retention plans are an inexpensive way of enhancing workplace productivity and engaging employees emotionally. Proficient employees keep the quality up and business operations run smoothly along with the cost saving in the longer run paper.

S.Guru Vignesh, V.Sarojini, S.Vetrive (Jan 2018),in “Employee Attrition and Employee RetentionChallenges & Suggestions” state that, retention plans are an inexpensive way of enhancing workplace productivity and engaging employees emotionally. Proficient employees keep the quality up and business operations run smoothly along with the cost saving in the longer run.

2.3 PROBLEM STATEMENT DEFINITION

Problem Statement

Problem	• A large company named XYZ, employs, at any given point of time, around 4000 employees. However, every year, around 15% of its employees leave the company and need to be replaced with the talent pool available in the job market. The management believes that this level of attrition (employees leaving, either on their own or because they got fired) is bad for the company
Current situation	• 1.The former employees' projects get delayed, which makes it difficult to meet timelines, resulting in a reputation loss among consumers and partners • 2.A sizeable department has to be maintained, for the purposes of recruiting new talent • 3.More often than not, the new employees have to be trained for the job and/or given time to acclimatise themselves to the company
Corporate consern	• the management has contracted an HR analytics firm to understand what factors they should focus on, in order to curb attrition. In other words, they want to know what changes they should make to their workplace, in order to get most of their employees to stay.
Needed...	• they want to know which of these variables is most important and needs to be addressed right away
Solution needed..	• You are required to model the probability of attrition using a logistic regression. The results thus obtained will be used by the management to understand what changes they should make to their workplace, in order to get most of their employees to stay.

A large company named XYZ, employs, at any given point of time, around 4000 employees. However, every year, around 15% of its employees leave the company and need to be replaced with the talent pool available in the job market. The management believes that this level of attrition (employees leaving, either on their own or because they got fired) is bad for the company, because of the following reasons -

1. The former employees' projects get delayed, which makes it difficult to meet timelines, resulting in a reputation loss among consumers and partners
2. A sizeable department has to be maintained, for the purposes of recruiting new talent
3. More often than not, the new employees have to be trained for the job and/or given time to acclimatise themselves to the company

Hence, the management has contracted an HR analytics firm to understand what factors they should focus on, in order to curb attrition. In other words, they want to know what changes they should make to their workplace, in order to get most of their employees to stay. Also, they want to know which of these variables is most important and needs to be addressed right away.

Since you are one of the star analysts at the firm, this project has been given to you.

Goal of the case study

You are required to model the probability of attrition using a logistic regression. The results thus obtained will be used by the management to understand what changes they should make to their workplace, in order to get most of their employees to stay.

3. IDEATION & PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS

3.2 IDEATION & BRAINSTORMING

Step-1: Team Gathering, Collaboration and Select the Problem Statement

Step-2: Brainstorm, Idea Listing and Grouping

Step-3: Idea Prioritization

3.3 PROPOSED SOLUTION

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	A large company named XYZ, employs, at any given point of time, around 4000 employees. However, every year, around 15% of its employees leave the company and need to be replaced with the talent pool available in the job market. The management believes that this level of attrition (Employees leaving, either on their own or because they got fired) is bad for the company, because of the following reasons - 1. The former employees' projects get delayed, which makes it difficult to meet timelines, resulting in a reputation loss among consumers and partners 2. A sizeable department has to be maintained, for the purposes of recruiting new talent 3. More often than not, the new employees have to be trained for the job and/or given time to acclimatise themselves to the company Hence, the management has contracted an HR analytics firm to understand what factors they should focus on, in order to curb attrition. In other words, they want to know what changes they should make to their workplace, in order to get most of their employees to stay. Also, they want to know which of these variables is most important and needs to be addressed right away

2.	Idea / Solution description	You are required to model the probability of attrition using a logistic regression. The results thus obtained will be used by the management to understand what changes they should make to their workplace, in order to get most of their employees to stay
3.	Novelty / Uniqueness	The solution will give idea or changes that they should make to their workplace, in order to get most of their employees to stay. Also, they will come to know which of these variables is most important and needs to be addressed right away
4.	Social Impact / Customer Satisfaction	1. The former employees' projects will not be delayed, which makes it to produce on time, resulting in a good reputation among consumers and partners 2. A sizeable department will be maintained, for the purposes of recruiting new talent 3. the new employees will be trained for the job and/or given time to acclimatise themselves to the company
5.	Business Model (Revenue Model)	If there is no attrition in the company, then the revenue and the profit of the company gets increased.
6.	Scalability of the Solution	Analysis and Models will be helpful in understanding the reason for attrition and the steps to be taken by the company to reduce it

3.4 PROBLEM SOLUTION FIT

4. REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENTS:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Dataset	Sign up through Kaggle Registration through Gmail
FR-2	Uploading the dataset	Sign up into IBM Cloud Account Invitation through mail id Sign up into IBM Cognos Analytics Invitation through mail id

FR-3	Data Visualization Charts	Sign up into IBM Cognos Analytics Invitation through mail id
FR-4	Coding	Sign in Jupyter Python coding - Jupyter
FR-4	Modelling and testing	Sign in Python Direct Collaboration through google collabs

4.2 NON-FUNCTIONAL REQUIREMENTS:

Following are the non-functional requirements of the proposed solution.

FR No.	**Non-Functional Requirement**	**Description**
NFR-1	**Usability**	The dataset is obtained from the external sources must be safe and recommended for analysis
NFR-2	**Security**	Organizations must protect their most critical business assets—your data—against unauthorized or unwanted use. They must combine people, processes, and technology to protect data throughout its lifecycle. Use a unified platform that integrates data security information across your entire enterprise and that ensures scalability on environments of any size across public cloud, on-premises, and hybrid cloud deployment

NFR-3	**Reliability**	The analysis gives suggestions and steps that can be carried to whole company's attrition problem, as a long-time solution
NFR-4	**Performance**	The performance of the analysis must be solving the problem fully, so that it gives a permanent solution to the problem faced
NFR-5	**Availability**	The dataset is analysed and solution is given to the problem faced and the solution must be available for the full process
NFR-6	**Scalability**	Data is growing at an exponential rate. Keeping up with new data sources across environments creates complexity at an unprecedented scale

5. PROJECT DESIGN

5.1 DATA FLOW DIAGRAMS

1. User configures credentials for the Watson Natural Language Understanding service and starts the app.
2. User selects data file to process and load.
3. Extracted text is passed to Watson NLU for enrichment.

4. Enriched data is visualized in the UI using the D3.js library
5. Python cloud is used for coding
6. The outcomes are analysed and modelled in Python
7. The external data is obtained as solution

5.2 SOLUTION & TECHNICAL ARCHITECTURE

SOLUTION ARCHITECTURE

TECHNICAL ARCHITECTURE

S.No	Component	Description	Technology
	User Interface Web UI	Web UI is used for the user interaction	HTML, CSS, JavaScript
	Application Logic-1 IBM Watson Cloud Account	IBM Watson® Studio empowers data scientists, developers and analysts to build, run and manage AI models, and optimize decisions anywhere on IBM Cloud Pak® for Data.	human speech for meaning and syntax.
	Application Logic-2 IBM Watson Cognos Analytics	IBM® Cognos® Business Intelligence is an integrated business intelligence suite that provides a wide range of functionality to help you understand your organization's data.	Java. JRE (Java Runtime Environment) to function.
	Application Logic-3 Python	Google is quite aggressive in AI research. Over many years, Google developed AI framework called **TensorFlow** and a development tool called **Colaboratory**. Today TensorFlow is open-sourced and since 2017, Google made Colaboratory free for public use. Colaboratory is now known as Google Colab or simply **Colab**.	Google Colaboratory (Google Colab) is a free cloud-based framework with a Jupyter notebook environment with free access to CPU/GPU/TPU
1.	Application Logic-4 Jupyter	The Jupyter Notebook is the original web application for creating and sharing computational documents. It offers a simple, streamlined, document-centric experience.	Jupyter Notebook is built using several open-source libraries, including IPython, ZeroMQ, Tornado, jQuery, Bootstrap, and MathJax.

6.	External API-1 Dataset	Kaggle website is used to get the dataset. Kaggle allows users to find and publish data sets, explore and build models in a web-based data-science environment, work with other data scientists and machine learning engineers, and enter competitions to solve data science challenges.	Python and C++ Framework: Keras and PyTorch
7.	Database IBM Cloud	With IBM Cloud IaaS, organizations can deploy and access virtualized IT resources -- such as compute power, storage and networking -- over the internet.	IaaS – Infrastructure as a Service

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
	Open-Source Frameworks	The dataset is been obtained from third party Kaggle website. Kaggle website is used to get the dataset. Kaggle allows users to find and publish data sets, explore and build models in a web-based data-science environment, work with other data scientists and machine learning engineers, and enter competitions to solve data science challenges.	Opensource framework - Python and C++ Framework: Keras and PyTorch

	Security Implementations	• Application Logic-1 IBM Watson Cloud Account IBM Watson® Studio empowers data scientists, developers and analysts to build, run and manage AI models, and optimize decisions anywhere on IBM Cloud Pak® for Data. • Application Logic-2 IBM Watson Cognos Analytics IBM® Cognos® Business Intelligence is an integrated business intelligence suite that provides a wide range of functionality to help you understand your organization's data. • Application Logic-3 Python Google is quite aggressive in AI research. Over many years, Google developed AI framework called **TensorFlow** and a development tool called **Colaboratory**. Today TensorFlow is open-sourced and since 2017, Google made Colaboratory free for public use. Colaboratory is now known as Google Colab or simply **Colab** • Application Logic-4 Jupyter The Jupyter Notebook is the original web application for creating and sharing computational documents. It offers a simple, streamlined, document-centric experience.	SHA-256, Encryptions, IAM Controls, OWASP,human speech for meaning and syntax. Java. JRE (Java Runtime Environment) to function. C++, JavaScript, Qt framework for its graphical user interface Google Colaboratory (Google Colab) is a free cloud-based framework with a Jupyter notebook environment with free access to CPU/GPU/TPU Jupyter Notebook is built using

			several opensource libraries, including IPython, ZeroMQ, Tornado,

			jQuery, Bootstrap, and MathJax.
	Scalable Architecture	The Public cloud infrastructure architecture illustrates the IBM Cloud platform, which can be used to support scalable, secure, and resilient workloads. The infrastructure services include networks, compute, storage, security, and management.	Java. JRE (Java Runtime Environment) to function. C++, JavaScript, Qt framework for its graphical user interface

	Availability	IBM® Data Replication for Availability enables high-speed data replication for business continuity across Db2® databases, Db2 Warehouse, Db2 Warehouse in IBM Integrated Analytics System appliances, and Db2 Warehouse on Cloud. The software enables continuous availability, including disaster recovery by synchronizing transactions over both row- and column-organized tables, whether on the same platform, across the data center, or around the world in an active-active configuration. It offers near real-time asynchronous data replication from a primary database server to one or more standby replicas for workload balancing or shifting workloads during planned outages, while also dramatically reducing the time to recovery for unplanned outages.	C++, JavaScript
	Performance	progressive employers should be mindful of the ethical standards they adhere to while utilizing this information. Collecting and analyzing workforce data without appropriatecommunication and purpose may cause unease and distrust among employees	HTML, CSS, JavaScript human speech for meaning and syntax.

5.3 USER STORIES

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can registerIBM Cloud Account by entering my email, password, and confirming my password. As a user, I will receive confirmation email once I have registered for the application	I can access my account / dashboard I can receive confirmation email & click confirm	Low	Sprint-1

Customer (Mobile user)	Registration	USN-2	As a user, I can register IBM Cognos Analytics -by entering my email, password, and confirming my password As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm.I can access my account / dashboard	High	Sprint-2
Customer (Mobile user)	Registration	USN-3	Jupyter is signed in through google collabs	I can receive confirmation email & click confirm.I can access my account / dashboard	High	Sprint-2
Customer Third party	Login	USN-4	As a user, I can register for the application through Kaggle	I can register & access the dashboard with Gmail Login.I can receive confirmation email & click confirm	Medium	Sprint-3
Customer Cloud	Login and Register	USN-5	As a user, I can register for the python through Gmail	I can register & access the dashboard with Gmail Login. I can receive confirmation email & click confirm	High	Sprint-4

6. PROJECT PLANNING & SCHEDULING

6.1 SPRINT PLANNING & ESTIMATION

Product Backlog, Sprint Schedule, and Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members

Sprint-1	To model the probability of attrition using logistic regression	USN-1	As a user, I can register for the application by entering my email, through google collabs	2	High	1)Sriman Balaji. R 2)Srishuwetha. A. M 3)Sandeepkuamr. S 4)Suraj Kumar. R
Sprint-1	Business understanding, importing packages and understanding the data	USN-2	As a user, I can register for the application by entering my email, through google collabs	1	High	1)Sriman Balaji. R 2)Srishuwetha. A. M 3)Sandeepkuamr. S 4)Suraj Kumar. R
Sprint-2	Data Understanding & Data preparation	USN-3	As a user, I can register for the application by entering my email, through google collabs	2	Low	1)Sriman Balaji. R 2)Srishuwetha. A. M 3)Sandeepkuamr. S 4)Suraj Kumar. R
Sprint-3	Data Understanding And Data preparation	USN-4	As a user, I can register for the application by entering my email, through google collabs	2	Medium	1)Sriman Balaji. R 2)Srishuwetha. A. M 3)Sandeepkuamr. S 4)Suraj Kumar. R

Sprint-4	EDA, Model Building and Model Evaluation	USN-5	As a user, I can register for the application by entering my email, through google collabs	1	High	1)Sriman Balaji. R 2)Srishuwetha. A. M 3)Sandeepkuamr. S 4)Suraj Kumar. R

Project Tracker, Velocity & Burndown Chart:

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

Velocity:

AV = sprint duration / velocity = 20/10 = 2

BURNDOWN CHART:

VELOCITY CHART:

6.2 SPRINT DELIVERY SCHEDULE

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

6.3 REPORTS FROM JIRA

BURNDOWN CHART:

VELOCITY CHART:

WORK REPORT:

START
CORPORATE EMPLOYEE ATTRITION ANALYSIS
TO DO
ACCEPTANCE TESTING
IN - PROGRESS
DELIVERY OF SPRINT 4
DONE
DELIVERY OF SPRINT 3
DELIVERY OF SPRINT 2
DELIVERY OF SPRINT 1
PROJECT DEVELOPMENT PHASE
MILESTONES
SPRINT DELIVERY PLAN
PROJECT PLANNING PHASE
CUSTOMER JOURNEY MAP
DATA FLOW DIAGRAM & USER STORIES
SOLUTION REQUIRED
TECHNOLOGY STACK (STACK &ARCHITECTURE
PROJECT DESIGN PHASE -II
PROPOSED SOLUTION
PROBLEM SOLUTION FIT
SOLUTION ARCHITECTURE
PROJECT DESIGN PHASE - I
LITERATURE SURVEY
EMPATHY MAP
PROBLEM STATEMENT
BRAINSTROMING IDEAS
IDEATION PHASE

7. CODING & SOLUTIONING

7.1 Feature 1

DATASET :

- Employee Attrition Analysis (Logistic Regression Model)

Employee Attrition Analysis (Logistic Regression Model)

https://www.kaggle.com/vjchoudhary7/hr-analytics-case-study

DATA UNDERSTANDING:

The data received for the analysis can be divided into 4 broad categories -

- General Data – General data, acquired from HR
- Employee Survey Data – Data collected from yearly employee survey
- Manager Survey Data – Data collected from yearly manager survey
- Biometric Data – Daily in and out times for each employee, collected using biometric attendance machines

General Data
Age
Attrition (Yes/No)
Department
Education Field

Manager Survey Data
Job Involvement
Performance Rating

Employee Survey Data
Environment Satisfaction
Job Satisfaction
Work Life Balance

Biometric Data
In Time
Out Time

UNDERSTANDING THE DATASET:

Let us try to understand each field of the data (general_data.csv)

Below are the values each column has. The column names are pretty self-explanatory.

1. AGE Numerical Value
2. ATTRITION Employee leaving the company (0=no, 1=yes)
3. BUSINESS TRAVEL (1=No Travel, 2=Travel Frequently, 3=Travel Rarely)
4. DEPARTMENT (1=HR, 2=R&D, 3=Sales)

5. DISTANCE FROM HOME Numerical Value - THE DISTANCE FROM WORK TO HOME
6. EDUCATION Numerical Value. (1 'Below College' 2 'College' 3 'Bachelor' 4 'Master' 5 'Doctor')
7. EDUCATION FIELD (1=HR, 2=LIFE SCIENCES, 3=MARKETING, 4=MEDICAL SCIENCES, 5=OTHERS, 6= TECHNICAL)
8. EMPLOYEE COUNT Numerical Value
9. EMPLOYEE ID Numerical Value
10. GENDER (1=FEMALE, 2=MALE)
11. JOB LEVEL Numerical Value
12. JOB ROLE (1=HR REP, 2=HR, 3=LAB TECHNICIAN, 4=MANAGER, 5= MANAGING DIRECTOR, 6= RESEARCH DIRECTOR, 7= RESEARCH SCIENTIST, 8=SALES EXECUTIVE, 9= SALES REPRESENTATIVE)
13. MARITAL STATUS (1=DIVORCED, 2=MARRIED, 3=SINGLE)
14. MONTHLY INCOME Numerical Value - MONTHLY SALARY
15. NUMCOMPANIES WORKED Numerical Value - NO. OF COMPANIES WORKED AT
16. OVER 18 (1=YES, 2=NO)
17. PERCENT SALARY HIKE Numerical Value - PERCENTAGE INCREASE IN SALARY
18. STANDARD HOURS Numerical Value - STANDARD HOURS
19. STOCK OPTIONS LEVEL Numerical Value - STOCK OPTIONS (Higher the number, the more stock option an employee has)
20. TOTAL WORKING YEARS Numerical Value - TOTAL YEARS WORKED
21. TRAINING TIMES LAST YEAR Numerical Value - HOURS SPENT TRAINING
22. YEARS AT COMPANY Numerical Value - TOTAL NUMBER OF YEARS AT THE COMPANY
23. YEARS SINCE LAST PROMOTION Numerical Value - LAST PROMOTION
24. YEARS WITH CURRENT MANAGER Numerical Value - YEARS SPENT WITH CURRENT MANAGER

b. Let us try to understand about each field of the data (employee_survey_data.csv)

1. Employee ID
2. Environment Satisfaction (1 'Low' 2 'Medium' 3 'High' 4 'Very High')
3. Job Satisfaction (1 'Low' 2 'Medium' 3 'High' 4 'Very High')
4. Work Life Balance (1 'Bad', 2 'Good', 3 'Better', 4 'Best')

c. Let us try to understand about each field of the data (manager_survey_data.csv)

1. Employee ID
2. Job Involvement (1 'Low' 2 'Medium' 3 'High' 4 'Very High')
3. Performance Rating (1 'Low', 2 'Good', 3 'Excellent', 4 'Outstanding')

SOLUTION REQUIRED:

- To model the probability of attrition using a logistic regression
- Business Understanding
- Data Understanding – sources of the data, meaning of the data
- Data preparation & EDA
- Model Building
- Model Evaluation
- Data Visualization charts
- Dashboard Creation

METHODOLOGY USED:

- Predictive modelling of attrition
- Recommending ways for company XYZ to decrease its level of attrition

LIBRARIES USED FOR THIS CODE:

ibm corporate employee attrition analysis.ipynb

File Edit View Insert Runtime Tools Help Last saved at 8:18 PM

Comment Share

+ Code + Text

Connect Editing

```
from sklearn import svm
from sklearn.linear_model import LogisticRegression
from  sklearn.svm import SVC
from sklearn.metrics import log_loss, make_scorer
import math
from sklearn.metrics import roc_auc_score
from sklearn.metrics import auc
from sklearn.metrics import accuracy_score
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report
from sklearn.metrics import roc_curve
from sklearn.metrics.classification import accuracy_score, log_loss
from sklearn.calibration import CalibratedClassifierCV
from prettytable import PrettyTable
```

Data Present in Google DRive MOunting to Google Collab

```
[ ] from google.colab import drive
    drive.mount('/content/drive')

    #!ls "/content/drive/My Drive/Kaggle_dataset/HR_analytics/PA-I_Case_Study_HR_Analytics"

    !cp "/content/drive/My Drive/Kaggle_dataset/HR_analytics/PA-I_Case_Study_HR_Analytics/general_data.csv" "general_data.csv"
    !cp "/content/drive/My Drive/Kaggle_dataset/HR_analytics/PA-I_Case_Study_HR_Analytics/employee_survey_data.csv" "employee_survey_data.csv"
    !cp "/content/drive/My Drive/Kaggle_dataset/HR_analytics/PA-I_Case_Study_HR_Analytics/manager_survey_data.csv" "manager_survey_data.csv"
    !cp "/content/drive/My Drive/Kaggle_dataset/HR_analytics/PA-I_Case_Study_HR_Analytics/in_time.csv" "in_time.csv"
    !cp "/content/drive/My Drive/Kaggle_dataset/HR_analytics/PA-I_Case_Study_HR_Analytics/out_time.csv" "out_time.csv"
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
```

```
import warnings
warnings.filterwarnings("ignore")
import re  import os
import datetime as dt

import numpy as np  import
pandas as pd  import
matplotlib.pyplot as plt  import
seaborn as sns  import csv

from sklearn.feature_extraction.text import CountVectorizer
from sklearn.linear_model import SGDClassifier  from sklearn
import metrics
from  sklearn.metrics  import  f1_score,precision_score,recall_score
from sklearn import svm
from sklearn.linear_model import LogisticRegression  from
sklearn.svm import SVC
from sklearn.metrics import log_loss, make_scorer  import
math
from sklearn.metrics import roc_auc_score  from
sklearn.metrics import auc
from sklearn.metrics import accuracy_score  from
sklearn.metrics import confusion_matrix  from
sklearn.metrics import classification_report  from
sklearn.metrics import roc_curve
```

```
from sklearn.metrics.classification import accuracy_score,
log_loss  from sklearn.calibration import CalibratedClassifierCV
from prettytable import PrettyTable
```

TO READ ALL THE CSV FILES:

CODING:

```
from google.colab import drive  drive.mount('/content/drive')

#!ls "/content/drive/My
    Drive/Kaggle_dataset/HR_analytics/PAI_Case_Study_HR_Analytics"

!cp "/content/drive/My Drive/Kaggle_dataset/HR_analytics/PA-
    I_Case_Study_HR_Analytics/general_data.csv" "general_data.csv"
!cp "/content/drive/My Drive/Kaggle_dataset/HR_analytics/PA-
    I_Case_Study_HR_Analytics/employee_survey_data.csv"
    "employee_survey_data.csv"
!cp "/content/drive/My Drive/Kaggle_dataset/HR_analytics/PA-
    I_Case_Study_HR_Analytics/manager_survey_data.csv"
    "manager_survey_data.csv"
!cp                                                    "/content/drive/My
    Drive/Kaggle_dataset/HR_analytics/PAI_Case_Study_HR_Analytics/in_
    time.csv" "in_time.csv"
!cp "/content/drive/My Drive/Kaggle_dataset/HR_analytics/PA-

    I_Case_Study_HR_Analytics/out_time.csv" "out_time.csv"
```

OUTPUT:

```
Drive already mounted at /content/drive; to attempt to forcibly
    remount, call drive.mount("/content/drive", force_remount=True).
```

READING AND UPLOADING CSV FILES

CODING:

```
generalData_df     =  pd.read_csv("general_data.csv")
employee_survey_df =
pd.read_csv("employee_survey_data.csv")
manager_survey_df  =
pd.read_csv("manager_survey_data.csv")
intime_df          =   pd.read_csv("in_time.csv")
outtime_df         =  pd.read_csv("out_time.csv")

print(generalData_df.shape ,  employee_survey_df.shape
    ,  manager_survey_df.shape)
print(employee_survey_df.columns.tolist())
print(manager_survey_df.columns.tolist())
```

OUTPUT:

```
4410, 24) (4410, 4) (4410, 3)
['EmployeeID', 'EnvironmentSatisfaction', 'JobSatisfaction', 'WorkLifeBalance']
['EmployeeID', 'JobInvolvement', 'PerformanceRating'
```

MERGING OF DATA:

```
df1=pd.merge(generalData_df,employee_survey_df,on='EmployeeID')
final_df=pd.merge(df1,manager_survey_df,on='EmployeeID')

if(len(final_df.columns.tolist())  == (len(generalData_df.columns.tolist()) +
    len(employee_survey_df.columns.tolist()) + len(manager_survey_df.columns.tolist())) -
    2):
```

```
print("OK")
```

OUTPUT:

```
OK
```

CODING:

	Age	Attrition	BusinessTravel	Department	DistanceFromHome	Education	EducationField	EmployeeCount	EmployeeID	Gender	JobLevel	JobRole	MaritalStatus	MonthlyIncome	Num
0	51	No	Travel_Rarely	Sales	6	2	Life Sciences	1	1	Female	1	Healthcare Representative	Married	131160	
1	31	Yes	Travel_Frequently	Research & Development	10	1	Life Sciences	1	2	Female	1	Research Scientist	Single	41890	
2	32	No	Travel_Frequently	Research & Development	17	4	Other	1	3	Male	4	Sales Executive	Married	193280	
3	38	No	Non-Travel	Research & Development	2	5	Life Sciences	1	4	Male	3	Human Resources	Married	83210	
4	32	No	Travel_Rarely	Research & Development	10	1	Medical	1	5	Male	1	Sales Executive	Single	23420	

```
final_df.head()
```

OUTPUT:

```
AgeAttritionBusinessTravelDepartmentDistanceFromHomeEducationEducationFieldEmpl
    oveeCountEmployeeIDGenderJobLevelJobRoleMaritalStatusMonthlyIncomeNumComp
    aniesWorkedOver18PercentSalaryHikeStandardHoursStockOptionLevelTotalWorking
    YearsTrainingTimesLastYearYearsAtCompanyYearsSinceLastPromotionYearsWithCur
    rManagerEnvironmentSatisfactionJobSatisfactionWorkLifeBalanceJobInvolvementPer
    formanceRating051NoTravel  RarelySales62Life Sciences11Female1Healthcare
    RepresentativeMarried1311601.0Y11801.061003.04.02.033131YesTravel  Frequently
    Research & Development101Life Sciences12Female1Research
    ScientistSingle418900.0Y23816.035143.02.04.024232NoTravel  FrequentlyResearch
    & Development174Other13Male4Sales
    ExecutiveMarried1932801.0Y15835.025032.02.01.033338NoNon-TravelResearch &
    Development25Life Sciences14Male3Human
    ResourcesMarried832103.0Y118313.058754.04.03.023432NoTravel  RarelyResearch
    & Development101Medical15Male1Sales
    ExecutiveSingle234204.0Y12829.026044.01.03.033
```

FINDING NULL'S

CODING:

```
print("finding whether is there is any nulls in the final_df  if
    yes                    how                    many
    {}".format(final_df.isnull().sum().sum()))
final_df.dropna(inplace=True)
print("finding whether is there is any nulls in after droping the
    final_df                if          yes          how          many
    {}".format(final_df.isnull().any().sum()))
```

OUTPUT:

```
finding whether is there is any nulls in the final_df  if yes how many  111  finding
    whether is there is any nulls in after droping the  final_df  if yes how many  0
```

CHECKING WHETHER DUPLICATES PRESENT IN FINAL_DF

CODING:

```
final_df_dupliactedRows=final_df[final_df.duplicated( keep=False)].shape[0]
if(final_df_dupliactedRows==0):
  print("there is No duplicate elements in
Final_df")   else:    print("there is  duplicate
elements in Final_df")
```

OUTPUT:

```
there is No duplicate elements in Final_df
```

EDA

We will cover how to visually analyse:

- Numerical variables with histograms,
- Categorical variables with count plots,
- Relationships between numerical variables with scatter plots, joint plots, and pair plots, and
- Relationships between numerical and categorical variables with box-and-whisker plots and complex conditional plots

UNIVARIANT ANALYSIS

FOR ALL COLUMNS PLOTTING COUNT PLOTS

CODING:

```
#For all columns doing countplots
```

```
fig = plt.figure(figsize=(26,30))   for idx,i in
enumerate(final_df.columns.tolist()):
ax=plt.subplot(8,4,idx+1)
   sns.countplot(x=i,data=final_df,ax=ax)
plt.show()
```

OUTPUT:

FOR NUMERICAL COLUMNS UNIVARIANT ANALYSIS IS DONE

CODING:

```
#Divide into NumericalColumns
columns=final_df.columns.tolist()
num_col_eda=['Age','DistanceFromHome','PercentSalaryHike','MonthlyIncome','TotalWorking
    Years','YearsAtCompany']
final_df[num_col_eda].hist(figsize=(10,10))
plt.show()
```

OUTPUT:

INFERENCES

Key Observation from Above Plot are

- Except Age most of the Columns are in Skew Distribution form
- Age Feature Distribution is almost Normal Distribution
- As logistic regression does not require independent variables to be normal distributed .so i am not changing distribution of features which are skewed into the normal Distribution

DIVIDING CATEGORICAL COLUMNS INTO ORDINAL_COLUMNS AND NOMINAL_COLUMNS AND

PROCEDDING UNIVARIANT ANALYSIS FOR THEM CODING:

```
#Divide into CategoricalColumns
num_attrition=num_col_eda+['Attrition']
Education=['Below-College','College','Bachelor','Master','Doctor']
EnvironmentSatisfaction=['Low','Medium','High','Very High']
JobInvolvement=['Low','Medium','High','Very High']
JobSatisfaction=['Low','Medium','High','Very High']
RelationshipSatisfaction=['Low','Medium','High','Very High']
PerformanceRating=['Low','Good','Excellent','Outstanding']
WorkLifeBalance=['Bad','Good','Better','Best']

val=[Education,EnvironmentSatisfaction,JobInvolvement,JobSatisfaction,PerformanceRating,
      WorkLifeBalance]
cat1=['Education','EnvironmentSatisfaction','JobInvolvement','JobSatisfaction','PerformanceR
ating','WorkLifeBalance']

cat_col_eda=set(columns) -
set(num_col_eda)
cat_col_eda=set(cat_col_eda) - set(cat1)   fig
= plt.figure(figsize=(16,20))
```

FOR ORDINAL COLUMNS UNIVARIANT ANALYSIS CODING:

```
#2. chaning nums as x_ticks to   categorys as
x_ticks   fig = plt.figure(figsize=(30,28))   for idx,i
in enumerate(zip(cat1,val)):
   #crosstab = pd.crosstab(index=final_df[i[0]], columns=final_df["Attrition"])
ax=plt.subplot(6,4,idx+1)
   #crosstab.plot(kind="bar",stacked=True,ax=ax)
#sns.countplot(x=i[0],data=final_df,ax=ax)
sns.countplot(final_df[i[0]],hue=final_df['Attrition'],ax=ax)
ax.set(xticks=range(len(i[1])), xticklabels=[j for j in i[1]])   plt.show()
```

OUTPUT:

OUTPUT:

```
there is no paricular distinction
```

FOR NOMINAL_COLUMN DOING UNIVARIANT ANALYSIS CODING:

```
#Divide into CategoricalColumns   #1. with categorys as x_ticks   fig =
plt.figure(figsize=(20,26))   for idx,i in enumerate(cat_col_eda):
crosstab = pd.crosstab(index=final_df[i], columns=final_df["Attrition"])
ax=plt.subplot(6,4,idx+1)
crosstab.plot(kind="bar",stacked=True,ax=ax)
   #sns.countplot(x=i[0],data=final_df,ax=ax)
   #ax.set(xticks=range(len(i[1])), xticklabels=[j for j in i[1]])
plt.show()
```

OUTPUT:

CODING:

```
#2. chaning nums as x_ticks to   categorys as x_ticks   fig =
plt.figure(figsize=(20,26))   for idx,i in enumerate(zip(cat1,val)):
crosstab = pd.crosstab(index=final_df[i[0]], columns=final_df["Attrition"])
ax=plt.subplot(6,4,idx+1)
crosstab.plot(kind="bar",stacked=True,ax=ax)
#sns.countplot(x=i[0],data=final_df,ax=ax)
   ax.set(xticks=range(len(i[1])), xticklabels=[j for j in i[1]])
plt.show()
```

OUTPUT:

CODING:

```
#for i in num_attrition:
#crosstab = pd.crosstab(index=final_df[i], columns=final_df["Attrition"])
#num_attrition=num_col_eda+['Attrition']
fig = plt.figure(figsize=(20,26))   for idx,i in enumerate(num_attrition):
crosstab = pd.crosstab(index=final_df[i],
columns=final_df["Attrition"])    ax=plt.subplot(6,4,idx+1)
   #sns.boxplot(x=i,data=final_df,ax=ax)

 crosstab.plot(kind="bar",stacked=True,ax=ax)
```

OUTPUT:

FINDING OUTLIER IN NUMERICAL DATA CODING:

```
#Box Plot for finding "Outiler" in our
data  fig = plt.figure(figsize=(20,26))
for idx,i in enumerate(num_col_eda):
ax=plt.subplot(6,4,idx+1)
   sns.boxplot(x=i,data=final_df,ax=ax)
```

OUTPUT:

RESULTS FROM ABOVE GRAPH:

- from above Boxplots, we are trying to find is there any outliers in Numerical columns
- We can Observe outliers on Monthly Income, Total Working Years and Years at Company Columns that those columns don't outliers Because there is highly possibilities on occurring

PAIR-PLOT CODING:

```
#Pair-Plot
sns.pairplot(final_df[num_attrition], hue = 'Attrition')
```

OUTPUT:

By Observing above plot ,for any considering any two columns we cannot get any conclusioin on Attrition feature

```
for i in num_attrition:
sns.FacetGrid(final_df,hue="Attrition",size=3).map(sns.distplot,i).add_legend()
plt.show()
```

FEATURE EXTRACTION

CODING:

```
final_df.head()
```

OUTPUT:

AgeAttritionBusinessTravelDepartmentDistanceFromHomeEducationEducationFieldEmployeeCount EmployeeIDGenderJobLevelJobRoleMaritalStatusMonthlyIncomeNumCompaniesWorkedOver18Per centSalaryHikeStandardHoursStockOptionLevelTotalWorkingYearsTrainingTimesLastYearYearsAt CompanyYearsSinceLastPromotionYearsWithCurrManagerEnvironmentSatisfactionJobSatisfaction WorkLifeBalanceJobInvolvementPerformanceRatingage_fsage_fs1051NoTravel_RarelySales62Life Sciences11Female1Healthcare RepresentativeMarried1311601.0Y11801.061003.04.02.0335040-60131YesTravel_FrequentlyResearch & Development101Life Sciences12Female1Research ScientistSingle418900.0Y23816.035143.02.04.0243020-40232NoTravel_FrequentlyResearch & Development174Other13Male4Sales ExecutiveMarried1932801.0Y15835.025032.02.01.0333020-40338NoNon-TravelResearch & Development25Life Sciences14Male3Human ResourcesMarried832103.0Y118313.058754.04.03.0233020-40432NoTravel_RarelyResearch & Development101Medical15Male1Sales ExecutiveSingle234204.0Y12829.026044.01.03.0333020-40

```
#lambda x: x*10 if x<2 else (x**2 if x<4 else x+10)
#final_df['age_fs']=final_df['Age'].map(lambda x: "20" if(x<30) else ("30" if(30<=x<40) else
      ("40" if(40<=x<50) else "50")))
final_df['age_fs1']=final_df['Age'].map(lambda x: "20-40" if(x<40) else "40-60")
#sns.FacetGrid(final_df,hue="Attrition",size=3).map(sns.distplot,'age_fs').add_legend()
#plt.show()
#sns.FacetGrid(final_df,hue="Attrition",size=3).map(sns.distplot,'Age').add_legend()
```

```
#final_df['age_fs1']=final_df['Age'].map(lambda x: "20-40" if(x<40) else "40-
60")   sns.countplot(final_df["age_fs1"],hue=final_df['Attrition'])   plt.show()
```

FINDING CORRELATION

CORRELATION ON NUMERICAL FEATURES

CODING:

```
fig, ax = plt.subplots(figsize=(10,7))        # Sample figsize in inches
sns.heatmap(final_df_numerical_data.corr(),cmap='YlOrRd',linewidths=.5,ax=ax)
```

OUTPUT:

There is no higher dependency between column to columns from above heat Map Plot

CATEGORICAL VARIABLES CORRELATION

HYPOTHESIS TESTING CONDITIONS

Our hypotheses will be:

1. **Null Hypothesis (H0)**

H0: There is no relationship between 2 categorial variables ie Both features or variables are independent of each other

1. **Alternate Hypothesis (H1)**

H1: There is Relationship between 2 categorical variables .ie Both features or variables are independent of each other

CODING:

ibm corporate employee attrition analysis.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

- H1 :: There is Relationship between 2 categorical varibles .ie .. Both features or varibles are independent of each other

```
#categorical=[i for i in final_df.columns.tolist() if(final_df[i].dtype==object)]
#categorical.remove("Over18")

import scipy.stats
from scipy.stats import chi2
results=[]
lst=[]
#final=[]
input_features=[]
chisqr_result=[]
for i in categorical:
  #print("*"*6 + i + "*"*6)
  final=[]
  for j in categorical:
    #print("*"*6 + j + "*"*6)
    #print("*"*6 + i +"--" + j + "*"*6)
    #Contingency Table
    contingency_table=pd.crosstab(final_df[i],final_df[j])
    #print('contingency_table :-\n',contingency_table)

    #Observed Values
    Observed_Values = contingency_table.values
    #print("Observed Values :-\n",Observed_Values)

    #Expected Values
    #import scipy.stats
    b=scipy.stats.chi2_contingency(contingency_table)
    Expected_Values = b[3]
    #print("Expected Values :-\n",Expected_Values)
```

ibm corporate employee attrition analysis.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

```
    contingency_table=pd.crosstab(final_df[i],final_df[j])
    #print('contingency_table :-\n',contingency_table)

    #Observed Values
    Observed_Values = contingency_table.values
    #print("Observed Values :-\n",Observed_Values)

    #Expected Values
    #import scipy.stats
    b=scipy.stats.chi2_contingency(contingency_table)
    Expected_Values = b[3]
    #print("Expected Values :-\n",Expected_Values)


    #Degree of Freedom
    no_of_rows=Observed_Values.shape[0]
    no_of_columns=Observed_Values.shape[1]
    df=(no_of_rows-1)*(no_of_columns-1)
    #print("Degree of Freedom:-",df)

    #Significance Level 5%
    alpha=0.05


    #chi-square statistic - χ2
    #from scipy.stats import chi2
    chi_square=sum([(o-e)**2./e for o,e in zip(Observed_Values,Expected_Values)])
    chi_square_statistic=chi_square[0]+chi_square[1]
    #print("chi-square statistic:-",chi_square_statistic)


    #critical_value
    critical_value=chi2.ppf(q=1-alpha,df=df)
```

ibm corporate employee attrition analysis.ipynb

File Edit View Insert Runtime Tools Help All changes saved

```
chi_square_statistic=chi_square[0]+chi_square[1]
#print("chi-square statistic:-",chi_square_statistic)


#critical_value
critical_value=chi2.ppf(q=1-alpha,df=df)
#print('critical_value:',critical_value)

#p-value
p_value=1-chi2.cdf(x=chi_square_statistic,df=df)
#print('p-value:',p_value)


#print('Significance level: ',alpha)
#print('Degree of Freedom: ',df)
#print('chi-square statistic:',chi_square_statistic)
#print('critical_value:',critical_value)
#print('p-value:',p_value)

#lst1=[df,chi_square_statistic,critical_value,p_value]
lst1=[df,chi_square_statistic,p_value]

#compare chi_square_statistic with critical_value and p-value which is the probability of getting chi-square>0.09 (chi_square_statistic)
if(chi_square_statistic>=critical_value):
  #print("Reject H0,There is a relationship between 2 categorical variables")
  test_stat=1
else:
  #print("Retain H0,There is no relationship between 2 categorical variables")
  test_stat=0

if(p_value<=alpha):
    ##print("Reject H0,There is a relationship between 2 categorical variables")
```

ibm corporate employee attrition analysis.ipynb

File Edit View Insert Runtime Tools Help All changes saved

```
#compare chi_square_statistic with critical_value and p-value which is the probability of getting chi-square>0.09 (chi_square_statistic)
if(chi_square_statistic>=critical_value):
  #print("Reject H0,There is a relationship between 2 categorical variables")
  test_stat=1
else:
  #print("Retain H0,There is no relationship between 2 categorical variables")
  test_stat=0

if(p_value<=alpha):
    ##print("Reject H0,There is a relationship between 2 categorical variables")
    p_val=1
else:
    #print("Retain H0,There is no relationship between 2 categorical variables")
    p_val=0

if((test_stat==1) and (p_val==1) ):
  final_output=1
else:
  final_output=0

lst.append(lst1)
final.append(final_output)
input_features.append((i,j))


results.append(lst)
chisqr_result.append(final)

print(input_features)
print(results)
print(chisqr_result)
```

```
#categorical=[i for i in final_df.columns.tolist()
if(final_df[i].dtype==object)]
#categorical.remove("Over18")

import
scipy.stats  from
scipy.stats
import chi2
results=[]  lst=[]
#final=[]
input_features=[]
chisqr_result=[]
for i in
categorical:
  #print("*"*6 +
i     +     "*"*6)
final=[]     for j
in categorical:
   #print("*"*6 + j + "*"*6)
   #print("*"*6 + i +"--" + j + "*"*6)
```

```
    #Contingency Table
    contingency_table=pd.crosstab(final_df[i],final_df[j])
#print('contingency_table :-\n',contingency_table)
```

```
    #Observed Values
    Observed_Values = contingency_table.values
    #print("Observed Values :-\n",Observed_Values)

    #Expected Values        #import
scipy.stats
    b=scipy.stats.chi2_contingency(contingency_table)
    Expected_Values = b[3]
    #print("Expected Values :-\n",Expected_Values)

    #Degree of Freedom      no_of_rows=Observed_Values.shape[0]
no_of_columns=Observed_Values.shape[1]      df=(no_of_rows-
1)*(no_of_columns-1)
    #print("Degree of Freedom:-",df)

    #Significance Level 5%      alpha=0.05

    #chi-square statistic - χ2        #from
scipy.stats import chi2
    chi_square=sum([(o-e)**2./e for o,e in zip(Observed_Values,Expected_Values)])
chi_square_statistic=chi_square[0]+chi_square[1]      #print("chi-square statistic:-
",chi_square_statistic)

    #critical_value      critical_value=chi2.ppf(q=1-alpha,df=df)
    #print('critical_value:',critical_value)

    #p-value
    p_value=1-chi2.cdf(x=chi_square_statistic,df=df)      #print('p-value:',p_value)

    #print('Significance level: ',alpha)
    #print('Degree of Freedom: ',df)
    #print('chi-square statistic:',chi_square_statistic)
    #print('critical_value:',critical_value)
    #print('p-value:',p_value)

    #lst1=[df,chi_square_statistic,critical_value,p_value]
lst1=[df,chi_square_statistic,p_value]

    #compare chi_square_statistic with critical_value and p-value which is the probability of
       getting chi-square>0.09 (chi_square_statistic)
    if(chi_square_statistic>=critical_value):
      #print("Reject H0,There is a relationship between 2 categorical variables")
test_stat=1      else:
      #print("Retain H0,There is no relationship between 2 categorical variables")
test_stat=0

    if(p_value<=alpha):
        ##print("Reject H0,There is a relationship between 2 categorical variables")
p_val=1      else:
```

```
        #print("Retain H0,There is no relationship between 2 categorical variables")
p_val=0

    if((test_stat==1) and (p_val==1)
):       final_output=1      else:
     final_output=0

    lst.append(lst1)
final.append(final_output)
input_features.append((i,j))


  results.append(lst)
chisqr_result.append(final)

print(input_features)
print(results)
print(chisqr_result)
```

```
# plot the heatmap
sns.heatmap(chisqr_result,
      xticklabels=categorical,
      yticklabels=categorical,annot=True)
```

7.2 FEATURE 2

TRAIN-TEST SPLIT CODING:

```
from sklearn.model_selection import GridSearchCV
from sklearn.model_selection import RandomizedSearchCV
from sklearn.model_selection import TimeSeriesSplit
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression

attrition_lable=final_df['Attrition']
final_df=final_df.drop('Attrition',1)
```

```
attrition_lable=attrition_lable.map(lambda x : 1 if(x=='Yes') else 0)

X_train, X_test, y_train, y_test =
    train_test_split(final_df,attrition_lable,test_size=0.20,stratify=attrition_lable,
    random_state=42)
X_train, X_cv, y_train, y_cv = train_test_split(X_train,y_train,test_size=0.20,stratify=y_train,
    random_state=42)
```

CONVERTING CATEGORICAL COLUMNS DATA INTO NUMERICAL DATA

USING GET DUMMIES CODING:

```
#foe i in [X_train,X_cv,X_test]:
X_train=pd.get_dummies(X_train)
X_cv=pd.get_dummies(X_cv)
X_test=pd.get_dummies(X_test)

print(X_train.shape , X_cv.shape , X_test.shape)
```

```
(2752, 48) (688, 48) (860, 48)
```

USING COUNT VECTORIZES

ibm corporate employee attrition analysis.ipynb

File Edit View Insert Runtime Tools Help Saving...

+ Code + Text

2. Using CountVectorizes

```
#categorical
#[ X_train[i].value_counts().keys().tolist() for i in categorical]

#final_df.applymap(lambda x: j.replace("-","_").replace(" ","_"))
for i in categorical:
  #print(i)
  X_train[i]=X_train[i].map(lambda x: x.replace("-","_").replace(" ","_"))
  X_cv[i]=X_cv[i].map(lambda x: x.replace("-","_").replace(" ","_"))
  X_test[i]=X_test[i].map(lambda x: x.replace("-","_").replace(" ","_"))
  #print(df[i])

#X_train
[ X_test[i].value_counts().keys().tolist() for i in categorical]

from sklearn.feature_extraction.text import CountVectorizer

vect = CountVectorizer()

x_train_MaritalStatus= vect.fit_transform(X_train['MaritalStatus'])
x_test_MaritalStatus = vect.transform(X_test['MaritalStatus'])
x_cv_MaritalStatus  = vect.transform(X_cv['MaritalStatus'])

x_train_MaritalStatus.shape , x_cv_MaritalStatus.shape , x_test_MaritalStatus

vect = CountVectorizer()

x_train_BusinessTravel= vect.fit_transform(X_train['BusinessTravel'])
x_test_BusinessTravel = vect.transform(X_test['BusinessTravel'])
x_cv_BusinessTravel  = vect.transform(X_cv['BusinessTravel'])
```

ibm corporate employee attrition analysis.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

```
x_train_MaritalStatus= vect.fit_transform(X_train['MaritalStatus'])
x_test_MaritalStatus = vect.transform(X_test['MaritalStatus'])
x_cv_MaritalStatus  = vect.transform(X_cv['MaritalStatus'])

x_train_MaritalStatus.shape , x_cv_MaritalStatus.shape , x_test_MaritalStatus

vect = CountVectorizer()

x_train_BusinessTravel= vect.fit_transform(X_train['BusinessTravel'])
x_test_BusinessTravel = vect.transform(X_test['BusinessTravel'])
x_cv_BusinessTravel  = vect.transform(X_cv['BusinessTravel'])

x_train_BusinessTravel.shape , x_cv_BusinessTravel.shape , x_test_BusinessTravel.shape
vect.get_feature_names()

x_train_BusinessTravel.toarray()


lst1=[]
for i in categorical:
  vect = CountVectorizer()

  x_train_vect= vect.fit_transform(final_df[i])

  x_train_col_name=pd.DataFrame(x_train_vect.toarray(),columns=vect.get_feature_names())

  lst1.append(x_train_col_name)

print(lst1)


horizontalStack = pd.concat([surveySub, surveySubLast10], axis=1)
```

ibm corporate employee attrition analysis.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

```
lst1=[]
for i in categorical:
  vect = CountVectorizer()

  x_train_vect= vect.fit_transform(final_df[i])

  x_train_col_name=pd.DataFrame(x_train_vect.toarray(),columns=vect.get_feature_names())

  lst1.append(x_train_col_name)

print(lst1)


horizontalStack = pd.concat([surveySub, surveySubLast10], axis=1)
x_train_col_name=pd.DataFrame(x_train_vect.toarray(),columns=vect.get_feature_names())

from sklearn.feature_extraction.text import CountVectorizer

vect = CountVectorizer()

x_train_BusinessTravel= vect.fit_transform(X_train['MaritalStatus'])
x_test_BusinessTravel = vect.transform(X_test['MaritalStatus'])
x_cv_BusinessTravel  = vect.transform(X_cv['MaritalStatus'])
#col=['JobRole_'+i for i in vect.get_feature_names()]

pd.DataFrame(x_train_BusinessTravel.toarray(),columns=col)


final_df.columns.tolist()
```

```
#categorical
#[ X_train[i].value_counts().keys().tolist() for i in categorical]

#final_df.applymap(lambda x: j.replace("-
","_").replace(" ","_"))   for i in categorical:    #print(i)
  X_train[i]=X_train[i].map(lambda x: x.replace("-","_").replace(" ","_"))
  X_cv[i]=X_cv[i].map(lambda x: x.replace("-","_").replace(" ","_"))
  X_test[i]=X_test[i].map(lambda x: x.replace("-","_").replace(" ","_"))
#print(df[i])

#X_train
[ X_test[i].value_counts().keys().tolist() for i in categorical]

from sklearn.feature_extraction.text import CountVectorizer

vect = CountVectorizer()

x_train_MaritalStatus=
vect.fit_transform(X_train['MaritalStatus'])
x_test_MaritalStatus                                                    =
vect.transform(X_test['MaritalStatus'])    x_cv_MaritalStatus
= vect.transform(X_cv['MaritalStatus'])

x_train_MaritalStatus.shape , x_cv_MaritalStatus.shape , x_test_MaritalStatus

vect = CountVectorizer()

x_train_BusinessTravel=
vect.fit_transform(X_train['BusinessTravel'])
x_test_BusinessTravel                                                   =
vect.transform(X_test['BusinessTravel'])    x_cv_BusinessTravel
= vect.transform(X_cv['BusinessTravel'])

x_train_BusinessTravel.shape , x_cv_BusinessTravel.shape ,
x_test_BusinessTravel.shape   vect.get_feature_names()

x_train_BusinessTravel.toarray()

lst1=[]   for
i in
categorical:
```

```
vect = CountVectorizer()

x_train_vect= vect.fit_transform(final_df[i])

x_train_col_name=pd.DataFrame(x_train_vect.toarray(),columns=vect.get_feature_names())

lst1.append(x_train_col_name)

print(lst1)

horizontalStack = pd.concat([surveySub, surveySubLast10], axis=1)
x_train_col_name=pd.DataFrame(x_train_vect.toarray(),columns=vect.get_feature_names())

from sklearn.feature_extraction.text import CountVectorizer

vect = CountVectorizer()

x_train_BusinessTravel= vect.fit_transform(X_train['MaritalStatus'])
x_test_BusinessTravel = vect.transform(X_test['MaritalStatus'])
x_cv_BusinessTravel = vect.transform(X_cv['MaritalStatus'])
#col=['JobRole_'+i for i in vect.get_feature_names()]

pd.DataFrame(x_train_BusinessTravel.toarray(),columns=col)

final_df.columns.tolist()
```

CHECKING DISTRIBUTION ON THE LABEL IN TEST, TRAIN, CV DATA

CODING:

```
# it returns a dict, keys as class labels and values as the number of data points in that class  train_class_distribution = y_train.value_counts()
```

```
test_class_distribution = y_test.value_counts()   cv_class_distribution =
y_cv.value_counts()

my_colors = 'rgbkymc'
train_class_distribution.plot(kind='bar')   plt.xlabel('Class')
plt.ylabel('Data points per Class')
plt.title('Distribution of yi in train data')
plt.grid()   plt.show()

sorted_yi = np.argsort(-train_class_distribution.values)   for i
in sorted_yi:
   print('Number of data points in class', i, ':',train_class_distribution.values[i], '(',
      np.round((train_class_distribution.values[i]/y_train.shape[0]*100), 3), '%)')


print('-'*80)   my_colors =
'rgbkymc'
test_class_distribution.plot(kind='bar')
plt.xlabel('Class')   plt.ylabel('Data points
per Class')   plt.title('Distribution of yi in
test data')   plt.grid()   plt.show()

sorted_yi = np.argsort(-test_class_distribution.values)   for i
in sorted_yi:
   print('Number of data points in class', i+1, ':',test_class_distribution.values[i], '(',
      np.round((test_class_distribution.values[i]/y_test.shape[0]*100), 3), '%)')

print('-'*80)   my_colors =
'rgbkymc'
cv_class_distribution.plot(kind='bar')
plt.xlabel('Class')
plt.ylabel('Data points per Class')
plt.title('Distribution of yi in cross validation data')
plt.grid()   plt.show()

sorted_yi = np.argsort(-train_class_distribution.values)  for i in
sorted_yi:
   print('Number of data points in class', i+1, ':',cv_class_distribution.values[i], '(',
      np.round((cv_class_distribution.values[i]/y_cv.shape[0]*100), 3), '%)')
```

ibm corporate employee attrition analysis.ipynb
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
Comment Share
Connect Editing
for i in sorted_yi:
print('Number of data points in class', i+1, ':',cv_class_distribution.values[i], '(', np.round((cv_class_distribution.values[i]/y_cv.shape[0]*100), 3),
Distribution of yi in train data
Data points per Class
Class
Number of data points in class 0 : 2307 (83.83 %)
Number of data points in class 1 : 445 (16.17 %)
Distribution of yi in test data
Data points per Class
Class
Number of data points in class 1 : 721 (83.837 %)
Number of data points in class 2 : 139 (16.163 %)
Distribution of yi in cross validation data
Data points per Class
Class
Number of data points in class 1 : 577 (83.866 %)
Number of data points in class 2 : 111 (16.134 %)

OUTPUT:

Distribution of yi in train data
Data points per Class
2000
1500
1000
500
0
0
1
Class

Number of data points in class 0 : 2307 (83.83 %) Number of data points in class 1 : 445 (16.17 %)

Number of data points in class 1 : 721 (83.837 %)
Number of data points in class 2 : 139 (16.163 %)

Number of data points in class 1 : 577 (83.866 %)
Number of data points in class 2 : 111 (16.134 %)

FUNCTIONS FOR ML MODELS

```
import math   from sklearn.metrics
import roc_auc_score   from
sklearn.metrics import auc
from sklearn.metrics import accuracy_score
from sklearn.metrics import confusion_matrix
from sklearn.metrics import
classification_report   from sklearn.metrics
import precision_score
```

```
from sklearn.metrics import recall_score
from sklearn.metrics import f1_score
from sklearn.metrics import roc_curve
from prettytable import PrettyTable
```

FUNCTION FOR HYPER PARAMETER TUNING

```
def  bestHypermeterFunc(algo,x_train,y_train,x_cv,y_cv,x_test,y_test,verborse,kernel_select=N
      one):

  print(kernel_select)

  #Defining values for HyperParameter
alpha = [10 ** x for x in range(-6, 1)]
  C_log=[math.log(i) for i in alpha]
  #Storing Error in Array
cv_log_error_array = []

      for i in
alpha:

    if(verborse==1):
```

```
    print("for alpha =", i)

   if(algo=='Logistic_Regression'):       clf = SGDClassifier(alpha=i,
penalty='l2', loss='log', random_state=42)

   elif(algo=='SVM'):         clf = SVC(C=i,kernel=kernel_select , probability=True,
class_weight='balanced')

   clf.fit(x_train, y_train)
   sig_clf = CalibratedClassifierCV(clf, method="sigmoid")
sig_clf.fit(x_train, y_train)
   sig_clf_probs = sig_clf.predict_proba(x_cv)
cv_log_error_array.append(log_loss(y_cv, sig_clf_probs, labels=clf.classes_, eps=1e-15))
if(verborse==1):
      print("Log Loss :",log_loss(y_cv, sig_clf_probs))

  if(verborse==1):
   fig, ax = plt.subplots()      ax.plot(C_log, cv_log_error_array,c='g')      for i, txt in
enumerate(np.round(cv_log_error_array,3)):
ax.annotate((C_log[i],str(txt)[:4]), (C_log[i],np.round(cv_log_error_array[i],2)))
plt.grid()
   plt.title("Cross Validation Error for each
alpha")      plt.xlabel("Alpha i's")
plt.ylabel("Error measure")      plt.show()

  return cv_log_error_array,alpha
```

DESIGNING ML AGO FOR THE BEST FIT HYPER PARAMETER VALUE

ibm corporate employee attrition analysis.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

Connect Editing

```
elif(algo=='SVM'):
  clf = SVC(C=alpha[best_alpha],kernel=kernel_select,probability=True, class_weight='balanced')

clf.fit(x_train, y_train)
sig_clf = CalibratedClassifierCV(clf, method="sigmoid")
sig_clf.fit(x_train, y_train)

#predicting output Labels for all train,CV and Test Data
predict_train=sig_clf.predict(x_train)
predict_cv=sig_clf.predict(x_cv)
predict_test=sig_clf.predict(x_test)

#Finding LogLoss Metrics For all Data using predictProb values
predictProb_train = sig_clf.predict_proba(x_train)
logloss_train=log_loss(y_train, predictProb_train, labels=clf.classes_, eps=1e-15)
print('For values of best alpha = ', alpha[best_alpha], "The train log loss is:",logloss_train)

predictProb_cv = sig_clf.predict_proba(x_cv)
logloss_cv=log_loss(y_cv, predictProb_cv, labels=clf.classes_, eps=1e-15)
print('For values of best alpha = ', alpha[best_alpha], "The cross validation log loss is:",logloss_cv)

predictProb_test = sig_clf.predict_proba(x_test)
logloss_test=log_loss(y_test, predictProb_test, labels=clf.classes_, eps=1e-15)
print('For values of best alpha = ', alpha[best_alpha], "The test log loss is:",logloss_test)

if(verborse==1):
#calling ploting function
  plot_confusion_matrix(y_test,predict_test)

return predict_train,predict_cv,predict_test,logloss_train,logloss_cv,logloss_test,clf
```

```
def main_model(algo,x_train,y_train,x_cv,y_cv,x_test,y_test,verborse,kernel_select=None):

  #calling bestHypermeterFunc and that Function Return's the CVLog_Error Array,alpha
cv_log_error_array,alpha =
bestHypermeterFunc(algo,x_train,y_train,x_cv,y_cv,x_test,y_test,verborse,kernel_select)

  print(kernel_select)

  print(cv_log_error_array,alpha)
print("******bestHypermeterFunc Completed********")

  #Finding Best HyperParameter  Index  from cv_log_error_array using np.argmin()
  #from that index find its value
best_alpha =
np.argmin(cv_log_error_array)    print("for
best_alpha =", alpha[best_alpha])

  #Actual ML ALgorithm Running
if(algo=='Logistic_Regression'):
   clf = SGDClassifier(alpha=alpha[best_alpha], penalty='l2', loss='log', random_state=42)

  elif(algo=='SVM'):
   clf = SVC(C=alpha[best_alpha],kernel=kernel_select,probability=True,
      class_weight='balanced')
```

```
  clf.fit(x_train, y_train)
  sig_clf = CalibratedClassifierCV(clf, method="sigmoid")
```

```
  sig_clf.fit(x_train, y_train)

  #predicting output Labels for all train,CV and Test Data
predict_train=sig_clf.predict(x_train)
predict_cv=sig_clf.predict(x_cv)     predict_test=sig_clf.predict(x_test)

  #Finding LogLoss Metrics For all Data using predictProb values     predictProb_train
= sig_clf.predict_proba(x_train)
  logloss_train=log_loss(y_train, predictProb_train, labels=clf.classes_, eps=1e-15)
print('For values of best alpha = ', alpha[best_alpha], "The train log loss is:",logloss_train)

  predictProb_cv = sig_clf.predict_proba(x_cv)
  logloss_cv=log_loss(y_cv, predictProb_cv, labels=clf.classes_, eps=1e-15)     print('For
values of best alpha = ', alpha[best_alpha], "The cross validation log loss is:",logloss_cv)

  predictProb_test = sig_clf.predict_proba(x_test)
  logloss_test=log_loss(y_test, predictProb_test, labels=clf.classes_, eps=1e-15)     print('For
values of best alpha = ', alpha[best_alpha], "The test log loss is:",logloss_test)

  if(verborse==1):     #calling
ploting function
    plot_confusion_matrix(y_test,predict_test)

  return predict_train,predict_cv,predict_test,logloss_train,logloss_cv,logloss_test,clf
```

Function for plotting Confusion Matrix

```
#0=N0 , 1=Yes  def
plot_confusion_matrix(y_test,pred):

 labels = [0,1]
```

```
 #confusion_matrix
 C = confusion_matrix(y_test,pred)

 #Recall MAtrix
A =(((C.T)/(C.sum(axis=1))).T)

 #Precision MAtrix
B =(C/C.sum(axis=0))

 # representing A in heatmap format    lst=[C,A,B]
 lst_names=["Confusion Matrix","Recall","Precision"]
     for i in zip(lst,lst_names):      print("-
"*20, i[1], "-"*20)
plt.figure(figsize=(7,4))
   sns.heatmap(i[0], annot=True, cmap="YlGnBu", fmt=".3f", xticklabels=labels,
      yticklabels=labels)
   plt.xlabel('Predicted                    Class')
plt.ylabel('Original Class')      plt.show()
```

Function for Plotting Pretty Table for Metrics

ibm corporate employee attrition analysis.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

Function for Ploting Pretty Table for Metrics

```
[ ] from prettytable import PrettyTable

    def metrics(algo,X_train_scalar,y_train,X_cv_scalar,y_cv,X_test_scalar,y_test,verbrose,kernel_select=None):

      predict_train,predict_cv,predict_test,logloss_train,logloss_cv,logloss_test,clf=main_model(algo,X_train_scalar,y_train,X_cv_scalar,y_cv,X_test_scalar,y_test,verbrose,kernel_select)

      accuracy_train=accuracy_score(y_train,predict_train)
      accuracy_cv=accuracy_score(y_cv,predict_cv)
      accuracy_test=accuracy_score(y_test,predict_test)

      precision_train=precision_score(y_train,predict_train)
      precision_cv=precision_score(y_cv,predict_cv)
      precision_test=precision_score(y_test,predict_test)

      recall_train=recall_score(y_train,predict_train)
      recall_cv=recall_score(y_cv,predict_cv)
      recall_test=recall_score(y_test,predict_test)

      F1Score_train=f1_score(y_train,predict_train)
      F1Score_cv=f1_score(y_cv,predict_cv)
      F1Score_test=f1_score(y_test,predict_test)

      x = PrettyTable()
      x.field_names = ["Metric", "Train_value" , "CV_value","Test_value"]

      x.add_row([ "Log-Loss" ,   np.round(logloss_train,3)  ,   np.round(logloss_cv,3)    ,   np.round(logloss_test,3) ])
      x.add_row([ "Precision" ,  np.round(precision_train,3) ,   np.round(precision_cv,3)  ,  np.round(precision_test,3) ])
```

Show desktop

ibm corporate employee attrition analysis.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

```
      accuracy_train=accuracy_score(y_train,predict_train)
      accuracy_cv=accuracy_score(y_cv,predict_cv)
      accuracy_test=accuracy_score(y_test,predict_test)

      precision_train=precision_score(y_train,predict_train)
      precision_cv=precision_score(y_cv,predict_cv)
      precision_test=precision_score(y_test,predict_test)

      recall_train=recall_score(y_train,predict_train)
      recall_cv=recall_score(y_cv,predict_cv)
      recall_test=recall_score(y_test,predict_test)

      F1Score_train=f1_score(y_train,predict_train)
      F1Score_cv=f1_score(y_cv,predict_cv)
      F1Score_test=f1_score(y_test,predict_test)

      x = PrettyTable()
      x.field_names = ["Metric", "Train_value" , "CV_value","Test_value"]

      x.add_row([ "Log-Loss" ,   np.round(logloss_train,3)  ,   np.round(logloss_cv,3)    ,   np.round(logloss_test,3) ])
      x.add_row([ "Precision" ,  np.round(precision_train,3) ,   np.round(precision_cv,3)  ,  np.round(precision_test,3) ])
      x.add_row([ "Recall" ,     np.round(recall_train,3) ,      np.round(recall_cv,3)     ,  np.round(recall_test,3)])
      x.add_row([ "F1-Score" ,   np.round(F1Score_train,3) ,     np.round(F1Score_cv,3)    , np.round(F1Score_test,3)])
      x.add_row([ "Accuracy" ,    np.round(accuracy_train,3)  ,   np.round(accuracy_cv,3)    ,   np.round(accuracy_test,3) ])

      print(x)

      print("Starting Get Important Features Function")
      get_imp_features(clf,kernel_select)
```

```
from prettytable import PrettyTable

def metrics(algo,X_train_scalar,y_train,X_cv_scalar,y_cv,X_test_scalar,y_test,verbrose,kernel_sel
    ect=None):

 predict_train,predict_cv,predict_test,logloss_train,logloss_cv,logloss_test,clf=main_model(algo
    ,X_train_scalar,y_train,X_cv_scalar,y_cv,X_test_scalar,y_test,verbrose,kernel_select)

 accuracy_train=accuracy_score(y_train,predict_train)
accuracy_cv=accuracy_score(y_cv,predict_cv)
accuracy_test=accuracy_score(y_test,predict_test)

 precision_train=precision_score(y_train,predict_train)
precision_cv=precision_score(y_cv,predict_cv)
precision_test=precision_score(y_test,predict_test)

 recall_train=recall_score(y_train,predict_train)
recall_cv=recall_score(y_cv,predict_cv)    recall_test=recall_score(y_test,predict_test)

 F1Score_train=f1_score(y_train,predict_train)
 F1Score_cv=f1_score(y_cv,predict_cv)
 F1Score_test=f1_score(y_test,predict_test)

 x = PrettyTable()
 x.field_names = ["Metric", "Train_value" , "CV_value","Test_value"]

 x.add_row([ "Log-Loss"
      , np.round(logloss_train,3) , np.round(logloss_cv,3)  , np.round(logloss_test,3) ])
 x.add_row([ "Precision" , np.round(precision_train,3)
    , np.round(precision_cv,3) , np.round(precision_test,3) ])
 x.add_row([ "Recall" ,   np.round(recall_train,3)
    ,    np.round(recall_cv,3)    , np.round(recall_test,3)])
 x.add_row([ "F1-Score" , np.round(F1Score_train,3) ,   np.round(F1Score_cv,3)    ,
    np.round(F1Score_test,3)])
 x.add_row([ "Accuracy"
    , np.round(accuracy_train,3) , np.round(accuracy_cv,3)  , np.round(accuracy_test,3) ])
```

```
    print(x)

 print("Starting Get Important Features Function")    get_imp_features(clf,kernel_select)
```

Function to Print Important Features

```
def get_imp_features(clf,kernel_select):
    print(kernel_select)    if((kernel_select == 'rbf' ) or
(kernel_select == 'poly')):



  print(" Kernel is {} so, we cannot get the important Features using Coef_
    Funtion".format(kernel_select))



else:

  coefs=sorted(zip(clf.coef_[0],X_train.columns.tolist()))
feat=X_train.columns.tolist()

  top10Negative=coefs[:10]      top10Postive=coefs[::-
1][:10]

  res_neg=pd.DataFrame(top10Negative,columns=['Values','Top10NegativeFeatures'])
res_pos=pd.DataFrame(top10Postive,columns=['Values','Top10PostiveFeatures'])
res=pd.concat([res_neg,res_pos],axis=1)

  print("*"*20)
#print(len(feat))
  #print(len(coefs))
    feat=[i[1] for i in
coefs]      coefs1=[i[0] for i
in coefs]
plt.figure(figsize=(10,14))
  plt.barh(range(len(feat)), coefs1,
align='center')      plt.yticks(range(len(feat)),
feat)      plt.show()

 #return res
```

Standard Scalar Function

```
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()
scaler.fit(X_train)
X_train_scalar=scaler.transform(X_train)
X_cv_scalar=scaler.transform(X_cv)
X_test_scalar=scaler.transform(X_test)
```

With Out Balanced Data

LOGISTIC REGRESSION USING FUNCTION

```
metrics("Logistic_Regression",X_train_scalar,y_train_balance,X_cv_scalar,y_cv,X_test_scalar,y_te
    st,0)

None  None
    [0.6237285769755854, 0.6146666927567451, 0.5896041032669407,
    0.5859322390340136, 0.5825722438150815, 0.5846176931817395,
    0.6017441506328953] [1e-06, 1e-05, 0.0001, 0.001, 0.01, 0.1, 1]
    ******bestHypermeterFunc Completed********   for best_alpha =
    0.01
    For values of best alpha =  0.01 The train log loss is: 0.5237113321410725
    For values of best alpha =  0.01 The cross validation log loss is: 0.5825722438150815
    For values of best alpha =  0.01 The test log loss is: 0.5472738641985359
    +----------+----------+---------+----------+
    |  Metric  | Train_value | CV_value | Test_value |
    +----------+----------+---------+----------+
    | Log-Loss |   0.524   | 0.583  |  0.547   |
    | Precision |   0.725   | 0.287  |  0.324   |
    |  Recall  |   0.772   |  0.64  |  0.698   |
    | F1-Score |   0.748   | 0.397  |  0.443   |
    | Accuracy |   0.739   | 0.686  |  0.716   |
    +----------+----------+---------+----------+
```

```
Starting Get Important Features Function
None
********************
```

SVM

WITH BALANCED DATA:

For Balancing Data using SMOTE function from Imblearn Package

```
linear
    [0.6143143363763562, 0.6143143337374273, 0.6142136356700172,
```

```
metrics("SVM",X_train_scalar,y_train_balance,X_cv_scalar,y_cv,X_test_scalar,y_test,1,"linear")
linear
    for alpha = 1e-06
    Log Loss : 0.6143143363763562
    for alpha = 1e-05
    Log Loss : 0.6143143337374273
    for alpha = 0.0001
    Log Loss : 0.6142136356700172
    for alpha = 0.001
    Log Loss : 0.5871537845084024
    for alpha = 0.01
    Log Loss : 0.5817866567511131
    for alpha = 0.1
    Log Loss : 0.583009140383031
    for alpha = 1
    Log Loss : 0.5832164963645
```

```
0.5871537845084024, 0.5817866567511131, 0.583009140383031,
0.5832164963645] [1e-06, 1e-05, 0.0001, 0.001, 0.01, 0.1, 1]
******bestHypermeterFunc
Completed********   for best_alpha = 0.01
For values of best alpha =  0.01 The train log loss is: 0.5278361295624395
For values of best alpha =  0.01 The cross validation log loss is:
0.5817866567511131
For values of best alpha =  0.01 The test log loss is: 0.5474462705866496   --------
------------ Confusion Matrix --------------------
```

```
+-----------+-------------+----------+-----------+
|  Metric  | Train_value | CV_value | Test_value |
+-----------+-------------+----------+-----------+
| Log-Loss |   0.528   | 0.582  |  0.547   |
| Precision |   0.732   | 0.289  |  0.329   |
|  Recall  |   0.774   | 0.622  |  0.698   |
| F1-Score |   0.752   | 0.394  |  0.447   |
| Accuracy |   0.745   | 0.692  |  0.721   |
+-----------+-------------+----------+-----------+
Starting Get Important Features Function
linear
********************
```

YearsSinceLastPromotion
NumCompaniesWorked
BusinessTravel_Travel_Frequently
MaritalStatus_Single
EducationField_Human Resources
PerformanceRating
JobRole_Research Director
EducationField_Life Sciences
Department_Human Resources
JobRole_Sales Executive
JobRole_Sales Representative
JobRole_Research Scientist
Department_Sales
JobRole_Laboratory Technician
JobRole_Human Resources
Gender_Female
EmployeeID
StandardHours
Over18_Y
EmployeeCount
EducationField_Medical
Gender_Male
StockOptionLevel
PercentSalaryHike
EducationField_Other
EducationField_Marketing
JobRole_Healthcare Representative
JobLevel
Department_Research & Development
MonthlyIncome
BusinessTravel_Travel_Rarely
Education
JobInvolvement
YearsAtCompany
MaritalStatus_Married
JobRole_Manufacturing Director
DistanceFromHome
JobRole_Manager
MaritalStatus_Divorced
EducationField_Technical Degree
WorkLifeBalance
TrainingTimesLastYear
BusinessTravel_Non-Travel
TotalWorkingYears
Age
EnvironmentSatisfaction
JobSatisfaction
YearsWithCurrManager
−0.4
−0.2
0.0
0.2
0.4

RBF Kernal

r

b
f
f
o
r
a
l
p
h
a
=
1
e
-
0
6
L
o
g
L
o
s
s
:
0
.
5
7
3
7
6
6
1
9

5
0
8
4
8
7
8
6
f
o
r
a
l
p
h
a
=
1
e
-
0
5
L
o
g
L
o
s
s
:
0
.
5
7
3
7
6
6
1
9
8
2
6
6
9
2
2
1
f
o
r
a
l
p

h
a
=
0
.
0
0
0
1
L
o
g
L
o
s
s
:
0
.
5
7
3
7
6
6
1
9
2
0
5
6
6
7
3
6
f
o
r
a
l
p
h
a
=
0
.
0
0
1
L
o
g
L

o
s
s
:
0
.
5
7
3
7
6
6
1
9
5
5
4
7
5
8
8
8
f
o
r
a
l
p
h
a
=
0
.
0
1
L
o
g
L
o
s
s
:
0
.
5
7
3
5
6
1
1
7
1

	8000113 for alpha = 0.1 LogLoss: 0.475145163975406 24 for alpha

```
=
1
L
o
g
L
o
s
s
:
0
.
2
7
8
1
0
7
2
9
1
7
3
5
5
7
8
3
5
```

```
rbf
    [0.5737661950848786, 0.5737661982669221, 0.5737661920566736,
```

```
0.5737661955475888, 0.5735611718000113, 0.47514516397540624,
0.27810729173557835]  [1e-06,  1e-05,  0.0001,  0.001,  0.01,  0.1,  1]
******bestHypermeterFunc Completed********  for best_alpha = 1
For values of best alpha =  1 The train log loss is: 0.06707151930965102
For values of best alpha =  1 The cross validation log loss is: 0.27810729173557835
For values of best alpha =  1 The test log loss is: 0.21711277243059424  --------------------
Confusion Matrix --------------------
```

```
+-----------+-------------+----------+------------+
|  Metric  | Train_value | CV_value | Test_value |
+-----------+-------------+----------+------------+
| Log-Loss |   0.067   | 0.278  |  0.217   |
| Precision |   0.985   | 0.722  |  0.814   |
|  Recall  |   0.975   | 0.631  |  0.662   |
| F1-Score |    0.98   | 0.673  |   0.73   |
| Accuracy |    0.98   | 0.901  |  0.921   |
+-----------+-------------+----------+------------
+  Starting Get Important Features
Function   rbf
 Kernel is rbf so, we cannot get the important Features using Coef_ Funtion
```

```
import pandas as pd   import
matplotlib.pyplot as plt
a=pd.DataFrame([ ['NaiveBayes',0.5192,1.163 ,1.195],
              ['KNN',  0.6288 , 1.006 , 1.083] ,
              ['LR ClassBlance',   0.4406 , 0.986 , 1.006] ,
              ['LR NO ClassBlance', 0.4347 , 1.0114  ,1.0298],
             ['SVM LINEAR',  0.4828 ,  1.0236 , 1.056],
             ['RF with Response coding ', 0.056  , 1.373 , 1.383],
            ['RF with OneHotEncoding', 0.8369 , 1.1908 , 1.1781 ],
             ['SlackModels', 0.5296, 1.124  , 1.19],
             ['max voting', 0.821, 1.167 , 1.196],
            ['LR NO ClassBlance[Count Vectorize UNIGRAM]', 0.6099, 1.157 , 1.11 ],
['LR NO ClassBlance[Count Vectorize BIGRAM]', 0.8382, 1.2165 , 1.2227 ]],
columns=["MODELS","Train LogLoss","CV LogLoss","Test LogLoss"],
           )

#plt.figure(figsize=(1,1))
plt.rcParams["figure.figsize"] = [14,9]
a.set_index('MODELS', inplace=True)
a.plot(kind='barh')
plt.xlabel("LogLoss",fontsize=20)
plt.ylabel("MOdels",fontsize=20)
plt.xticks(fontsize=14)
```

```
#rotation=90)
plt.yticks(fontsize=14)  plt.show()
```

ibm corporate employee attrition analysis.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

Connect Editing

```
import pandas as pd
import matplotlib.pyplot as plt
a=pd.DataFrame([ ['NaiveBayes',0.5192,1.163 ,1.195],
                 ['KNN',  0.6288 , 1.006 , 1.083] ,
                 ['LR ClassBlance',   0.4406 , 0.986 , 1.006] ,
                 ['LR NO ClassBlance', 0.4347 , 1.0114  ,1.0298],
                ['SVM LINEAR',  0.4828 ,  1.0236 , 1.056],
                ['RF with Response coding ', 0.056  , 1.373 , 1.383],
               ['RF with OneHotEncoding', 0.8369 , 1.1908 , 1.1781 ],
                ['SlackModels', 0.5296, 1.124  , 1.19],
                ['max voting', 0.821, 1.167 , 1.196],
               ['LR NO ClassBlance[Count Vectorize UNIGRAM]', 0.6099, 1.157 , 1.11 ],
                ['LR NO ClassBlance[Count Vectorize BIGRAM]', 0.8382, 1.2165 , 1.2227 ]],
               columns=["MODELS",'Train LogLoss','CV LogLoss','Test LogLoss'],
              )

#plt.figure(figsize=(1,1))
plt.rcParams["figure.figsize"] = [14,9]
a.set_index('MODELS', inplace=True)
a.plot(kind='barh')
plt.xlabel("LogLoss",fontsize=20)
plt.ylabel("MOdels",fontsize=20)
plt.xticks(fontsize=14) #rotation=90)
plt.yticks(fontsize=14)
plt.show()
```

Comparsion of Mertrics(LogLoss and F1Score) with all ML-Models along with Data (which is balanced using SMOTE and Unbalanced Data)

Comparsion of Mertrics(LogLoss and F1Score) with all ML-Models along with Data (which is balanced using SMOTE and Unbalanced Data)

```
a=pd.DataFrame([ ['LogisticRegression',0.382 ,0.547],
                 ['SVM_linear', 0.38 , 0.547 ] ,
                 ['SVM_RBF', 0.22 ,0.217 ] ,
                 ['SVM_Polynomial', 0.193 ,0.198] ],


columns=["MODELS",'TestLogLoss_withOutBalanceData','TestLogLoss_BalanceData'],
          )
plt.rcParams["figure.figsize"] = [7,5]
a.set_index('MODELS', inplace=True)
a.plot(kind='barh')
plt.xlabel("LogLoss",fontsize=20)
plt.ylabel("MOdels",fontsize=20)
plt.xticks(fontsize=14)
#rotation=90)
plt.yticks(fontsize=14)  plt.show()
```

```
a=pd.DataFrame([ ['LogisticRegression',0.154  ,0.443],
            ['SVM_linear',  0.137  , 0.447 ] ,
            ['SVM_RBF', 0.749  ,0.73 ] ,
            ['SVM_Polynomial', 0.845   ,0.837] ],

           columns=["MODELS",'TestF1Score_withOutBalanceData','TestF1Score_BalanceD
ata'],
          )
plt.rcParams["figure.figsize"] = [7,5]
a.set_index('MODELS', inplace=True)
a.plot(kind='barh')
plt.xlabel("F1SCore",fontsize=20)
plt.ylabel("MOdels",fontsize=20)
plt.xticks(fontsize=14)
#rotation=90)
plt.yticks(fontsize=14)   plt.show()
```

7.3 DATABASE SCHEMA

DATA VISUALIZATION CHARTS AND DASHBOARD CREATION

Using the given dataset, we need to create various graphs and charts to highlight the insights and visualizations. For the given problem statment, try to build the following visualizations that suit the solution requirements.

- Employee Attrition by Age
- Attrition by Business Travel
- Attrition by Department, Job Role, Education Level and Marital Status
- Attrition by Salary Hike Percent
- Attrition by No. of Companies Worked
- Attrition by Income Groups
- Attrition by Work Experience Groups
- Dashboard of Attrition of Employees based on Employment details

IBM COGNOS ANALYTICS

To create data visualization charts and dashboard we need to login into IBM Cognos analytics. IBM® Cognos® Analytics integrates reporting, modeling, analysis, dashboards, stories, and event management so that you can understand your organization data, and make effective business decisions. This tool is used to give better understanding about the dataset.

STEPS TO CREATE VISUALIZATION CHARTS AND DASHBOARD CREATION

- Uploading of data
- Preparing the data
- Exploration of data
- Creation of Visualization Charts
- Dashboard creation

LOADING THE DATASET :

PREPARING THE DATA & EXPLORATION OF DATA

CREATION OF VISUALIZATION CHARTS

- EMPLOYEE ATTRITION STATUS:

- INFERENCES :

We can understand from the above pie chart that 16.1% of people are willing to leave and 83.3% say no to it

- ATTRITION BY BUSINESS TRAVEL

- INFERENCES :

We can understand from the above column chart that 468 people are willing to leave

- ATTRITION BY DEPARTMENT, JOB ROLE, EDUCATION LEVEL AND MARITAL STATUS

- INFERENCES :

We can understand from the above 4 division charts , people are willing to leave % is higher.

- ATTRITION BY SALARY HIKE PERCENT

- INFERENCE:

We can understand from the above charts that % of people leaving and staying.

- ATTRITION BY NO. OF COMPANIES WORKED

- INFERENCES:

We can understand from the above charts that % of people leaving and staying.

- ATTRITION BY INCOME GROUPS

- INFERENCE:

We can understand from the above charts that % of people leaving and staying.

- ATTRITION BY WORK EXPERIENCE GROUPS

- INFERENCES:

We can understand from the above charts that % of people leaving and staying.

- DASHBOARD OF ATTRITION OF EMPLOYEES BASED ON EMPLOYMENT DETAILS

- INFERENCES

We can understand from the above charts that % of people leaving and staying.

FINDING OF THIS PROJECT:

A total of 24 variables, collected from 4 sources were used to predict the probability of an employee leaving the company in the next year, using a logistic regression model

- Logistic Regression Model* is able to correctly identify 77% of employees that were likely to churn
- It is also able to identify employees that are not likely to churn, with 77% accuracy

➤ KS Statistic falls in 3rd decile (top 30%)

- Hence, it would be beneficial to target 30% of your employees most likely to leave, and work on making them stay.
- Targeting fewer employees (top 20% or top 10%) will not identify enough employees likely to leave

Targeting more employees (top 40% or top 50%) will be inefficient

➤ Predicting Attrition – Model is able to capture 75% of employees likely to leave

- Model is able to identify 75% of the employees likely to leave in the first 3 deciles

➤ Predicting Attrition – Model performs 2.5 times better than a random reach out

- Using the model offers a "lift" of 2.5 for the 3rd decile

▯ RECOMMENDATIONS – WHAT FACTORS MAKE EMPLOYEES STAY/LEAVE? (1/4)

➤ EXPERIENCE

- Employees that have worked for a total of 7 years or less are more likely to leave*
- Employees that have worked for a total of 10 years or more are more likely to stay*

➤ AGE

- Employees aged 36 years and above are more likely to stay*

- Employees aged 32 years and below are more likely to leave*

RECOMMENDATIONS – WHAT FACTORS MAKE EMPLOYEES STAY/LEAVE? (2/4)

➤ TRAINING

- Employees that got 3 or more training sessions last year are more likely to stay*
- Employees that got 2 or fewer training sessions last year are more likely to leave*

➤ YEARS WITH CURRENT MANAGER

- Employees that have spent 3 years or more under the same manager are more likely to stay*
- Employees that have spent 2 years or less under the same manager are more likely to leave*

RECOMMENDATIONS – WHAT FACTORS MAKE EMPLOYEES STAY/LEAVE? (3/4)

➤ JOB SATISFACTION

- Employees that have medium, high or very high levels of job satisfaction, are more likely to stay*
- Employees that have low levels of job satisfaction, are more likely to leave*

➤ ENVIRONMENT SATISFACTION

- Employees that have medium, high or very high levels of environment satisfaction, are more likely to stay*

- Employees that have low levels of environment satisfaction, are more likely to leave*

RECOMMENDATIONS – WHAT FACTORS MAKE EMPLOYEES STAY/LEAVE? (4/4)

➤ AVERAGE WORK HOURS

- Employees that, on average work for 7.3 hours or less, are more likely to stay*
- Employees that, on average work for 8.2 hours or more, are more likely to leave*

➤ WORK LIFE BALANCE

- Employees that rated their work life balance as good, better or best, are more likely to stay**
- Employees that rated their work life balance as bad, are more likely to leave**

▫ RECOMMENDATIONS – FACTORS THAT SURPRISINGLY DON'T AFFECT ATTRITION

- ◊ ▶Monthly Income and Percent Salary Hike do not affect attrition*

RECOMMENDATIONS

CURRENT EMPLOYEES:

- Work life balance should be improved
- Work environment should be improved
- The manager of an employee should not be changed very often
- Employees should be provided relevant training regularly, especially for its younger employees

FUTURE EMPLOYEES (CHANGES IN HIRING PROCESS):

- The company should follow either one of the strategies given below –
    - Hire older people with decent work experience
    - Hire young people and train them appropriately
- It could also opt for a combination of the two

8. TESTING

8.1 Test Cases

Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Commnets	TC for Automation(Y/N)	BUG ID	Executed By
Hypothesis Condition	Functional	General_data.c	Hypothesis condition has two parameter 0 or 1	Cleaning and preparation of data	1.To check and prepare the data 2.To write python codes for Hypothesis condition	https://www.kaggle.com/datasets/vjchoudhary7/hr-analytics-case-study General_data.csv, Employee_Survey_Data.csv, Manager_Survey_data.csv	Hypothesis value for each parameter either 0 or 1	Working as expected	Pass	Steps are clear and coding is right	Output of the test is displayed as result	nul	1)Bheulah G.L. 2)Ashwini A. 3)Sameera Banu N. 4)Shaheenah M.
Train Test Split	Functional	General_data.csv, Employee_Survey_Data.csv, Manager_Survey_data.csv	Each parameter of general data survey is made as parameter for test split	Cleaning and preparation of data	1.To test each data for test split 2.We need to write python code for each test split	https://www.kaggle.com/datasets/vjchoudhary7/hr-analytics-case-study General_data.csv, Employee_Survey_Data.csv, Manager_Survey_data.csv	The should be passed as package	Working as expected	pass	Steps are clear and coding is written right	The test case is passed	nul	1)Bheulah G.L. 2)Ashwini A. 3)Sameera Banu N. 4)Shaheenah M.
Hyper Parameter Testing	Functional	General_data.csv, Employee_Survey_Data.csv, Manager_Survey_data.csv	checking the condition for developing model	Cleaning and preparation of data	1.To prepare and clean the data and 2.To write python codes for each parameters	https://www.kaggle.com/datasets/vjchoudhary7/hr-analytics-case-study General_data.csv, Employee_Survey_Data.csv, Manager_Survey_data.csv	The should be passed as package	Working as expected	pass	Steps are clear and coding is written right	The test case is passed	nul	1)Bheulah G.L. 2)Ashwini A. 3)Sameera Banu N. 4)Shaheenah M.
Confusion Matrix	Functional	General_data.csv, Employee_Survey_Data.csv, Manager_Survey_data.csv	Overall necessary prediction can be made	Cleaning and preparation of data	1.To prepare the data 2. To write python code for each parameter	https://www.kaggle.com/datasets/vjchoudhary7/hr-analytics-case-study General_data.csv, Employee_Survey_Data.csv, Manager_Survey_data.csv	The parameter required for modelling can be identified and result is seen as heat plot	Working as expected	pass	Steps are clear and coding is written right	Output is seen as Heat plot	nul	1)Bheulah G.L. 2)Ashwini A. 3)Sameera Banu N. 4)Shaheenah M.
EDA	Functional	General_data.csv, Employee_Survey_Data.csv, Manager_Survey_data.csv	Development of Model	Cleaning and preparation of data	1.To find the parameter required for modelling 2. To write python code	https://www.kaggle.com/datasets/vjchoudhary7/hr-analytics-case-study General_data.csv, Employee_Survey_Data.csv, Manager_Survey_data.csv	The case is passed and the result is seen as bar graph	Working as expected	pass	Steps are clear and coding is written right	The output is seen as bar graph	nul	1)Bheulah G.L. 2)Ashwini A. 3)Sameera Banu N. 4)Shaheenah M.
Logistic Regression	Functional	General_data.csv, Employee_Survey_Data.csv, Manager_Survey_data.csv	Outcome of the Model	Cleaning and preparation of data	1. To find the required data for Regression Modelling 2. To write python coding	https://www.kaggle.com/datasets/vjchoudhary7/hr-analytics-case-study General_data.csv, Employee_Survey_Data.csv, Manager_Survey_data.csv	The case is passed and the result is seen as bar graph	Working as expected	pass	Steps are clear and coding is written right	The output is seen as bar graph	nul	1)Bheulah G.L. 2)Ashwini A. 3)Sameera Banu N. 4)Shaheenah M.
SVM Model	Functional	Survey_Data.c	Modelling test	Cleaning and preparation of data	1. From the Regression Modelling , the resulting parameter will be used for SVM modelling. 2.To write python coding	https://www.kaggle.com/datasets/vjchoudhary7/hr-analytics-case-study General_data.csv, Employee_Survey_Data.csv, Manager_Survey_data.csv	The case is passed and the result is seen	Working as expected	pass	Steps are clear and coding is written right	The output is seen as bar graph	nul	1)Bheulah G.L. 2)Ashwini A. 3)Sameera Banu N. 4)Shaheenah M.

8.2 USER ACCEPTANCE TESTING

UAT Execution & Report Submission

1. Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the [ProductName] project at the time of the release to User Acceptance Testing (UAT).

1. Defect Analysis

This report shows the number of resolved or closed bugs at each severity level, and how they were resolved

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Model	1	2	1	0	3
Duplicate	1	0	0	0	1
External	2	0	0	1	3
Fixed	7	2	3	0	12

Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	1	0	0	1
Totals	11	5	6	2	23

1. **Test Case Analysis**

This report shows the number of test cases that have passed, failed, and untested

Section	Total Cases	Not Tested	Fail	Pass
Hypothesis Condition	2	0	0	2
Train Test Split	5	2	0	3
Hyper Tuning Parameter Test	4	0	0	4
Confusion Matrix	1	0	0	1
Logistic Regression	1	0		1
Final Report Output	6	2	0	4
SVM Model	1	0	0	1

9. RESULTS

9.1 PERFORMANCE METRICS

Model Performance Testing:

Project team shall fill the following information in model performance testing template.

S.No.	Parameter	Values	Screenshot
	Metrics	**Regression Model:** R2 score - 79% **Classification Model:** Confusion Matrix - 79% Accuracy Score- 76% & Classification Report – 1.Correlation 2.Confusion matrix 3.Logistic regression 4.Linear SVM 5.RBF Kernel 6.Poly Kernel	

YearsSinceLastPromotion
NumCompaniesWorked
BusinessTravel_Travel_Frequently
MaritalStatus_Single
JobRole_Research Director
EducationField_Human Resources
JobRole_Sales Executive
EducationField_Life Sciences
PercentSalaryHike
Department_Human Resources
EmployeeID
Gender_Male
PerformanceRating
JobRole_Healthcare Representative
EducationField_Medical
JobRole_Laboratory Technician
Department_Research & Development
JobRole_Research Scientist
StandardHours
Over18_Y
EmployeeCount
JobRole_Human Resources
EducationField_Marketing
Department_Sales
JobRole_Sales Representative
Gender_Female
Education
DistanceFromHome
JobInvolvement
StockOptionLevel
BusinessTravel_Travel_Rarely
MonthlyIncome
YearsAtCompany
JobLevel
EducationField_Technical Degree
EducationField_Other
JobRole_Manager
MaritalStatus_Married
JobRole_Manufacturing Director
MaritalStatus_Divorced
TrainingTimesLastYear
WorkLifeBalance
BusinessTravel_Non-Travel
Age
EnvironmentSatisfaction
JobSatisfaction
TotalWorkingYears
YearsWithCurrManager
−0.4
−0.2
0.0
0.2

ibm corporate employee attrition analysis.ipynb
File Edit View Insert Runtime Tools Help
+ Code + Text
Connect
SVM
Using Linear Kernal

ibm corporate employee attrition analysis.ipynb
Using RBF Kernel
Using Poly Kernel

ibm corporate employee attrition analysis.ipynb
Using Poly Kernel

ibm corporate employee attrition analysis.ipynb
With Balanced Data : FOr Balancing Data using SMOTE function from Imblearn Package

ibm-corporate-employee-attrition-analysis.ipynb
File Edit View Insert Runtime Tools Help
+ Code + Text
Connect
metrics("logistic_Regression",X_train_scaler,y_train_balanced,X_cv_scaler,y_cv,X_test_scaler,y_test,0)
None
None
[0.42572053975004, 0.41484465276391, 0.5094845926487, 0.5093125984836, 0.5025720015015, 0.5041707101799, 0.49134410611051] [1e-06, 1e-05, 0.0001, 0.001, 0.01,
*****bestparameterhas completed*****
For best_alpha = 0.01
For values of best alpha = 0.01 The train log loss is: 0.5137511234035
For values of best alpha = 0.01 The cross validation log loss is: 0.5025720015015
For values of best alpha = 0.01 The test log loss is: 0.5471708410159
Metric	Train value	CV value	Test value
Log-Loss	0.514	0.503	0.547
Precision	0.725	0.287	0.534
Recall	0.775	0.04	0.098
F1-Score	0.748	0.707	0.443
Accuracy	0.739	0.886	0.774
Starting get Important Features function
None
****
NumOfCompaniesWorked
BusinessTravel_Travel_Frequently
MaritalStatus_Single
EducationField_Human Resources
JobRole_Research Director
Department_Human Resources
EducationField_Life Sciences
JobRole_Sales Executive
PercentSalaryHike
Employee35
Gender_Male

ibm-corporate-employee-attrition-analysis.ipynb
File Edit View Insert Runtime Tools Help
+ Code + Text
Connect
+ Linear SVM
metrics("SVM",X_train_scaler,y_train_balanced,X_cv_scaler,y_cv,X_test_scaler,y_test,1,"linear")
linear
for alpha = 1e-06
for alpha = 1e-05
for alpha = 0.0001
for alpha = 0.001
for alpha = 0.01
for alpha = 0.1
for alpha = 1
Cross Validation Error for each alpha

ibm-corporate-employee-attrition-analysis.ipynb
File Edit View Insert Runtime Tools Help
+ Code + Text
Connect
+ Linear SVM
metrics("SVM",X_train_scaler,y_train_balanced,X_cv_scaler,y_cv,X_test_scaler,y_test,1,"linear")
Alpha i=
linear
for alpha = 1e-06
for alpha = 1e-05
for alpha = 0.0001
for alpha = 0.001
for alpha = 0.01
for alpha = 0.1
for alpha = 1
*****bestparameterhas completed*****
For best_alpha = 0.01
Confusion Matrix
Recall

ibm corporate employee attrition analysis.ipynb
File Edit View Insert Runtime Tools Help
+ Code + Text
Linear SVM
Predicted Class
Predicted Class
Metric	Train_value	CV_value	Test_value
Log-Loss	0.589	0.580	0.547
Precision	0.772	0.689	0.529
Recall	0.774	0.622	0.698
F1-Score	0.762	0.598	0.607
Accuracy	0.745	0.681	0.721
Starting Get Important Features Function

ibm corporate employee attrition analysis.ipynb
File Edit View Insert Runtime Tools Help
+ Code + Text
Cross Validation Error for each alpha
Confusion Matrix

ibm corporate employee attrition analysis.ipynb
File Edit View Insert Runtime Tools Help
+ Code + Text
Cross Validation Error for each alpha

2.	Tune the Model	Hyperparameter Tuning – 80% Validation Method – 1.predictive analysis 2.svm model 3. poly kernel 4. RBF Kernel 5.regression analysis	

ibm corporate employee attrition analysis.ipynb
Function for HyperParameter Tuning

ibm corporate employee attrition analysis.ipynb
designing ML Algo For the Best Fit HyperParameter Value

ibm corporate employee attrition analysis.ipynb
designing ML Algo For the Best Fit HyperParameter Value

Model Performance Testing:

Project team shall fill the following information in model performance testing template.

S.No.	Parameter	Screenshot / Values

1.	Dashboard design	No of Visulizations / Graphs – 8 LOADING THE DATASET: PREPARING THE DATA & EXPLORATION OF DATA:

CREATION OF VISUALIZATION CHARTS

EMPLOYEE ATTRITION STATUS:

ATTRITION BY BUSINESS TRAVEL:

CREATION OF VISUALIZATION CHARTS

EMPLOYEE ATTRITION STATUS:

ATTRITION BY BUSINESS TRAVEL:

ATTRITION BY SALARY HIKE PERCENT:

		  ITION BY WORK EXPERIENCE GROUPS: BOARD OF ATTRITION OF EMPLOYEES BASED MPLOYMENT DETAILS
2.	Data Responsiveness	▯ Employee Attrition by Age ▯ Attrition by Business Travel ▯ Attrition by Department, Job Role, Education Level and Marital Status ▯ Attrition by Salary Hike Percent

		▯ Attrition by No. of Companies Worked

		• Attrition by Income Groups • Attrition by Work Experience Groups Dashboard of Attrition of Employees based on Employment details
3.	Amount Data to Rendered (DB2 Metrics)	General_data.csv, Employee_Survey_Data.csv, Manager_Survey_data.csv
4.	Utilization of Data Filters	Grouping Sections Auto general
5.	Effective User Story	No of Scene Added - 8
6.	Descriptive Reports	No of Visulizations / Graphs – 4

Model Performance Testing:

Project team shall fill the following information in model performance testing template.

S.No.	Parameter	Values	Screenshot
1.	Model Summary Regression analysis	for best_alpha = 0.01 For values of best alpha = 0.01 The train log loss is: 0.3571556733065044 For values of best alpha = 0.01 The cross validation log loss is: 0.38726151334759856 For values of best alpha = 0.01 The test log loss is: 0.3819696610792824 +----------+----------+---------+----------+ | Metric | Train value | CV value | Test value | +----------+----------+---------+----------+ | Log-Loss | 0.357 | 0.387 | 0.382 | | Precision | 0.771 | 0.824 | 0.706 | | Recall | 0.144 | 0.126 | 0.086 | | F1-Score | 0.242 | 0.219 | 0.154 | | Accuracy | 0.855 | 0.855 | 0.847 |	
	SVM Model	linear linear [0.4001705021624625, 0.40017050214069255, 0.4001705020945628, 0.3930644655688551, 0.392335542092979, 0.392432524331171, 0.3927347253426151] [1e-06, 1e-05, 0.0001, 0.001, 0.01, 0.1, 1] ******bestHypermeterFunc Completed******** for best_alpha = 0.01 For values of best alpha = 0.01 The train log loss is: 0.3605788274237615 For values of best alpha = 0.01 The cross validation log loss is: 0.392335542092979 For values of best alpha = 0.01 The test log loss is: 0.37951616221945994 +----------+----------+---------+----------+ | Metric | Train value | CV value | Test value | +----------+----------+---------+----------+ | Log-Loss | 0.361 | 0.392 | 0.38 | | Precision | 0.652 | 0.75 | 0.5 | | Recall | 0.097 | 0.135 | 0.079 | | F1-Score | 0.168 | 0.229 | 0.137 | | Accuracy | 0.846 | 0.853 | 0.838 | +----------+----------+---------+----------+ **********	

---------- Recall ----------

---------- Precision ----------

Metric	Train value	CV value	Test value
Log-Loss	0.528	0.582	0.547
Precision	0.732	0.289	0.329
Recall	0.774	0.622	0.698
F1-Score	0.752	0.394	0.447
Accuracy	0.745	0.692	0.721

Starting Get Important Features Function
linear
****

Validation Accuracy – 76%

3.	Confidence Score (Only Yolo Projects)	Confidence Score – for alpha = 1e-06 Log Loss : 0.5544280465616621 for alpha = 1e-05 Log Loss : 0.5544279653425883 for alpha = 0.0001 Log Loss : 0.5544279626082605 for alpha = 0.001 Log Loss : 0.554427965596575 for alpha = 0.01 Log Loss : 0.5520345548670238 for alpha = 0.1 Log Loss : 0.4446536394141772 for alpha = 1 Log Loss : 0.23267354410235444 poly [0.5544280465616621, 0.5544279653425883, 0.5544279626082605, 0.554427965596575, 0.5520345548670238, 0.4446536394141772, 0.23267354410235444] [1e-06, 1e-05, 0.0001, 0.001, 0.01, 0.1, 1] ******bestHypermeterFunc Completed******* for best_alpha = 1 For values of best alpha = 1 The train log loss is: 0.05843126754768401 For values of best alpha = 1 The cross validation log loss is: 0.23267354410235444 0.23267354410235444 For values of best alpha = 1 The test log loss is: 0.19790587725653494 ---------- Confusion Matrix ---------- ---------- Recall ----------	

---------- Precision ----------

Metric	Train_value	CV_value	Test_value
Log-Loss	0.058	0.233	0.198
Precision	0.985	0.802	0.807
Recall	0.99	0.838	0.871
F1-Score	0.988	0.819	0.837
Accuracy	0.988	0.94	0.945

Starting Get Important Features Function
poly
Kernel is poly so, we cannot get the important Features using Coef_ Funtion

10. ADVANTAGES & DISADVANTAGES

CURRENT **High Turnover**	ALTERNATIVE **High Retention**
• Cost: 16-212% employee annual salary per lost employee • Lost productivity • Disrupted work environment • Low morale • Increased miscommunication • More mistakes or accidents • Low employee engagement and little employee-driven improvement	• Greater profits (from increased sales as well as saving on costs) • Better workplace culture • Stronger employee relationships • Improved communication • Higher workplace morale • Happier customers and more promoters • Employee-driven innovation

11. CONCLUSION

The following suggestion are given based on the analysis and modelling result:

CURRENT EMPLOYEES:

- Work life balance should be improved
- Work environment should be improved
- The manager of an employee should not be changed very often
- Employees should be provided relevant training regularly, especially for its younger employees

FUTURE EMPLOYEES (CHANGES IN HIRING PROCESS):

- The company should follow either one of the strategies given below –
    - Hire older people with decent work experience
    - Hire young people and train them appropriately
- It could also opt for a combination of the two

12. FUTURE SCOPE

The future scope of the research is that these analysis and modelling helps in forecasting the cause of employee disengagement, enables HR managers develop long-term strategies to reduce attrition, Competitive measures to enhance company brand image, Develops and shapes drills that benefit both the management and the employees. The scope of this research can be extended to many numbers of samples and to other working fields other than corporates

13. APPENDIX

Nowadays, employee attrition became a serious issue regarding a company's competitive advantage. It's very expensive to find, hire and train new talents. It's more cost-effective to keep the employees a company already has. A company needs to maintain a pleasant working atmosphere to make their employees stay in that company for a longer period. A few years back it was done manually but it is an era of machine learning and data analytics. Now, a company's HR department uses some data analytics tool to identify which areas to be modified to make most of its employees to stay.

Why are we using logistic regression to analyze employee attrition?

Whether an employee is going to stay or leave a company, his or her answer is just binomial i.e. it can be “YES” or “NO”. So, we can see our dependent variable Employee Attrition is just a categorical variable. In the case of a dependent categorical variable, we can not use linear regression, in that case, we have to use “LOGISTIC REGRESSION“.

Methodology

Here, I am going to use 5 simple steps to analyze Employee Attrition using R software

1. DATA COLLECTION
2. DATA PRE PROCESSING
3. DIVIDING THE DATA into TWO PARTS “TRAINING” AND “TESTING”
4. BUILD UP THE MODEL USING “TRAINING DATA SET”
5. DO THE ACCURACY TEST USING “TESTING DATA SET”

Data Exploration

A large company named XYZ, employs, at any given point of time, around 4000 employees. However, every year, around 15% of its employees leave the company. Since the attrition level is too high, the management wants to use predictive modelling to bring it down.

Hence, the objectives of the analysis are to:

•Help company XYZ identify current employees that are very likely to leave

•Recommend ways for company XYZ to decrease its attrition level in the future

The analysis is divided into three parts:

•Data Understanding – Source of data, patterns in the data

•Predictive modelling of attrition

•Recommending ways for company XYZ to decrease its level of attrition

A quick look at the dataset:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	Attrition	BusinessTra	DailyRate	Departmen	DistanceFrc	Education	EducationFi	EmployeeC	EmployeeN	Environmer	Gender	HourlyRate	JobInvolver	JobLevel	JobRole	JobSatisfact	MaritalStat
2	Yes	Travel_Rare	1102	Sales	1	2	Life Science	1	1	2	Female	94	3	2	Sales Execu	4	Single
3	No	Travel_Frec	279	Research &	8	1	Life Science	1	2	3	Male	61	2	2	Research Sc	2	Married
4	Yes	Travel_Rare	1373	Research &	2	2	Other	1	4	4	Male	92	2	1	Laboratory	3	Single
5	No	Travel_Frec	1392	Research &	3	4	Life Science	1	5	4	Female	56	3	1	Research Sc	3	Married
6	No	Travel_Rare	591	Research &	2	1	Medical	1	7	1	Male	40	3	1	Laboratory	2	Married
7	No	Travel_Frec	1005	Research &	2	2	Life Science	1	8	4	Male	79	3	1	Laboratory	4	Single
8	No	Travel_Rare	1324	Research &	3	3	Medical	1	10	3	Female	81	4	1	Laboratory	1	Married
9	No	Travel_Rare	1358	Research &	24	1	Life Science	1	11	4	Male	67	3	1	Laboratory	3	Divorced
10	No	Travel_Frec	216	Research &	23	3	Life Science	1	12	4	Male	44	2	3	Manufactur	3	Single
11	No	Travel_Rare	1299	Research &	27	3	Medical	1	13	3	Male	94	3	2	Healthcare	3	Married
12	No	Travel_Rare	809	Research &	16	3	Medical	1	14	1	Male	84	4	1	Laboratory	2	Married
13	No	Travel_Rare	153	Research &	15	2	Life Science	1	15	4	Female	49	2	2	Laboratory	3	Single
14	No	Travel_Rare	670	Research &	26	1	Life Science	1	16	1	Male	31	3	1	Research Sc	3	Divorced
15	No	Travel_Rare	1346	Research &	19	2	Medical	1	18	2	Male	93	3	1	Laboratory	4	Divorced
16	Yes	Travel_Rare	103	Research &	24	3	Life Science	1	19	3	Male	50	2	1	Laboratory	3	Single
17	No	Travel_Rare	1389	Research &	21	4	Life Science	1	20	2	Female	51	4	3	Manufactur	1	Divorced
18	No	Travel_Rare	334	Research &	5	2	Life Science	1	21	1	Male	80	4	1	Research Sc	2	Divorced
19	No	Non-Travel	1123	Research &	16	2	Medical	1	22	4	Male	96	4	1	Laboratory	4	Divorced
20	No	Travel_Rare	1219	Sales	2	4	Life Science	1	23	1	Female	78	2	4	Manager	4	Married
21	No	Travel_Rare	371	Research &	2	3	Life Science	1	24	4	Male	45	3	1	Research Sc	4	Single
22	No	Non-Travel	673	Research &	11	2	Other	1	26	1	Female	96	4	2	Manufactur	3	Divorced

	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE	AF	AG	AH	AI
1	MonthlyIncome	MonthlyRate	NumCompaniesW	Over18	OverTime	PercentSala	Performanc	Relationshi	StandardHc	StockOptio	TotalWorki	TrainingTin	WorkLifeBa	YearsAtCor	YearsInCur	YearsSincel	YearsWithC	Age
2	5993	19479	8	Y	Yes	11	3	1	80	0	8	0	1	6	4	0	5	41
3	5130	24907	1	Y	No	23	4	4	80	1	10	3	3	10	7	1	7	49
4	2090	2396	6	Y	Yes	15	3	2	80	0	7	3	3	0	0	0	0	37
5	2909	23159	1	Y	Yes	11	3	3	80	0	8	3	3	8	7	3	0	33
6	3468	16632	9	Y	No	12	3	4	80	1	6	3	3	2	2	2	2	27
7	3068	11864	0	Y	No	13	3	3	80	0	8	2	2	7	7	3	6	32
8	2670	9964	4	Y	Yes	20	4	1	80	3	12	3	2	1	0	0	0	59
9	2693	13335	1	Y	No	22	4	2	80	1	1	2	3	1	0	0	0	30
10	9526	8787	0	Y	No	21	4	2	80	0	10	2	3	9	7	1	8	38
11	5237	16577	6	Y	No	13	3	2	80	2	17	3	2	7	7	7	7	36
12	2426	16479	0	Y	No	13	3	3	80	1	6	5	3	5	4	0	3	35
13	4193	12682	0	Y	Yes	12	3	4	80	0	10	3	3	9	5	0	8	29
14	2911	15170	1	Y	No	17	3	4	80	1	5	1	2	5	2	4	3	31
15	2661	8758	0	Y	No	11	3	3	80	1	3	2	3	2	2	1	2	34
16	2028	12947	5	Y	Yes	14	3	2	80	0	6	4	3	4	2	0	3	28
17	9980	10195	1	Y	No	11	3	3	80	1	10	1	3	10	9	8	8	29
18	3298	15053	0	Y	Yes	12	3	4	80	2	7	5	2	6	2	0	5	32
19	2935	7324	1	Y	Yes	13	3	2	80	2	1	2	2	1	0	0	0	22
20	15427	22021	2	Y	No	16	3	3	80	0	31	3	3	25	8	3	7	53
21	3944	4306	5	Y	Yes	11	3	3	80	0	6	3	3	3	2	1	2	38
22	4011	8232	0	Y	No	18	3	4	80	1	5	5	2	4	2	1	3	24
23	3407	6986	7	Y	No	23	4	2	80	0	10	4	3	5	3	0	3	36
24	11994	21293	0	Y	No	11	3	3	80	0	13	4	3	12	6	2	11	34
25	1232	19281	1	Y	No	14	3	4	80	0	0	6	3	0	0	0	0	21
26	2960	17102	2	Y	No	11	3	3	80	0	8	2	3	4	2	1	3	34

Take a look:

Data preparation

- ## Detect the missing values:
- We have to see if there are any missing values in the dataset.

```
anyNA(JOB_Attrition)
```

- Result: FALSE; i.e. there are no missing values in our data set ” JOB_Attrition”

- ## Change the data types:
- First of all, we have to change the data type of the dependent variable “Attrition”. It is given as “Yes” and “No” form i.e. it is a categorical variable. To make a proper model we have to convert it into numeric form. To do so, we will assign value 1 to “Yes” and value 0 to “No” and convert it into numeric.

```
JOB_Attrition$Attrition[JOB_Attrition$Attrition=="Yes"]=1
JOB_Attrition$Attrition[JOB_Attrition$Attrition=="No"]=0
JOB_Attrition$Attrition=as.numeric(JOB_Attrition$Attrition)
```

- Next, we will change all “character” variables into “Factor”

- There are 8 character variables: Business Travel, Department, Education, Education Field, Gender, Job role, Marital Status, Over Time. There column numbers are 2,4,6,7,11,15,17,22 respectively.

```
JOB_Attrition[,c(2,4,6,7,11,15,17,22)]=lapply(JOB_Attrition[,c(2,4,6,7,11,15,17,22)],as.factor)
```

- Lastly, there is one other variable ” Over 18” which has all inputs as “Y”. It is also a character variable. We will transform into numeric as it has only one level so transforming into factor will not provide a good result. To do so, we will assign value 1 to “Y” and transform it into numeric.

```
JOB_Attrition$Over18[JOB_Attrition$Over18=="Y"]=1
JOB_Attrition$Over18=as.numeric(JOB_Attrition$Over18)
```

Splitting the dataset into “training” and “testing”

In any regression analysis, we have to split the dataset into 2 parts:

1. TRAINING DATA SET

2. TESTING DATA SET

With the help of the Training data set we will build up our model and test its accuracy using the Testing Data set.

```
set.seed(1000)
ranuni=sample(x=c("Training","Testing"),size=nrow(JOB_Attrition),replace=T,prob=c(0.7,0.3)
)
TrainingData=JOB_Attrition[ranuni=="Training",
]
TestingData=JOB_Attrition[ranuni=="Testing",]
nrow(TrainingData) nrow(TestingData)
```

We have successfully split the whole data set into two parts. Now we have 1025 Training data & 445 Testing data.

Building up the model

We are now going to build up the model following some simple steps as follows:

1. Identify the independent variables
2. Incorporate the dependent variable “Attrition” in the model
3. Transform the data type of model from “character” to “formula”
4. Incorporate TRAINING data into the formula and build the model

```
independentvariables=colnames(JOB_Attrition[,2:35])
independentvariables
Model=paste(independentvariables,collapse="+")
Model
Model_1=paste("Attrition~",Model)
Model_1
class(Model_1)
formula=as.formula(Model_1)
formula
```

Output:

```
> formula
Attrition ~ BusinessTravel + DailyRate + Department + DistanceFromHome +
    Education + EducationField + EmployeeCount + EmployeeNumber +
    EnvironmentSatisfaction + Gender + HourlyRate + JobInvolvement +
    JobLevel + JobRole + JobSatisfaction + MaritalStatus + MonthlyIncome +
    MonthlyRate + NumCompaniesWorked + Over18 + OverTime + PercentSalaryHike +
    PerformanceRating + RelationshipSatisfaction + StandardHours +
    StockOptionLevel + TotalWorkingYears + TrainingTimesLastYear +
    WorkLifeBalance + YearsAtCompany + YearsInCurrentRole + YearsSinceLastPromotion +
    YearsWithCurrManager + Age
```

Next, we will incorporate “Training Data” into the formula using the “glm” function and build up a logistic regression model.

```
Trainingmodel1=glm(formula=formula,data=TrainingData,family="binomial")
```

Now, we are going to design the model by the “Stepwise selection” method to fetch significant variables of the model. Execution of the code will give us a list of output where the variables are added and removed based on our significance of the model. The AIC value at each level reflects the goodness of the respective model. As the value keeps dropping it leads to a better fitting logistic regression model.

The application of the summary on the final model will give us the list of final significant variables and their respective important information.

```
Trainingmodel1=step(object = Trainingmodel1,direction = "both")
summary(Trainingmodel1)
```

```
                                Estimate Std. Error z value Pr(>|z|)
(Intercept)                      2.39506    1.54588   1.549 0.121306
BusinessTravelTravel_Frequently  2.27473    0.54975   4.138 3.51e-05 ***
BusinessTravelTravel_Rarely      1.22619    0.51430   2.384 0.017117 *
DistanceFromHome                 0.04415    0.01299   3.400 0.000675 ***
EducationFieldLife Sciences     -0.64408    0.97046  -0.664 0.506894
EducationFieldMarketing          0.31719    1.03345   0.307 0.758906
EducationFieldMedical           -0.78963    0.97159  -0.813 0.416378
EducationFieldOther             -0.57202    1.07902  -0.530 0.596023
EducationFieldTechnical Degree   0.24200    0.99122   0.244 0.807121
EnvironmentSatisfaction         -0.43843    0.10244  -4.280 1.87e-05 ***
GenderMale                       0.43233    0.22638   1.910 0.056162 .
JobInvolvement                  -0.52721    0.14711  -3.584 0.000339 ***
JobRoleHuman Resources           1.52115    0.81078   1.876 0.060634 .
JobRoleLaboratory Technician     1.33100    0.54552   2.440 0.014692 *
JobRoleManager                   0.41330    0.79632   0.519 0.603753
JobRoleManufacturing Director    0.47421    0.62340   0.761 0.446848
JobRoleResearch Director       -15.78918  703.79094  -0.022 0.982101
JobRoleResearch Scientist        0.58586    0.54412   1.077 0.281608
JobRoleSales Executive           0.75873    0.55817   1.359 0.174041
JobRoleSales Representative      1.42601    0.63391   2.250 0.024479 *
JobSatisfaction                 -0.34296    0.09875  -3.473 0.000515 ***
MaritalStatusMarried             0.47517    0.31249   1.521 0.128365
MaritalStatusSingle              1.45219    0.32090   4.525 6.03e-06 ***
NumCompaniesWorked               0.19962    0.04821   4.140 3.47e-05 ***
OverTimeYes                      1.92209    0.23446   8.198 2.44e-16 ***
PercentSalaryHike               -0.04655    0.03199  -1.455 0.145653
RelationshipSatisfaction        -0.35975    0.10090  -3.565 0.000363 ***
TotalWorkingYears               -0.11388    0.03519  -3.236 0.001213 **
TrainingTimesLastYear           -0.22353    0.09093  -2.458 0.013964 *
WorkLifeBalance                 -0.34294    0.14989  -2.288 0.022138 *
YearsAtCompany                   0.13557    0.04963   2.731 0.006309 **
YearsInCurrentRole              -0.19590    0.05846  -3.351 0.000805 ***
YearsSinceLastPromotion          0.14191    0.05045   2.813 0.004909 **
YearsWithCurrManager            -0.10590    0.05871  -1.804 0.071286 .
Age                             -0.02582    0.01623  -1.591 0.111630
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 891.31  on 1024  degrees of freedom
Residual deviance: 581.72  on  990  degrees of freedom
AIC: 651.72

Number of Fisher Scoring iterations: 17
```

From our above result we can see, Business travel, Distance from home, Environment satisfaction, Job involvement, Job satisfaction, Marital status, Number of companies worked, Over time, Relationship satisfaction, Total working

years, Years at the company, years since last promotion, years in the current role all these are most significant variables in determining employee attrition. If the company mostly looks after these areas then there will be a lesser chance of losing an employee.

A quick visualization to see how much these variables affect "attrition"

Here I have used Tableau for these visualizations; isn't it beautiful? This software just makes our work easier.

Now, we can perform the Hoshmer-Lemeshow goodness of fit test on the data set, to judge the accuracy of the predicted probability of the model.

The hypothesis is:

H0: The model is a good fit.

H1: The model is not a good fit.

If, p-value>0.05 we will accept H0 and reject H1.

To perform the test in R we need to install the mkMisc package.

```
HLgof.test(fit=Trainingmodel1$fitted.values,obs=Trainingmodel1$y)
```

```
        Hosmer-Lemeshow H statistic

data:  Trainingmodel1$fitted.values and Trainingmodel1$y
X-squared = 11.555, df = 8, p-value = 0.1722
```

Here, we can see the p-value is greater than 0.05, hence we will accept H0. Now, it is proved that our model is a well fitted one.

Generating a ROC curve for training data

Another technique to analyze the goodness of fit of logistic regression is the ROC measures(Receiver Operating characteristics). The ROC measures are sensitivity, 1Specificity, False Positive, and False Negative. The two measures we use extensively are Sensitivity and Specificity. The sensitivity measures the goodness of accuracy of the model while specificity measures the weakness of the model. To do this in R we need to install a package pROC.

```
troc=roc(response=Trainingmodel1$y,predictor = Trainingmodel1$fitted.values,plot=T)
troc$auc
```

The area under the curve: 0.8759

Interpretation of the figure:

The plot of these two measures gives us a concave plot which shows as sensitivity is increasing 1-specificity is increasing but at a diminishing rate. The C-value(AUC) or the value of the concordance index gives the measure of the area under the ROC curve. If c=0.5 then it would have meant that the model can not perfectly discriminate between 0 and 1 responses. Then it implies that the initial model can not perfectly say which employees are going to leave and who are going to stay.

But, here we can see our c-value is far greater than 0.5. It is 0.8759. Our model can perfectly discriminate between 0 and 1. Hence, we can successfully conclude it is a well-fitted model.

Creating the classification table for the training data set:

```
trpred=ifelse(test=Trainingmodel1$fitted.values>0.5,yes = 1,no=0)
table(Trainingmodel1$y,trpred)
```

The above code states, the predicted value of the probability greater than 0,.5 then the value of the status is 1 else it is 0. based on this criterion this code relabels ‘Yes’ and ‘No’ Responses of “Attrition”. Now, it is important to understand the percentage of predictions that match the initial belief obtained from the data set. Here we will compare (1-1) and (0-0) pair.

We have 1025 training data. We have predicted {(839+78)/1025}*100=89% correctly.

Comparing the result with testing data:

We will now compare the model with testing data. It is much like an accuracy test.

```
testpred=predict.glm(object=Trainingmodel1,newdata=TestingData,type = "response")
testpred
tsroc=roc(response=TestingData$Attrition,predictor = testpred,plot=T)
tsroc$auc
```

Now, We have incorporated Testing data into the training model and will see the ROC.

The area under the curve: 0.8286(c-value). It is also far higher than 0.5. It is also a well-fitted model.

Creating the classification table for the testing data set

```
testpred=ifelse(test=testpred>0.5,yes=1,no=0)
table(TestingData$Attrition,testpred)
```

```
> table(TestingData$Attrition,testpred)
   testpred
      0   1
  0 362   7
  1  48  28
```

We have 445 Testing data. we have correctly predicted {(362+28)/445}*100=87.64%.

Consequently, we can say, our logistic regression model is a very good fitted model. Any employee attrition data set can be analyzed using this model.

What do you think is it a good model? Comment below

CONCLUSION:

We have successfully learned how to analyze employee attrition using “LOGISTIC REGRESSION” with the help of R software. Only with a couple of codes and a proper data set, a company can easily understand which areas needed to look after to make the workplace more comfortable for their employees and restore their human resource power for a longer period.