

Corporate Employee Attrition Analysis

A PROJECT REPORT

Submitted By

Team ID : PNT2022TMID21456.

Team Leader : NAGANATHAN M (9177119D126),

Team Member : AATHISHWARAN D (9177119D113),

Team Member : MEIYAPPAN A (9177119D125),

Team Member : SAHEEL AQTHAR S (9177119D131).

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

ELECTRONICS AND COMMUNICATION ENGINEERING.

THIAGARAJAR COLLEGE OF ENGINEERING,

MADUARI-625015.

TABLE OF CONTENTS

CHAPTER NO	TITLE	PAGE NO
1	INTRODUCTION	4
1.1	Project Overview	4
1.2	Purpose	4
2	LITERATURE SURVEY	5
2.1	Existing problem	5
2.2	References	5
2.3	Problem Statement Definition	6
3	IDEATION & PROPOSED SOLUTION	7
3.1	Empathy Map Canvas	7
3.2	Ideation & Brainstorming	8
3.3	Proposed Solution	11
3.4	Problem Solution fit	11
4	REQUIREMENT ANALYSIS	12
4.1	Functional requirement	12
4.2	Non-Functional requirements	13

CHAPTER NO	TITLE	PAGE NO
5	PROJECT DESIGN	14
5.1	Data Flow Diagrams	14
5.2	Solution & Technical Architecture	14
5.3	User Stories	15
6	PROJECT PLANNING & SCHEDULING	16
6.1	Sprint Planning & Estimation	16
6.2	Sprint Delivery Schedule	17
6.3	Reports from JIRA	17
7	CODING & SOLUTIONING	21
8	TESTING	28
8.1	Test Cases	28
8.2	User Acceptance Testing	28
9	RESULTS	29
10	ADVANTAGES & DISADVANTAGES	32
11	CONCLUSION	33
12	FUTURE SCOPE	33
13	APPENDIX	34

1. INTRODUCTION

1.1 Project overview

Employee attrition has become a vital problem across the world. It is one of the crucial issues faced by business leaders within companies where they lose the most talented employees. A good employee is always an asset to the organization and their resignation can lead to various problems like financial losses, overall performance, and loss of acquired knowledge. Furthermore, hiring new employees is far exorbitant, taxing, and time-consuming in comparison to recruiting the existing one. It is very time-consuming to recruit a new employee as it takes him months for training, adjusting to the culture, rules, and environment. Therefore, upcoming trends and technology using Machine Learning Algorithms must be exploited for the benefit of business organizations. Knowing the reason beforehand for the employee attrition, companies can mitigate this loss. This analysis provides a conclusive review of employee attrition from the data set IBM HR Analytics Employee Attrition Performance.

1.2 Purpose

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 managers in IT

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

1. LITERATURE SURVEY

2.1 Existing Problem

The Existing system includes only few attributes for analysis and also deals with qualitative observations and simple statistical analysis. The qualitative observations deal with data and can be observed through human senses. They do not involve measurements or number. Due to the increase in IOT and connected device, we now have access to so much of data and along with it an increase needs to manage and understand data.

2.2 References

1. From Big Data to Deep Data to support people analytics for employee attrition prediction, Nesrine Ben Yahia, Hlel Jihen, Ricardo Colomo-Palacio(2021)

2. Machine Learning Approach for Employee Attrition Analysis. Dr. R. S. Kamath | Dr. S. S. Jamsandekar | Dr. P. G. Naik ,Published in International Journal of Trend in Scientific Research and Development (ijtsrd), (March 2019)

3. Investigation of early career teacher attrition(ECT) and the impact of induction programs in Western Australia, Janine E. Wyatt, Michael O'Neill (2021)

2.3 Problem Statement Definition

- To create a dashboard and perform analysis of employee attrition in corporates using IBM Cognos analytics platform.
- To reduce the employee attrition rate through data analytics, data visualization by analysing the major factors that causes attrition.

3. IDEATION AND PROPOSED SOLUTION

3.1 Empathy Map Canvas

Template

Empathy map

Use this framework to develop a deep, shared understanding and empathy for other people. An empathy map helps describe the aspects of a user's experience, needs and pain points, to quickly understand your users' experience and mindset.

[Share template feedback](#)

1

Build empathy

The information you add here should be representative of the observations and research you've done about your users.

Says

what friends say
what boss say
what influencers say

Thinks and Feel?

what really counts major
preoccupation worries and
aspirations.

Says

- what friends say
- what boss say
- what influencers say
- Employees compensation
- Employees have been approached when you're not even connected
- Employees want to know what the future holds for them
- The gap between managers and employees
- Employees have been approached when you're not even connected
- Employees want to know what the future holds for them
- The gap between managers and employees

Thinks and Feel?

- what really counts major preoccupation worries and aspirations.

Gain

- "Want" needs measures of success
- obstacles
- Do's

Pain

- Fears frustrations obstacles


Need some
inspiration?

See a detailed version
of this template by
Michael van der
Meulen

[Open example](#)

3.2 Ideation & Brainstorming

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



Brainstorm & idea prioritization

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

- 10 minutes to prepare
- 1 hour to collaborate
- 2-6 people recommended

[Show template feedback](#)

Before you collaborate

A little bit of preparation goes a long way with this session. Here's what you need to do to get going.

10 minutes

- Team gathering**
Define who should participate in the session and send an invite. Share relevant information or pre-work ahead.
- Set the goal**
Think about the problem you'll be focusing on solving in the brainstorming session.
- Learn how to use the facilitation tools**
Use the Facilitator Superpowers to run a happy and productive session.

[Open entire](#)

Team Leader - Naganathan.M

Team Members: Arathivaran.D
Meiyappan.A
Sahel Akhtar.S

Define your problem statement

What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your first session.

5 minutes

PROBLEM

How might we [your problem statement]?

Key rules of brainstorming

To run an smooth and productive session:

- Stay in topic.
- Encourage wild ideas.
- Defer judgment.
- Listen to others.
- Go for volume.
- If possible, be visual.

Step-2: Brainstorm, Idea Listing and Grouping.

2 Brainstorm

Write down any ideas that come to mind that address your problem statement.

10 minutes

Tip
You can reuse a sticky note and for the past 1 hour to about 10 to 15 minutes.

Argument M	Antithesis D	Synthesis S	Solved Problem S
Analyze the workforce	Build teams according to personalities	Goal setting and engagement	Hire and fire the right people
Check each person's ability to work	Analyze the strength and weakness of each member.	Without goals is a struggle for most of the employees	Spend time defining skills, values and personalities that work best for the team.
Team with innovative ideas	Motivation of the employee to work	Self Actualization	The right fit could stay for years.
Because we get a lot of ideas from our customers, we have the same idea as our competitors. We need to get more ideas from our customers and our employees.	If you build a team that only focuses on making money, you may get a lot of money, but the chances that your team will stay together are slim.	Even needs: Pride and feeling of accomplishment.	Get rid of these people before they poison the waterhole.
When managing people, it can be difficult to remember who you've given feedback to, and when the feedback was.	If you tell your team that they're doing well, they may never improve.	Safety needs: Security and safety	Many employees who can't handle pressure, stress, culture should be fired.
Analysis can also happen through conversations with managers to those who are high performing groups if you don't have a system in place for checking feedback yet.	There are many ways to get decisions made and more people involved, but also take time to include people with a different point of view.	Physiological needs.	Engage your team and let them know that you're not just a manager, you're a coach and a mentor. It's not just about the money.

3 Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. Group all sticky notes that have been grouped, give each cluster a sentence like later. If a cluster is bigger than a sticky note, try and see if you can break it up into smaller sub-groups.

20 minutes

Tip
Add a sticky note to each cluster to make it easier to find, share, organize and collaborate around ideas as they evolve.

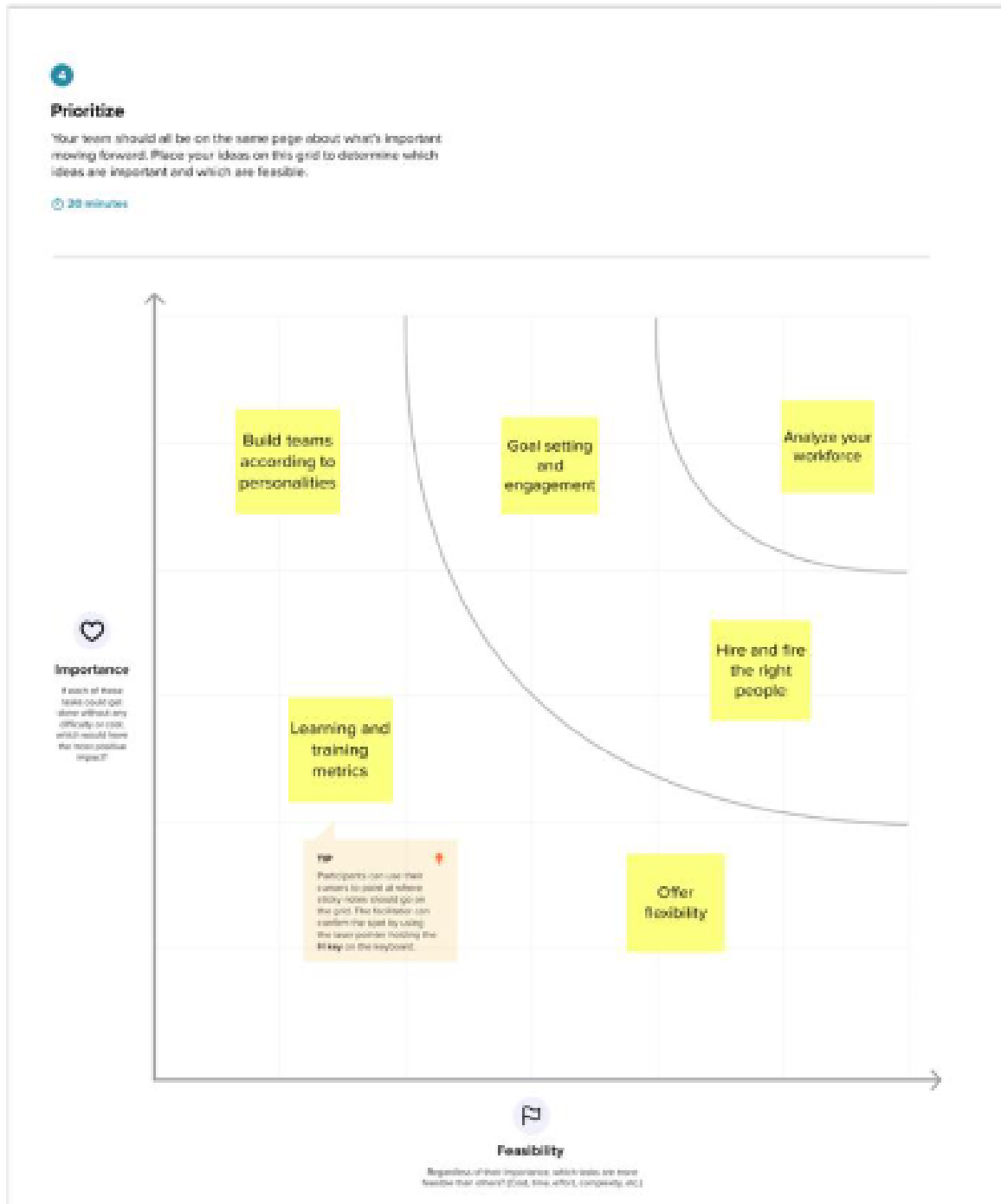
Turn out, your employees are adults. Working isn't fun, working remotely, flexible time off for family needs, parity a no-brainer. Have all the these increase employee satisfaction, and good employees stay productive anyway.

Overworked employees have little time for learning and staying motivated, so that's bad. A company that values learning and development sends the signal to employees that they want people to build careers, not just do a job. Learning and being can add value for the company, making it easier to perform better.

We wish all of our employees were intrinsically motivated to do a good job for its own sake, but without goals to meet and exceed, many of us struggle to get started. And you can't solve the problem just by throwing money at it.

Hire as far as you sound productive, but spend more time finding and ensuring that the benefits and pay you offer line up with regional and industry benchmarks.

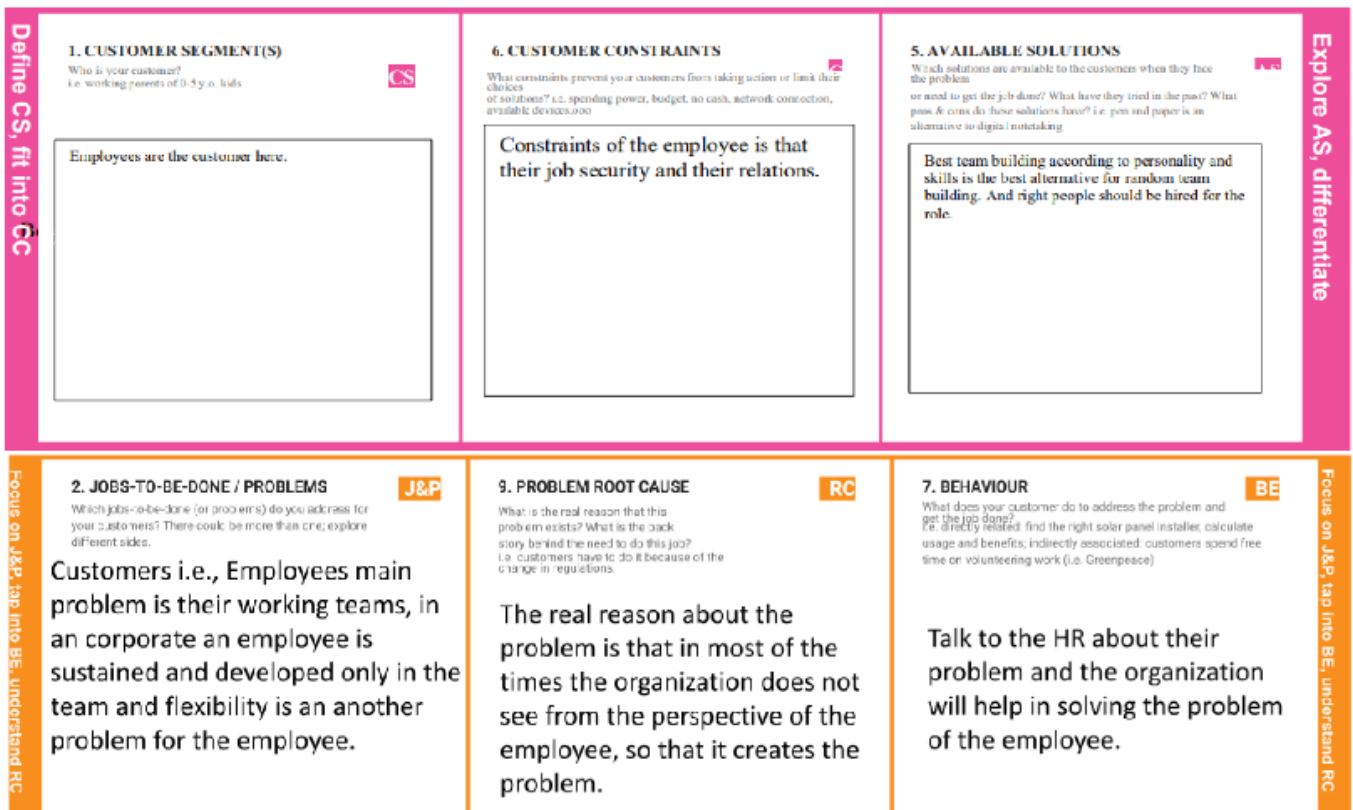
Step-3: Idea Prioritization.



3.3 Proposed Solution

The Existing system includes only few attributes for analysis and also deals with qualitative observations and simple statistical analysis. The qualitative observations deal with data and can be observed through human senses. They do not involve measurements or number. Due to the increase in IOT and connected device, we now have access to so much of data and along with it an increase needs to manage and understand data.

3.4 Problem Solution fit



3. TRIGGERS What triggers customers to act? inequality makes employee trigger and also the colleagues working with them, who are not fit for their role.	10. YOUR SOLUTION Solution to this problem is that, analyzing the workforce, building teams according to personalities, goal setting and engagement, learning and training metrics, hire and fire the right people and offer flexibility to the employees.	8. CHANNELS of BEHAVIOUR 8.1 ONLINE What kind of actions do customers take online? Through online mode, an employee can mail to the HR about the problem he/she is facing in the organization.
4. EMOTIONS: BEFORE / AFTER How do customers feel when they face a problem or in job and afterwards? Employees feel insecure and not in the environment of working when they face a problem and afterwards if employee discusses with the organization i.e., HR, may solve the problem of the employee.		8.2 OFFLINE What kind of actions do customers take offline? In offline mode, the employee can directly talk to the HR or the organization head about their problem and can be solved accordingly.

4. REQUIREMENT ANALYSIS

4.1 Functional requirement

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail Registration through LinkedIn
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	Account Creation	Create an account in the Profile Dashboard
FR-4	Input Credentials	Uploading your dataset Analyzing the attrition rate using dashboard
FR-5	Processing Methods	Using IBM Cognos Analytics Dashboard Using Prediction algorithm to find attrition rate
FR-6	Output Credentials	Using the Dashboard and Algorithm they know about the employee attrition and way to reduce the employee attrition
FR-7	Report preparation	Record the outcome of the algorithm in document.

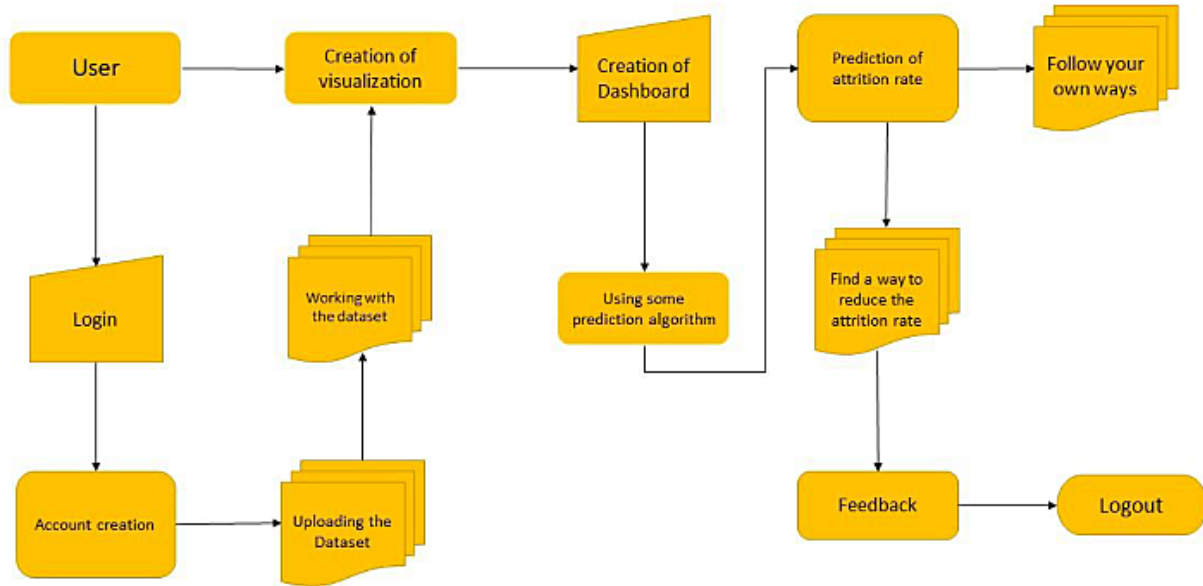
4.2 Non-Functional requirements

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

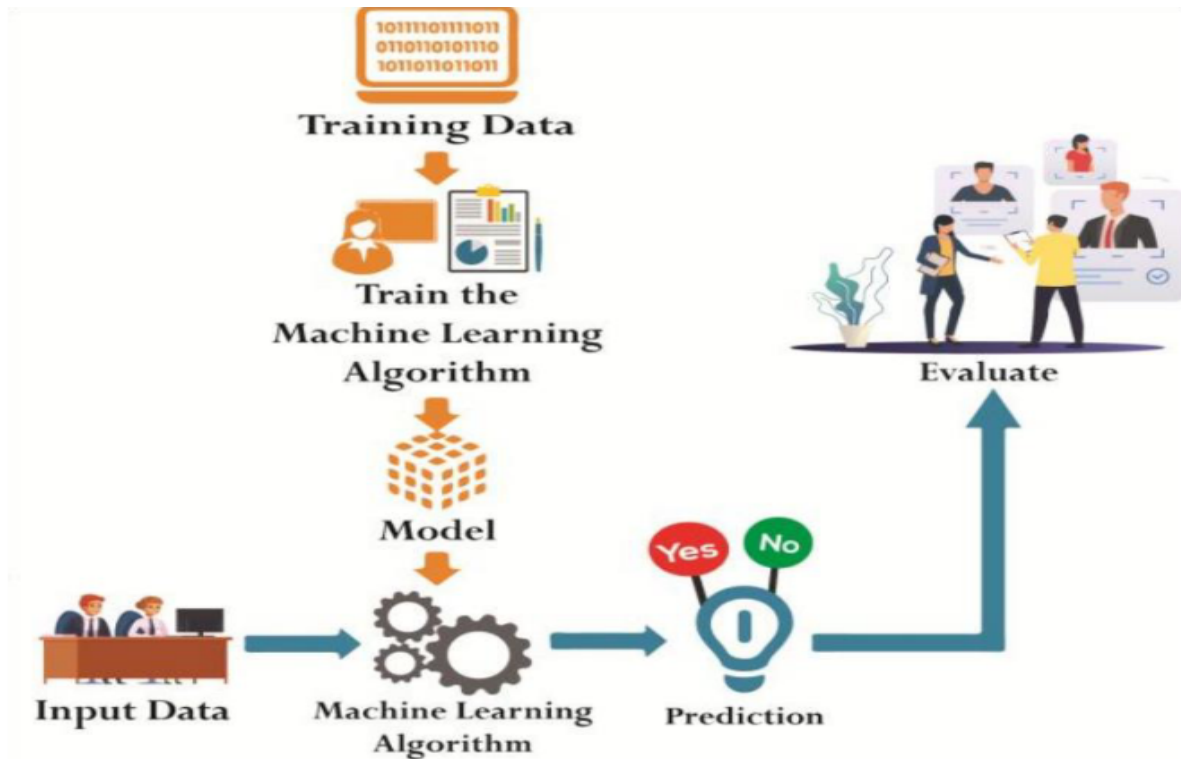
FR No.	Non-Functional Requirement	Description
NFR-1	Usability	The user can be able to interact with the system user friendly. The system is build with a simple modules and algorithms.
NFR-2	Security	Access permissions for the particular system information may only be changed by the system's data administrator. The user's data must be having an high security measures.
NFR-3	Reliability	The database update process must roll back all related updates when any update fails. The dataset will not be modified by anyone only the user can be able to modify the dataset.
NFR-4	Performance	The performance of the dashboard is flexible to every user's. The front-page load time must be no more than 2 seconds for users that access the website using an LTE mobile connection.
NFR-5	Availability	New module deployment mustn't impact front page, dashboard and check out pages availability and mustn't take longer than one hour. The rest of the pages that may experience problems must display a notification with a timer showing when the system is going to be up again.
NFR-6	Scalability	The website attendance limit must be scalable enough to support 200,000 users at a time. The dashboard is scalable for the companies when their employee's dataset is used for analysis. The model can successfully predict the futuristic approach and suggests preventive measures.

5. PROJECT DESIGN

5.1 Data Flow Diagrams



5.2 Solution & Technical Architecture



5.3 User Stories

Use the below template to list all the user stories for the product.

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Web user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
		USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail	I can register & access the dashboard with Gmail Login	Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password	I can access my account / dashboard	High	Sprint-1
	Dashboard	USN-6	Uploading the Dataset	I can be able to upload my dataset	High	Sprint 2
		USN-7	Working With Dataset	I can be able to access my dashboard	High	Sprint 2
		USN-8	Visualization	I can be able to view the visual attrition rate of my dataset	High	Sprint 3
		USN-9	Working with Dashboard	I can be able to view the various views of the attrition rate	High	Sprint 3
Customer Care Executive		USN-10	Asking Help / Feedback	I can be able to ask help if I can face any issues or problems while using the webpage	Medium	Sprint 4
Administrator		USN-11	Managing the Database	I can assure that my data is in secure state	High	Sprint 4
		USN-12	Managing the over all process	I can assure that my data and process is going good	High	Sprint 4

6. PROJECT PLANNING

6.1 Sprint Planning & Estimation

Product Backlog, Sprint Schedule, and Estimation (4 Marks)

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	2	High	Naganathan M
Sprint-1		USN-2	As a user, I will receive confirmation email once I have registered for the application	1	High	Saheel Aqthar S
Sprint-2		USN-3	As a user, I can register for the application through Facebook	2	Low	Aathishwaran D
Sprint-1		USN-4	As a user, I can register for the application through Gmail	2	Medium	Meiyappan A
Sprint-1	Login	USN-5	As a user, I can log into the application by entering email & password	2	High	Naganathan M
Sprint-2	Dashboard	USN-6	As a user, I can able to access the dashboard	4	Medium	Saheel Aqthar S
Sprint-2		USN-7	As a user, I can able to upload my dataset through dashboard	2	High	Aathishwaran D
Sprint-3		USN-8	As a user, I can able to done a Data Pre-processing	3	Medium	Meiyappan A
Sprint-3		USN-9	As a user, I can able to build a model for my dataset – Train the model	4	Low	Naganathan M

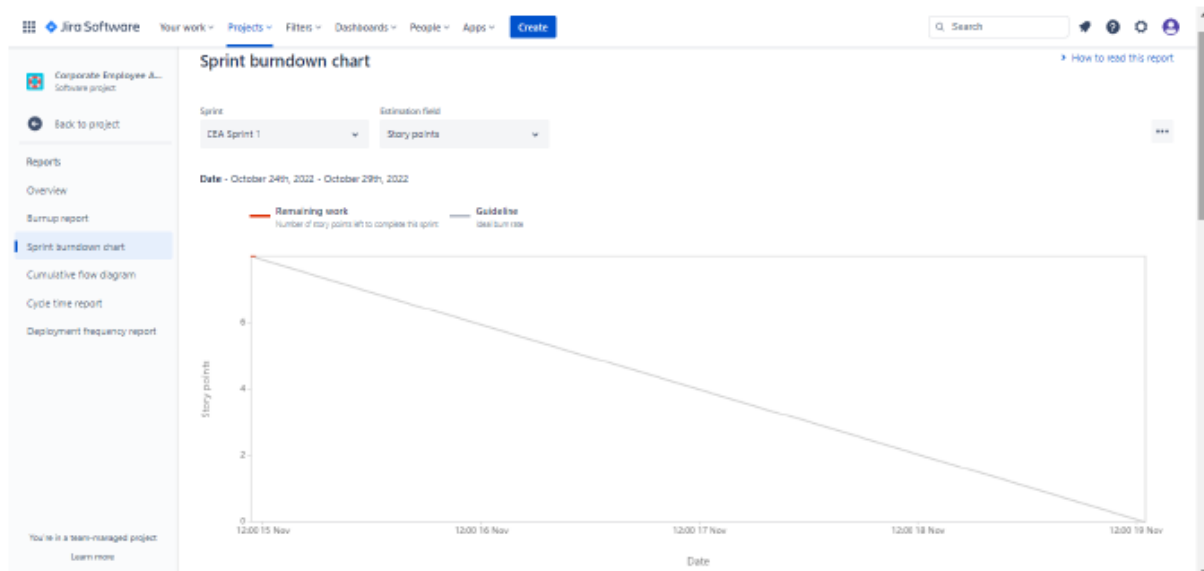
Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-3		USN-10	As a user, I can able to test my model	4	Low	Saheel Aqthar S
Sprint-3		USN-11	As a user, I can able to evaluate my performance	3	Medium	Aathishwaran D
Sprint-4		USN-12	As a user, I can able find a prediction of my dataset attrition rate using algorithm	5	High	Meiyappan A
Sprint-4		USN-13	As a user, I can able view the visualization of my dataset in the dashboard	5	High	Naganathan M
Sprint-2		USN-14	As a user, I can to ask the help to the development team	3	Low	Saheel Aqthar S
Sprint-4	Database	USN-15	As a user, I can assure that my information are in the safe state	5	Medium	Aathishwaran D
Sprint-2	Logout	USN-16	As a user, I can able to logout the page with my presence	2	Medium	Saheel Aqthar S

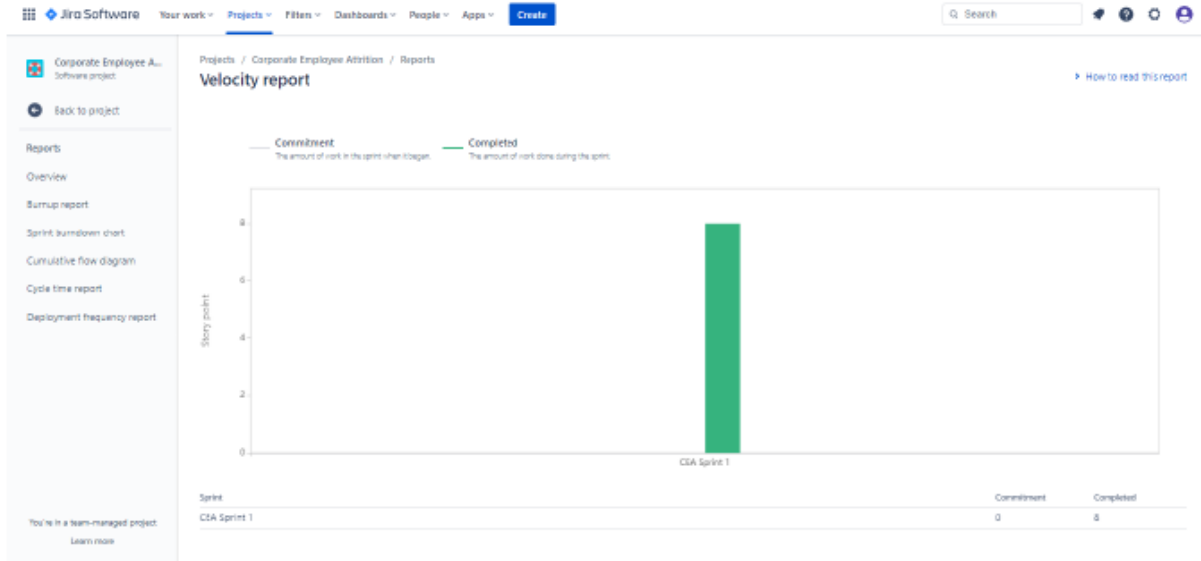
6.2 Sprint Delivery Schedule

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	7	6 Days	24 Oct 2022	29 Oct 2022	7	29 Oct 2022
Sprint-2	13	6 Days	31 Oct 2022	05 Nov 2022	13	05 Nov 2022
Sprint-3	14	6 Days	07 Nov 2022	12 Nov 2022	14	12 Nov 2022
Sprint-4	15	6 Days	14 Nov 2022	19 Nov 2022	15	19 Nov 2022

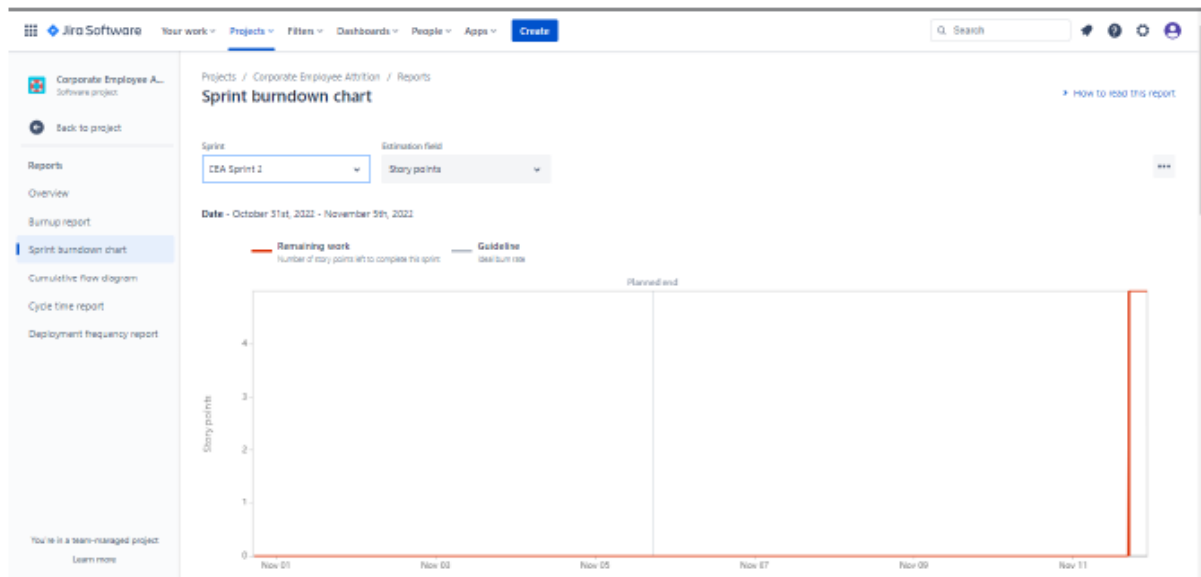
6.3 Reports from JIRA

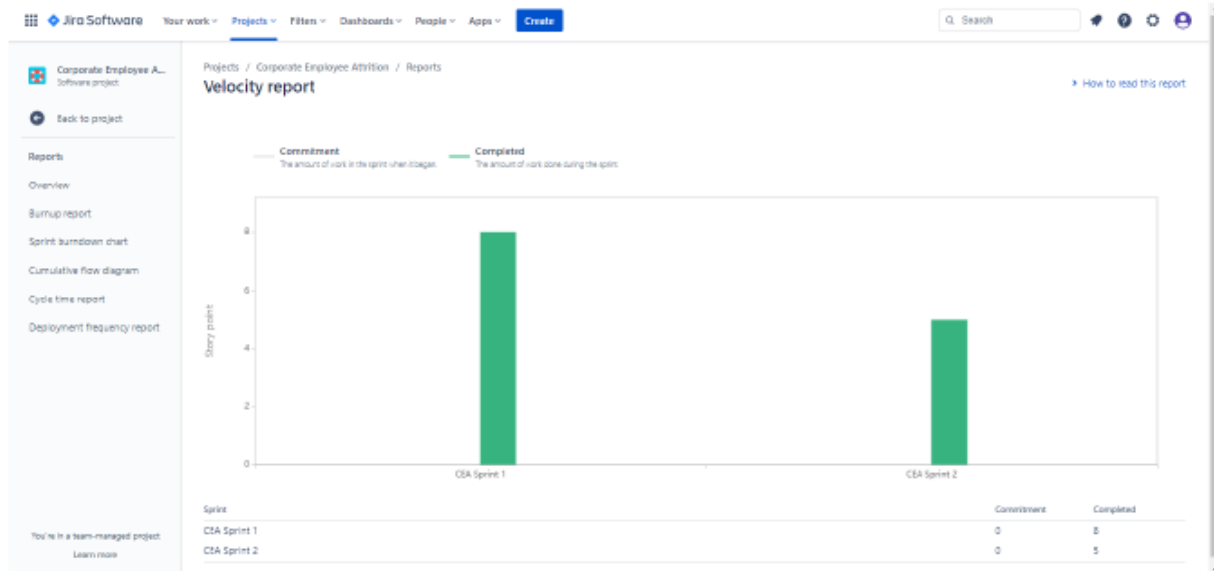
Sprint 1:



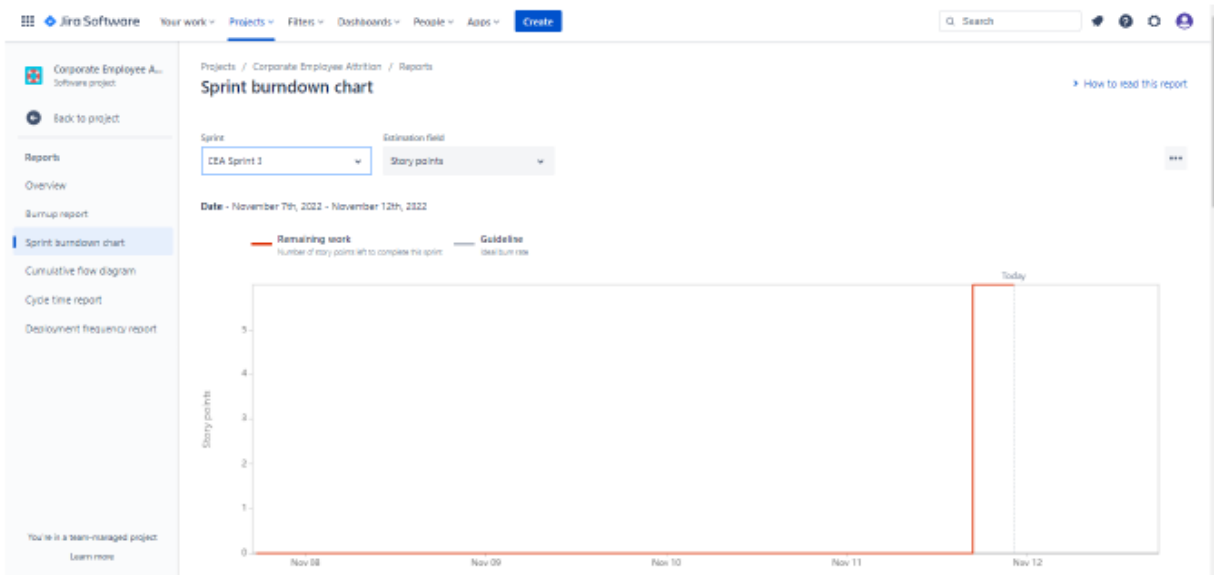


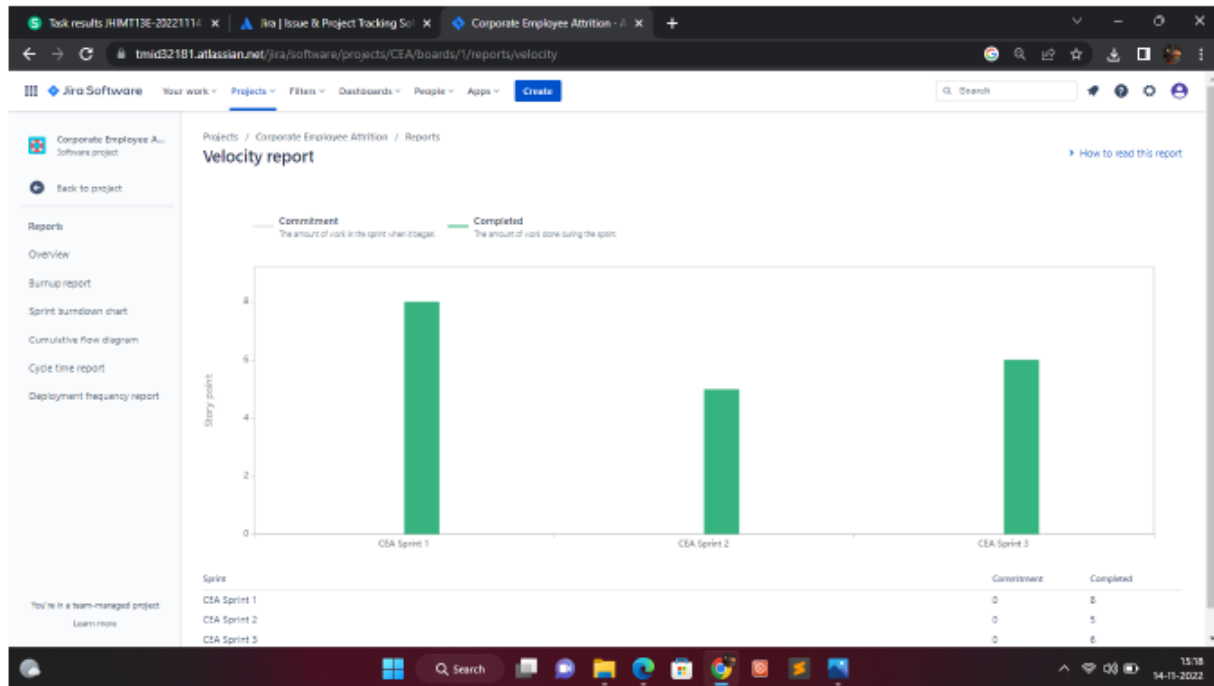
Sprint 2:



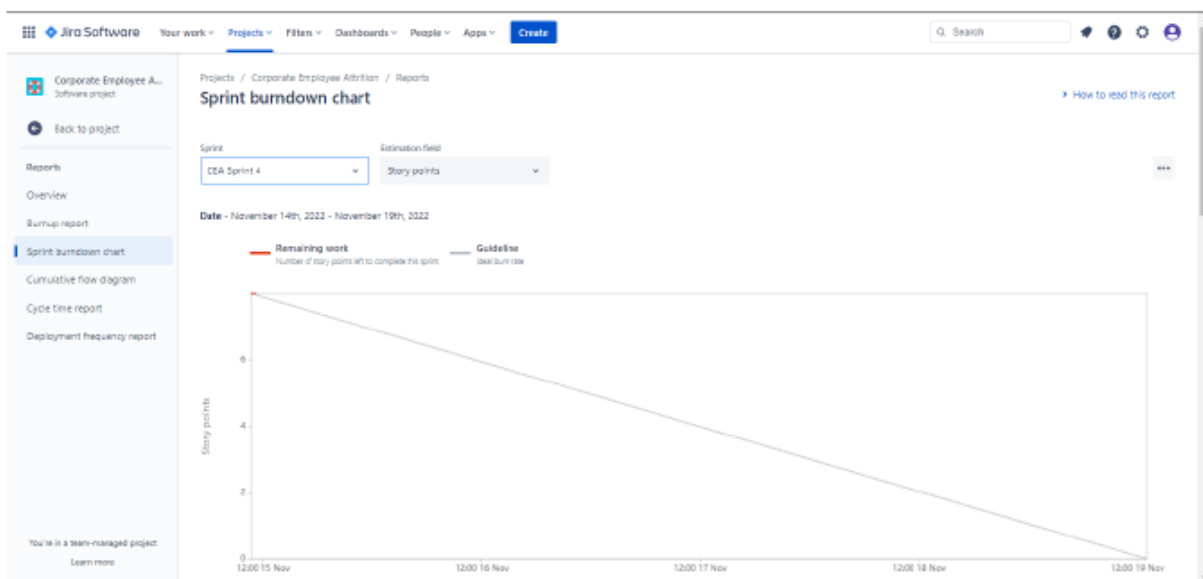


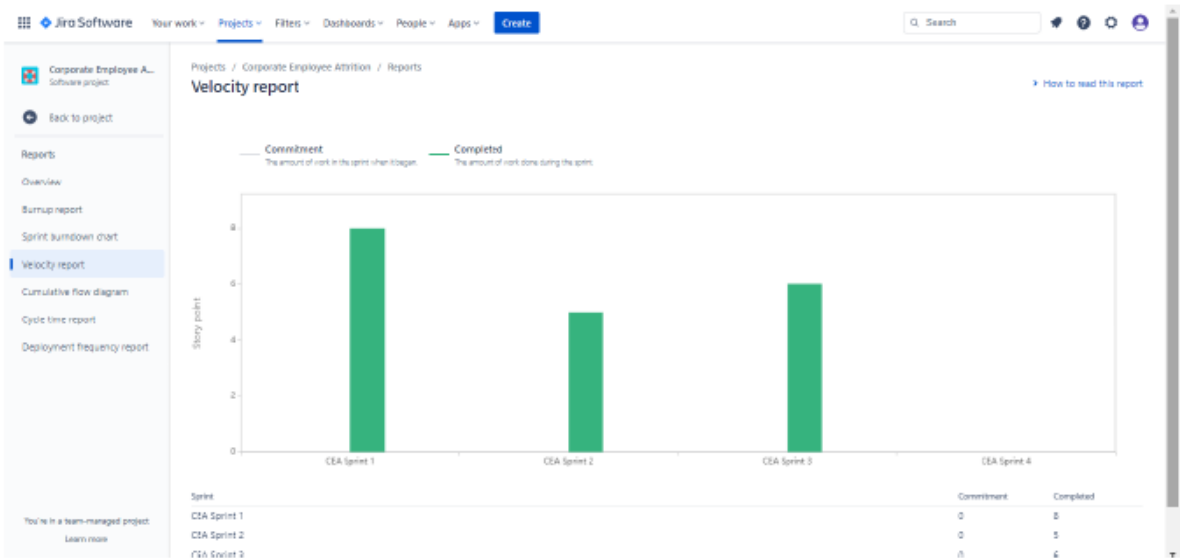
Sprint 3 :



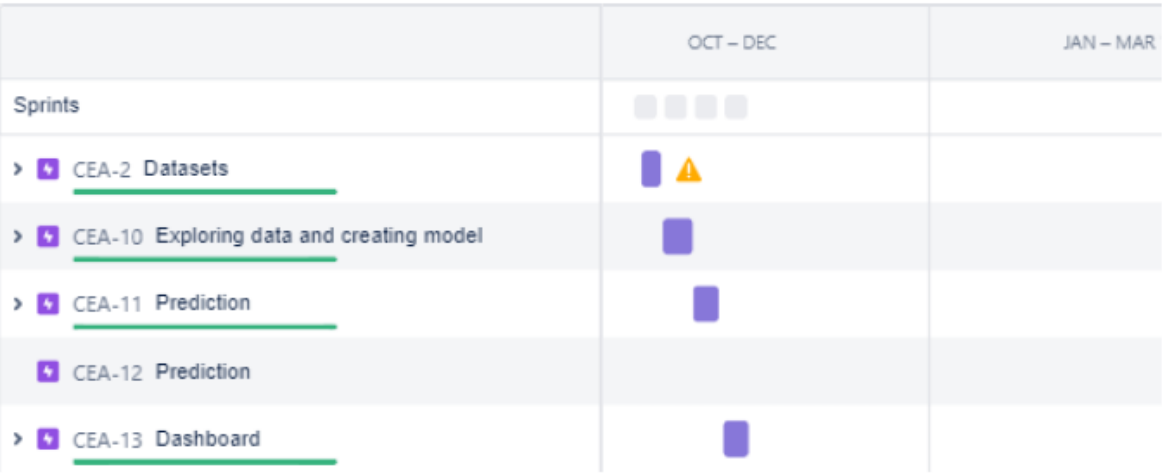


Sprint 4:





Road Map :



7. CODING & SOLUTIONING

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

#GENERAL
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

```

#FEATURE ENGINEERING
from sklearn.preprocessing import LabelEncoder
from imblearn.over_sampling import SMOTE
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
#MODEL SELECTION
from sklearn.model_selection import KFold
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import GridSearchCV
#MODEL
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC
from sklearn.tree import DecisionTreeClassifier
#MODEL SCORES
from sklearn.metrics import confusion_matrix , accuracy_score
, classification_report
#FEATURE IMPORTANCE
from sklearn.inspection import permutation_importance

path = '/content/drive/MyDrive/Colab Notebooks/HR-Employee-Attrition.csv'
df =pd.read_csv(path)
df

df.shape

df.info()

df.select_dtypes('int64' , 'float64').columns

cat_cols = df.select_dtypes('object').columns
cat_cols

df.describe().T

df

for cat in cat_cols:
    print(cat , '-> ' , df[cat].unique())
    print()

print("All columns Unique values count")
for col in df:
    print(col, len(df[col].unique()), sep=': ')

plt.figure(figsize =(14,5))
plt.subplot(1,2,1)

```

```

sns.countplot(df['Attrition'], color='b', hue=df['Gender'])
plt.title('Attrition by Gender')
plt.subplot(1,2,2)
plt.pie(df['Attrition'].value_counts(), colors=['r', 'c'], explode=[0,0.1],
autopct='%0.2f', labels=['No', 'Yes'])
plt.title('Attrition')

plt.figure(figsize=(16,4))
plt.subplot(1,3,1)
sns.distplot(df['Age'], color='m')
plt.title('Age')
plt.subplot(1,3,2)
sns.stripplot(x='Gender', y='Age', data=df, palette="Set2")
plt.title('Gender vs Age')
plt.subplot(1,3,3)
sns.countplot('Gender', data=df, color='c')
plt.title('Gender')
plt.tight_layout()

plt.figure(figsize=(14,13))
plt.subplot(2,1,1)
sns.countplot(y='JobRole', data=df, palette='winter_r')
plt.title('JOB ROLE')
plt.subplot(2,1,2)
sns.countplot(y='JobRole', data=df, palette='winter_r', hue=df['Attrition'])

plt.figure(figsize=(14,5))
plt.subplot(1,2,1)
sns.countplot('Department', data=df, hue='Attrition', palette='gist_rainbow_r')
plt.subplot(1,2,2)
plt.pie(df['Department'].value_counts(), autopct='%0.2f', colors=['r', 'c', 'g'], labels=['Research & Development', 'Sales', 'Human Resources'], explode=[0,0.1,0])

#HANDLING CATEGORICAL OUTPUT VARIABLE
df['Attrition'].replace({'Yes':1, 'No':0}, inplace=True)
df['Attrition'].head()

plt.figure(figsize=(14,10))
plt.subplot(2,2,1)
sns.countplot(df['JobSatisfaction'], hue=df['Attrition'], palette='Accent_r')
plt.subplot(2,2,2)
sns.countplot(df['EnvironmentSatisfaction'], hue=df['Attrition'], palette='Accent')
plt.subplot(2,2,3)

```

```

sns.countplot(df['JobInvolvement'], hue = df['Attrition'], palette='brg_r')
plt.subplot(2, 2, 4)
sns.countplot(df['PerformanceRating'], hue = df['Attrition'],
, palette='twilight_r')

plt.figure(figsize = (20 , 8))
sns.boxplot(x = 'JobRole', y = 'MonthlyIncome', data = df, hue = 'Attrition',
, color = 'red')

lt.figure(figsize = (12, 10))
plt.subplot(2, 1, 1)
sns.boxplot(x = 'MaritalStatus', y = 'RelationshipSatisfaction', data = df, hue
= 'Attrition', color = 'g')
plt.subplot(2, 1, 2)
sns.boxplot(df['JobLevel'], df['MonthlyIncome'], hue = df['Attrition'],
, palette='Reds_r')

col = ['YearsInCurrentRole', 'YearsSinceLastPromotion', 'YearsWithCurrManager',
, 'YearsAtCompany']
plt.figure(figsize = (10 , 10))
for i, c in enumerate(col):
    plt.subplot(2 , 2, i+1)
    sns.distplot(df[c], color = 'b')

plt.figure(figsize = (16 , 16))
sns.heatmap(df.corr(), cmap = 'ocean', cbar = True, annot = True)

no_use = []
for col in df.columns:
    if(len(df[col].unique()) == 1):
        no_use.append(col)
no_use

df.drop(columns = no_use, axis = 1, inplace = True)

df.columns

y_n_type = []
others = []
for col in df.select_dtypes('object').columns:
    if(len(df[col].unique()) == 2):
        y_n_type.append(col)

y_n_type

df['Gender'].replace({'Male':1, 'Female':0}, inplace = True)
df['OverTime'].replace({'Yes':1, 'No':0}, inplace = True)

```



```

others = df.select_dtypes('object').columns
others

le = LabelEncoder()
for col in others:
    df[col] = le.fit_transform(df[col])

df.select_dtypes('object').columns

x = df.drop('Attrition' ,axis =1)
y = df['Attrition']

print(x.shape ,y.shape)

sns.countplot(df['Attrition'])

(df.Attrition.value_counts()/1470)*100

smote = SMOTE(sampling_strategy='minority')
x ,y = smote.fit_resample(x ,y)
print(x.shape ,y.shape)
y.value_counts()
sns.countplot(y ,palette='viridis')
plt.title('Now Class is Balanced')

x_train , x_test , y_train ,y_test = train_test_split(x , y, test_size=0.2 ,
random_state= 52)
print(x_train.shape)

#scaling the data
sc = StandardScaler()
x_train = sc.fit_transform(x_train)
x_test = sc.transform(x_test)
x_train

k = KFold(n_splits = 5)

lr_model = LogisticRegression()
lr_score = cross_val_score(lr_model , x_train , y_train ,cv = k ,scoring =
'neg_mean_squared_error')
lr_score.mean()

rf_model = RandomForestClassifier()
rf_score = cross_val_score(rf_model , x_train , y_train ,cv = k ,scoring =
'neg_mean_squared_error')
rf_score.mean()

svm_model = SVC()

```

```

svm_score = cross_val_score(svm_model , x_train , y_train ,cv = k ,scoring =
'neg_mean_squared_error')
svm_score.mean()

dt_model = DecisionTreeClassifier()
dt_score = cross_val_score(dt_model , x_train , y_train ,cv = k ,scoring =
'neg_mean_squared_error')
dt_score.mean()

plt.figure(figsize = (14 , 6))
plt.subplot(1,2,1)
x = ['Logistic Regression','Random Forest' , 'Support Vector' , 'Decision Tree']
y = [lr_score.mean() , rf_score.mean() , svm_score.mean() , dt_score.mean()]
plt.title('Neg Mean square error for Models')
sns.barplot(y,x,palette="viridis")
plt.subplot(1,2,2)
plt.plot(x , y,marker = 'o' , color = 'r',mfc = 'b' , ms = 8 )
plt.title('Neg Mean square error')

#we obtained less less -ve mena sq error for SVC and random forest
#lets try building model with both of them

model_params ={
    'RandomForestClassifier':
    {
        'model':RandomForestClassifier(),
        'param':
        {
            'n_estimators':[10 ,50 ,100,130],
            'criterion':['gini' , 'entropy'],
            'max_depth':range(4,8,1),
            'max_features':['auto' , 'log2']
        }
    },
    'SVC':
    {
        'model':SVC(),
        'param':
        {
            'C':[1,20],
            'gamma':[1,0.1],
            'kernel':['rbf']
        }
    }
}

scores =[]

```

```

for model_name , mp in model_params.items():
    model_sel = GridSearchCV(estimator= mp['model'] ,param_grid= mp['param']
, cv = 4 ,return_train_score=False)
    model_sel.fit(x_train,y_train)
    scores.append({
        'model':model_name,
        'best_score':model_sel.best_score_,
        'best_params':model_sel.best_params_
    })
scores

svm_model = SVC(C=20 ,gamma=0.1 ,kernel='rbf')
svm_model.fit(x_train ,y_train)
ytest_pred = svm_model.predict(x_test)
ytrain_pred = svm_model.predict(x_train)
accuracy_score(y_test ,ytest_pred)

print(classification_report(y_test , ytest_pred))

print(classification_report(y_train , ytrain_pred))

sns.heatmap(confusion_matrix(y_test ,ytest_pred) ,annot = True ,cmap ='ocean')

sns.heatmap(confusion_matrix(y_train ,ytrain_pred) ,annot = True ,cmap
='Spectral_r')

from sklearn.inspection import permutation_importance
perm_importance = permutation_importance(svm_model, x_test, y_test)
perm_importance

perm_importance.importances_mean

df.columns

cols = ['Age', 'BusinessTravel', 'DailyRate', 'Department',
'DistanceFromHome', 'Education', 'EducationField', 'EmployeeNumber',
'EnvironmentSatisfaction', 'Gender', 'HourlyRate', 'JobInvolvement',
'JobLevel', 'JobRole', 'JobSatisfaction', 'MaritalStatus',
'MonthlyIncome', 'MonthlyRate', 'NumCompaniesWorked', 'OverTime',
'PercentSalaryHike', 'PerformanceRating', 'RelationshipSatisfaction',
'StockOptionLevel', 'TotalWorkingYears', 'TrainingTimesLastYear',
'WorkLifeBalance', 'YearsAtCompany', 'YearsInCurrentRole',
'YearsSinceLastPromotion', 'YearsWithCurrManager']

features = np.array(cols)
plt.figure(figsize = (14 ,10))
sorted_idx = perm_importance.importances_mean.argsort()
sns.barplot( perm_importance.importances_mean[sorted_idx] ,features[sorted_idx])

```

```
)  
plt.xlabel("Permutation Importance")  
plt.title('FEATURE IMPORTANCE')
```

8. TESTING

8.1 Test Cases

1. Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issue of corporate employee attrition at the time of the release.

8.2 User Acceptance Testing

2. 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 Design	3	2	0	0	5
Duplicate	4	0	2	0	6
External	3	2	0	0	5
Fixed	1	0	1	0	2
Not Reproduced	0	3	3	0	6
Skipped	0	0	3	2	5
Won't Fix	0	0	1	0	1
Totals	11	7	10	2	30

3. Test Case Analysis

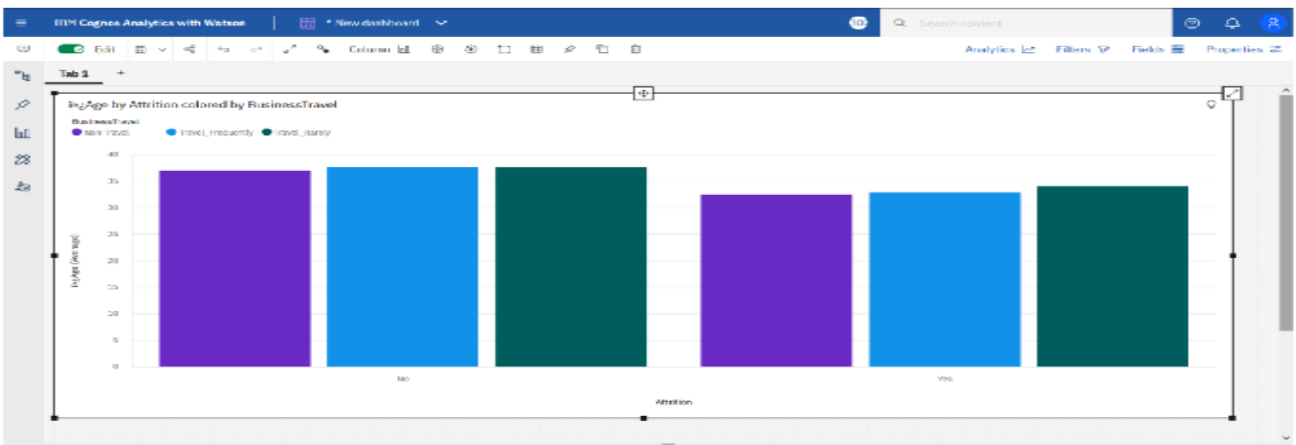
Section	Total Cases	Not Tested	Fail	Pass
Login Page	1	0	0	1
Employee Attrition Details	1	0	0	1

Database	2	0	0	2
Dashboard	1	0	0	1
Visualize the data	8	0	0	8
Logistic Regression	4	0	0	4

9. RESULTS

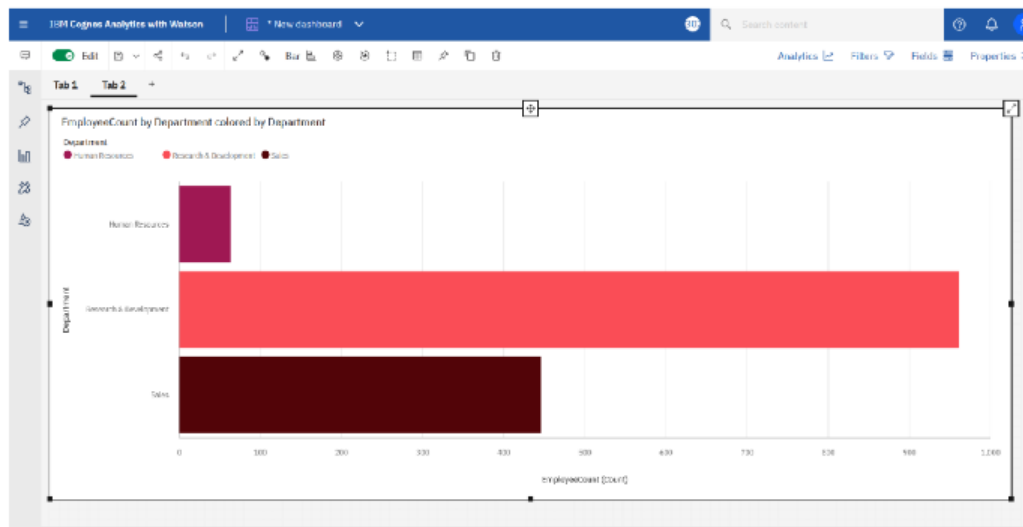
9.1 Performance Metrics

1. ATTRITION STATUS BY AGE:



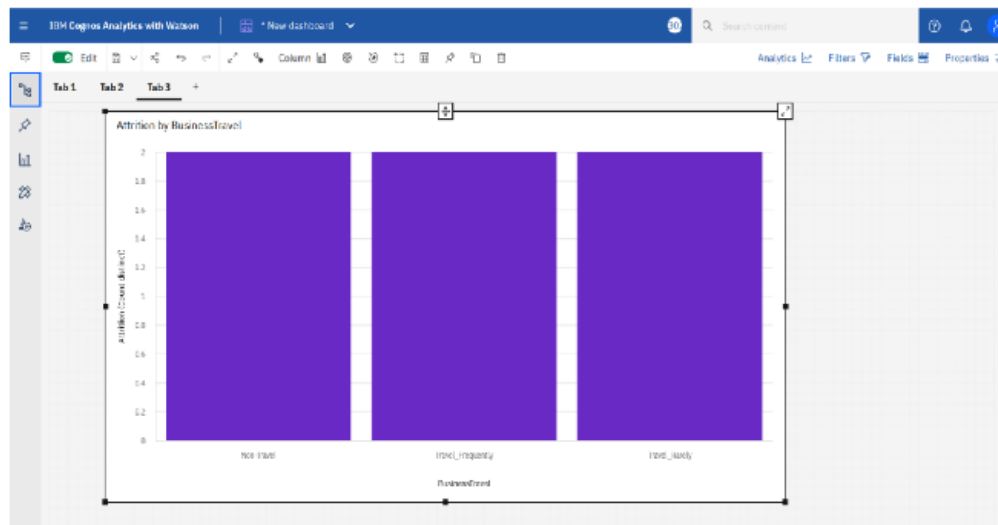
This visualization explains about the attrition status by age prediction through column chart.

2.EMPLOYEE COUNT BY DEPARTMENT:



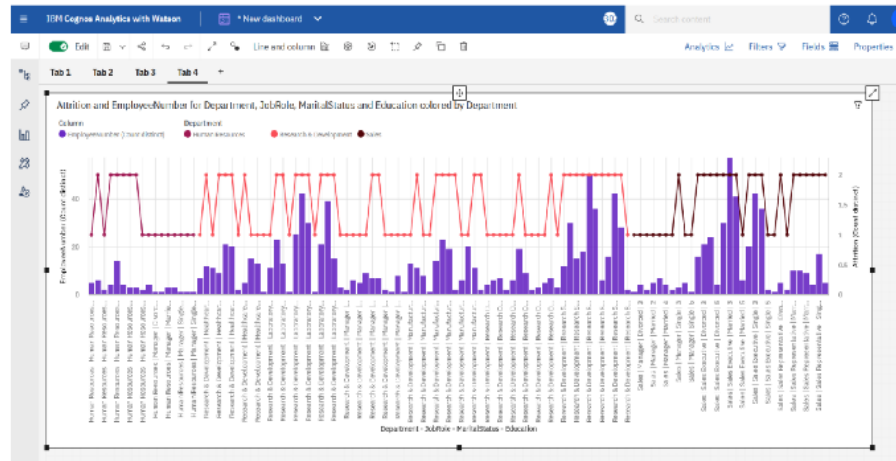
The visualized bar chart will clearly examine the employee count analysed by different departments.

3.ATTRITION BASED ON BUSINESS TRAVEL:



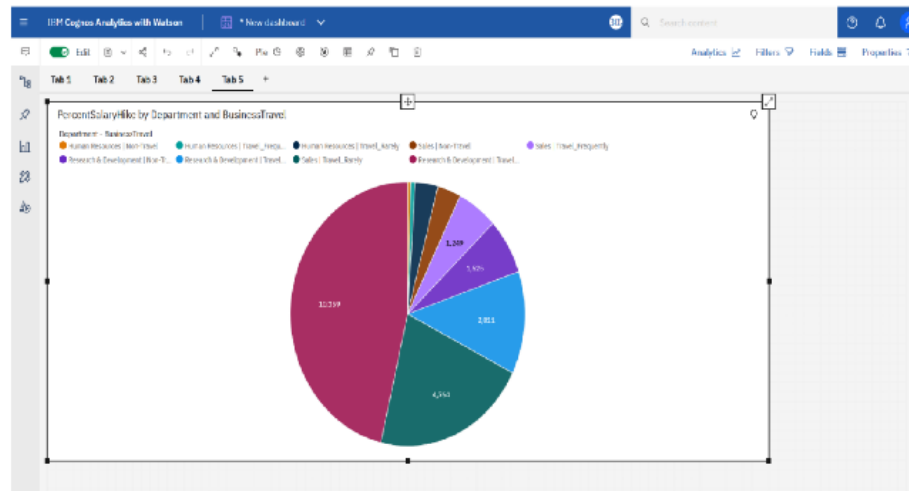
Visualization performed using the waterfall chart to view the attrition based on business travel.

4. ATTRITION BASED ON DEPARTMENT, JOBROLE, EDUCATION & MARITAL STATUS:



With the help of employee data set, the above visualized Line and Column chart explicit the attrition of employees based on department, job role, education and marital status which helps to analyze further implementation.

5. ATTRITION BASED ON SALARY HIKE PERCENTAGE:



Using the employee data set, I predict the visualization using pie chart which may use to find the attrition between the salary to percentage hike.

9. ADVANTAGES & DISADVANTAGES

9.1 Advantages

Data Collection :

The study is conducted among working IT professionals of two different categories. This categorization mainly was focused on experience level and role in the organization. It was important to know the views of candidates who seek for the job for various reasons as well as the views of interviewers involved in the process of hiring the candidates. The research study involves reference of both primary and secondary data. Primary Data Primary data is collected through a field survey with the help of a structured self-administrated Questionnaire. The survey consisted of close ended questions by the means of convenience sampling. The scaling technique installed in the questionnaire is 5-point rating scale. Total 120 respondent were IT professionals belonging to the organizations from Nagpur, Pune and Mumbai cities in Maharashtra. Secondary Data Secondary data is collected by referring to the Journals, research papers and published data in the form of books and newspapers.

Type of Research :

The research paper adopted the descriptive research design methodology. Sample Design, Sample Size and Sampling Method The sample selected for the study is an Indian Information Technology Industry. The nature of the sample is restricted to working professionals in Information Technology sector and is collected through the convenience sampling technique. The sample size was 120 respondents.

9. CONCLUSION

Employees as well as organizations must be clear with their expectations regarding the job profile. Any sort of mismatch leads to discrepancy and employees may fail to perform at their job. This eventually leads to attrition. Organizations should state the requirements and expectations unambiguously. This helps candidates decide upon to accept the job position or not. This eventually avoids further conflicts in the employment terms.

10. FUTURE SCOPE

Research findings suggest that attrition reasons in IT organizations primarily revolve around professional growth and challenges in the organization. Although economic factors happen to be the most influential factor, professionals may settle for second best criteria of their preference that is career growth and supportive work policies in the organization. On the other hand, candidates who aspire to have a better job than the one in hand are more interested in securing the next job. Young talent wants to work on latest technology and functional domain. IT professionals who are young career makers are less influenced by Brand name or geographical area. Most of the IT professionals look for challenging role and position in the organization. Candidates as well as senior professionals believe that challenging work motivate them to maintain the interest in the work life. Employees as well as organizations must be clear with their expectations regarding the job profile. Any sort of mismatch leads to discrepancy and employees may fail to perform at their job. This eventually leads to attrition. Organizations should state the requirements and expectations unambiguously. This helps candidates decide upon to accept the job position or not. This eventually avoids further conflicts in the employment

terms. Further this research can make more detailed conclusions over “mapping of candidates’ expectations with organizations’ requirement” by collecting the data focusing on all the steps of recruitment and selection process.

11. APPENDIX

11.1 Source Code

```
In [2]: path = '/content/HR-Employee-Attrition.csv'
df = pd.read_csv(path)
df
```

```
Out[2]:
```

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EmployeeCount	EmployeeNumber	...	Re
0	41	Yes	Travel_Rarely	1102	Sales	1	2	Life Sciences	1	1	...	
1	49	No	Travel_Frequently	279	Research & Development	8	1	Life Sciences	1	2	...	
2	37	Yes	Travel_Rarely	1373	Research & Development	2	2	Other	1	4	...	
3	33	No	Travel_Frequently	1392	Research & Development	3	4	Life Sciences	1	5	...	
4	27	No	Travel_Rarely	591	Research & Development	2	1	Medical	1	7	...	
...	
1465	36	No	Travel_Frequently	884	Research & Development	23	2	Medical	1	2061	...	
1466	39	No	Travel_Rarely	613	Research & Development	6	1	Medical	1	2062	...	
1467	27	No	Travel_Rarely	155	Research & Development	4	3	Life Sciences	1	2064	...	
1468	49	No	Travel_Frequently	1023	Sales	2	3	Medical	1	2065	...	
1469	34	No	Travel_Rarely	628	Research & Development	8	3	Medical	1	2068	...	

1470 rows x 35 columns

```
In [3]: df.shape
```

```
Out[3]: (1470, 35)
```

```
In [4]: df.info()
```

In [4]: `df.info()`

```
RangeIndex: 1470 entries, 0 to 1469
Data columns (total 35 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   Age                                   1470 non-null   int64
1   Attrition                           1470 non-null   object
2   BusinessTravel                       1470 non-null   object
3   DailyRate                            1470 non-null   int64
4   Department                           1470 non-null   object
5   DistanceFromHome                     1470 non-null   int64
6   Education                             1470 non-null   int64
7   EducationField                       1470 non-null   object
8   EmployeeCount                        1470 non-null   int64
9   EmployeeNumber                       1470 non-null   int64
10  EnvironmentsSatisfaction              1470 non-null   int64
11  Gender                               1470 non-null   object
12  HourlyRate                           1470 non-null   int64
13  JobInvolvement                       1470 non-null   int64
14  JobLevel                             1470 non-null   int64
15  JobRole                              1470 non-null   object
16  JobSatisfaction                      1470 non-null   int64
17  MaritalStatus                        1470 non-null   object
18  MonthlyIncome                        1470 non-null   int64
19  MonthlyRate                          1470 non-null   int64
20  NumCompaniesWorked                   1470 non-null   int64
21  Over18                               1470 non-null   object
22  OverTime                             1470 non-null   object
23  PercentsSalaryHike                   1470 non-null   int64
24  PerformanceRating                    1470 non-null   int64
25  RelationshipsSatisfaction             1470 non-null   int64
26  StandardHours                        1470 non-null   int64
27  StockOptionLevel                     1470 non-null   int64
28  TotalWorkingYears                    1470 non-null   int64
29  TrainingTimesLastYear                1470 non-null   int64
30  WorkLifeBalance                      1470 non-null   int64
31  YearsAtCompany                       1470 non-null   int64
32  YearsInCurrentRole                   1470 non-null   int64
33  YearsSinceLastPromotion              1470 non-null   int64
34  YearsWithCurrManager                 1470 non-null   int64
dtypes: int64(26), object(9)
memory usage: 402.1+ KB
```

In [5]: `df.select_dtypes('int64', 'float64').columns`

```
Out[5]: Index(['Age', 'DailyRate', 'DistanceFromHome', 'Education', 'EmployeeCount',
              'EmployeeNumber', 'EnvironmentsSatisfaction', 'HourlyRate',
              'JobInvolvement', 'JobLevel', 'JobSatisfaction', 'MonthlyIncome',
              'MonthlyRate', 'NumCompaniesWorked', 'PercentsSalaryHike',
              'PerformanceRating', 'RelationshipsSatisfaction', 'StandardHours',
              'StockOptionLevel', 'TotalWorkingYears', 'TrainingTimesLastYear',
              'WorkLifeBalance', 'YearsAtCompany', 'YearsInCurrentRole',
              'YearsSinceLastPromotion', 'YearsWithCurrManager'],
              dtype='object')
```

In [6]: `cat_cols = df.select_dtypes('object').columns`
`cat_cols`

```
Out[6]: Index(['Attrition', 'BusinessTravel', 'Department', 'EducationField', 'Gender',
              'JobRole', 'MaritalStatus', 'Over18', 'OverTime'],
              dtype='object')
```

In [7]: `df.describe().T`

Out[7]:

	count	mean	std	min	25%	50%	75%	max
Age	1470.0	36.923810	9.135373	18.0	30.00	36.0	43.00	60.0
DailyRate	1470.0	802.485714	403.509100	102.0	465.00	802.0	1157.00	1499.0
DistanceFromHome	1470.0	9.192517	8.106864	1.0	2.00	7.0	14.00	29.0
Education	1470.0	2.912925	1.024165	1.0	2.00	3.0	4.00	5.0
EmployeeCount	1470.0	1.000000	0.000000	1.0	1.00	1.0	1.00	1.0
EmployeeNumber	1470.0	1024.865306	602.024335	1.0	491.25	1020.5	1555.75	2068.0
EnvironmentSatisfaction	1470.0	2.721769	1.093082	1.0	2.00	3.0	4.00	4.0
HourlyRate	1470.0	65.891156	20.329428	30.0	48.00	66.0	83.75	100.0
JobInvolvement	1470.0	2.729932	0.711561	1.0	2.00	3.0	3.00	4.0
JobLevel	1470.0	2.063946	1.106940	1.0	1.00	2.0	3.00	5.0
JobSatisfaction	1470.0	2.728571	1.102846	1.0	2.00	3.0	4.00	4.0
MonthlyIncome	1470.0	6502.931293	4707.956783	1009.0	2911.00	4919.0	8379.00	19999.0
MonthlyRate	1470.0	14313.103401	7117.786044	2094.0	8047.00	14235.5	20461.50	26999.0
NumCompaniesWorked	1470.0	2.693197	2.498009	0.0	1.00	2.0	4.00	9.0
PercentSalaryHike	1470.0	15.209524	3.659938	11.0	12.00	14.0	18.00	25.0
PerformanceRating	1470.0	3.153741	0.360824	3.0	3.00	3.0	3.00	4.0
RelationshipSatisfaction	1470.0	2.712245	1.081209	1.0	2.00	3.0	4.00	4.0
StandardHours	1470.0	80.000000	0.000000	80.0	80.00	80.0	80.00	80.0
StockOptionLevel	1470.0	0.793878	0.852077	0.0	0.00	1.0	1.00	3.0
TotalWorkingYears	1470.0	11.279592	7.780782	0.0	6.00	10.0	15.00	40.0
TrainingTimesLastYear	1470.0	2.799320	1.289271	0.0	2.00	3.0	3.00	6.0
WorkLifeBalance	1470.0	2.761224	0.706476	1.0	2.00	3.0	3.00	4.0
YearsAtCompany	1470.0	7.008163	6.126525	0.0	3.00	5.0	9.00	40.0
YearsInCurrentRole	1470.0	4.229252	3.623137	0.0	2.00	3.0	7.00	18.0
YearsSinceLastPromotion	1470.0	2.187755	3.222430	0.0	0.00	1.0	3.00	15.0
YearsWithCurrManager	1470.0	4.123129	3.568136	0.0	2.00	3.0	7.00	17.0

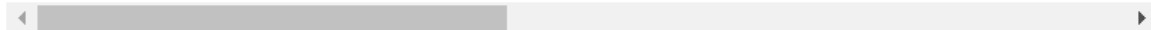
In [8]:

df

Out[8]:

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EmployeeCount	EmployeeNumber	...	Re
0	41	Yes	Travel_Rarely	1102	Sales	1	2	Life Sciences	1	1	...	
1	49	No	Travel_Frequently	279	Research & Development	8	1	Life Sciences	1	2	...	
2	37	Yes	Travel_Rarely	1373	Research & Development	2	2	Other	1	4	...	
3	33	No	Travel_Frequently	1392	Research & Development	3	4	Life Sciences	1	5	...	
4	27	No	Travel_Rarely	591	Research & Development	2	1	Medical	1	7	...	
...	
1465	36	No	Travel_Frequently	884	Research & Development	23	2	Medical	1	2061	...	
1466	39	No	Travel_Rarely	613	Research & Development	6	1	Medical	1	2062	...	
1467	27	No	Travel_Rarely	155	Research & Development	4	3	Life Sciences	1	2064	...	
1468	49	No	Travel_Frequently	1023	Sales	2	3	Medical	1	2065	...	
1469	34	No	Travel_Rarely	628	Research & Development	8	3	Medical	1	2068	...	

1470 rows × 35 columns



In [9]:

```

for cat in cat_cols:
    print(cat, '-> ', df[cat].unique())
    print()

Attrition -> ['Yes' 'No']

BusinessTravel -> ['Travel_Rarely' 'Travel_Frequently' 'Non-Travel']

Department -> ['Sales' 'Research & Development' 'Human Resources']

EducationField -> ['Life Sciences' 'Other' 'Medical' 'Marketing' 'Technical Degree'
'Human Resources']

Gender -> ['Female' 'Male']

JobRole -> ['Sales Executive' 'Research Scientist' 'Laboratory Technician'
'Manufacturing Director' 'Healthcare Representative' 'Manager'
'Sales Representative' 'Research Director' 'Human Resources']

MaritalStatus -> ['Single' 'Married' 'Divorced']

Over18 -> ['Y']

OverTime -> ['Yes' 'No']

```

In [10]:

```

print("All columns Unique values count")
for col in df:
    print(col, len(df[col].unique()), sep=': ')

```

```

All columns Unique values count
Age: 43
Attrition: 2
BusinessTravel: 3
DailyRate: 886
Department: 3
DistanceFromHome: 29
Education: 5
EducationField: 6
EmployeeCount: 1
EmployeeNumber: 1470
EnvironmentSatisfaction: 4
Gender: 2
HourlyRate: 71
JobInvolvement: 4
JobLevel: 5
JobRole: 9
JobSatisfaction: 4
MaritalStatus: 3
MonthlyIncome: 1349
MonthlyRate: 1427
NumCompaniesWorked: 10
Over18: 1
OverTime: 2
PercentSalaryHike: 15
PerformanceRating: 2
RelationshipSatisfaction: 4
StandardHours: 1
StockOptionLevel: 4
TotalWorkingYears: 40
TrainingTimesLastYear: 7
WorkLifeBalance: 4
YearsAtCompany: 37
YearsInCurrentRole: 19
YearssinceLastPromotion: 16
YearsWithCurrManager: 18

```

In [21]:

```

no_use = []
for col in df.columns:
    if(len(df[col].unique()) ==1):
        no_use.append(col)
no_use

```

Out[21]: ['EmployeeCount', 'Over18', 'StandardHours']

```
In [23]: df.columns
```

```
Out[23]: Index(['Age', 'Attrition', 'BusinessTravel', 'DailyRate', 'Department',
               'DistanceFromHome', 'Education', 'EducationField', 'EmployeeNumber',
               'EnvironmentSatisfaction', 'Gender', 'HourlyRate', 'JobInvolvement',
               'JobLevel', 'JobRole', 'JobSatisfaction', 'MaritalStatus',
               'MonthlyIncome', 'MonthlyRate', 'NumCompaniesWorked', 'OverTime',
               'PercentSalaryHike', 'PerformanceRating', 'RelationshipSatisfaction',
               'StockOptionLevel', 'TotalWorkingYears', 'TrainingTimesLastYear',
               'WorkLifeBalance', 'YearsAtCompany', 'YearsInCurrentRole',
               'YearsSinceLastPromotion', 'YearsWithCurrManager'],
              dtype='object')
```

```
In [24]: y_n_type = []
         others = []
         for col in df.select_dtypes('object').columns:
             if len(df[col].unique()) == 2:
                 y_n_type.append(col)

         y_n_type
```

```
Out[24]: ['Gender', 'OverTime']
```

```
In [25]: df['Gender'].replace({'Male':1, 'Female':0}, inplace = True)
         df['OverTime'].replace({'Yes':1, 'No':0}, inplace = True)
```

CATEGORICAL FEATURES ENCODING

```
In [26]: others = df.select_dtypes('object').columns
         others
```

```
Out[26]: Index(['BusinessTravel', 'Department', 'EducationField', 'JobRole',
               'MaritalStatus'],
              dtype='object')
```

```
In [32]: le = LabelEncoder()
         for col in others:
             df[col] = le.fit_transform(df[col])
```

```
In [33]: df.select_dtypes('object').columns
```

```
Out[33]: Index([], dtype='object')
```

```
In [34]: x = df.drop('Attrition', axis = 1)
         y = df['Attrition']

         print(x.shape, y.shape)

(1470, 31) (1470,)
```

```
In [36]: (df.Attrition.value_counts()/1470)*100
```

```
Out[36]: 0    83.877551
         1    16.122449
         Name: Attrition, dtype: float64
```

```
In [39]: smote = SMOTE(sampling_strategy='minority')
         x, y = smote.fit_resample(x, y)

         print(x.shape, y.shape)

(2466, 31) (2466,)
```