

EMPLOYEE ATTRITION PREDICTION USING MACHINE LEARNING

Main Project Report

Submitted by

JUNED ANSAR P

Reg no:FIT20MCA-2064

*Submitted in partial fulfillment of the requirements for the award of the
degree of*

Master of Computer Applications

Of

A P J Abdul Kalam Technological University



Focus on Excellence

FEDERAL INSTITUTE OF SCIENCE AND TECHNOLOGY (FISAT)®

ANGAMALY-683577, ERNAKULAM(DIST)

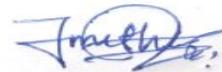
JULY 2022

DECLARATION

I hereby declare that the report of this project work, submitted to the Department of Computer Applications, Federal Institute of Science and Technology (**FISAT**), Angamaly in partial fulfillment of the award of the degree of Master of Computer Application is an authentic record of my original work.

The report has not been submitted for the award of any degree of this university or any other university.

Date :



Place: Angamaly

Juned Ansar P



FEDERAL INSTITUTE OF SCIENCE AND TECHNOLOGY (FISAT)®

An ISO 9001:2015 Certified, NAAC ('A' Grade) Accredited Institution with NBA Accredited Programmes
(Approved by AICTE – Affiliated to APJ Abdul Kalam Technological University, Kerala)

Owned and managed by Federal Bank Officers' Association Educational Society

HORMIS NAGAR, MOOKKANNOOR P.O., ANGAMALY - 683 577, ERNAKULAM DT., KERALA, INDIA.

Tel: (O) 0484-2725272 Fax: 0484 – 2725250 E-mail: mail@fisat.ac.in Website: www.fisat.ac.in

11TH July 2022

TO WHOMSOEVER IT MAY CONCERN

This is to certify that Mr./Ms. JUNED ANSAR. P (Reg. No. FIT20MCA-2064) has successfully completed his/her Main Project with the title "EMPLOYEE ATTRITION PREDICTION USING MACHINE LEARNING", in the Department of Computer Applications, FISAT, during the period from 30th March 2022 to 11th July 2022.

Dr DEEPA MARY MATHEWS
HEAD OF THE DEPARTMENT



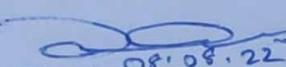
FEDERAL INSTITUTE OF SCIENCE AND TECHNOLOGY (FISAT)®
ANGAMALY, ERNAKULAM-683577



CERTIFICATE

This is to certify that the project report titled "**EMPLOYEE ATTRITION PREDICTION USING MACHINE LEARNING**" submitted by **JUNED ANSAR P.** (Reg No: **FIT20MCA-2064**) towards partial fulfillment of the requirements for the award of the degree of Master of Computer Applications is a record of bonafide work carried out by him during the year 2022.

Project Guide


Dr. Sujesh P Lal
08.08.22

Head of the Department


Dr. Deepa Mary Mathews



Submitted for the viva-voce held on at

Examiner :

ACKNOWLEDGEMENT

Gratitude is a feeling which is more eloquent than words, more silent than silence. To complete this project work I needed the direction, assistance and co-operation of various individuals, which is received in abundance with the grace of God.

I hereby express my deep sense of gratitude to **Dr. Manoj George** Principal, FISAT and **Dr. C Sheela** Vice principal, FISAT for allowing me to utilize all the facilities of the college.

My sincere thanks to **Dr. Deepa Mary Mathews** HOD, Department of Computer Applications FISAT , who had been a source of inspiration. I express heartiest thanks to **Dr. Sujesh P Lal** my internal guide for his encouragement and valuable suggestion. I express my heartiest gratitude to my scrum master **Ms. Manju Joy** and the faculty members in my Department for encouragement and constructive suggestions and comment during the project work.

Finally I wish to express my thanks to my parents, friends and well-wishers who extended their help in one way or other in preparation of my project. Besides all, I thank GOD for everything.

ABSTRACT

Employees are considered the backbone of an organization, college, or any business-related area. The success or failure of the organization depends on the employees who work sincerely for an organization. There is no doubt that talented employees will take the company to the top position. Employee turnover is the main problem facing the organization's skilled and experienced employees leaving the organization consistently.

The paper mainly predicts the dissatisfaction factor of employees and what reason to change their job. Once the dissatisfaction factor is identified the organization can take the action accordingly and also reduce the employee attrition rate. Employee attrition is the loss of staff in any organization caused by resignations. Employees are an organization's most significant resource. It's important to understand whether the workers are unhappy or if there are any other factors contributing to their departure from the specific position. Employees today are keen to switch organizations in search of better chances. However, if they quit their employment suddenly, the company could suffer a great loss. New hiring will cost time and money, and it will take some time for the newly employed people to generate a profit for the firm. One of the most significant issues that many firms face is the retention of talented and dedicated workers. Consequently, we can definitely lessen this issue by increasing employee satisfaction and creating a desirable workplace.

This technology may identify which employees are most likely to leave a company and for what reason, allowing management to adopt a number of preventative measures to keep people on board and lower attrition. Motivating employees, exposing them to fresh roles, soliciting ongoing employee input, etc., are a few employee retention tactics to reduce attrition.

Contents

1	INTRODUCTION	9
2	PROOF OF CONCEPT	10
2.1	Existing System	11
2.2	Proposed System	11
3	IMPLEMENTATION	12
3.1	Technology Used	14
3.1.1	Spyder	14
3.1.2	Flask (Python Framework)	14
3.1.3	Python	15
3.2	Algorithm	16
3.2.1	Random Forest	16
3.3	System Architecture	16
3.4	Work flow of the system	18
3.5	Dataset	20
3.6	Data Base Design	20
3.6.1	Tables	21
3.7	Modules	24
3.8	Admin module	24
3.8.1	Admin Login	24
3.8.2	Approve or Reject	24
3.8.3	View Employee details	25

3.8.4	Send notifications	25
3.8.5	Attrition result	25
3.9	Employee module	25
3.9.1	Registration	25
3.9.2	Employee Login	25
3.9.3	View details	26
3.9.4	Attend questioner	26
3.9.5	View Notification	26
3.10	ML Module	26
3.10.1	Data set Collection	26
3.10.2	Data prepossessing	26
3.10.3	Data Transformation	26
3.10.4	Splitting data	27
3.10.5	Model Training	27
3.10.6	Model Evaluation and Testing	27
4	RESULT ANALYSIS	28
4.1	Results	29
4.2	Accuracy	32
4.2.1	Trained model accuracy	32
4.2.2	Graph Results	33
5	CONCLUSION AND FUTURE SCOPE	39
5.1	Conclusion	40
5.2	Future Scope	40
6	APPENDIX	41
6.1	Source Code	42
6.1.1	main.py	42
6.1.2	App.py	46
6.2	Screen Shots	61

Chapter 1

INTRODUCTION

Employees are considered the backbone of an organization, college, or any business-related area. The success or failure of the organization depends on the employees who work sincerely for an organization. There is no doubt that talented employees will take the company to the top position. Employee turnover is the main problem facing the organization's skilled and experienced employees leaving the organization consistently.

The paper mainly predicts the dissatisfaction factor of employees and what reason to change their job. Once the dissatisfaction factor has been identified the organization can take the action accordingly and also reduce the employee attrition rate. Employee attrition is the loss of staff in any organization caused by resignations. Employees are an organization's most significant resource. It's important to understand whether the workers are unhappy or if there are any other factors contributing to their departure from the specific position. Employees today are keen to switch organizations in search of better chances.

However, if they quit their employment suddenly, the company could suffer a great loss. New hiring will cost time and money, and it will take some time for the newly employed people to generate a profit for the firm. One of the most significant issues that many firms face is the retention of talented and dedicated workers. Consequently, we can definitely lessen this issue by increasing employee satisfaction and creating a desirable workplace.

Chapter 2

PROOF OF CONCEPT

A lot of studies have been made on attrition prediction analysis in the literature. The major focus was on predicting employee attrition. Researchers have applied machine learning classification models like logistic regression, random forests, support vector machines, and others to analyze the attributes that impact the attrition rate. For instance, Srivastava[1] et al presented a framework that predicts employee churn by analyzing the behaviors of employees and attributes with the help of machine learning techniques. Setiawan[2] et al through their work found variables that have a major impact on employee attrition.

Employee Attrition is mainly the normal flow of people out of an organization, due to career or job change, relocation, illness, and so on [3]. Employee Attrition is the percentage of employees leaving the organization for what so ever reasons. Employees can leave the organization for many personal as well as professional reasons. So basically there are two types of turnover, one is voluntary turnover which is decided by the employee, and the other type of turnover is decided by the company and that is why it is called involuntary turnover [4]. Involuntary turnover generally happens when the performance of the employee is not up to expectations. Retention is also necessary for the growth and stability of an organization [4]. The high attrition rate causes when there are more employment opportunities in the market. Currently, employee attrition is one of the major issues faced by HR managers. There are so many working employees who are not satisfied due to one

of the aspects which are not fulfilled by the organization which results in a higher attrition rate.

2.1 Existing System

- Retaining employees and experience has become a challenge for organizations in recent years.
- There aren't any existing methods to predict employee attrition.
- To avoid being blind-sided and improve certain elements of the organization, predictive modeling is viewed as a great tool for human resource departments.

2.2 Proposed System

- The proposed work predicts attrition of employees working in Fisat college by exploring classification algorithms and does performance analysis.
- The employees will answer the questions provided in the questionnaire.
- The data is fed into model which will predict the probability of attrition.
- This probability can be used for further analysis

Chapter 3

IMPLEMENTATION

EMPLOYEE ATTRITION PREDICTION USING MACHINE LEARNING is an user friendly web application system. The project involved "WAFn – UseC_HR – Employee – Attrition.csv" dataset to train the model. The dataset was taken from the kaggle site. First of all cleaning of dataset is done for which the prepossessing techniques such as describe, info, checking any null values etc.. are used. The describe method displays basic statistical information on the numerical values of the dataset like mean, standard deviation, percentile, min and max etc. The info() method is basically shows the number of entries in the dataset, the column names and the column types in this project the dataset include 1470 entries and 35 columns and checking the dataset any null value its done using the isnull() method. This will list number of null values in each column. In this project there are no null values in any of the 35 columns.

Then plot several graphs generated by the system with respect to attrition such as age vs attrition , overtime vs attrition, corelation of the dataset columns etc. after drop the unneeded columns before to training the dataset.

The proposed system uses the different machine learning algorithms to build the model and it chooses the best accuracy model. Here employee dataset is used to predict the attrition which includes all past and present records of the employees. Then the dataset splitting is done. In this 75 percentage data was used for training and 25 percentage data were used for testing the data. The main aim of training is done to predict correctly as often as possible. The test data is used to see how well the machine can predict new answers and to validate machine learning model behavior. After splitting the dataset, Random forest model is used to predict the employee attrition.

The project uses Spyder to develop the data science code and Flask server as framework to connect the model code and the UI. The back-end is python and server is python flask.

3.1 Technology Used

3.1.1 Spyder

Spyder is an open-source cross-platform integrated development environment (IDE) for scientific programming in the Python language. Spyder integrates with a number of prominent packages in the scientific Python stack, including NumPy, SciPy, Matplotlib, pandas, IPython, SymPy and Cython, as well as other open-source software. For a user to write a code in python and save it on your desktop an IDE is required. Although to execute codes on your system only requirement is to install python and then access it from windows command prompt. But using an IDE it improves the user's interactivity with python. An IDE provides the option to write a code in a script and then test the output of the same by running the code in IDE itself and the output is displayed in the output window. If a user wants to understand the working of some specific function, they can refer the help window for documentation inside IDE and user can also try to run single liner codes in the console.

3.1.2 Flask (Python Framework)

Flask is a web framework. This means flask provides you with tools, libraries and technologies that allow you to build a web application. This web application can be some web pages, a blog, a wiki or go as big as a web-based calendar application or a commercial website.

Flask is part of the categories of the micro-framework. Micro-framework are normally framework with little to no dependencies to external libraries. This has pros and cons. Pros would be that the framework is light, there are little dependency to update and watch for security bugs, cons is that some time you will have to do more work by yourself or increase yourself the list of dependencies by adding plugins.

3.1.3 Python

Python is an interpreter, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

Often, programmers fall in love with Python because of the increased productivity it provides. Since there is no compilation step, the edit-test-debug cycle is incredibly fast. Debugging Python programs is easy: a bug or bad input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception. When the program doesn't catch the exception, the interpreter prints a stack trace. A source level debugger allows inspection of local and global variables, evaluation of arbitrary expressions, setting breakpoints, stepping through the code a line at a time, and so on. The debugger is written in Python itself, testifying to Python's introspective power. On the other hand, often the quickest way to debug a program is to add a few print statements to the source: the fast edit-test-debug cycle makes this simple approach very effective.

Python is a multi-paradigm programming language. Object-oriented programming and structured programming are fully supported, and many of its features support functional programming and aspect-oriented programming (including by meta programming and Meta objects (magic methods)). Many other paradigms are supported via extensions, including design by contract and logic programming. Python uses dynamic typing, and a combination of reference counting and a cycle-detecting garbage collector for memory management. It also features dynamic name resolution (late binding), which binds method and variable names during program execution.

3.2 Algorithm

3.2.1 Random Forest

Random Forest is used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to enhance the performance of the model. Instead of depending on one decision tree, the random forest takes the prediction from each tree and prediction which have majority of votes will be the final output. As the number of trees increases the accuracy also increases and prevents it from the over fitting problem.

How it works ?

Step 1:Start

Step 2:Get the dataset and convert it to a data frame.

Step 3:Drop the unnecessary features.

Step 4:Normalize the dataset.

Step 5:Create a random forest model for employee attrition and fit the model.

Step 6:Using the model, predict the employee attrition of a person.

Step 7:Display the predicted result.

Step 8:Stop.

3.3 System Architecture

The diagram that describes the operation of the system (figure 3.1). First trained the data then train the machine learning algorithm in these project use Random forest algorithm after training build the model using these model test the unseen data(input data) then the model predict the output that's yes/ no then evaluate the result.

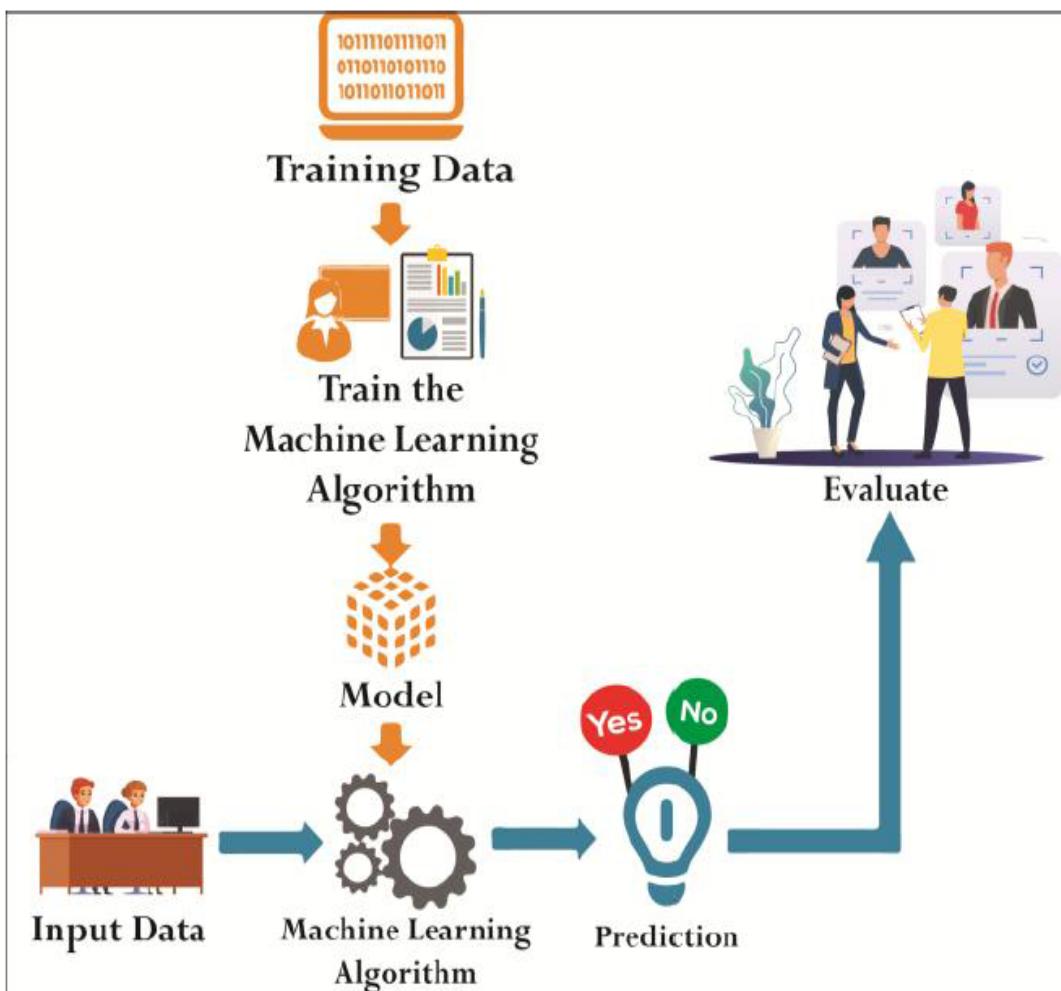


Figure 3.1: Structure or Architecture of the system

3.4 Work flow of the system

Workflow of a system that describes the flow and the process of the system (figure 3.2).

The process starts with the data collection the employees are registered through the user interface the registered details are stored in the database then check for any missing values in the data if the checking result is yes the control again goes to the registration page. if the checking result is correct, then analyses and prepossessing the data, then choose the machine learning algorithm to predict the employee attrition.

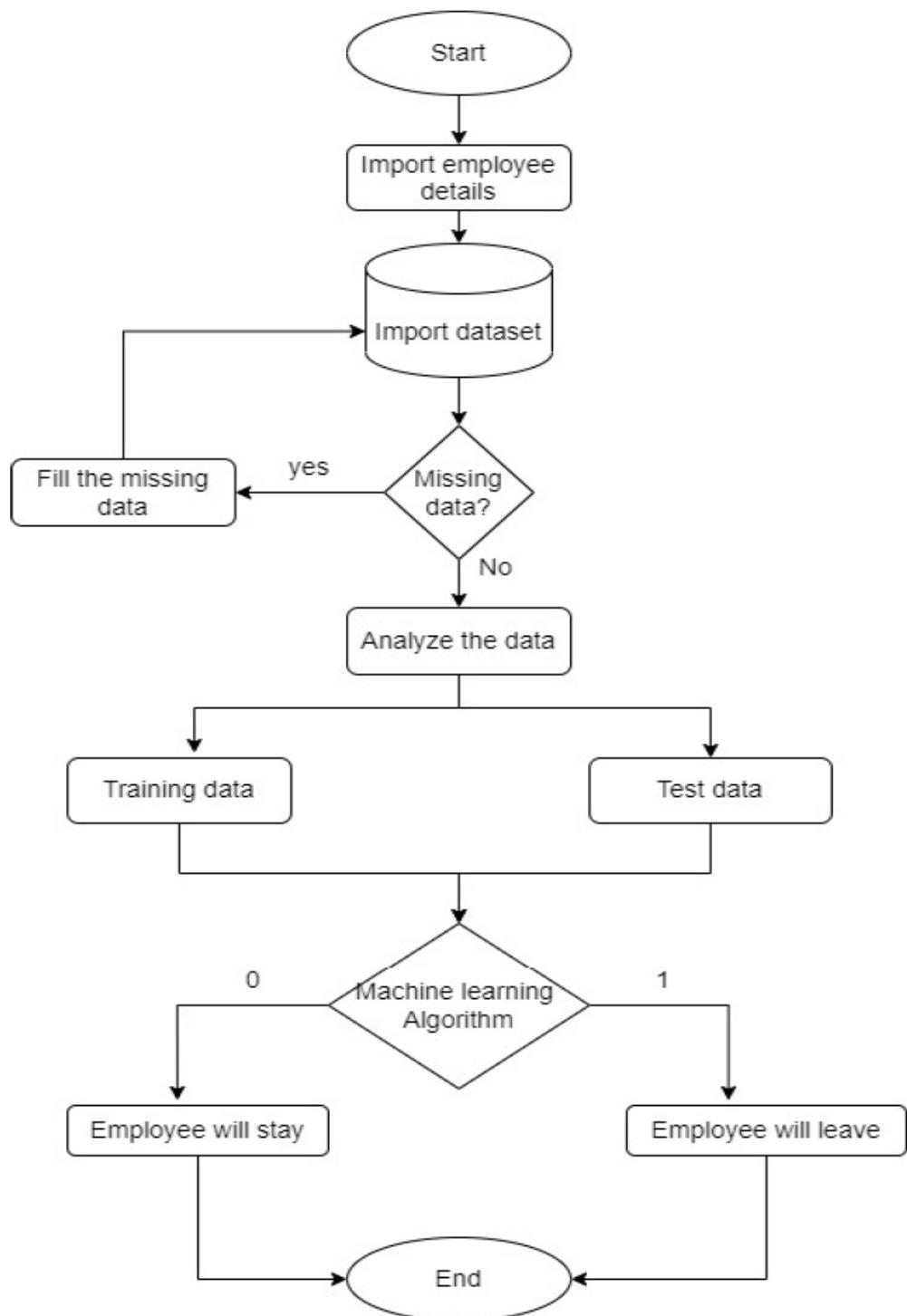


Figure 3.2: workflow of the system

3.5 Dataset

The dataset is the main part of the machine learning project. In this project use the "WA_Fn – UseC_HR – Employee – Attrition.csv" dataset to prediction. It include 1470 data tuple and 35 features such as Age , 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 . the datasets contain all the necessary attributes of employees to predict employee attrition such as Employee id, Satisfaction Level, Last Evaluation, Number of Projects, Average Monthly Hours, Time spent in Company (Years), Departments, Left, Promotion last 5 years, salary level.

The site from which the dataset searched is:

- Kaggle.com

3.6 Data Base Design

A data base is a collection of inter related data stored with minimum redundancy to serve many quickly and efficiently. The general way is to make information accessing easy, quick, inexpensive and flexible for the user. In data base design several objectives are considered controlling redundancy, ease of learning and use, data dependence, more information at low cost, accuracy and integrity are some of them

3.6.1 Tables

Mainly there are five tables that are registration,register,login,attrition and notify. There is an unique id for each table.

- Register table: All the employee details like name ,phone, gender, address, DOB, pin, qualification, profession, email, state and password are stored in this table. Totally there are eleven columns and one foreign key that's auto-increment are there.
- Login table: In login table login details are stored. login details like email, password, status, type.The status and type determine whether to open the employee home page or the admin home page.
- Attrition table: The prediction results are stored in this table and its also store the email id of employee.
- Notify table:This table stores all the messages the admin sent to the employees and the Gmail id of the employees to know which employee sent the message.
- Registration table: The admin details are stored in registration table.

Column Name	Data Type
id	int(pk)
name	varchar(30)
phone	char(25)
gender	varchar(8)
date	date
pin	int
state	varchar(20)
qualification	varchar(20)
profession	varchar(100)
email	varchar(300)
password	varchar(20)

Table 3.1: Register table

Column Name	Data Type
Id	int(pk)
email	varchar(300)
password	varchar(20)
type	varchar(20)
status	varchar(20)

Table 3.2: Login table

Column Name	Data Type
id	int (pk)
email	varchar(50)
attrition	varchar(50)

Table 3.3: Attrition table

Column Name	Data Type
gmail-id	varchar(80)
message	varchar(500)

Table 3.4: Notify Table

Column Name	Data Type
id	int(pk)
name	varchar(30)
email	varchar(30)
password	varchar(30)
type	varchar(20)

Table 3.5: Registration Table

3.7 Modules

There are two main modules in the "EMPLOYEE ATTRITION PREDICTION USING MACHINE LEARNING " project. The mainly 3 module that are Admin,Employee and ML module.

3.8 Admin module

The overall control is done by the admin. The admin is the person who sits at the head of an organization.If it is in a college, this would be done by the system administrator

3.8.1 Admin Login

The logged in admin reaches the admin home through the admin login page.The admin's logging credentials are already saved in the the database logging table.Employees and admin login through the same login page. At the time of login, the type and status of the login person is checked. If the type is admin and the status is user, it goes to the admin page, and if the type is employee and the status is user, it goes to the employee page.

3.8.2 Approve or Reject

The information registered by the employees through the employee registration page is stored in the database and that information also goes to the approve or reject page in the admin. There the admin decides whether to approve the registered user or employees or not. Only admin approved users or employees can login.only the approved user can log in to the employee page.

3.8.3 View Employee details

Admin can view all details of registered employee through the employees page. The details like name, id, email id, job position, date of birth etc..

3.8.4 Send notifications

Admin can send messages or any kind of job-related notice to the employee through the notice page

3.8.5 Attrition result

Admin can see the results of employee attrition prediction based on employee questioner attendance.

3.9 Employee module

3.9.1 Registration

An employee registration form is used by employers to track information about new hires. It is a helpful way to keep track of all information about the new .

3.9.2 Employee Login

Admin and admin login through the same login page. At the time of login, the type and status of the login person is checked. If the type is admin and the status is user, it goes to the admin page, and if the type is employee and the status is user, it goes to the employee page.

3.9.3 View details

The employees can view their own details.

3.9.4 Attend questioner

It is the main part of the system. Employees answer questions through a questionnaire.

The model predicts attrition based on answers to questions

3.9.5 View Notification

Employees can see messages sent by admin.

3.10 ML Module

3.10.1 Data set Collection

The dataset collection is very important for the project. In this project use the "WA_Fn – UseC_HR – Employee – Attrition.csv" dataset to prediction. It include 1470 data tuple and 35 features.

3.10.2 Data prepossessing

Next step to explore and analysis the data set. Firstly read the data set. Then analyse the data set using the describe method and check if any null values are present our data set using "isnull()" function. After the testing the isnull function the dataset no longer contains null values. The dataset is normalized to a standardized form.

3.10.3 Data Transformation

Before run the models, we have to convert all categorical values (text values) to numerical values. In this step converting all categorical values to the numeric. This is done using the

label encoder function .

3.10.4 Splitting data

In this project 75 percentage of the data for training and 25 percentage of the data for testing. The splitting is done-using the *train,test,split()* function .

3.10.5 Model Training

After the splitting the data set ,it was applied to the data model. Here, to predict the employee attrition using Random Forest() model.

3.10.6 Model Evaluation and Testing

The confusion Matrix, R square and classification report are used see the accuracy of the model.A confusion matrix is a table that is often used to describe the performance of a classification model (or "classifier") on a set of test data for which the true values are known.

A classification report is a performance evaluation metric in machine learning. It is used to show the precision, recall, F1 Score, and support of the trained classification model.

Chapter 4

RESULT ANALYSIS

The project was completed successfully in time. This web application predicts employee attrition using the machine learning algorithm(Random forest). the employees enter the details to register and log in to the employee home and the employees answer questions through a questionnaire. The model predicts attrition based on answers to questions.

4.1 Results

	Age	DailyRate	DistanceFromHome	Education	EmployeeCount	EmployeeNumber	E
count	1470.000000	1470.000000	1470.000000	1470.000000	1470.0	1470.000000	
mean	36.923810	802.485714	9.192517	2.912925	1.0	1024.865306	
std	9.135373	403.509100	8.106864	1.024165	0.0	602.024335	
min	18.000000	102.000000	1.000000	1.000000	1.0	1.000000	
25%	30.000000	465.000000	2.000000	2.000000	1.0	491.250000	
50%	36.000000	802.000000	7.000000	3.000000	1.0	1020.500000	
75%	43.000000	1157.000000	14.000000	4.000000	1.0	1555.750000	
max	60.000000	1499.000000	29.000000	5.000000	1.0	2068.000000	

8 rows × 26 columns

Figure 4.1: Description of the dataset

The describe method displays basic statistical information on the numerical values of the dataset like mean, standard deviation, percentile, min and max.

```
df.info()

<class 'pandas.core.frame.DataFrame'>
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  EnvironmentSatisfaction 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  PercentSalaryHike 1470 non-null    int64  
 24  PerformanceRating 1470 non-null    int64  
 25  RelationshipSatisfaction 1470 non-null    int64  
 26  StandardHours     1470 non-null    int64  
 27  StockOptionLevel   1470 non-null    int64
```

Figure 4.2: Information of the dataset

The info() method is basically shows the number of entries in the dataset, the column names and the column types in this project the dataset include 1470 entries and 35 columns. Among these columns are 26 integer data types and 9 object data types.

▶	Age	0
▶	Attrition	0
▶	BusinessTravel	0
▶	DailyRate	0
▶	Department	0
▶	DistanceFromHome	0
▶	Education	0
▶	EducationField	0
▶	EmployeeCount	0
▶	EmployeeNumber	0
▶	EnvironmentSatisfaction	0
▶	Gender	0
▶	HourlyRate	0
▶	JobInvolvement	0
▶	JobLevel	0
▶	JobRole	0
▶	JobSatisfaction	0
▶	MaritalStatus	0
▶	MonthlyIncome	0
▶	MonthlyRate	0
▶	NumCompaniesWorked	0
▶	Over18	0
▶	Overtime	0
▶	PercentSalaryHike	0
▶	PerformanceRating	0
▶	RelationshipSatisfaction	0
▶	StandardHours	0
▶	StockOptionLevel	0
▶	TotalWorkingYears	0
▶	TrainingTimesLastYear	0
▶	WorkLifeBalance	0
▶	YearsAtCompany	0
▶	YearsInCurrentRole	0
▶	YearsSinceLastPromotion	0
▶	YearsWithCurrManager	0

Figure 4.3: Checking null values

The `isnull()` method is basically used to check if there are any null values in the dataset.

This will list number of null values in each column. In this project there are no null values in any of the 35 columns.

4.2 Accuracy

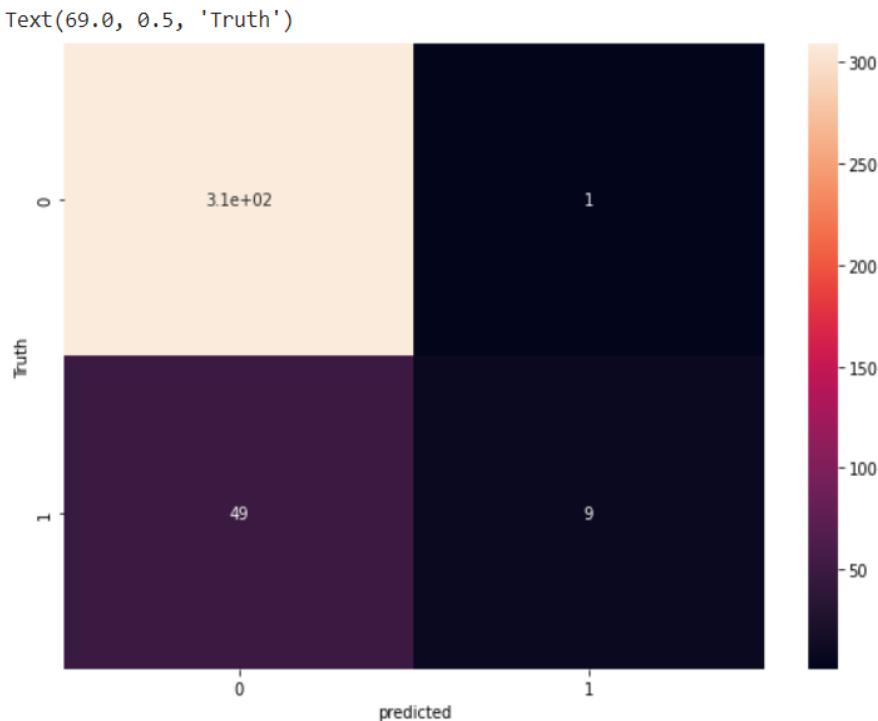


Figure 4.4: Confusion matrix of Random Forest model

There are many way to check the accuracy of the model.in this project one is used confusion matrix it show how well the model performed.it is showing almost 88 percentage accuracy which is good.

4.2.1 Trained model accuracy

I trained my dataset in two or more different data models then I got the maximum accuracy with Random Forest model.So I choose the models to train my data.

Used Models	Accuracy
Random forest	88%
Decision Tree	81%
SVM	86%
KNN	86%

Table 4.1: Trained model accuracy

4.2.2 Graph Results



The image shows a correlation matrix visualization. The matrix is a square grid of numbers representing the correlation coefficient between pairs of variables. The diagonal elements are all 1.000000, indicating perfect correlation with themselves. The variables listed on the rows and columns are: Age, DailyRate, DistanceFromHome, Education, EnvironmentSatisfaction, HourlyRate, JobInvolvement, JobLevel, JobSatisfaction, MonthlyIncome, MonthlyRate, NumCompaniesWorked, PercentSalaryHike, PerformanceRating, RelationshipSatisfaction, StockOptionLevel, TotalWorkingYears, and TrainingTimesLastYear. The values in the matrix range from -0.04985 to 0.680381.

	Age	DailyRate	DistanceFromHome	Education	EnvironmentSatisfaction	HourlyRate	J
Age	1.000000	0.010661	-0.001686	0.208034	0.010146	0.024287	
DailyRate	0.010661	1.000000	-0.004985	-0.016806	0.018355	0.023381	
DistanceFromHome	-0.001686	-0.004985	1.000000	0.021042	-0.016075	0.031131	
Education	0.208034	-0.016806	0.021042	1.000000	-0.027128	0.016775	
EnvironmentSatisfaction	0.010146	0.018355	-0.016075	-0.027128	1.000000	-0.049857	
HourlyRate	0.024287	0.023381	0.031131	0.016775	-0.049857	1.000000	
JobInvolvement	0.029820	0.046135	0.008783	0.042438	-0.008278	0.042861	
JobLevel	0.509604	0.002966	0.005303	0.101589	0.001212	-0.027853	
JobSatisfaction	-0.004892	0.030571	-0.003669	-0.011296	-0.006784	-0.071335	
MonthlyIncome	0.497855	0.007707	-0.017014	0.094961	-0.006259	-0.015794	
MonthlyRate	0.028051	-0.032182	0.027473	-0.026084	0.037600	-0.015297	
NumCompaniesWorked	0.299635	0.038153	-0.029251	0.126317	0.012594	0.022157	
PercentSalaryHike	0.003634	0.022704	0.040235	-0.011111	-0.031701	-0.009062	
PerformanceRating	0.001904	0.000473	0.027110	-0.024539	-0.029548	-0.002172	
RelationshipSatisfaction	0.053535	0.007846	0.006557	-0.009118	0.007665	0.001330	
StockOptionLevel	0.037510	0.042143	0.044872	0.018422	0.003432	0.050263	
TotalWorkingYears	0.680381	0.014515	0.004628	0.148280	-0.002693	-0.002334	
TrainingTimesLastYear	-0.019621	0.002453	-0.036942	-0.025100	-0.019359	-0.008548	

Figure 4.5: Correlation of the data set

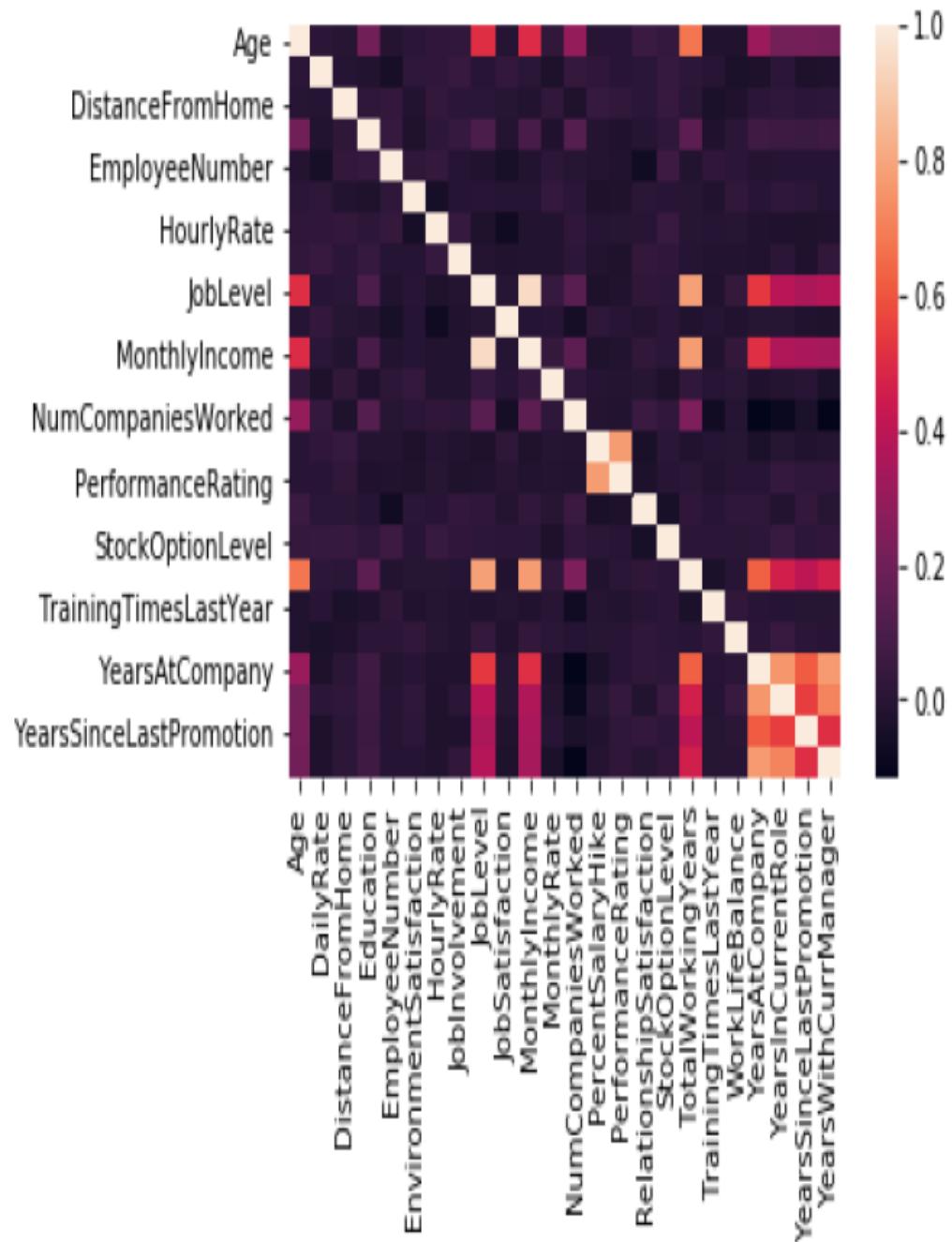


Figure 4.6: Heat map of data set

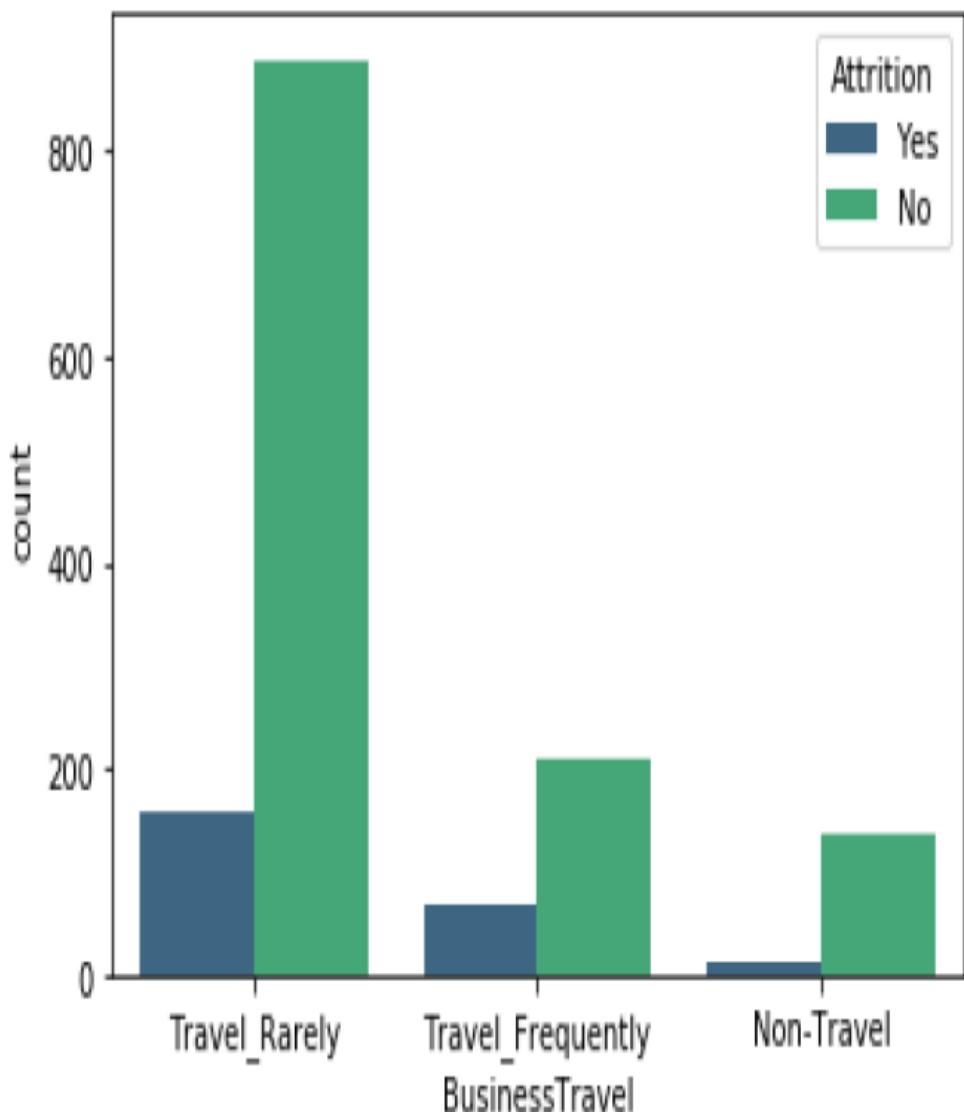


Figure 4.7: Traveling vs Attrition

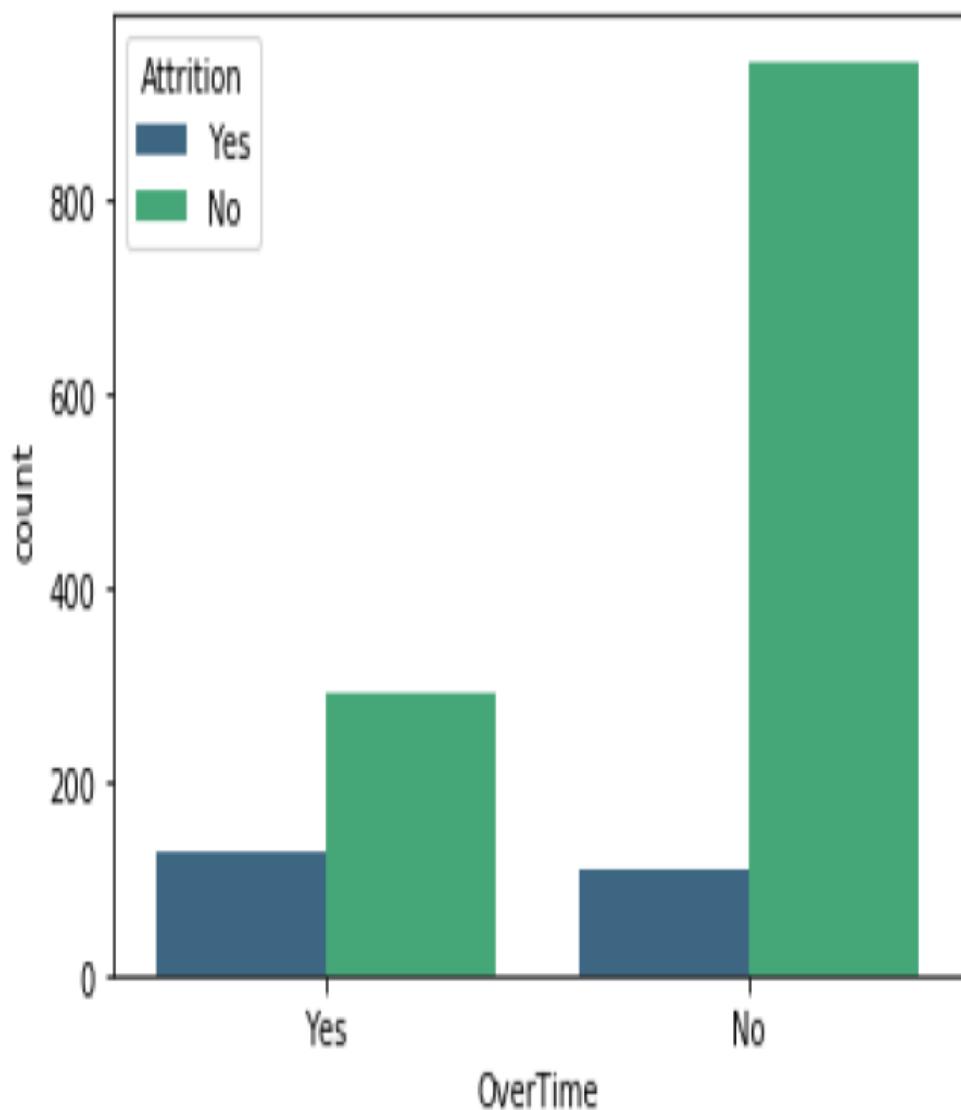


Figure 4.8: Overtime vs Attrition

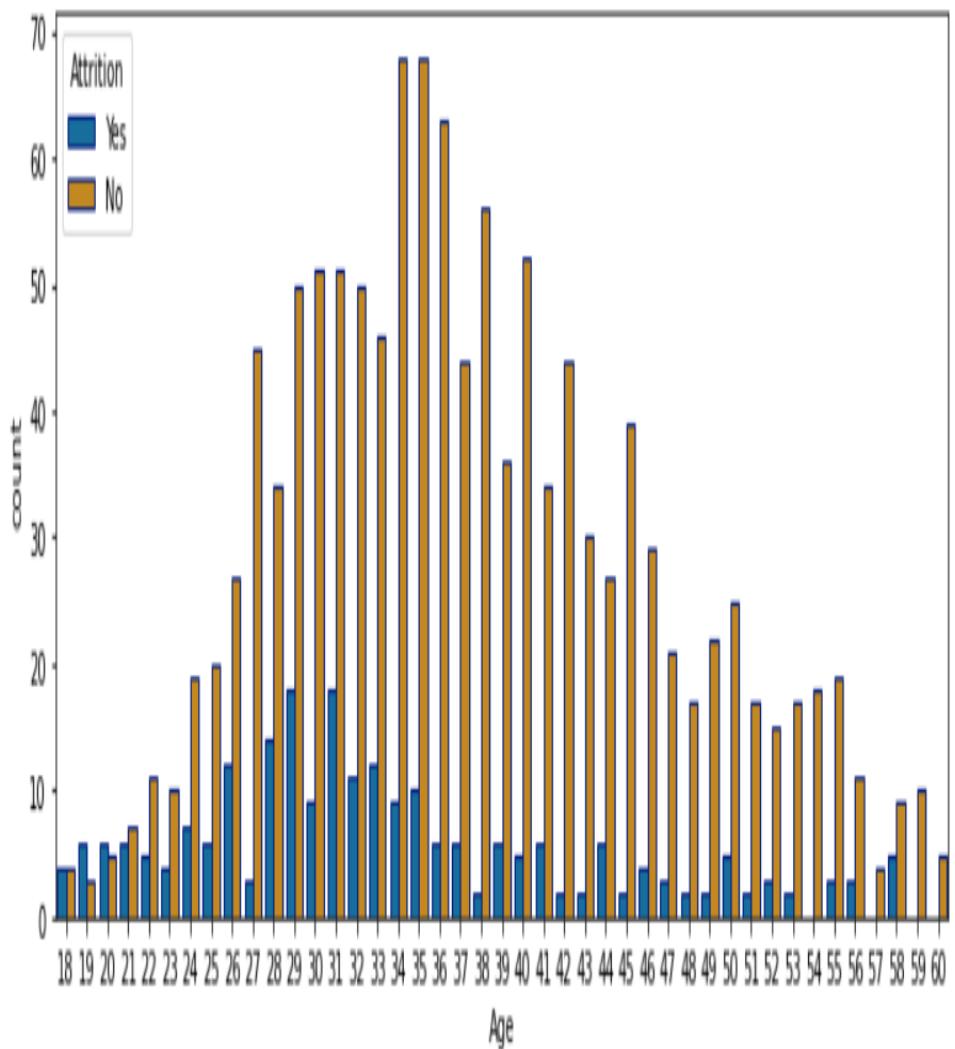


Figure 4.9: Age vs Attrition

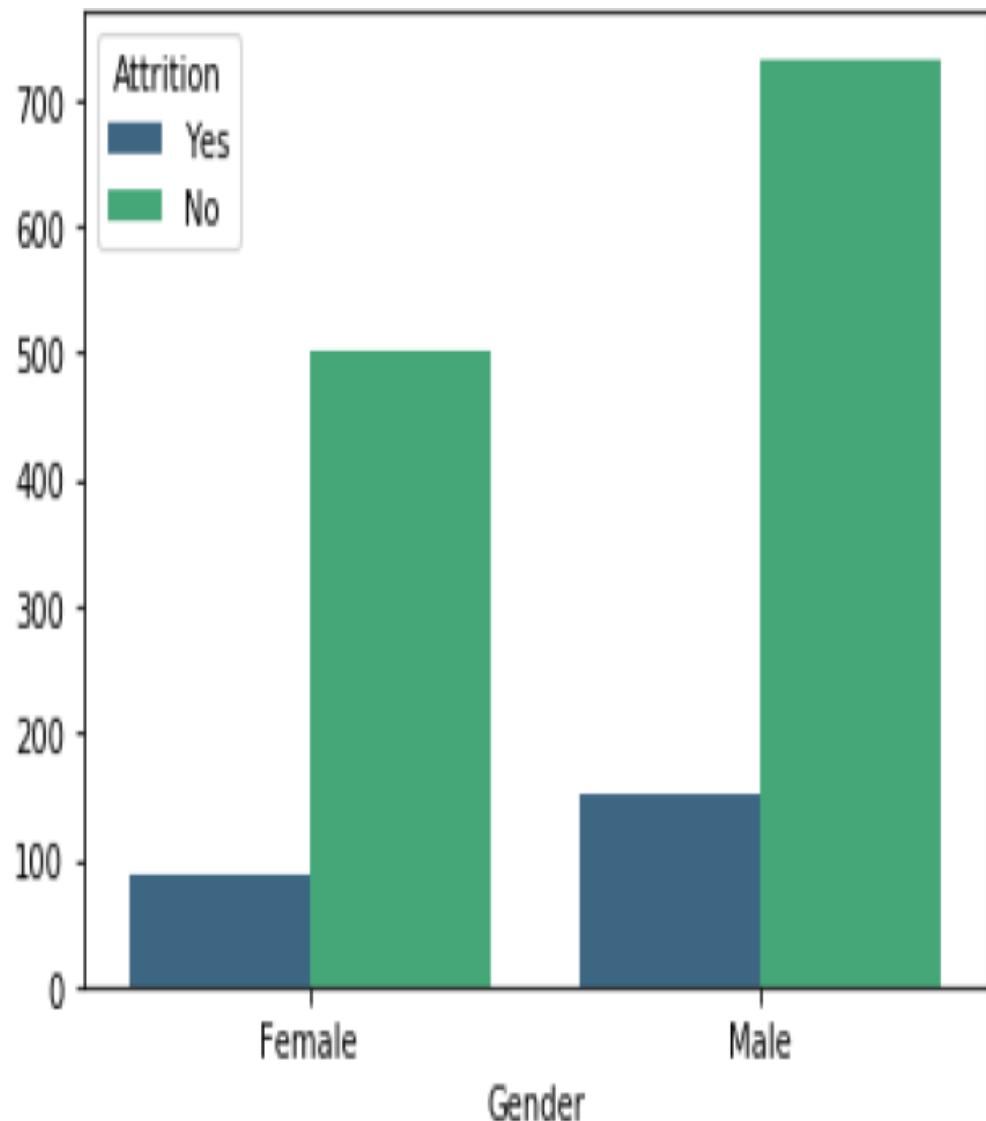


Figure 4.10: Gender vs Attrition

Chapter 5

CONCLUSION AND FUTURE SCOPE

5.1 Conclusion

Staff retention is important for business as it reduces costly training costs and saves company time. I developed several staff retention strategies after extensive research using speculative modeling strategies: Job satisfaction level and workplace quality have a beneficial effect on retaining employee. Employee retention opportunities increase with higher levels of employment and workplace satisfaction. Married men and women are more likely to be stable than those who are single or divorced. Therefore, it is beneficial for this organization to employ married men and women. The chances of diminishing are increased with lower incomes and longer stays in the company. Employees can be retained by providing timely promotions and various benefits. It is usually best to offer additional benefits or incentives

5.2 Future Scope

The application has got a lot of future scope.

- Add more data and features it make better prediction rate.
- Could also predict the reason for which the employee leaves the organization
- The website add more functions like contact number, college or company details and the news or the information inside the Employees can see the information about that college.

Chapter 6

APPENDIX

6.1 Source Code

```
[21] from sklearn.ensemble import RandomForestClassifier
    forest = RandomForestClassifier(n_estimators = 10, criterion = 'entropy', random_state = 0)
    forest.fit(X_train, Y_train)

    RandomForestClassifier(criterion='entropy', n_estimators=10, random_state=0)

[22] from sklearn.metrics import confusion_matrix

    cm = confusion_matrix(Y_test, forest.predict(X_test))

    TN = cm[0][0]
    TP = cm[1][1]
    FN = cm[1][0]
    FP = cm[0][1]

    print(cm)
    print('Model Testing Accuracy = "{}!'.format( (TP + TN) / (TP + TN + FN + FP)))
    print()# Print a new line

[[309  1]
 [ 49  9]]
Model Testing Accuracy = "0.8641304347826086!"
```

Figure 6.1: Random Forest Model code

6.1.1 main.py

```
#Import Libraries

import numpy as np

import pandas as pd

import seaborn as sns

from sklearn.model_selection import train_test_split

df=pd.read_csv("abc.csv")

df

df.isna().sum()

df.isnull().values.any()

df.describe()
```

```
df.info()

df['Attrition'].value_counts()

sns.countplot(df['Attrition'])

#Age vs attrition
#Show the number of employees that left and stayed
by age

import matplotlib.pyplot as plt
fig_dims = (12, 4)
fig, ax = plt.subplots(figsize=fig_dims)

#ax = axis
sns.countplot(x='Age', hue='Attrition', data = df,
palette="colorblind", ax = ax,
edgecolor=sns.color_palette("dark", n_colors = 1));

#Print all of the object data types and their unique values
for column in df.columns:
    if df[column].dtype == object:
        print(str(column) + ' : ' + str(df[column].unique()))
        print(df[column].value_counts())
        print("-----")

#Remove unneeded columns

#Remove the column EmployeeNumber
df = df.drop('EmployeeNumber', axis = 1) # A number
```

```
assignment

#Remove the column StandardHours
df = df.drop('StandardHours', axis = 1)
#Contains only value 80

#Remove the column EmployeeCount
df = df.drop('EmployeeCount', axis = 1)
#Contains only the value 1

#Remove the column EmployeeCount
df = df.drop('Over18', axis = 1)
#Contains only the value 'Yes'

#correlation of the columns
df.corr()

#Visualize the correlation
plt.figure(figsize=(14,14)) #14in by 14in
sns.heatmap(df.corr(), annot=True, fmt='.0%')

#conversion
from sklearn.preprocessing import LabelEncoder

for column in df.columns:
    if df[column].dtype == np.number:
        continue
    df[column] = LabelEncoder().
        fit_transform(df[column])

#Create a new column at the end of the dataframe
that contains the same value
```

```
df[ 'Age_Years' ] = df[ 'Age' ]  
#Remove the first column called age  
df = df.drop( 'Age' , axis = 1)  
#Show the dataframe  
df  
  
  
#Split the data  
X = df.iloc [ : , 1:df.shape [ 1 ] ].values  
Y = df.iloc [ : , 0 ].values  
  
  
X_train , X_test , Y_train , Y_test =  
train_test_split(X, Y, test_size = 0.25, random_state=0)  
  
  
Random forest clasifier  
  
  
from sklearn.ensemble import RandomForestClassifier  
forest = RandomForestClassifier  
(n_estimators=10,criterion='entropy',random_state=0)  
forest.fit(X_train, Y_train)  
  
  
from sklearn.metrics import confusion_matrix  
  
  
cm = confusion_matrix(Y_test, forest.predict(X_test))  
  
  
TN = cm[0][0]  
TP = cm[1][1]  
FN = cm[1][0]  
FP = cm[0][1]
```

```
print(cm)

print('Model Testing Accuracy = "{}!"'.
format((TP + TN)/(TP + TN + FN + FP)))

print()# Print a new line

%matplotlib inline

import matplotlib.pyplot as plt
import seaborn as sn
plt.figure(figsize=(10,7))
sn.heatmap(cm, annot=True)
plt.xlabel('predicted')
plt.ylabel('Truth')
```

6.1.2 App.py

```
import email
#from os import uname
#from importlib.metadata import files
import numpy as np
import pandas as pd
import csv
import sklearn
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import confusion_matrix
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score
```

```
import json

# Flask constructor

import flask

from DBConnection import Db

from flask import request

from flask import Flask , render_template , redirect

#from flask_session import Session

from DBConnection import Db

app = Flask(__name__ , template_folder='template')

u_name = ""

def eg(res):
    global u_name

    #u_name=session.get('uname')

    age = res.get('age')

    c_w = res.get('c_w')

    y_c = res.get('y_c')

    y_curr = res.get('y_curr')

    js = res.get('js')

    es = res.get('es')

    wb = res.get('wb')

    ji = res.get('ji')

    l_pro = res.get('l_pro')

    c_mng = res.get('c_mng')
```

```
wh = res.get('wh')
value1 = res.get('value1')
value2 = res.get('value2')
value3 = res.get('value3')
mi = res.get('mi')
gen = res.get('gen')
ms = res.get('ms')
bt = res.get('bt')
ot = res.get('ot')
t_w_y = res.get('t_w_y')
t_p = res.get('t_p')
dr = mi/20
hr = dr/wh
xyz= """

df = pd.read_csv(r"C:\\\\Users\\\\junaidsar\\\\Downloads\\\\Employee_attrition_prediction\\\\WA_Fn-UseC-HR-Employee-Attrition.csv")
# print(df.head(7))
df = df.drop('Over18', axis=1)
df = df.drop('EmployeeNumber', axis=1)
df = df.drop('StandardHours', axis=1)
df = df.drop('EmployeeCount', axis=1)
df = df.drop('Education', axis=1)
df = df.drop('JobLevel', axis=1)
df = df.drop('MonthlyRate', axis=1)
df = df.drop('PercentSalaryHike', axis=1)
df = df.drop('PerformanceRating', axis=1)
```

```
df = df.drop('RelationshipSatisfaction', axis=1)
df = df.drop('StockOptionLevel', axis=1)
df = df.drop('DistanceFromHome', axis=1)

for column in df.columns:
    if df[column].dtype == np.float64:
        continue
    df[column] = LabelEncoder().fit_transform(df[column])

# print(df.head(10))
df['Age_Years'] = df['Age']
df = df.drop('Age', axis=1)

x = df.iloc[:, 1:df.shape[1]].values
y = df.iloc[:, 0].values

# Split the dataset into 75% training and 25%
# testing
x_train, x_test, y_train, y_test =
train_test_split(x, y,
test_size=0.25, random_state=0)

# Use the random forest classifier module
forest = RandomForestClassifier(n_estimators=10,
criterion='entropy', random_state=0)
forest.fit(x_train, y_train)

# Get the accuracy on the training dataset
```

```
print(forest.score(x_train, y_train))

# Show the confusion matrix and accuracy score
for the
model on the test data
cm = confusion_matrix(y_test, forest.predict(x_test))

TN = cm[0][0]
TP = cm[1][1]
FN = cm[1][0]
FP = cm[0][1]

print(cm,u_name)
print('Random forestModel Testing Accuracy -> {}'
      .format((TP + TN) / (TP + TN + FN + FP)))

categ = ['Employee will stay', 'Employee will leave']
custom_dt = [[bt, dr, value1, value2, es, gen, hr, ji, value3,
              js, ms, mi, c_w, ot, t_w_y, t_p, wb, y_c, y_curr, l_pro, c_mng, age]]
print(categ[int(forest.predict(custom_dt))])

att = categ[int(forest.predict(custom_dt))]

db = Db()
#print(u_name)
sv = db.selectOne("select * from attrition where email="
                  +" "+ u_name + "''")
if sv is None:
    tmp = db.selectOne("select * from login where email=
```

```
    "" + u_name + """)

db.insert("insert into attrition VALUES(
    "+str(tmp['xy'])+", '"+u_name+"', '"+att+"')")

return '''<script> alert('Submitted**')</script>'''
else:

    print("You have already attended the questionnaire")
    return '''<script> alert('You have already attended
the questionnaire')</script>'''

@app.route('/')

def handle_data():
    return render_template('ind.html')

@app.route('/first_page')

def first_page():
    global u_name
    u_name = ""
    return render_template('ind.html')

'''@app.route('/adm_reg', methods=['get', 'post'])

def adm_reg():

    if request.method == "POST":

        name = request.form['name']
        email = request.form['email']
        passw = request.form['password']
```

```
typ = "admin"
status = "admin"
db = Db()
db.insert("insert into registration VALUES
(0,'" + name + "','" + email + "','" + passw + ',
'" + typ + "')")
temp1 = db.selectOne("select * from registration
where email=''" + email + "'")
xy = temp1[ 'id ']
db.insert("insert into login VALUES('" + str(xy) + ',
'" + email + "','" + passw + "','" + typ + "','" + status + "')")
return <script> alert('User Registered ');
window.location="/emp_log"</script>
else:
    return render_template('hr_reg.html')'''
```



```
@app.route('/emp_reg', methods=['get', 'post'])
def emp_reg():
    if request.method == "POST":
        name = request.form[ 'name' ]
        phone = str(request.form[ 'phone' ])
        gender = request.form[ 'fav_language' ]
        date = request.form[ 'birthday' ]
        city = request.form[ 'City' ]
        pin = str(request.form[ 'pin' ])
        state = request.form[ 'state' ]
        quali = request.form[ 'qual' ]
        prof = request.form[ 'prof' ]
```

```
email = request.form[ 'email' ]
passw = request.form[ 'password' ]

typ = "employee"
status = "pending"
db = Db()

db.insert("insert into register VALUES
(0,'" + name + "','" + phone + "','" + gender + ',
'" + date + "','" + city + "','" + pin + ',
'" + state + "','" + quali + "','" + prof + ',
'" + email + "','" + passw + "')")

temp = db.selectOne("select * from register where
email=''" + email + "'')")

xy = temp[ 'id' ]

db.insert("insert into login VALUES('' + str(xy) + '',
'" + email + "','" + passw + "','" + typ + ',
'" + status + "')")

return '''<script> alert('User Registered');
window.location= "/"</script>''

else:
    return render_template('cust_reg.html')

'''@app.route('/adm_log', methods=['get', 'post'])

def adm_log():
    global u_name
    if request.method == "POST":
        em = request.form[ 'username' ]
        print(em)
```

```
    pa = request.form[ 'pass' ]
    db = Db()
    ss = db.select("select * from login where
    user_name='" + em + "' and pass='" + pa + "'")
    if ss is not None:
        u_name = str(em)
        return redirect( '/adminhome' )
    else:
        return <script> alert('Not Found');
        window.location="/</script> // single quotes
else:
    return render_template( 'hr_log.html' )'''
```



```
@app.route( '/emp_log' , methods=[ 'get' , 'post' ])
def emp_log():
    global u_name
    if request.method == "POST":
        em = request.form[ 'username' ]
        pa = request.form[ 'pass' ]
        db = Db()
        ss = db.selectOne("select * from login where
        email='" + em + "' and passw='" + pa + "'")
        if ss is not None and ss[ 'status' ] == 'user':
            u_name = str(em)
            if ss[ 'typ' ] == 'admin':
                return redirect( '/adminhome' )
            else:
                #session[ 'uname' ] =em
```

```
        return redirect('/emphome')

    elif ss is not None and ss['status'] == 'pending':
        return '''<script> alert('No permission yet');
window.location="/</script>''

    else:
        return '''<script> alert('User Not Found');
window.location="/</script>''

else:
    return render_template('cust_log.html')

@app.route('/adminhome')
def adminhome():
    return render_template('dashboard_adm.html',
                           data=u_name)

@app.route('/emphome')
def emphome():
    return render_template('dashboard_emp.html',
                           data=u_name)

@app.route('/test', methods=['POST'])
def test():
    output = request.get_json()
    result = json.loads(output)
    print(result)
    eg(result)
```

```
    return result

@app.route('/view_users')
def view_users():
    db = Db()
    res = db.select("select * from register")
    return render_template('users.html', data=res)

@app.route('/emp_details')
def emp_details():
    db = Db()
    res = db.select("select * from attrition")
    return render_template('attrition_or_not.html',
                           data=res)

@app.route('/profile')
def profile():
    db = Db()
    res = db.select("select * from register where
                    email=''" + u_name + "'")
    return render_template('user.html', data=res)

@app.route('/questionnaire')
def questionnaire():
    db = Db()
```



```
#file = request.files.get('ab')
#if file:
#    df1 = pd.read_csv(files_csv["file"])
#    print(df1)

return render_template('addon.html')

@app.route('/test1', methods=['POST'])
def test1():
    output = request.get_json()
    result = json.loads(output)
    print(result)
    eg1(result)
    return result

def eg1(res):
    age = res.get('fp')

    return

@app.route('/approve')
def approve():
    db = Db()
    res = db.select("select * from login,register where
    login.status='pending' and login.email=register.email")
    return render_template('approve_user.html', data=res)

@app.route('/approveuser/<email>')
def approveuser(email):
```

```
db = Db()

res = db.update("update login set status='user'
where email=' + str(email) + ''')
return '''<script> alert('User Approved');
window.location="/approve"</script>'''

@app.route('/disapproveuser/<email>')
def disapproveuser(email):
    db = Db()

    db.delete("delete from login where email=
' + str(email) + ''")
    db.delete("delete from register where email=
' + str(email) + ''")
    return '''<script> alert('User Disapproved');
window.location="/approve"</script>'''

@app.route('/notifications', methods=['POST', 'GET'])
def notifications():
    db = Db()

    if request.method == "POST":
        em = request.form['g_id']
        mess = request.form['ta']
        db.insert("insert into notify values('' + em + '',
'' + mess + '')")
    return render_template('notifications.html')
```

```
@app.route('/notifications2', methods=['POST', 'GET'])

def notifications2():
    db = Db()

    ''' u_name = db.selectOne("select * from register where
    email=''" + email + "'") '''
    res = db.select("select * from notify where gmail_id=
    '" + u_name + "'")

    return render_template('notifications2.html', data=res)

@app.route('/fp')

def fp():
    return render_template('for_pass.html')

@app.route('/res_pass', methods=["POST", "GET"])

def res_pass():
    db = Db()

    em = request.form.get("m_id")
    mess = request.form.get("pass_wd")
    t1 = db.update("update register set password=
    '" + mess + "' where email=''" + str(em) + "'")
    t2 = db.update("update login set password=
    '" + mess + "' where email=''" + str(em) + "'")
    return redirect('\emp_log')

if __name__ == '__main__':
    app.run(debug=True)
```

6.2 Screen Shots

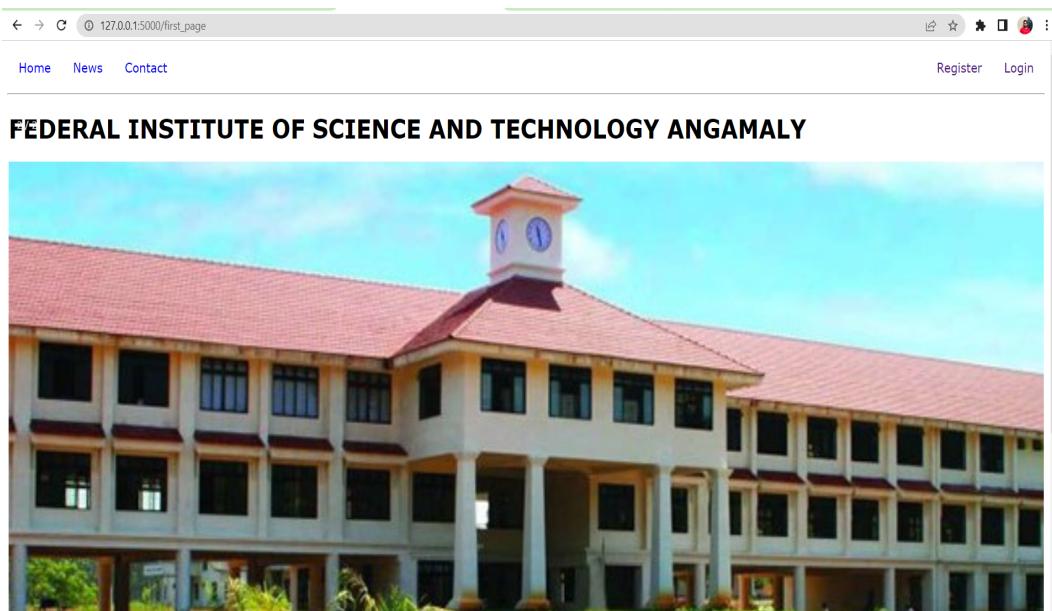


Figure 6.2: Home PAGE

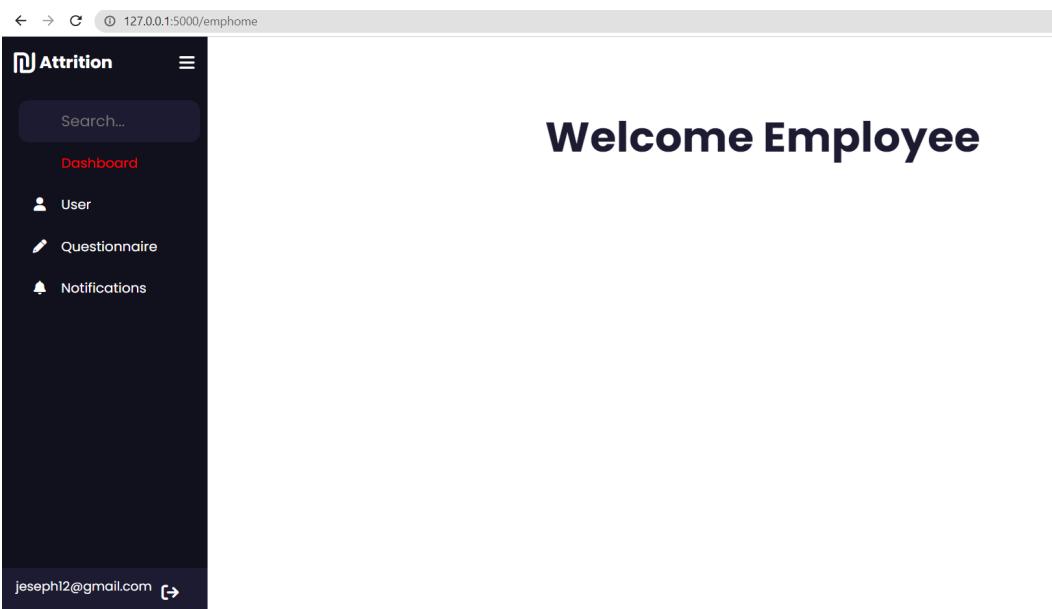


Figure 6.3: Employee Home PAGE

The screenshot shows a web application interface titled "Attrition". On the left, there is a dark sidebar with navigation links: "Search...", "Dashboard", "User Profile" (which is highlighted in red), "Questionnaire", and "Notifications". The main content area displays a table with the following data:

ID	Name	Email	Phone	Qualification	Profession	Birthday	State
201	joseph	joseph12@gmail.com	78965412	MCA	student	2022-06-17	Kerala

Figure 6.4: Employee profile page

The screenshot shows a web application interface titled "Attrition". On the left, there is a dark sidebar with navigation links: "Search...", "Dashboard", "User Profile" (highlighted in red), "Questionnaire" (highlighted in pink), and "Notifications". The main content area displays a form with the following fields:

- Age: [Input field]
- Gender: Male Female Other
- Marital status: Married Single
- Department: Teacher [Select dropdown]
- Education Field: MCA [Select dropdown]
- Job Role: Teacher [Select dropdown]
- Business Travel: Frequent Rare None
- Frequent Overtime work: Yes No
- Monthly Income: [Input field]

Figure 6.5: Employee Questioner page

The screenshot shows a web browser window with the URL 127.0.0.1:5000/notifications2. The page has a dark blue header with the 'Attrition' logo and a search bar. On the left, there's a sidebar with 'Dashboard', 'User Profile', 'Questionnaire', and 'Notifications' (which is highlighted in red). The main content area displays a table with two rows:

Date and Time	Message
Sunday 12:19pm 3 Jul 2022	Admin: welcome kiran
	Admin: your work is good keep going

Figure 6.6: View notification page

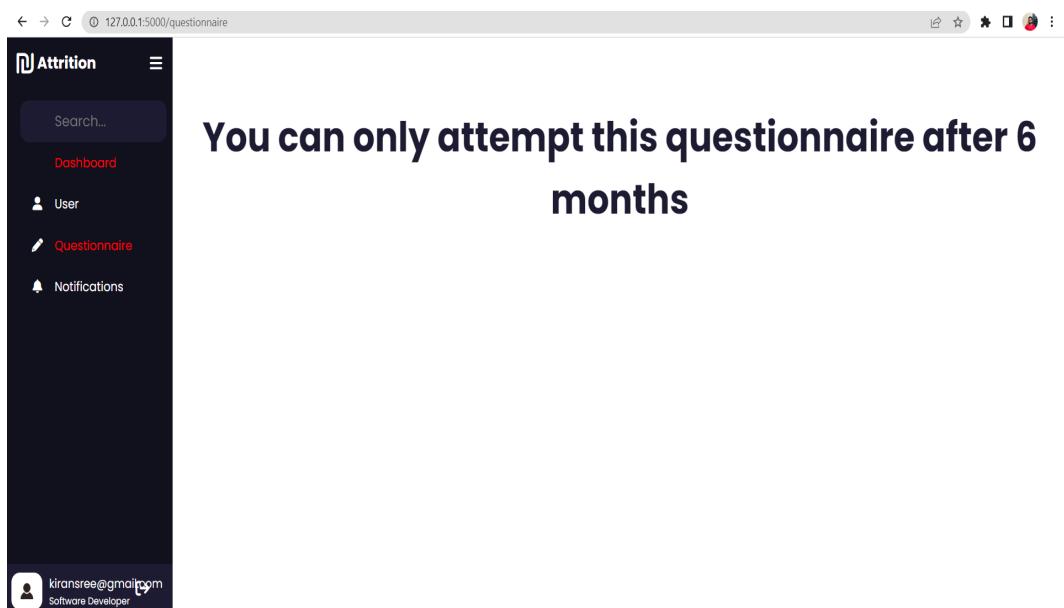


Figure 6.7: After attending the questioner

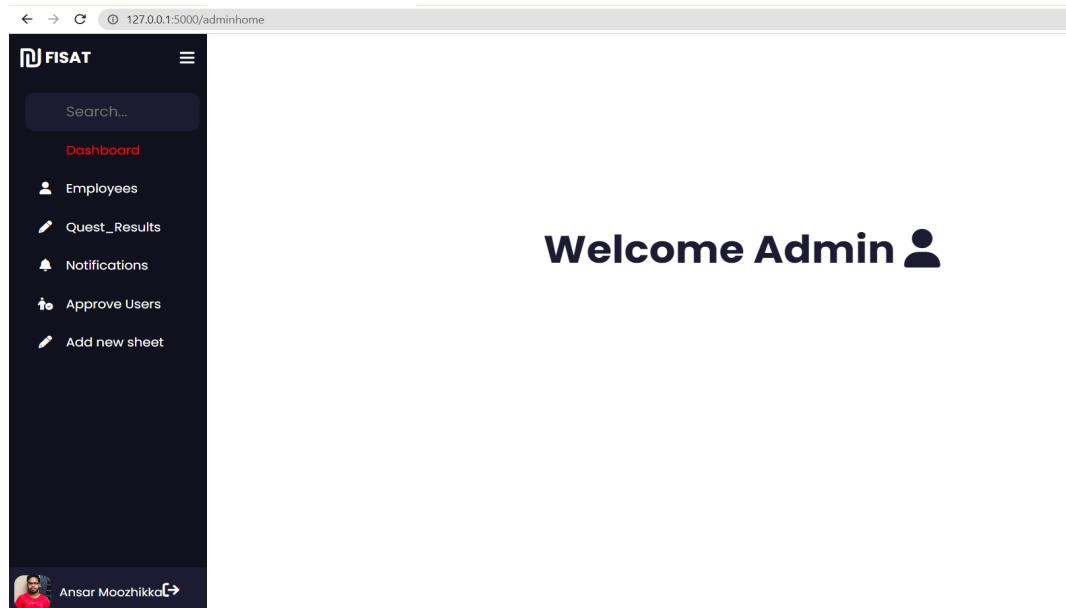


Figure 6.8: Admin home page

The screenshot shows a web application interface for viewing an employee profile. At the top, there is a header bar with navigation icons and the URL '127.0.0.1:5000/profile'. The main area has a dark sidebar on the left containing the Attrition logo, a search bar, and a list of menu items: Dashboard, User Profile, Questionnaire, and Notifications. On the right, there is a table displaying an employee's profile information. The table has columns for ID, Name, Email, Phone, Qualification, Profession, Birthday, and State. The data is as follows:

ID	Name	Email	Phone	Qualification	Profession	Birthday	State
201	joseph	jeseph12@gmail.com	78965412	MCA	student	2022-06-17	Kerala

Figure 6.9: Employee profile page

← → ⌂ 127.0.0.1:5000/employees

The screenshot shows a web application titled "Attrition". The left sidebar contains navigation links: Dashboard, Employees (highlighted in red), Quest_result, Notifications, Approve Users, and Add new sheet. A user profile picture and name "Ansar Moozhikka" are at the bottom. The main content area displays a table of employee details:

ID	Name	Email
104	Ansar.M	ansarmd@gmail.com
200	suttu	suttu@gmail.com
201	joseph	jeseph12@gmail.com
206	kiran karthikeyan	kiransree@gmail.com

Figure 6.10: Registered Employee details

← → ⌂ 127.0.0.1:5000/quest_res

The screenshot shows a web application titled "FISAT". The left sidebar contains navigation links: Dashboard, Employees (highlighted in red), Quest_result, Notifications, Approve Users, and Add new sheet. A user profile picture and name "Ansar Moozhikka" are at the bottom. The main content area displays a table of employee details:

SI.No	ID	Name	Email	Attrition Result
1	104	Ansar.M	ansarmd@gmail.com	Employee will stay
2	206	kiran karthikeyan	kiransree@gmail.com	Employee will stay

Figure 6.11: Registered Employee details

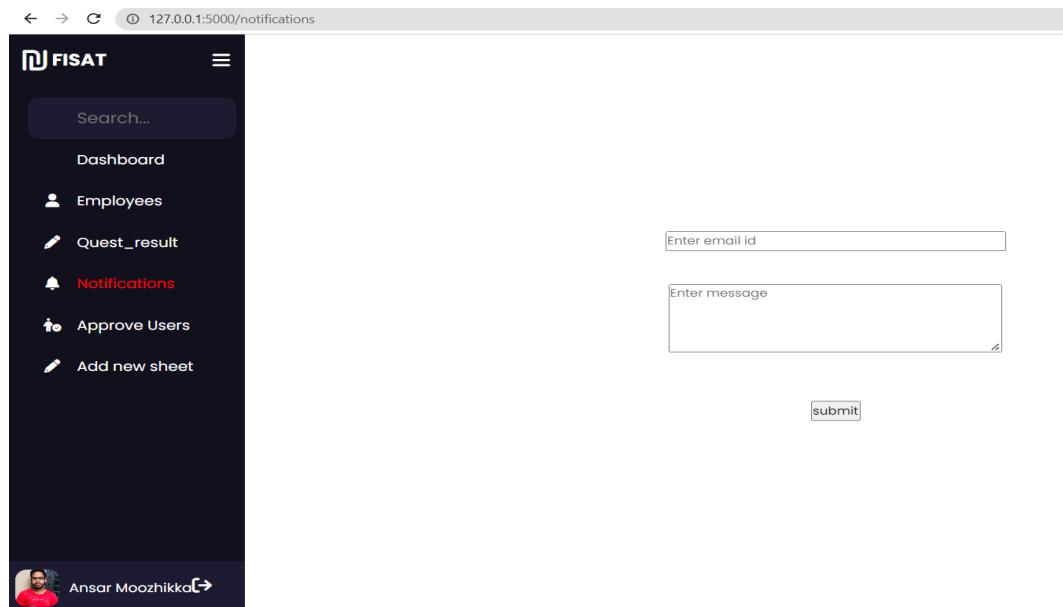


Figure 6.12: Send notification to the employees

The screenshot shows the Attrition application interface. The left sidebar has a dark theme with white icons and text. It includes a search bar, a dashboard link, and several menu items: Employees (which is highlighted in red), Quest_result, Notifications, Approve Users, and Add new sheet. Below the sidebar is a user profile for 'Ansar Moozhikka' with a right-pointing arrow. The main content area has a light background. At the top, there's a header with back, forward, and refresh buttons, and the URL '127.0.0.1:5000/approve'. To the right of the header is a toolbar with various icons. Below the toolbar is a table with the following data:

ID	Name	Email	Phone	Qualification	Profession	Birthday	State	Action
207	Manjima	manjumanji@gmail.com	985614723	Commerce	Teacher	1999-05-13	kerala	Approve Disapprove

Figure 6.13: Approve or Reject the registration

Chapter 7

REFERENCES

- [1] Srivastava, Devesh Kumar, and Priyanka Nair. "Employee attrition analysis using predictive techniques." International Conference on Information and Communication Technology for Intelligent Systems. Springer, Cham, 2017.
- [2] Setiawan, I., et al. "HR analytics: Employee attrition analysis using logistic regression." IOP Conference Series: Materials Science and Engineering. Vol. 830. No. 3. IOP Publishing, 2020
- [3] M.Sudheer Kumar, Obulesu Varikunta, K.Ramakrishna, "Employee Attrition and Retention Strategies in Manufacturing: An Empirical Study in Amara Raja Batteries Limited," International Journal of Innovative Technology and Exploring Engineering (IJITEE), Volume-8, Issue-7, May, 2019, pp. 2962-2968.
- [4] Diwakar Singh, "A Literature Review on Employee Retention with Focus on Recent Trends," International Journal of Scientific Research in Science and Technology (IJSRST 2019), Volume 6, Issue 1, pp. 425-431.
- [5] From Big Data to Deep Data to Support People Analytics for Employee Attrition Prediction NESRINE BEN YAHIA 1, JIHEN HLEL1 AND RICARDO COLOMO-

PALACIOS 2, (Senior Member, IEEE) 1RIADI Laboratory, National School of Computer Sciences, University of Manouba, Manouba 2010, Tunisia 2Computer Science Department, Østfold University College, 1783 Halden, Norway

- [6] Early Prediction of Employee Attrition using Data Mining Techniques Conference: 2018 IEEE 8th International Advance Computing Conference (IACC) Project: Employee Attrition
- [7] IBM Employee Attrition Analysis Shenghuan Yang Jiangxi University of Finance and Economics Md Tariqul Islam Syracuse University
- [8] An Improved Machine Learning-Based Employees Attrition Prediction Framework with Emphasis on Feature Selection Saeed Najafi-Zangeneh 1, Naser Shams-Gharneh 1, Ali Arjomandi-Nezhad 2 and Sarfaraz Hashemkhani Zolfani