**Intrusion Detection and Prevention using Honey pot Network for Cloud Security**

# Project Report Submitted

In Partial Fulfilment of the Requirements For the Degree of

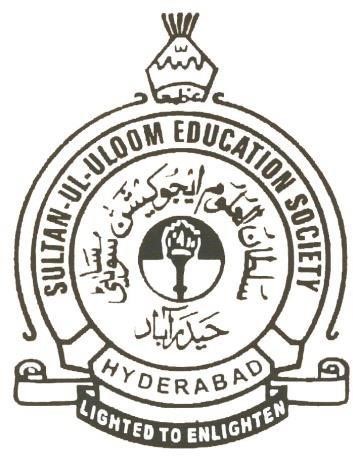
**BACHELOR OF ENGINEERING**

**IN**

# INFORMATION TECHNOLOGY

Submitted By

# Syed Talmeez (1604-16-737-097) Syed Aleem (1604-16-737-102) Syed Anas (1604-16-737-108)



**INFORMATION TECHNOLOGY DEPARTMENT**

# MUFFAKHAM JAH COLLEGE OF ENGINEERING& TECHNOLOGY

(Affiliated to Osmania University)

**Mount Pleasant, 8-2-249, Road No. 3, Banjara Hills, Hyderabad-34 2019-20**

# CERTIFICATE

It is certified that the work contained in the project report titled “Intrusion Detection and Prevention using Honey Pot Network for Cloud Computing,” by

|  |  |
| --- | --- |
| **Mr. Syed Talmeez** | **(1604-16-737-097)** |
| **Mr. Syed Aleem** | **(1604-16-737-102)** |
| **Mr. Syed Anas** | **(1604-16-737-108)** |

has been carried out under my/our supervision and that this work has not been submitted elsewhere for a degree”

|  |  |
| --- | --- |
| Project Supervisor | Signature of Head |
| Dr. Devasish Pal | **Dr. MOUSMI AJAYCHAURASIA** |
| Information Technology Dept. | Information Technology Dept. |
| MJ College of Engineering& Tech. | MJ College of Engineering &Tech. |
| Hyderabad –500 034. | Hyderabad – 500 034 |

## External Examiner

**ACKNOWLEDGEMENT**

With great pleasure we want to take this opportunity to express our heartfelt gratitude to all the people who helped in making this project work a grand success.

We are grateful to **Dr. Devasish Pal,** Professor, Information Technology Department for

his valuable suggestions and guidance during the execution of this project work.

We would like to thank **Dr. MOUSMI AJAY CHAURASIA**, Head of Information

Technology Department, for her moral support throughout the period of my study in MJCET.

We are highly indebted to, Principal MJCET for giving permission to carry out this project in

a successful and fruitful way.

We would like to thank the Teaching & Non- teaching staff of Information Technology

Department for their co-operation

Finally, we express our sincere thanks to **Prof. Basheer Ahmed**, Advisor cum Director, MJCET, for his continuous care. We sincerely acknowledge and thank all those who gave support directly and indirectly in completion of this project work.

**Syed Talmeez- 1604-17-737-097**

|  |  |
| --- | --- |
| **Syed Aleem-** | **1604-17-737-102** |
| **Syed Anas-** | **1604-17-737-108** |

# TABLE OF CONTENTS

**S.No. Title Page No.**

**TITLE PAGE CERTIFICATE ACKNOWLEGEMENT ABSTRACT**

## 1. INTRODUCTION 1-4

1.1 General 1

1.2 Problem Statement 2

1.3 Objectives 2

1.4 Types of Employee Turnover 3

1.4.1 Internal vs External Turnover 3

1.4.2 Skilled vs Unskilled Employees 3

1.4.3 Voluntary vs Involuntary Turnover 3

1.5 Causes of Employee Turnover 4

1.6 Organization of Report 4

## 2. LITERATURE SURVEY 5-12

2.1 Related research papers 5

2.1.1 Factors that lead to Employee Attrition 5

2.1.2 Classification Algorithms 6 a) Logistic Regression 6

### b) Naïve Bayes 6 c) Decision Tree 6

d) Random Forest 6 2.1.3 Stacking Classifier 6

2.1.4 Ensemble Methods 7

2.1.4.1 Controlling the tree size 7

2.1.4.2 Regression 7

2.1.4.3 Stacking with probability distributions 7

2.1.4.4 Discussion of Experimental Results 7

2.1.4.5 Comparison between various works 8

2.2 Technology Used 9

2.2.1 Stacking Classifier 9

2.2.2 Naïve Bayes 9

2.2.3 Decision Tree 9

2.2.4 Logistic Regression 10

2.2.5 Random Forest 10

2.2.6 Adaptive Boosting 10

2.2.7 Classifier Evaluation Index 11

2.2.8 Anaconda 12

2.2.9 Jupyter Notebook 12

## 3. SYSTEM ANALYSIS 13-23

3.1. Problem with Existing System 13

3.2. Proposed Work 14

3.2.1. Data set 14

3.2.2. Data Pre-processing 14

3.2.3. Test dataset and training dataset 14

3.2.4. Data Classification Techniques 15

3.2.4.1. Decision Tree 15

3.2.4.2. Support Vector Machine 16

3.2.4.3. Adaptive Boosting 16

3.2.4.4. Random Forest 17

3.3. Feasibility Study 18

3.3.1. Technical Feasibility 18

3.3.2. Economic Feasibility 18

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 3.3.3. Operational Feasibility |  |  |  | 19 |
| 3.4. Software Requirement Specification |  |  |  | 19 |
| 3.4.1. Introduction |  |  |  | 19 |
| 3.4.1.1. Purpose |  |  |  | 19 |
| 3.4.1.1 Scope |  |  |  | 19 |
| 3.4.2 Overall Description |  |  |  | 20 |
| 3.4.2.1 Product Perspective |  |  |  | 20 |
| 3.4.2.2 Product Functions |  |  |  | 21 |
| 3.4.2.3 User Class and Characteristics |  |  |  | 21 |
| i) Manager (User) |  |  |  | 21 |
| ii) Employee |  |  |  | 21 |
| 3.4.2.4 Operating Environment |  |  |  | 22 |
| 3.4.3 External Interface Requirements |  |  |  | 22 |
| 3.4.3.1 User Interface |  |  |  | 22 |
| 3.4.3.2 Software Interfaces |  |  |  | 23 |
| 3.5. COCOMO MODEL |  |  |  | 24 |
| **4. SYSTEM DESIGN** |  |  |  | **26-30** |
| 4.1 System Architecture |  |  |  | 26 |
| 4.2 Division of dataset |  |  |  | 27 |
| 4.3 Stacking and Meta Classifiers |  |  |  | 28 |
| 4.4 Use Case Diagram for Employee Attrition System | | |  | 29 |
| 4.5 Fish Bone Diagram for reasons of resignation | | |  | 29 |
| 4.6 Activity Diagram to prevent Attrition | | |  | 30 |

|  |  |  |  |
| --- | --- | --- | --- |
| **5. IMPLEMENTATION** |  |  | **31-59** |
| 5.1. Data Set Used |  |  | 31 |
| 5.2. Front End Execution |  |  | 32 |
| 1.2.1 Index.html |  |  | 32 |
| 1.2.2 Test.py |  |  | 40 |
| 5.3. Back End Execution |  |  | 42 |
| 5.3.1. EAF.py |  |  | 42 |
| 5.3.2. Stackupdated.py |  |  | 58 |
| **6. TESTING** |  |  | **60-62** |
|  |  |  |  |
| **7. OUTPUT SCREENS** |  |  | **63-65** |
| 7.3.Home Page of the Employee Attrition Predictor |  |  | 63 |
| 7.4.Employee with Low risk of Leaving the company |  |  | 64 |
| 7.5.Employee with High risk of Leaving the company |  |  | 65 |
| **CONCLUSION** |  |  | **66** |
| **FUTURE ENHANCEMENT** |  |  | **66** |
| **REFERENCES** |  |  | **67** |
| **Appendix 1** |  |  | **68** |
| **Appendix 2** |  |  | **69** |

# LIST OF FIGURES

|  |  |  |
| --- | --- | --- |
| **Figure Number** | **Figure Name** | **Page No.** |
| 1.1 | Reasons for Employee Attrition | 3 |
| 2.2.1 | Factors that lead to Employee Attrition | 5 |
| 2.1.4.5 | Attrition by Job Level | 8 |
| 2.2.8 | Anaconda Platform | 12 |
| 2.2.9 | Jupyter Notebook | 12 |
| 3.1 | Employee turnovers and its causes | 13 |
| 3.2.3.1 | Revelations from Work-force turnover study | 15 |
| 3.2.3.2 | Attrition performance evolution | 15 |
| 3.2.4.1 | Working of Decision Tree | 16 |
| 3.2.4.2 | Working of Support Vector Machine | 16 |
| 3.2.4.3 | Working of Adaptive Boosting | 17 |
| 3.2.4.4 | Working of Random Forest | 17 |
| 3.4.2.1 | Block Diagram of the product | 20 |
| 3.4.3.1 | User Interface of the Attrition Predictor | 22 |
| 4.1 | Flowchart for Employee Attrition | 24 |
| 4.2 | Flowchart for Data Set | 25 |
| 4.3 | Flowchart for Stacking and Meta classifier | 26 |
| 4.4 | Use Case Diagram for Employee Attrition | 27 |
| 4.5 | Reasons of resign using Fishbone Diagram | 27 |
| 4.6 | Activity Diagram to prevent Employee Attrition | 28 |
| 6.1 | Using Employee ID | 58 |
| 6.2 | Determining the Confidence Level | 58 |
| 6.3 | Graphical Representation of the data | 59 |
| 6.4 | Prediction of Attrition | 60 |
| 7.1 | Opening page of the Employee Attrition Predictor | 61 |
| 7.2 | Employee with Low risk of leaving the company | 62 |
| 7.3 | Employee with High risk of leaving the company | 63 |

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **Table No.** | **Table Name** | **Page No.** |
| 5.1.1 | Data Set used for predictions | 21 |
| 5.3.1 | Converting all data values into floating points | 45 |
| 5.3.2 | Comparison between Algorithms based on ROC and AUC values | 52 |
| 5.3.3 | Estimating the Feature Coefficients | 54 |

## Abstract

Now a days Employee Attrition prediction has become a major problem in organizations. Employee Turnover is a big issue for the organizations specially when trained, technical and key employees leave for better opportunities. This results in financial loss to replace a trained employee. Therefore, we use the current and past employee data to analyse the common reasons for employee attrition. Supervised machine learning methods are described, demonstrated and assessed for the prediction of employee turnover within an organization. In this study, numerical experiments for real and simulated human resources datasets representing organizations of small, medium- and large-sized employee populations are performed using, Logistic Regression and Random Forest, Stacking is used to achieve greater accuracy through AdaBoost, SVM and Decision Tree methods on the human resource data.

This machine learning algorithms are linearly combined with the weak classifiers which gives a final classifier of large accuracy in predicting the attrition. For this we implement feature selection method on the data and analyse the results to alert the organisation for attrition. This is helpful for the companies to predict employee attrition, and also helpful for their economic growth by reducing their human resource cost. The company after knowing this can make necessary retention plans to stop the employees from leaving by improving the feature that gives the maximum chance of turnover. This system can therefore be employed in other sectors like education, entertainment, health etc.

# 1. INTRODUCTION

## 1.1 Overview

An interdisciplinary field, **Data Science** deals with processes and systems, that are used to extract knowledge or insights from large amounts of data. Data science is a continuation of data analysis fields like [data mining,](https://www.educba.com/course/introduction-and-applications-of-data-mining/) statistics, predictive analysis. A vast field, data science uses a lot of theories and techniques that are a part of other fields like information science, mathematics, statics, [chemo metrics](https://en.wikipedia.org/wiki/Chemometrics) and Computer Science. Some of the methods used in data science includes probability models, [machine learning,](https://www.educba.com/course/machine-learning/) signal processing, data mining, statistical learning, [database,](https://www.educba.com/course/database-basics-2013/) data engineering, [visualization,](https://www.educba.com/10-best-data-visualization-tools/) pattern recognition and learning, uncertainty modelling, [computer programming](https://www.educba.com/fundamentals-of-computer-programming/) among others.

Data is everywhere, and the uses we are making out of it (science) are increasing and impacting society more and more. The Data Science Trends are largely a continuation of some of the biggest trends of the recent years including Big Data, Artificial Intelligence (AI), [Machine Learning](http://www.dataversity.net/2017-machine-learning-trends/) (ML), along with some newer technologies like [Block-chain,](http://www.dataversity.net/blockchain-can-used-secure-sensitive-data-storage/) Edge Computing, [Serverless Computing,](http://www.dataversity.net/tech-primer-serverless-computing-converging-faas/) [Digital Twins,](http://www.dataversity.net/modernizing-industries-iot-powered-digital-twin/) and others that employ various practices and techniques within the Data Science industry.

When there is a loss of employees in any organization, there are a lot of problems which are caused, starting from an empty position in the organization. Filling these positions is a lengthy process of interviewing candidates, training them and integrating them into teams. This makes retention of valuable talent essential to the smooth functioning of the organization. HR is constantly looking out for ways to predict which employee is unhappy with the current job in order to try and convince the employee to stay or to cushion the blow of loss of talent by looking for replacements. Accurate employee attrition prediction has tremendous monetary and productivity benefits for the organization and so machine learning is used to train classifier models using the dataset. The dataset is called IBM HR Analytics Employee Attrition & Performance, taken from Kaggle. The dataset consists of 35 variables such as Age, Daily Rate,

Hourly Rate, Job Satisfaction, Overtime and Monthly Income being some of the important factors that contribute to attrition. The target variable is Attrition which has two values - Yes and No. Supervised learning is a process of training the model to map the function between labelled input variables and target output variable. The target in this dataset is a binary variable and so classification is used to solve this problem. Classification is a supervised learning technique which uses a model to predict categorical values from the input data. Stacked Classifier combines multiple classifying algorithms which are trained on the training set and is used with a meta-classifier which predicts the output class using the output values from the Stack Classifier.

The data is used to train and ensemble Stacked Classifier consisting of three classifying algorithms: Support Vector Machine, Decision Tree Classifier and Adaptive Boosting with the meta classifier algorithm used being Logistic Regression for obtaining better accuracy. The dataset goes through cleaning, pre-processing, feature engineering, training and modeling to obtain an accuracy of 90% using Stacked Classifier.

## 1.2 Problem Statement

Evaluating the staff performance using the dataset consists of variables such as Age, Daily Rate, Hourly Rate, Job Satisfaction, Overtime and Monthly Income (being some of the important factors that contribute to attrition) for all the employees of an organisation to predict which employee is unhappy with the current job so that the HR can immediately start looking for their replacements.

## 1.3 Objectives

1. The main objectives of this study is to know the reasons-
2. To identify the rate at which attrition occurs
3. To know the reasons why attrition occurs
4. To identify the factors which make employees dissatisfy
5. To know the satisfactory level of employees towards their job and working conditions
6. To find the areas in the which the particular organisation of lagging behind



Figure.1.1. Reasons for Employee Attrition

## 1.4 Types of Employee Turnover

**1.4.1.** **Internal vs. external turnover**: Like recruitment, turnover can be classed as 'internal' or external. Internal turnover involves employees leaving their current position, and taking a new position with the same organization. Both positive (such as increased morale from the change of task and supervisor) and negative (such as project/relational disruption,) effects of internal turnover exist, and thus this form of turnover may be as important to monitor as its external counterpart.

**1.4.2.** **Skilled vs. unskilled employees**: Unskilled positions often have high turnover, and employees can generally be replaced without the organization or business incurring any loss of performance. The ease of replacing these employees provides little incentive to employers to offer generous employment contracts; conversely, contracts may strongly favour the employer and lead to increased turnover as employees seek, and eventually find, more favourable employment.

**1.4.3.** **Voluntary vs. Involuntary turnover**: Involuntary: In this case, the employee ceases to work for the company due to being laid off or terminated. It could be because the company is trying to cut costs, or the employee has violated company policy. Voluntary:

Voluntary turnover is when an employee terminates employment on their own accord. There are several possible causes like relocation of family, starting a family, taking care of an elderly relative.

## 1.5 Causes of Employee Turnover

There are several factors that cause high turnover within companies. This report will focus on voluntary turnover, because voluntary turnover is something that companies are more able to control.

Employees voluntarily quit for several reasons, specifically:

1. Pay is too low
2. Lack of benefits
3. Poor management
4. Less or no appreciation for work done
5. Less growth opportunities
6. Poor training

## 1.6 Organization of Report

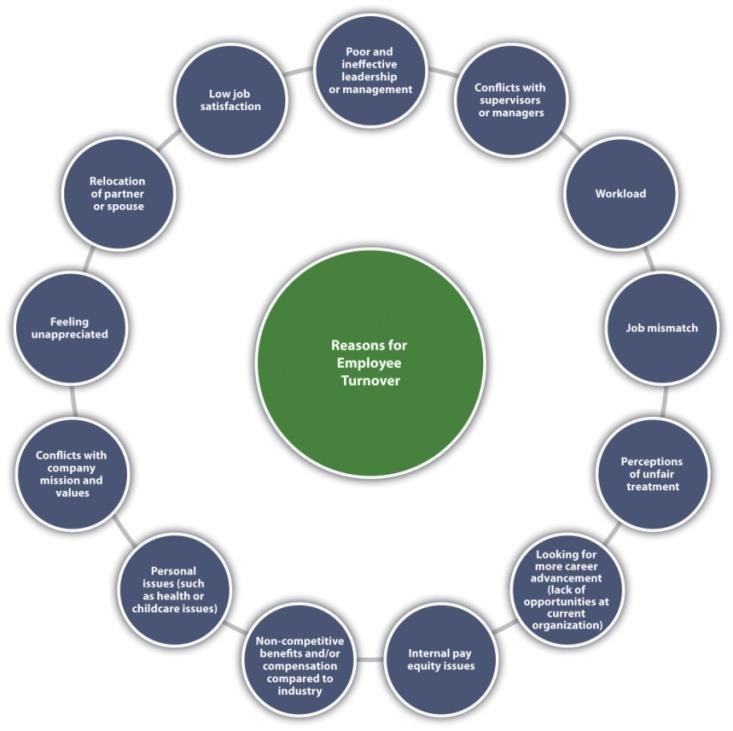
This rest of the report is laid out as follows. Chapter 2 Literature Survey presents an extensive review on the existing methods and technology applied along with the parameters involved. Chapter 3 System Analysis is an overview of the employee attrition prediction process and the proposed solution of our project along with all the technical specifications. A general framework of the proposed method is presented Chapter 4 i.e. System Design which is followed by conclusion, future scope and references section, also including appendix 1 and 2 at the end of the report.

# 2. LITERATURE SURVEY

## 2.1 Related research papers

### 2.1.1 Factors that lead to Employee Attrition

Uncover the factors that lead to employee attrition and explore important questions such as 'show me a breakdown of distance from home by job role and attrition' or 'compare average monthly income by education and attrition'. The various other factors are Salary hike, over-time, age, gender etc. This is a fictional data set created by IBM data scientists. [1]



### Figure.2.2.1. Factors that lead to Employee Attrition

#### 2.1.2 Classification algorithms

1. Logistic regression: It is a machine learning algorithm for classification. In this algorithm, the probabilities describing the possible outcomes of a single trial are modelled using a logistical function.
2. Naive Bayes: This algorithm is based on Bayes theorem with the assumption of independence between every pair of features. Naive Bayes classifiers work well in many real-world situations such as document classification and spam filtering.
3. Decision tree: It gives data of attributes together with its classes, a decision tree produces a sequence of rules that can be used to classify the data.
4. Random forest: Random forest classifier is a meta-estimator that fits a number of decision trees on various sub-samples of datasets and uses average to improve the predictive accuracy of the model and controls over-fitting. The sub-sample size is always the same as the original input sample size but the samples are drawn with replacement.[2]

#### 2.1.3 Stacking Classifier

Stacking is an ensemble learning technique to combine multiple classification models via a meta-classifier. The individual classification models are trained based on the complete training set; then, the meta-classifier is fitted based on the outputs-meta-features of the individual classification models in the ensemble. The meta-classifier can neither be trained on the predicted class labels or probabilities from the ensemble. Please note that this type of stacking is prone to over fitting due to information leakage. The related StackingCVClassifier.md does not derive the predictions for the 2nd-level classifier from the same dataset that was used for training the level1 classifiers and is recommended instead.

1. Simple staked classification
2. Using probabilities as meta-features
3. Stacked classification and Grid search
4. Stacking of classifier that operates on different feature subsets. [3]

#### 2.1.4 Ensemble methods

The goal of ensemble methods is to combine the prediction of several base estimators built with a given learning algorithm in order to improve generalizability/ robustness over a single estimator. Two families of ensemble methods re usually distinguished: one is the averaging method and the other is boosting method.

**2.1.4.1 Controlling the tree size**: The size of the regression tree base learners defines the level of variable interactions that can be captured by the gradient boosting model. In general, a tree of depth h can capture interactions of order h. There are two ways in which the size of the individual regression trees can be controlled.

**2.1.4.2 Regression**: It supports a number of different loss functions for regression, the default loss function for regression is at least squares. [4]

**2.1.4.3 Stacking with probability distributions**: Stacking with probability distributions and multi-response linear regression Ting and Witten (1999) stack base-level classifiers whose predictions are probability distributions (PDs) over the set of class values, rather than single class values. The meta-level attributes are thus the probabilities of each of the class values returned by each of the base level classifiers. The authors argue that this allows them to use not only the predictions, but also the confidence of the base-level classifiers. [5]

**2.1.4.4 Discussion of experimental results:** Most of the combining methods we consider are variants of stacking with MLR. Seewald(2002) presents empirical evidence that stacking with MLR (SMLR) performs worse onmulti-class datasets (as compared to two-class datasets). He cites the dimensionality ofthe meta-data as a probable cause, and argues that the reduction of this dimensionality by reducing the set of meta-level features helps (making SCMLR perform better).In our experiments, one way to increase the dimensionality of the meta-level data is to add more base-level classifiers. Note that the (relative) performance of SMLR decreases and that of SCMLR increases with the number of base-level classifiers. This is consistent withthe argument of Seewald about the dimensionality of the meta-data.

Another way to increase dimensionality is to add new meta-level attributes as in SMLRE. With a small number of base-level classifiers the effect of providing additional information about the certainty of predictions prevails. With a larger number of base-level classifiers, however, this effect is countered by the adverse effect of the increase of dimensionality of the meta-data. SMLRE thus provides only limited advantage over SMLR.

**2.1.4.5 Comparison between various works**: We compare our scheme with related work in terms of features, computation, and space requirement. We exclude the cost and space for those certificates verification in this comparison, as it may vary in different scenarios. In the earlier inventions of this project, the use of stacking classifier was not introduces whereas in this project the stacking classifier is employed here which performs the computation, by comparing all the algorithms and gives its result to meta classifier. The Meta Classifier gives the best algorithm depending on the high accuracy of such algorithm. Hence higher accuracy is achieved here.[6]

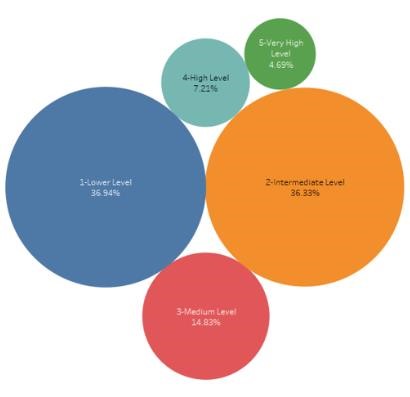


Figure.2.1.4.5 Attrition by job level

## 2.2 Technology Used

### 2.2.1 Stacking classifier

Stacking Classifier is an ensemble learning technique to combine multiple classification models via a meta-classifier. The StackingCV Classifier extends the standard stacking algorithm (implemented as Stacking Classifier) using cross-validation to prepare the input data for the level-2 classifier.

### 2.2.2 Naïve Bayes

Naïve Bayes Algorithm is a probability-based classification theorem which isbased on the Bayes Theorem. It can be used to predict the outcome of an occurring event withindependent conditions. The presence of one feature in a class will not affect the presence of any other feature but even if sometime the features depend on each other, these properties individually contribute to the probability hence called as Naive. In this algorithm, The probability of end result is encoded in the model along with the probability of the evidence variables occurring given that the end result occurs, As in Naïve Bayes is eager classifier and fast executing algorithm used for real time predictions and also has higher success rate compared to other algorithms.

### 2.2.3 Decision Tree

Decision Tree is a conventional algorithm used for performing classifications based on the decisions made in one stage. This algorithm works fast and the tree structure of the algorithm in easy understandable. Each node in the tree is the representation of an attribute which is being tested for making a decision, every branch is the representation of the output of that test, leaf nodes are the distributed sub classes. The part of the data set was trained to create a decision tree model and the trained model was used for prediction in the other part of the data set. The accuracy of the algorithm was established. Like all regression analyses, the logistic regression is a predictive analysis.

### 2.2.4 Logistic Regression

Logistic Regressionis used to describe data and to explain the relationship between one dependent binary variable and one or more nominal, ordinal, interval or ratio-level independent variables.

### 2.2.5 Random Forest

Random Foresttries to build multiple CART models with different samples and different initial variables. For instance, it will take a random sample of 100 observation and 5 randomly chosen initial variables to build a CART model. It will repeat the process (say) 10 times and then make a final prediction on each observation. Final prediction is a function of each prediction. This final prediction can simply be the mean of each prediction. The RF classification algorithm is used in two phases. First, the RF algorithm extracts subsamples from the original samples using the bootstrap re-sampling method and creates decision trees for each sample. Second, the algorithm classifies the decision trees and implements a simple vote, with the largest vote of the classification as the final result of the prediction.

The RF algorithm always includes two steps as follows:

1. Select the training set. Use the bootstrap random sampling method to retrieve K training sets from the original dataset (M properties), with the size of each training set the same as that of the original training set.
2. Build the RF model. Create a classification regression tree for each of the bootstrap training sets to produce K decision trees to form a “forest”; these trees are not pruned. Looking at the growth of each tree, this approach does not choose the best features as internal nodes for branches but rather the branching process is a random selection of all features.

### 2.2.6 Adaptive Boosting

The Adaptive Boosting (Ada Boost) algorithm works on the core principle of fitting a sequence of weak learners. It combines all the weak learners or any weak classifiers together with the increased weight to form a final classifier with the linear combination of weak classifiers. [7].

### 2.2.7 Classifier Evaluation Index

The common evaluation indices for the prediction model’s performance are accuracy (ACC), recall, precision (PPV), and the area under the curve (AUC). To calculate these indices, the confusion matrix is used. In the matrix, the columns represent the prediction categories and the sum of the value in the column is the data observations in the category. In addition, the rows in the matrix represent the actual categories and the sum of the values in the rows represents the data observations in that category. In this study, our focus is on whether or not there is employee turnover, which is a binary classification. Turnover is set as the positive category and no turnover set as the negative category. As shown in Table 1, TP denotes that the actual turnover is predicted as turnover; FN denotes that the actual turnover is predicted as no turnover; TN denotes that actual no turnover is predicted as no turnover and FP denotes that actual no turnover is predicted as turnover. Recall denotes the true positive rate (TPR) and the equation is

* 𝑅𝑒𝑐𝑎𝑙𝑙 = 𝑇𝑃𝑅 = 𝑇𝑃/(𝑇𝑃 + 𝐹𝑁) (1)

FPR denotes the false positive rate and the equation is

* 𝐹𝑃𝑅 = 𝐹𝑃/(𝐹𝑃 + 𝑇𝑁) (2)

Precision denotes the positive predictive value (PPV) and the equation is

* 𝑃𝑃𝑉 = 𝑇𝑃/(𝑇𝑃 + 𝐹𝑃) (3)

ACC denotes accuracy and the equation is

* 𝐴𝐶𝐶 = (𝑇𝑃 + 𝑇𝑁)/(𝑇𝑃 + 𝐹𝑃 + 𝐹𝑁+ 𝑇𝑁) (4)

In this study, we pay attention to the small classes (categorization features as turnover) in the problem of unbalanced classification. The main goal is to avoid misdiagnosis and minimize misdetection. Therefore, recall, the F-measure, the AUC, and the overall ACC will be used to evaluate the performance of the classification algorithm. [8]

### 2.2.8 Anaconda

It is a free and open source distribution of the Python and R programming languages for large-scale data processing, predictive analytics, and scientific computing. The distribution makes package management and deployment simple and easy. Matplotlib and lots of other useful (data) science tools form part of the distribution. [9]



### Figure.2.2.8 Anaconda Platform

#### 2.2.9 Jupyter Notebook

The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text. Uses include: data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much more.



Figure.2.2.9 Jupyter Notebook

# 3. SYSTEM ANALYSIS

## 3.1 Problem with Existing System

In any organization, managing Human Resources is an important task. Loss of employees lowers the overall productivity of the team and is also financially costly. Attrition of employees leaves behind a void that is costly to fill Machine Learning can be utilized for predicting an employee’s attrition**.**Adding unstructured, textual data into a conventional attrition identification. The outcome is raise performance in attrition identification analysis. This study supportive for marketing decision makers to improved recognize customer those have probability to attrition. Limitations: In the existing systems they used only few of data mining techniques for data prediction. Employee attrition effects in financial, time and effort loss for organizations. It is a big issue since a trained and experienced employee is difficult to substitute and it is cost effective.



Figure.3.1. Employee turnovers and its causes

## 3.2 Proposed Work

The proposed system has the following:

### 3.2.1 Data set

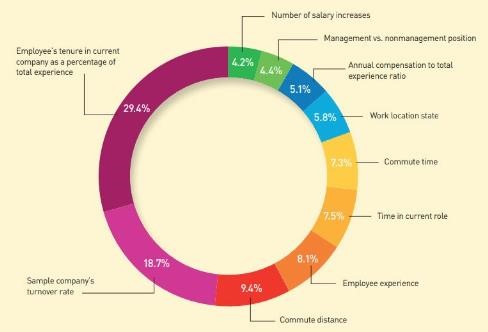
Data set is a collection of data. Most commonly a data set corresponds to the contents of a single database, where every column of the table represents a particular variable, and each row corresponds to a member of the dataset. For our project we take employee data from IBM which contains 1470 records and 35 fields including categorical and numeric features. Each record in the employee data set represents a single employee information and each field in the record represents a feature of that particular employee.

### 3.2.2 Data pre-processing

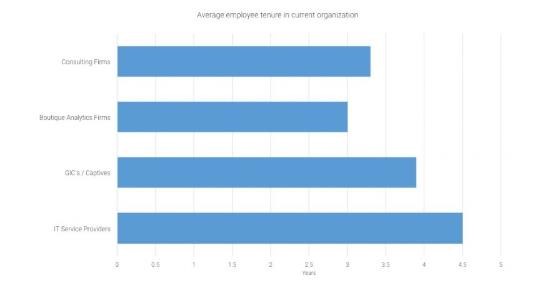
From the IBM employee dataset we implement a feature selection method to select the most important features of the dataset and divide total dataset into two sub datasets. One is test dataset another one is training dataset. That is if suppose any feature value in the record contain any null value or undefined or irrelevant value then separate that entire record from the original dataset and place that record into training dataset, else if the record contain perfect data with all features then place that into test dataset. Test dataset contain all important features to predict employee attrition or employee attrition and training dataset contain irrelevant data.

### 3.2.3 Test dataset and training dataset

Separating data into test datasets and training datasets is an important part of evaluating data mining models. By this separation of total data set into two data sets we can minimize the effects of data inconsistency and better understand the characteristics of the model. The test data set contains all the required data for data prediction and training data set contains all irrelevant data. Here we have 788 records in test dataset and 682 records in training dataset. We apply data classification and data prediction on the test dataset of 788 records.



### Figure.3.2.3.1. Revelations from Work-force turnover study

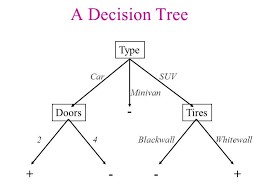


### Figure.3.2.3.2 Attrition performance evolution

**3.2.4 Data classification techniques**:

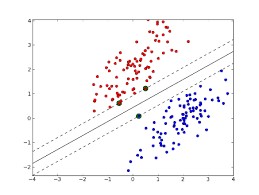
Data classification is the process of organizing data into categories for its most effective and efficient use. Data classification techniques are :

**3.2.4.1 Decision Tree:** It is tree structure that includes a root node, branches, and leaf nodes. Each internal node denotes a test on an attribute, each branched notes the outcome of a test, and each leaf node holds a class label.



### Figure 3.2.4.1 Working of Decision Tree

**3.2.4.2 Support Vector Machine:** In [machine learning,](https://en.wikipedia.org/wiki/Machine_learning) support-vector machines (SVMs also support-vector networks) aresupervised learnin[g](https://en.wikipedia.org/wiki/Supervised_learning) models with associated learning [algorithms](https://en.wikipedia.org/wiki/Algorithm) that analyse data used for [classification](https://en.wikipedia.org/wiki/Statistical_classification) and [regression analysis.](https://en.wikipedia.org/wiki/Regression_analysis) Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non[-probabilistic](https://en.wikipedia.org/wiki/Probabilistic_classification) [binary](https://en.wikipedia.org/wiki/Binary_classifier) [linear classifier.](https://en.wikipedia.org/wiki/Linear_classifier)

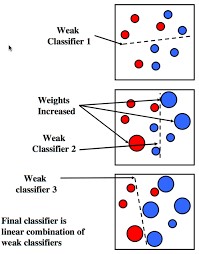


### Figure 3.2.4.2. Working of Support Vector Machine

**3.2.4.3 Adaptive Boosting:** The Adaptive Boosting (Ada Boost) algorithm works on the core principle of fitting a sequence of weak learners [7]. This algorithm uses a particular boost classifier as shown as Eq. 1. 𝐹(𝑥) = ∑ 𝑓𝑡(𝑥) 𝑇𝑡 = 1 represents a weak learner which takes an object x input and accordingly returns a value which indicates the class of that object.

𝐸𝑡 = ∑ [𝐹𝑡−1(𝑥𝑖) + 𝛼𝑡ℎ(𝑥𝑖)] 𝑖 Eq. (2). The sum training error Et is given as Eq. 2 and it is minimized as each weak learner produces ℎ(𝑥𝑖)the output hypothesis. For each iteration t, the

𝛼coefficient is assigned. 𝐹𝑡−1 is the boosted classifier. Eq. 1 and 2 are used for training. The weights 𝑤1 , 𝑤2 , .𝑤𝑁are applied to each of the training samples, this is known as boosting iteration. Initially all the weights are assigned by Eq. 3. 𝑤𝑖 = 1 / 𝑁 Eq. (3)



### Figure 3.2.4.3. Working of Adaptive Boosting

**3.2.4.4 Random Forest:** Random forest is a type of supervised machine learning algorithm based on [ensemble learning.](https://en.wikipedia.org/wiki/Ensemble_learning) Ensemble learning is a type of learning where you join different types of algorithms or same algorithm multiple times to form a more powerful prediction model. The [random forest](https://en.wikipedia.org/wiki/Random_forest) algorithm combines multiple algorithm of the same type i.e.

multiple decisiontrees, resulting in a forest of trees, hence the name "Random Forest".

The random forest algorithm can be used for both regression and classification tasks.

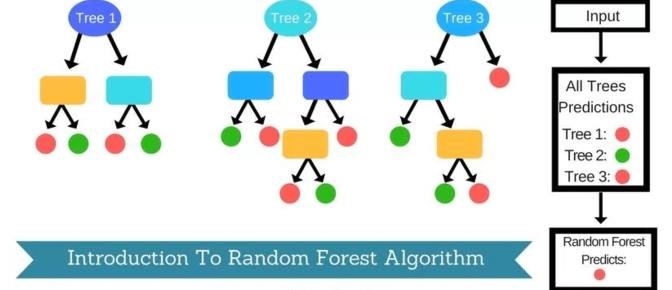


Figure 3.2.4.4. Working of Random Forest

## 3.3 Feasibility Study

Feasibility analysis is the process of confirming that a strategy, plan or design is possible and makes sense. This can be used to validate assumptions, constraints, decisions, approaches and business cases. For this project, we will be discussing the three major types of feasibility that are technical feasibility, operational feasibility and economic feasibility.

### 3.3.1 Technical Feasibility

A Technical feasibility study assesses the details of how you intend to deliver a product or service to customers. It is the process of validating the technology assumptions, architecture and design of a product or project. The proposed system consists of free-to-use software and it is platform independent. It can be used with different software versions by only doing minor changes to the code and has some different execution steps involved. This project is mainly used by many companies to reduce their financial costs by using retention schemes for the leaving employees.

### 3.3.2 Economical Feasibility

Economic feasibility is the cost and logistical outlook for a business project or endeavour. It is a kind of cost-benefit analysis of the examined project, which assesses whether it is possible to implement it or not. This project uses the Python libraries like Pandas, NumPy, SciPy, matplotlib, sklearn, etc using Anaconda and it also uses the Flask Environment. Python is a free, open-source programming language that is available for everyone to use. We have used the Jupyter notebook for the graphical representation of the algorithms and to find the accuracy of each individually. We then used Spyder notebook to implement the flask environment and the front end was executed through the web browser by running the flask on Anaconda PowerShell.

From this, we can say that the proposed system is economically feasible.

### 3.3.3 Operational Feasibility

Operational feasibility is a measure of how well a proposed system solves the problems. This system operates based on the various features of the employees and using the best algorithms accurately finds which of the employee will leave the organisation and which one will stay based on features like salary hike, distance from home, overtimes etc. This needs proper and valid data for the employee, with that kind of data it is highly feasible. Although a few precautions have to be kept in mind for each employee data record.

## 3.4 Software Requirement Specification

### 3.4.1 Introduction

#### 3.4.1.1 Purpose

The purpose of the Employee Attrition System is to evaluate the staff performance and determine which employee will be leaving the company based on the features given in the dataset used for a number of employees working in the company.

#### 3.4.1.2 Scope

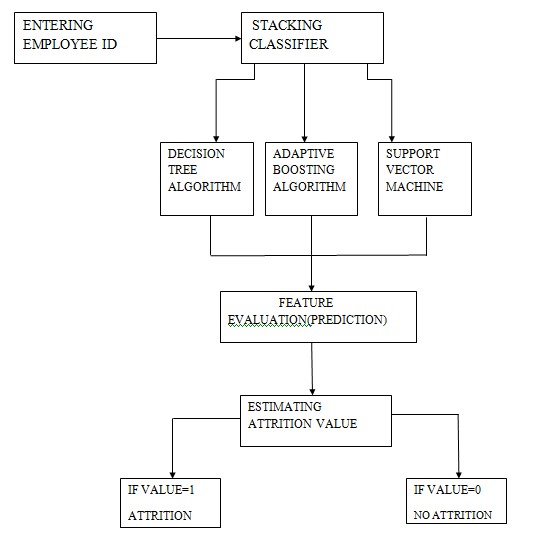
This system is used to evaluate and reduce the financial loss of the company by minimising the employee turnover rate. Hugh cost is imposed on the company due to increased cost of interviewing, hiring and training the new employees and these replacements can also lead to losing of excellent employees. Thus this system predicts the possibility of any attrition and the company can take measures before that and plan for any retention schemes to make the employee stay. This project uses data science and machine learning algorithms to predict with an accuracy of about 90% , whether there will attrition of the employees or not.

### 3.4.2 Overall Description

#### 3.4.2.1 Product Perspective

This product is mainly introduced for the companies and organisations to predict the employee attrition before it even happens to reduce the financial cost of the company. It uses the main features of all the employees stored in the record and these can be evaluated using the machine learning algorithms and the result can be estimated immediately on just entering the employee id. Through this we can stop any type of employee turnover by making necessary retention plans for employee retention.

Figure 3.4.2.1 shows the block diagram for the product and how the prediction is done is also shown briefly.



### Figure. 3.4.2.1 Block Diagram of the product

#### 3.4.2.2 Product Functions

The product’s goal is to predict employee attrition and alert the company to prepare retention plans for that particular employee or employees to avoid the turnover.

The functions of the products are as follows:

1. To take the input from the user to check about particular employee
2. Use the Stacking Classifier to combine the weak classifiers together
3. Using the algorithms perform feature evaluation i.e. cross-fold validation
4. Calculate an attrition value for the prediction which denotes attrition or no attrition
5. Defines low risk or high risk of the employee to leave the company

#### 3.4.2.3 User Classes and Characteristics

There are two user classes of this product as follows:

i. Manager (User): The person who is responsible to check the employee and stop him from leaving by making retention plans. This person has to keep a check on all the employees in the company and has to keep updating the records or the dataset. He should be the one operating the Employee Attrition Predictor to always note the status of the employees. ii. Employee: If the employee feels his needs are not satisfied by the company or the work place is too far from his home or any other problem, in that case he has an option of leaving the company. If the company doesn’t want to replace him, they need to make required arrangements for him so that it’ll be more convenient for him to work in the company. The company can make plans to increase his salary or provide any type of transportation facility for other employees to avoid their attrition due to this factor.

#### 3.4.2.4 Operating Environment

The operating environment for the system is as listed below.

1. Operating System: Windows 7 or above
2. Programming Language: Python
3. Programming Environment: Anaconda
4. Front End Environment: Flask
5. Dataset : IBM HR analytics employee attrition and performance

[WA\_Fn-UseC\_-HR-Employee-Attrition.csv]

#### 3.4.3 External Interface Requirements

##### 3.4.3.1 User Interface

This Interface for the system is used to enter the employee id on the output screen and the feature evaluation for that employee is done using the algorithms. The result of the prediction is displayed in the white space below and along with that a pie chart is also displayed to denote low risk or high risk for the employee to leave.



### Figure. 3.4.3.1 User Interface of the Attrition Predictor

#### 3.4.3.2 Software Interfaces

1. **Python** isinterpreted, high-level, general-purpose programming language. Python is [dynamically typed](https://en.wikipedia.org/wiki/Dynamic_programming_language) and [garbage-collected.](https://en.wikipedia.org/wiki/Garbage_collection_(computer_science)) It supports multiple [programming paradigms,](https://en.wikipedia.org/wiki/Programming_paradigms) including [structured](https://en.wikipedia.org/wiki/Structured_programming) (particularly, [procedural)](https://en.wikipedia.org/wiki/Procedural_programming), [object-oriented,](https://en.wikipedia.org/wiki/Object-oriented_programming)andfunctional programmin[g.](https://en.wikipedia.org/wiki/Functional_programming) Python is often described as a "batteries included" language due to its comprehensive [standard library.](https://en.wikipedia.org/wiki/Standard_library) It uses the following libraries:
2. **NumPy(**Numerical Python) is a perfect tool for scientific computing and performing basic and advanced array operations.
3. **Scipy** library includes modules for linear algebra, integration, optimization, and statistics. Its main functionality was built upon NumPy, so its arrays make use of this library. SciPy works great for all kinds of scientific programming projects (science, mathematics, and engineering). It offers efficient numerical routines such as numerical optimization, integration, and others in sub-modules.
4. **Pandas** is a library created to help developers work with "labelled" and "relational" data intuitively. It's based on two main data structures: "Series" (one-dimensional, like a list of items) and "Data Frames" (two-dimensional, like a table with multiple columns). Pandas allows converting data structures to Data Frame objects, handling missing data, and adding/deleting columns from Data Frame, imputing missing files, and plotting data with histogram or plot box. It’s a must-have for data wrangling, manipulation, and visualization.
5. **GridSearchCV** lets you combine an estimator with a grid search preamble to tune hyper-parameters. The method picks the optimal parameter from the grid search and uses it with the estimator selected by the user. GridSearchCV inherits the methods from the classifier.

## 3.5 COCOMO MODEL

Boehm proposed COCOMO (Constructive Cost Estimation Model) in 1981.COCOMO is one of the most generally used software estimation models in the world. COCOMO predicts the efforts and schedule of a software product based on the size of the software.

**The necessary steps in this model are:**

1. Get an initial estimate of the development effort from evaluation of thousands of lines of source code (KLOC).
2. Determine a set of 15 multiplying factors from various attributes of the project.
3. Calculate the effort estimate by multiplying the initial estimate with all the multiplying factors i.e., multiply the values in step1 and step2.

The initial estimate (also called nominal estimate) is determined by an equation of the form used in the static single variable models, using KLOC as the measure of the size. To determine the initial effort Ei in person-months the equation used is of the type is shown below

### Ei=a\*(KLOC)b

The value of the constant a and b are depends on the project type.

**In COCOMO, projects are categorized into three types:**

1. Organic
2. Semidetached
3. Embedded

ORGANIC: Relatively small, simple software projects in which small teams with good application experience work to a set of less than rigid requirements.

SEMI-DETACHED: An intermediate, (in size and complexity), a software project in which teams with mixed experience levels must meet a mix of rigid and less than rigid requirements.

EMBEDDED: A software project that must be developed within a set of tight hardware, software and operation constraints.

According to Boehm, software cost estimation should be done through three stages:

1. Basic Model
2. Intermediate Model
3. Detailed Model

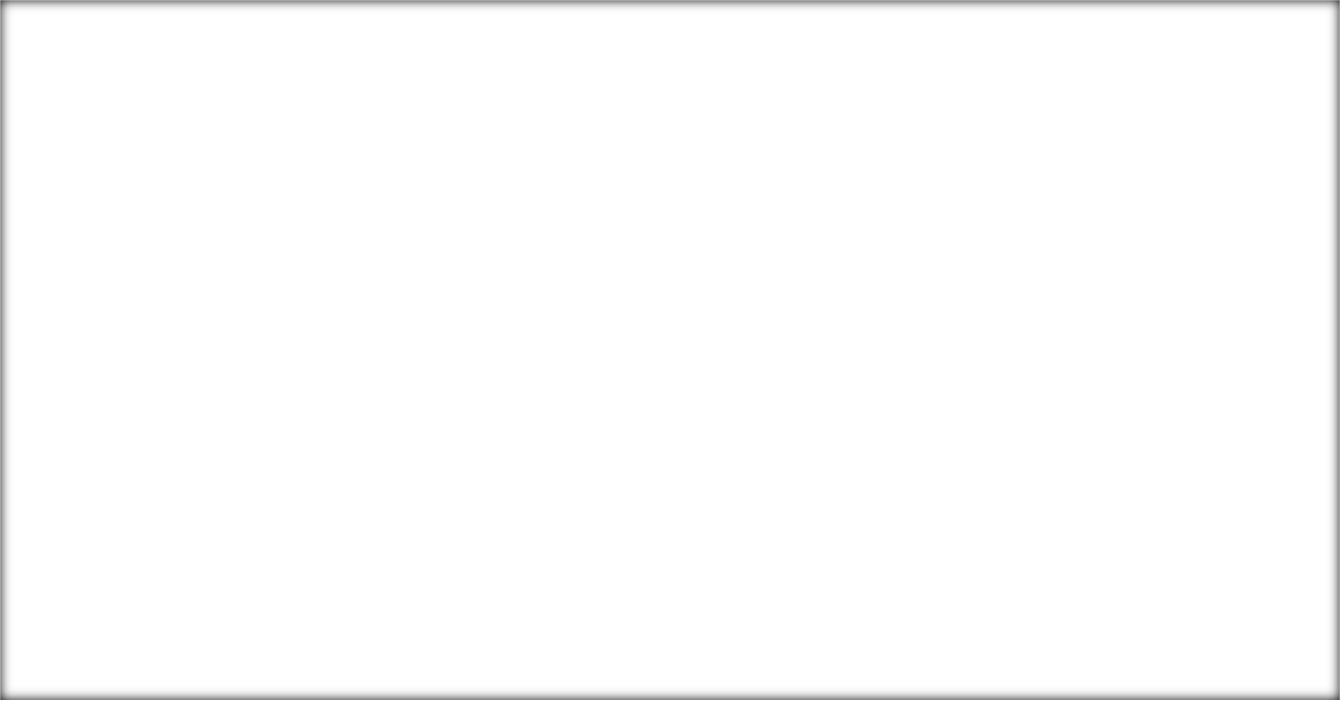
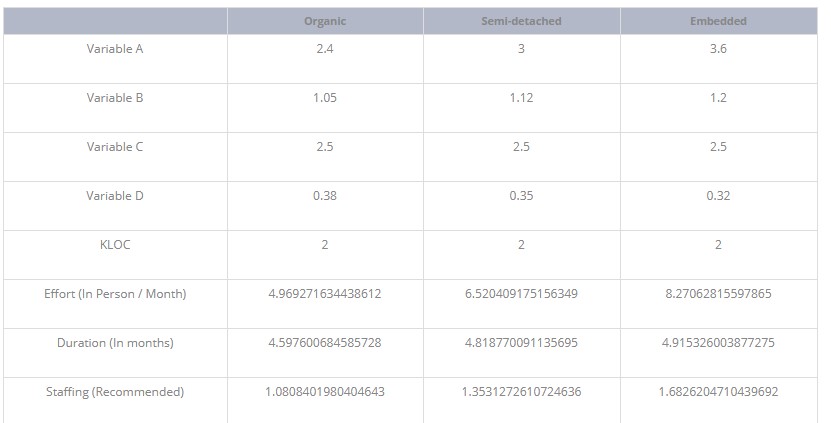
**Basic COCOMO Model:** The basic COCOMO model provides an accurate size of the project parameters. The following expressions give the basic COCOMO estimation model:

Effort = a\*KLOCb

Duration = c\*effortd Staffing = effort/duration

**THE ESTIMATION FOR OUR SYSTEM IS:**

**KLOC=2**



# 4. SYSTEM DESIGN

## 4.1 System Architecture

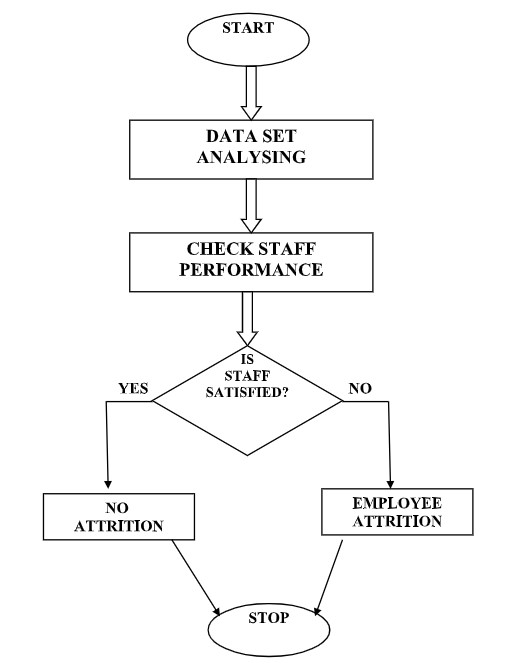


Figure.4.1. Flowchart for Employee Attrition

## 4.2 Division of data set

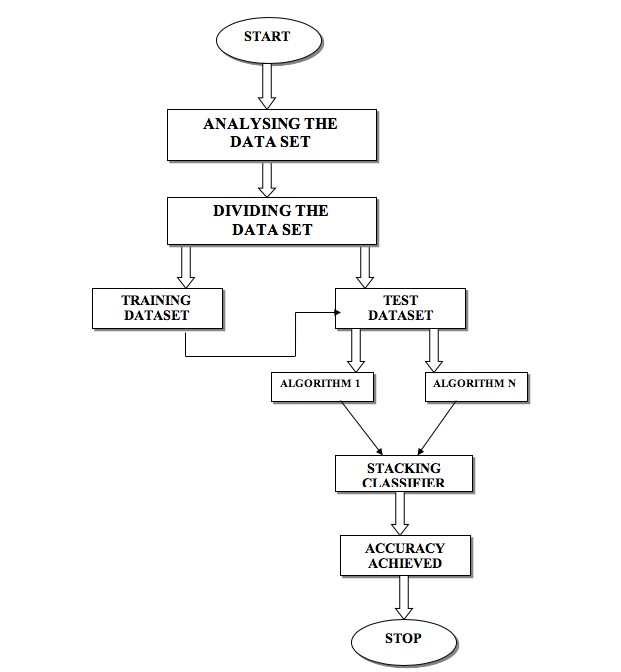


Figure.4.2. Flowchart for Data Set

## 4.3 Stacking and Meta Classifiers

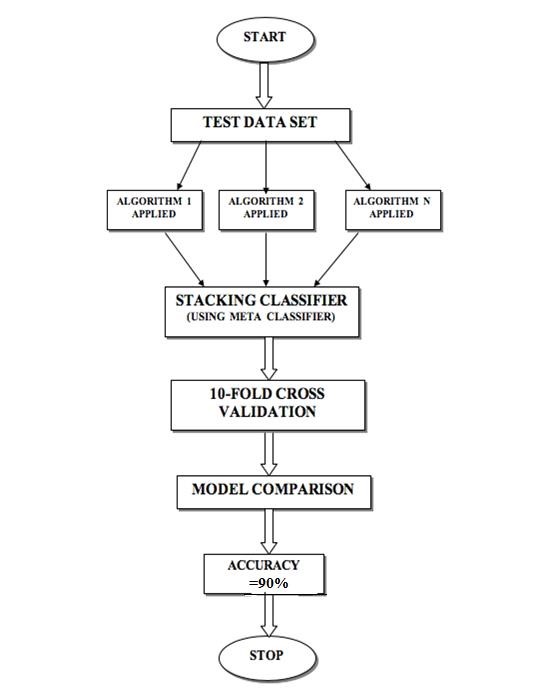


Figure.4.3. Flowchart for Stacking and Meta classifier

## 4.4 Use Case Diagram for Employee Attrition System

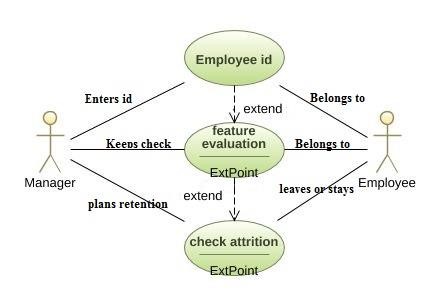


Figure.4.4. Use Case Diagram for Employee Attrition

## 4.5 Fish Bone Diagram for reasons of resignation

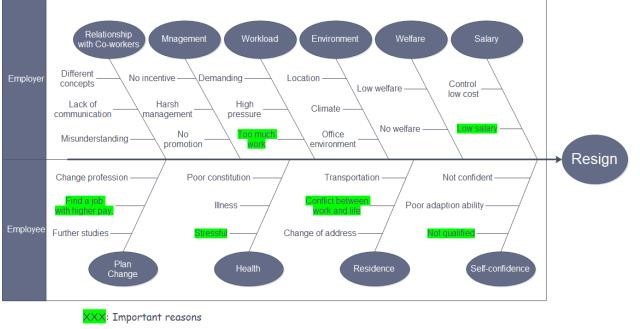


Figure.4.5. Reasons of resign using Fishbone Diagram

## 4.6 Activity Diagram to prevent Attrition

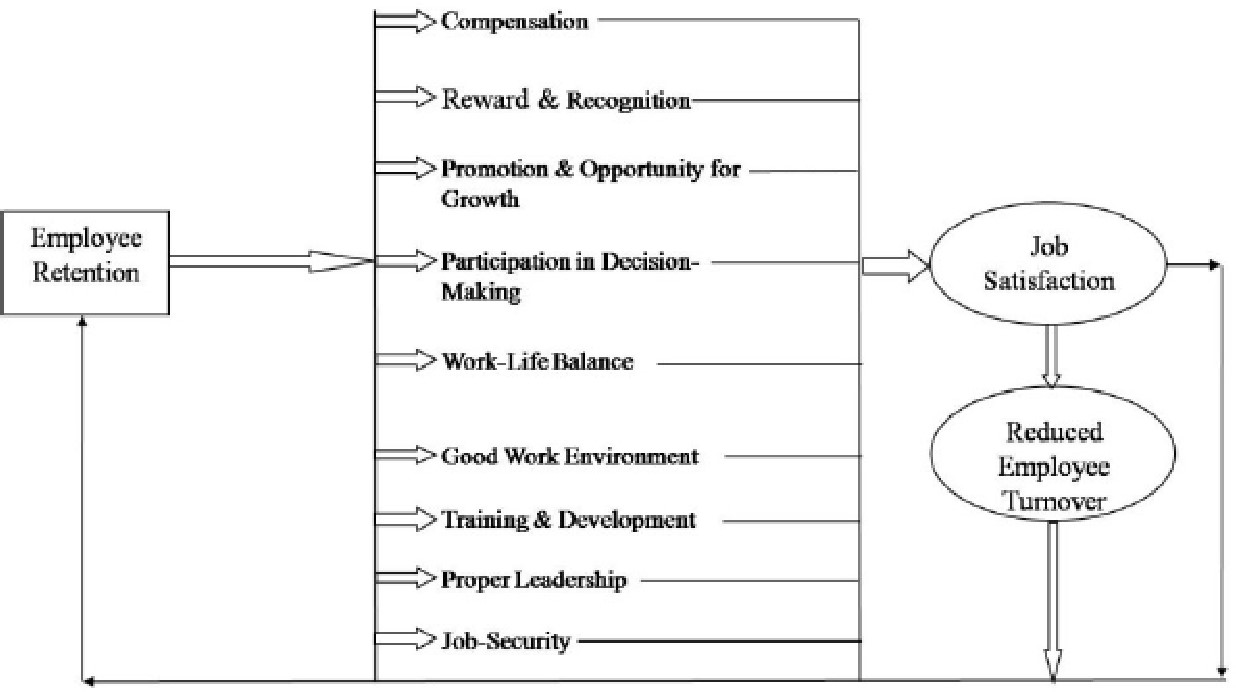
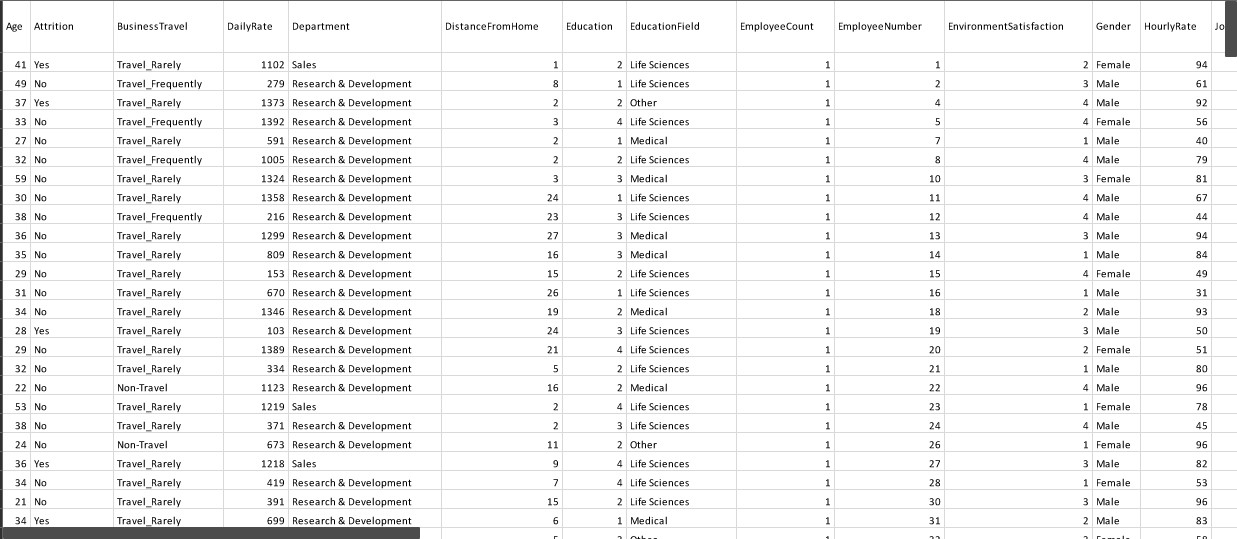


Figure.4.6. Activity Diagram to prevent Employee Attrition

# 5. IMPLEMENTATION

## 5.1 Data Set Used

IBM HR Analytics dataset containing data for 1,470 employees with 32 features



## .....Includes more 1,436 rows

### Table 5.1.1 Data Set used for predictions

**5.2 Front End Execution**

#### 5.2.1 Index.html

<style> .content { max-width: 700px; margin: auto; left-margin:20px; height:-260%; } .header { position:relatvie; top: 1px;

max-width: 100%; max-height: 150px;

right:1px; left:1px;

padding: 19px;

text-align: center; position: fixed; color: white; font-size: 47px; font-family: 'Arial'; background-color: black;

}

.id{ right:100px; padding: 40px; top:200px; font-size:40px; text-align: center; } .button { border: none; color: white; padding: 16px 16px;

text-align: center; text-decoration: none; display: inline-block; font-size: 16px; margin: 4px 4px; transition-duration: 0.5s; cursor: pointer;

}

.button { background-color: #34495E;

color: white; right:353px; font-size:14px; border: 2px solid #050b05; text-align: center; cursor: pointer; border-radius: 5px; padding: 5px; position:relative;

bottom:765px; left:270px

}

.button:hover { background-color: gray; color: white; border: 2px solid #050b05;

}

.button2 { left: 12px;

background-color: white;

color: black; font-size: 14px; border: 2px solid #050b05;

text-align: center; cursor: pointer; border-radius: 5px; position:relative; top:1px;

padding: 4px;

}

.button2:hover { background-color: #BDC3C7;

color: black;

}

#gr { left-margin:20px; right:5px; position:relative;

bottom:160px;

transition-duration: 0.5s;

} #text{ font-size:15px; padding:5px; position:relative; bottom:35px;

left:2px; text-align:left; }

#pie{ position:relative;

bottom:170px;

left:500px; }

</style>

<!DOCTYPE html>

<html>

<head>

<title>EAP Model as a Flask API</title>

<link rel="stylesheet" type="text/css" href="index.css">

<script src='https://cdnjs.cloudflare.com/ajax/libs/Chart.js/1.0.2/Chart.min.js'></script>

<link rel="stylesheet"

href="https://stackpath.bootstrapcdn.com/bootstrap/4.2.1/css/bootstrap.min.css" integrity="sha384-

GJzZqFGwb1QTTN6wy59ffF1BuGJpLSa9DkKMp0DgiMDm4iYMj70gZWKYbI706tWS" crossorigin="anonymous">

</head>

<body style="background-size:cover;background-image:url('/static/image3.jpg');">

<header>

<h1 class="header">Employee Attrition Predictor</h1>

</header><p class="id">

<form action="/predict" method="post"><br> Employee Id:

<input class="button2" type="text" name="Index" value=""> <br><br><br>

<textarea id="text" rows="4" cols="60">

{% if label1 %} {{ label1 }} {% endif %}

{% if label2 %} {{ label2 }} {% endif %}</textarea>

<p id="pie"> <canvas id="myChart">

</canvas> <script>

var pieData = [

{% for item, label, colors in set %}

{

value: {{item}}, label: "{{label}}", color : "{{colors}}"

},

{% endfor %}

];

// get bar chart canvas

var mychart = document.getElementById("myChart").getContext("2d");

steps = 10

max = {{max}} // draw bar chart

new Chart(mychart).Doughnut(pieData, {

options : {

label: "My First dataset",

circumference: Math.PI, rotation: 1.0 \* Math.PI, percentageInnerCutout: 10,

}});

</script>

</p>

<h2>{{ title }}</h2>

<p id="gr"> <canvas id="chart" width="600" height="400"></canvas>

<script> // bar chart data var barData = {

labels : [

{% for item in labelb %}

"{{ item }}",

{% endfor %}

],

datasets : [{

fillColor: "#34495E",

strokeColor: "black", pointColor: "rgba(151,187,205,1)", pointStrokeColor: "black", pointHighlightFill: "#fff", pointHighlightStroke: "rgba(151,187,205,1)",

bezierCurve : false,

data : [

{% for item in valueb %}

{{ item }},

{% endfor %}]

}

]

}

Chart.defaults.global.animationSteps = 16;

Chart.defaults.global.tooltipYPadding = 1;

Chart.defaults.global.tooltipCornerRadius = 0;

Chart.defaults.global.tooltipTitleFontStyle = "normal";

Chart.defaults.global.tooltipFillColor = "steelblue";

Chart.defaults.global.animationEasing = "easeOutBounce";

Chart.defaults.global.responsive = false;

Chart.defaults.global.scaleLineColor = "black";

Chart.defaults.global.scaleFontSize = 10;

// get bar chart canvas var mychart = document.getElementById("chart").getContext("2d");

steps = 20

max = {{ max }} // draw bar chart

var LineChartDemo = new Chart(mychart).Line(barData, {

scaleOverride: true, scaleSteps: steps, scaleStepWidth: 0.3, scaleStartValue: 0, scaleShowVerticalLines: true, scaleShowGridLines : true, barShowStroke : true, scaleShowLabels: true, bezierCurve: false,

});

</script></p>

<input class="button" type="submit" value="Predict">

</form>

</p>

</body>

</html>

**Code Description**: The following code gives us the homepage for the Employee Attrition Predictor along with the pie charts and the bar graphs after showing the result.

**5.2.2 Test.py** import flask from flask import Flask, request, render\_template from sklearn.externals import joblib import numpy as np from scipy import misc import pandas as pd import pickle import csv

from sklearn.externals import joblib app = Flask(\_\_name\_\_)

@app.route("/") @app.route("/index") def index():

return flask.render\_template('index.html')

@app.route('/predict', methods=['POST']) def make\_prediction(): if request.method=='POST': sample=[] x=request.form.get('Index')

x=int(x)

df\_fl = pd.read\_csv(r"C:\Users\HOME\Desktop\employee\WA\_Fn-UseC\_-HREmployee-Attrition.csv") max=df\_fl['Index'].max() max=int(max) if(x>max):

return render\_template('index.html', label1="No Employee with this Id ")

else:

samples=(df\_fl.iloc[x]) samples.pop('Index')

sampleDf= pd.DataFrame(samples) model = joblib.load('model.pk') prediction = model.predict(sampleDf.T) confidence = model.predict\_proba(sampleDf.T)

if(prediction==1):

label2 = "Confidence Level : "+str(round(confidence[0][1],2))

else:

label2 = "Confidence Level : "+str(round(confidence[0][0],2))

if(prediction[0]==1):

return render\_template('index.html',label1="This Employee is at a high risk of leaving the Company", label2=label2)

else:

return render\_template('index.html', label1="This Employee is not at risk of leaving the Company", label2=label2)

**Code Description**: The following code helps to calculate the prediction value and confidence level on the Flask Environment and displays the result on the output screen.

### 5.3 Back end Execution

#### 5.3.1 EAF.py

#Importing libraries import pandas as pd from pandas.plotting import scatter\_matrix from pandas import ExcelWriter from pandas import ExcelFile from openpyxl import load\_workbook import numpy as np from scipy.stats import norm, skew from scipy import stats import statsmodels.api as sm import warnings

warnings.filterwarnings('ignore')

# importing libraries for data visualisations import seaborn as sns from matplotlib import pyplot import matplotlib.pyplot as plt import matplotlib.pylab as pylab import matplotlib %matplotlib inline

color = sns.color\_palette() from IPython.display import display pd.options.display.max\_columns = None

# Standard plotly imports # from plotly import plotly as py import chart\_studio.plotly as py import plotly as py import plotly.figure\_factory as ff import plotly.graph\_objs as go from plotly.offline import download\_plotlyjs, init\_notebook\_mode, plot, iplot init\_notebook\_mode(connected=True)

#py.initnotebookmode(connected=True) # this code, allow us to work with offline plotly version

# Using plotly + cufflinks in offline mode import cufflinks as cf cf.set\_config\_file(offline=True) import cufflinks cufflinks.go\_offline(connected=True)

import seaborn as sns from matplotlib import pyplot import matplotlib.pyplot as plt import matplotlib.pylab as pylab import matplotlib %matplotlib inline color = sns.color\_palette() from IPython.display import display pd.options.display.max\_columns = None

#Standard plotly imports #from plotly import plotly as py import plotly as py import plotly.figure\_factory as ff import plotly.graph\_objs as go #from plotly.offline import download\_plotlyjs, init\_notebook\_mode, plot, iplot init\_notebook\_mode(connected=True)

#py.initnotebookmode(connected=True) # this code, allow us to work with offline plotly version

#|Using plotly + cufflinks in offline mode import chart\_studio.plotly as py import plotly.graph\_objects as go import cufflinks as cf cf.set\_config\_file(offline=True) import cufflinks cufflinks.go\_offline(connected=True)

from sklearn.preprocessing import OneHotEncoder, LabelEncoder

# from imblearn.over\_sampling import SMOTE # SMOTE

# sklearn modules for ML model selection from sklearn.model\_selection import train\_test\_split # import 'train\_test\_split' from sklearn.model\_selection import GridSearchCV from sklearn.model\_selection import RandomizedSearchCV from sklearn.model\_selection import ShuffleSplit from sklearn.model\_selection import KFold from sklearn.model\_selection import cross\_val\_score

# Libraries for data modelling

from sklearn import svm, tree, linear\_model, neighbors from sklearn import naive\_bayes, ensemble, discriminant\_analysis, gaussian\_process from sklearn.linear\_model import LogisticRegression from sklearn.tree import DecisionTreeClassifier from sklearn.neighbors import KNeighborsClassifier

from sklearn.discriminant\_analysis import LinearDiscriminantAnalysis from sklearn.naive\_bayes import GaussianNB from sklearn.svm import SVC from sklearn.neighbors import KNeighborsClassifier from xgboost import XGBClassifier

from sklearn.ensemble import RandomForestClassifier

# Common sklearn Model Helpers from sklearn import feature\_selection from sklearn import model\_selection from sklearn import metrics

# from sklearn.datasets import make\_classification # sklearn modules for performance metrics from sklearn.metrics import confusion\_matrix, classification\_report, precision\_recall\_curve from sklearn.metrics import auc, roc\_auc\_score, roc\_curve, recall\_score, log\_loss from sklearn.metrics import f1\_score, accuracy\_score, roc\_auc\_score, make\_scorer from sklearn.metrics import average\_precision\_score

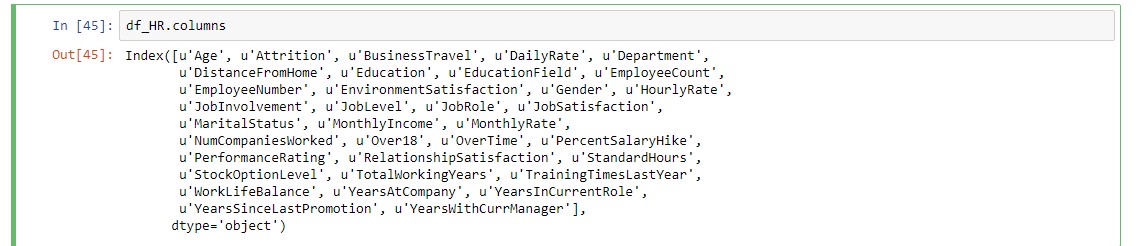
# importing misceallenous libraries import os import re import sys import timeit import string from datetime import datetime from time import time from dateutil.parser import parse

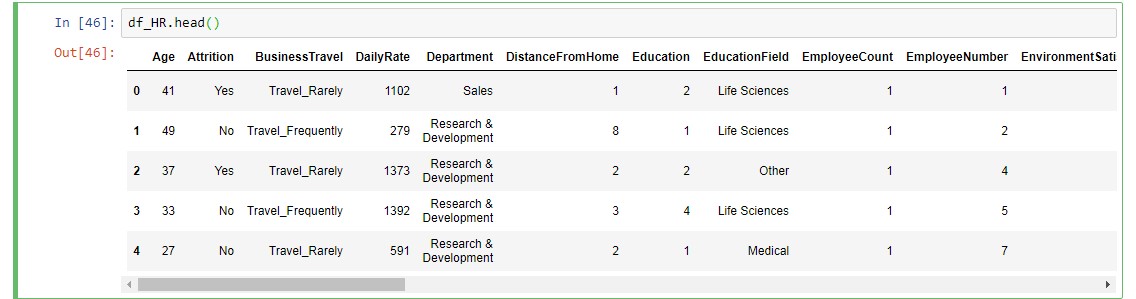
# ip = get\_ipython()

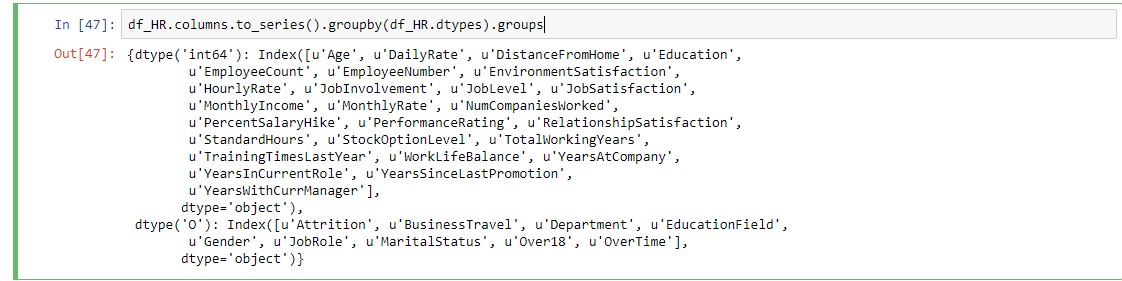
# ip.register\_magics(jupyternotify.JupyterNotifyMagics)



# Make a copy of the original sourcefile df\_HR = df\_sourcefile.copy()







# Columns datatypes and missing values df\_HR.info()

out[47]: <class 'pandas.core.frame.DataFrame'>

RangeIndex: 1470 entries, 0 to 1469 Data columns (total 35 columns):

Age 1470 non-null int64

Attrition 1470 non-null object

BusinessTravel 1470 non-null object

DailyRate 1470 non-null int64

Department 1470 non-null object

DistanceFromHome 1470 non-null int64

Education 1470 non-null int64

EducationField 1470 non-null object

EmployeeCount 1470 non-null int64

EmployeeNumber 1470 non-null int64

EnvironmentSatisfaction 1470 non-null int64

Gender 1470 non-null object

HourlyRate 1470 non-null int64

JobInvolvement 1470 non-null int64

JobLevel 1470 non-null int64

JobRole 1470 non-null object

JobSatisfaction 1470 non-null int64

MaritalStatus 1470 non-null object

MonthlyIncome 1470 non-null int64

MonthlyRate 1470 non-null int64

NumCompaniesWorked 1470 non-null int64

Over18 1470 non-null object

OverTime 1470 non-null object

PercentSalaryHike 1470 non-null int64

PerformanceRating 1470 non-null int64

RelationshipSatisfaction 1470 non-null int64

StandardHours 1470 non-null int64

StockOptionLevel 1470 non-null int64

TotalWorkingYears 1470 non-null int64

TrainingTimesLastYear 1470 non-null int64

WorkLifeBalance 1470 non-null int64

YearsAtCompany 1470 non-null int64

YearsInCurrentRole 1470 non-null int64

YearsSinceLastPromotion 1470 non-null int64 YearsWithCurrManager 1470 non-null int64 dtypes: int64(26), object(9) memory usage: 350.3+ KB

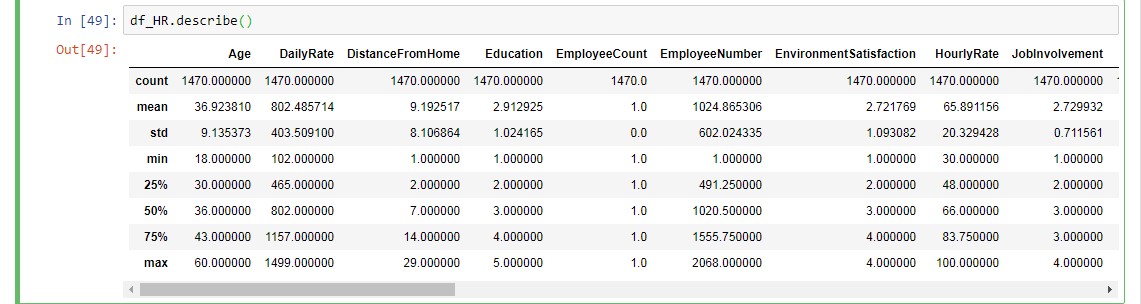
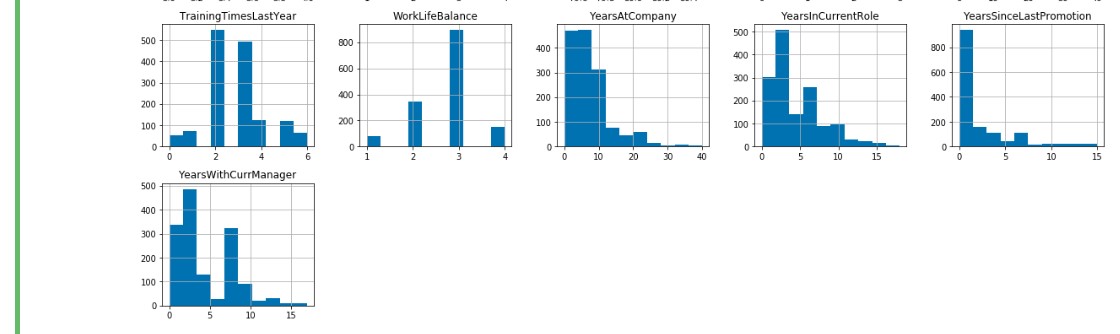
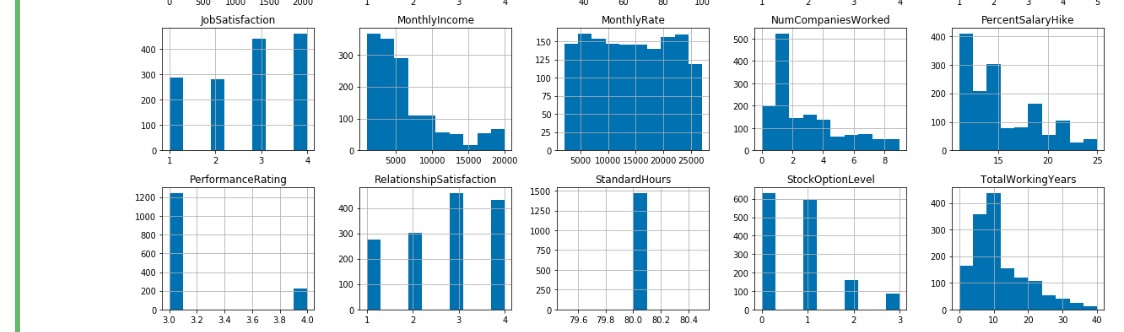
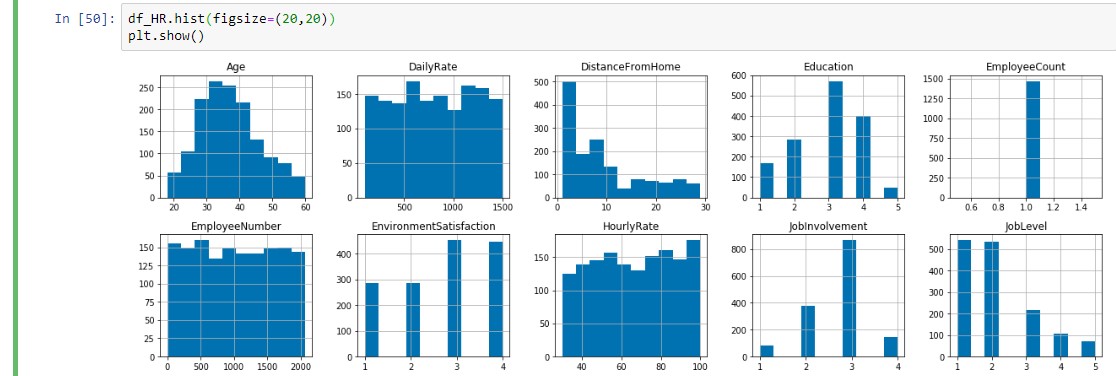
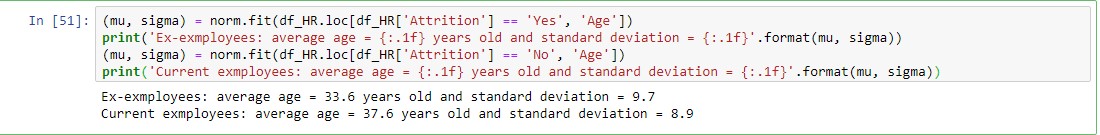
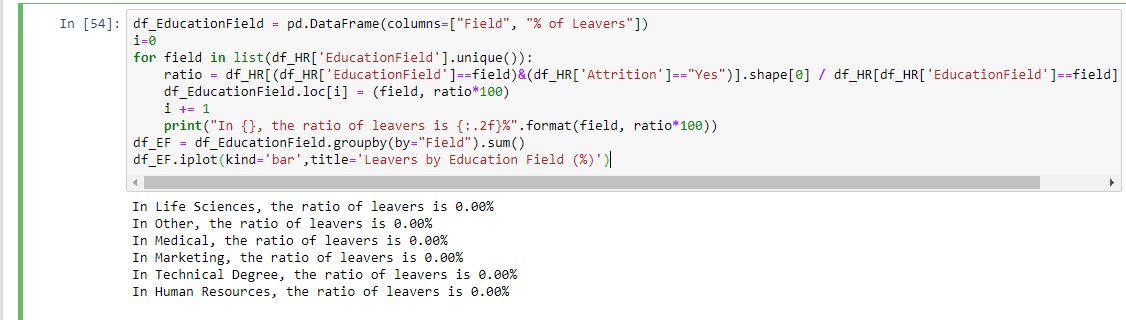
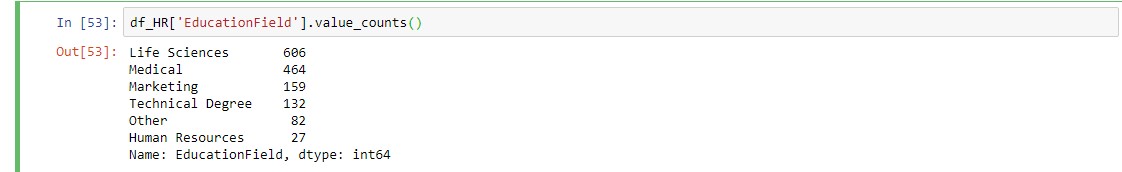
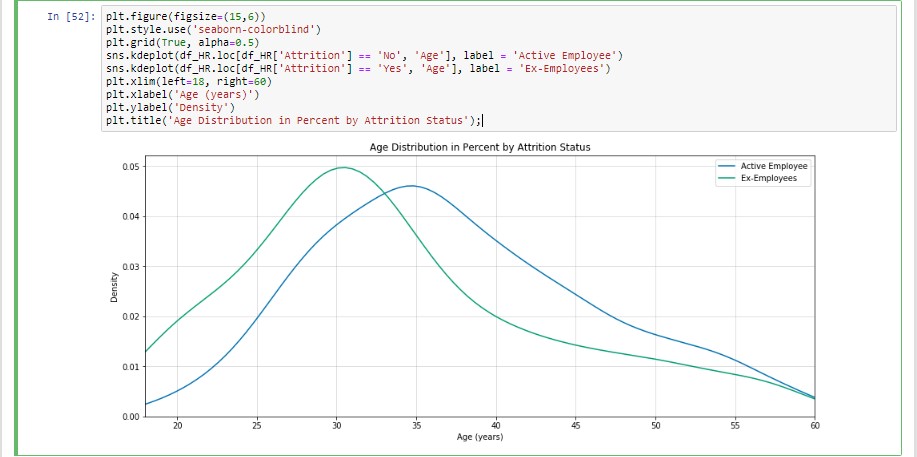
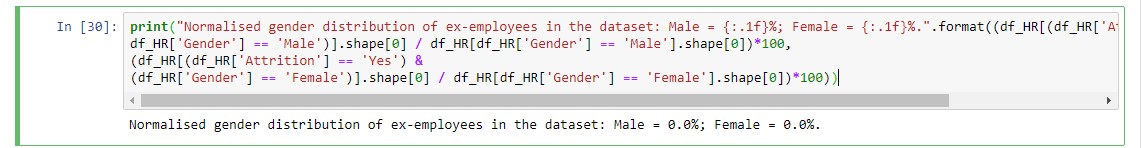


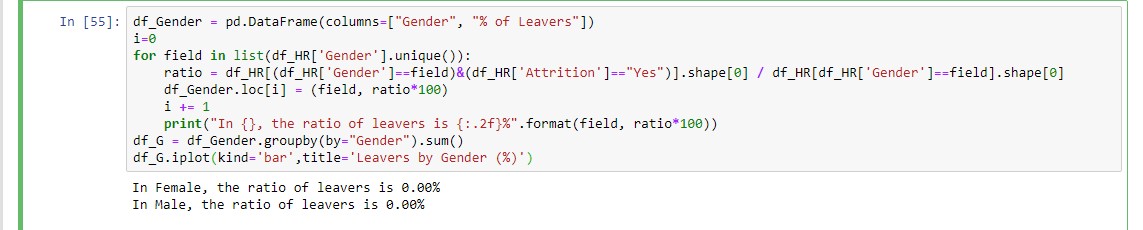
Figure. 5.3.1 Converting all data values into float

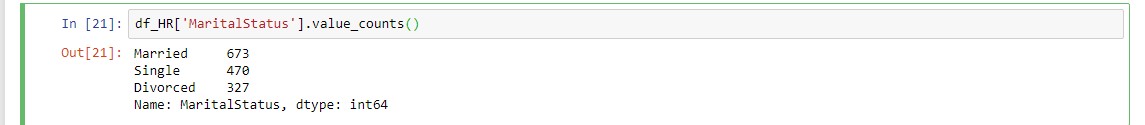


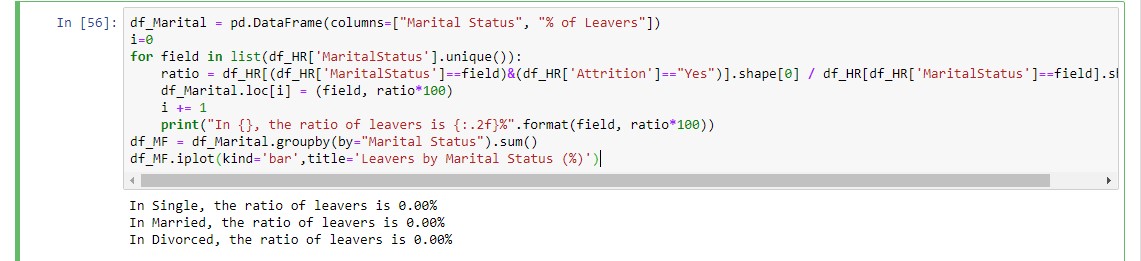


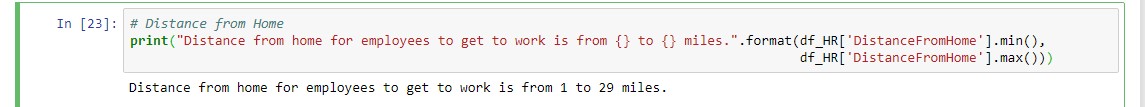


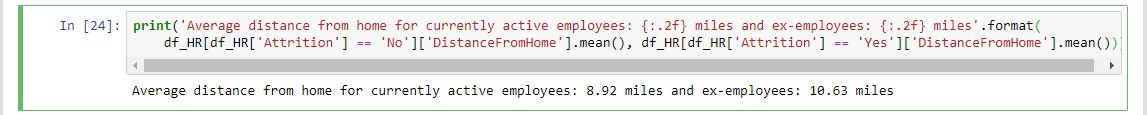


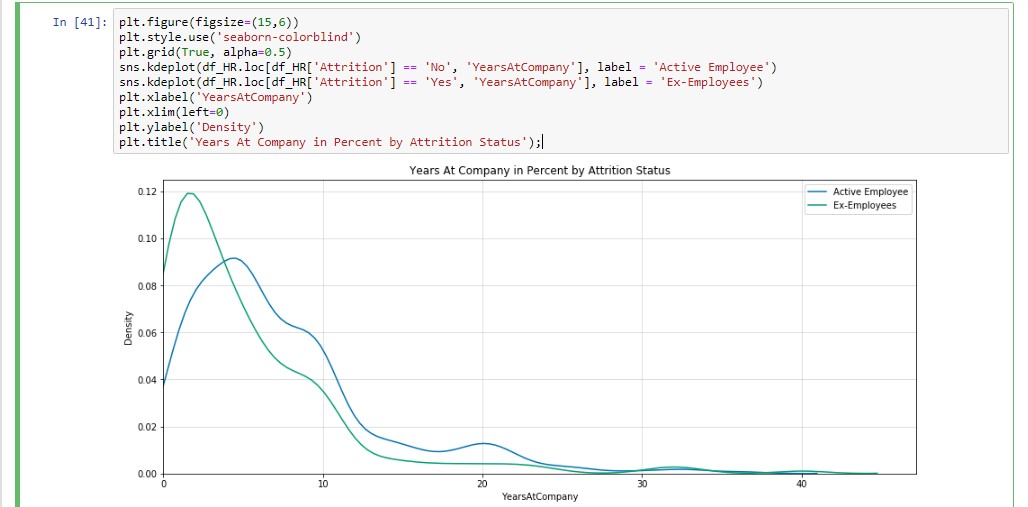




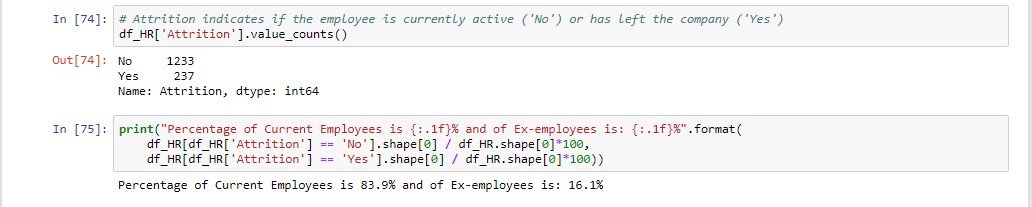


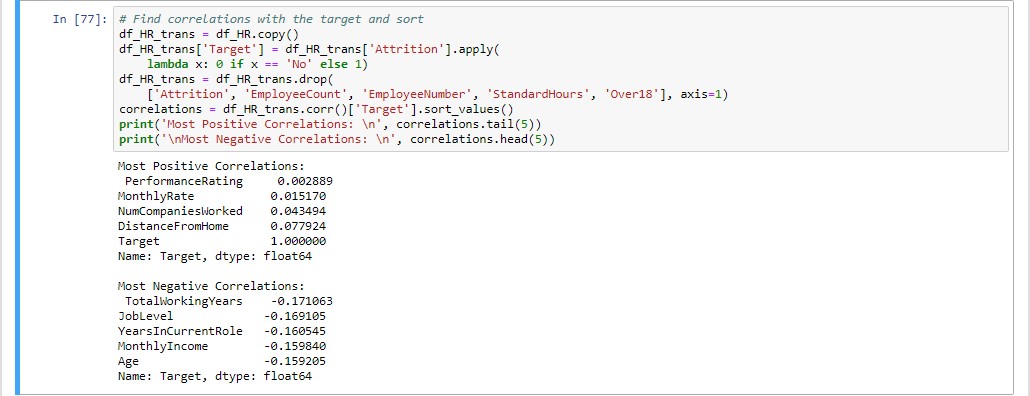






#the same is done for all other features



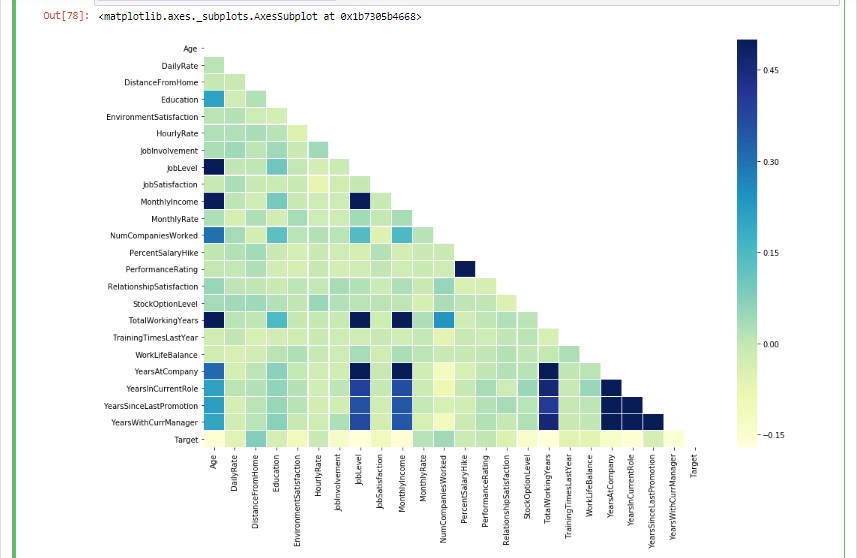


# Calculate correlations corr = df\_HR\_trans.corr() mask = np.zeros\_like(corr) mask[np.triu\_indices\_from(mask)] = True

# Heatmap

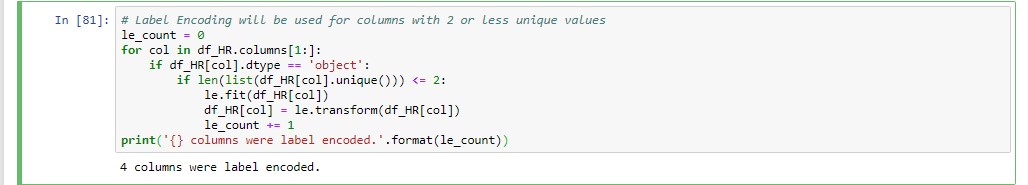
plt.figure(figsize=(15, 10)) sns.heatmap(corr, vmax=.5, mask=mask,

# annot=True, fmt='.2f', linewidths=.2, cmap="YlGnBu")

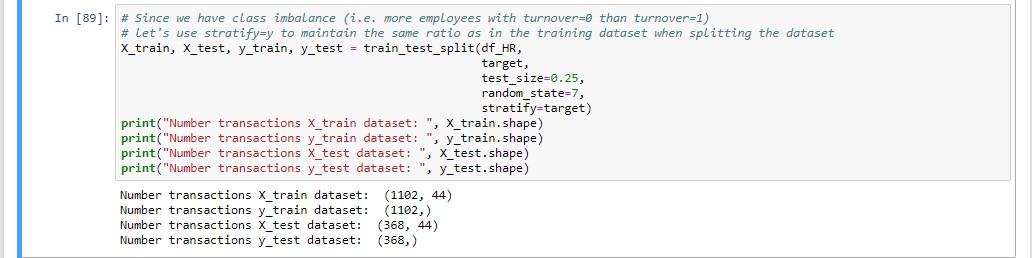
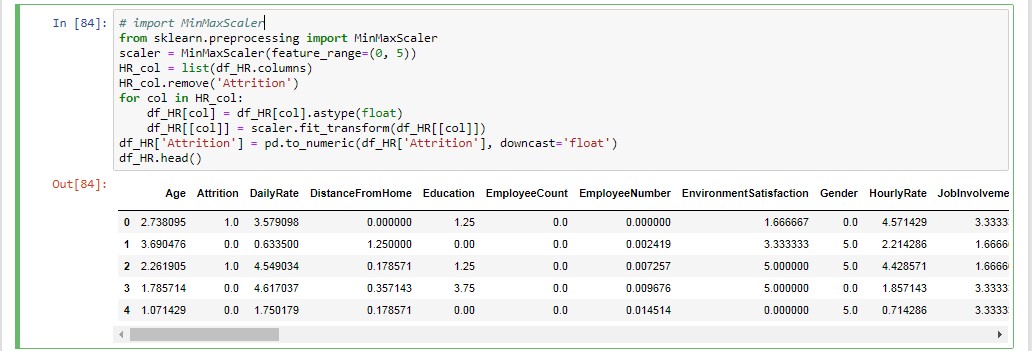


#Lable Encoding and creating a label encoder object

from sklearn.preprocessing import LabelEncoder, OneHotEncoder le = LabelEncoder()



df\_HR = pd.get\_dummies(df\_HR, drop\_first=True)

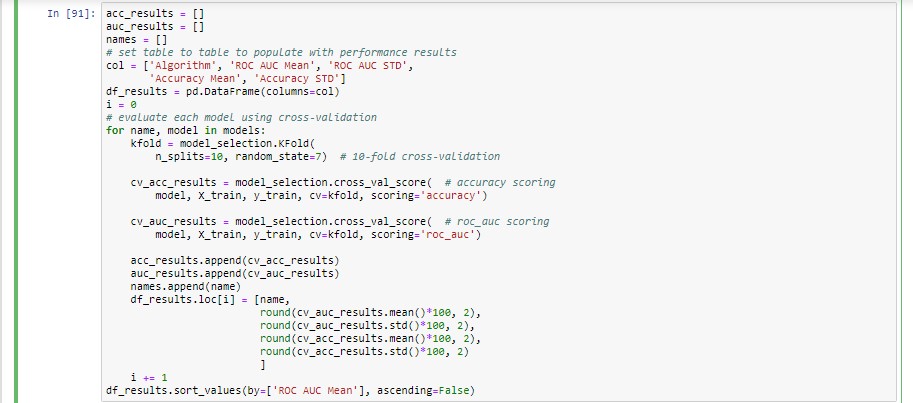


# selection of algorithms to consider and set performance measure models = []

models.append(('Logistic Regression', LogisticRegression(solver='liblinear', random\_state=7, class\_weight='balanced')))

models.append(('Random Forest', RandomForestClassifier( n\_estimators=100, random\_state=7))) models.append(('SVM', SVC(gamma='auto', random\_state=7))) models.append(('KNN', KNeighborsClassifier())) models.append(('Decision Tree Classifier',

DecisionTreeClassifier(random\_state=7))) models.append(('Gaussian NB', GaussianNB()))



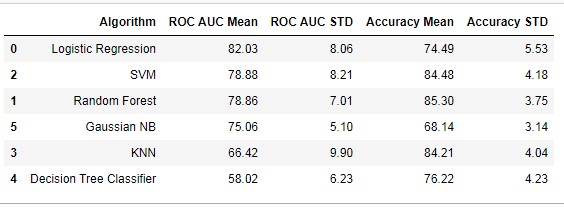
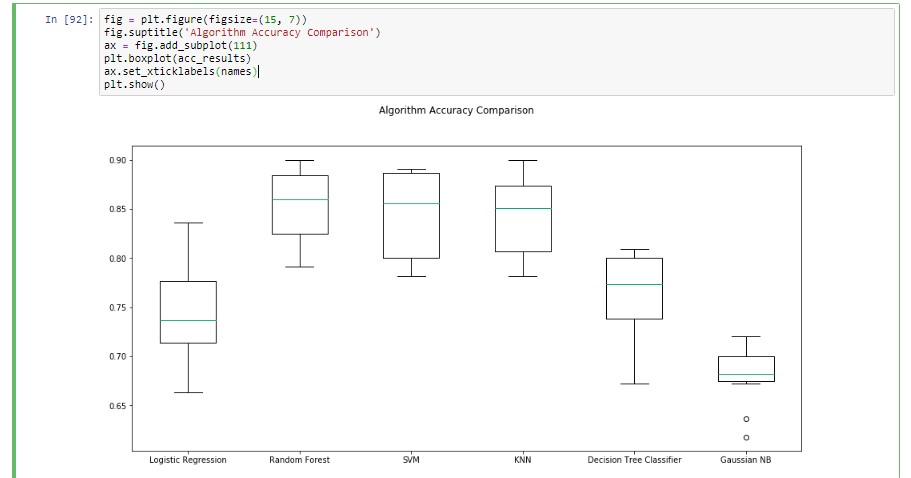
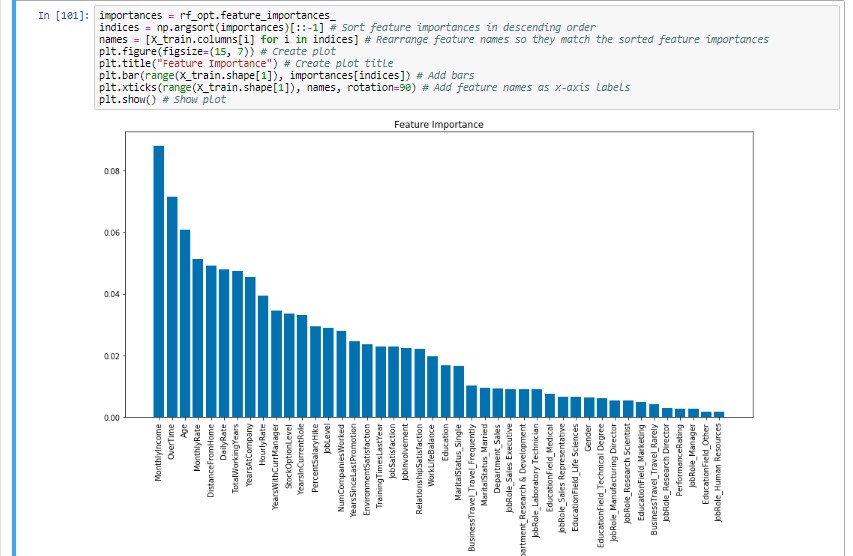
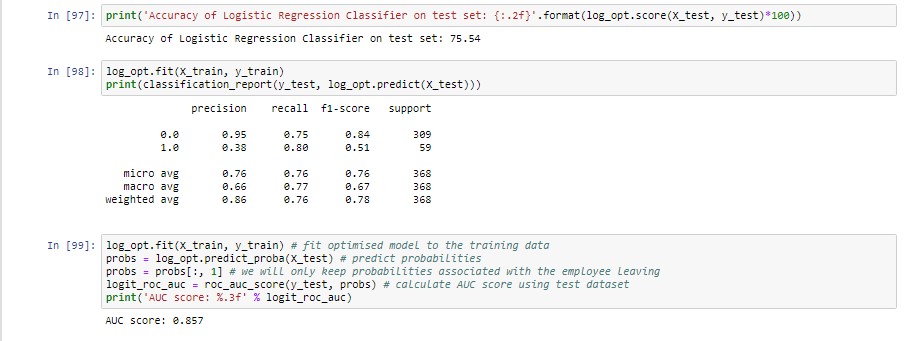
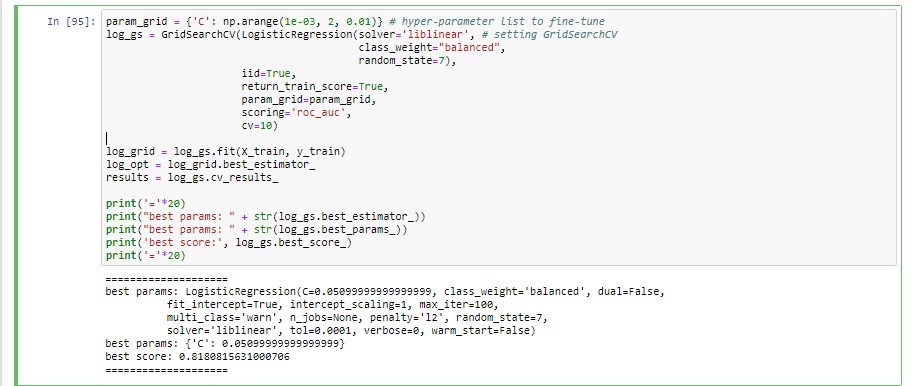
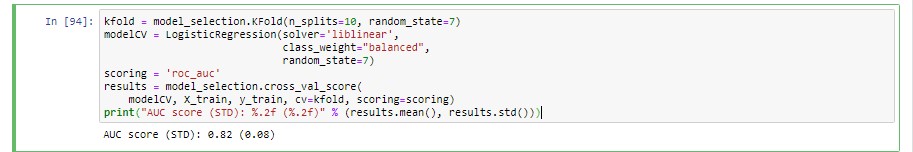
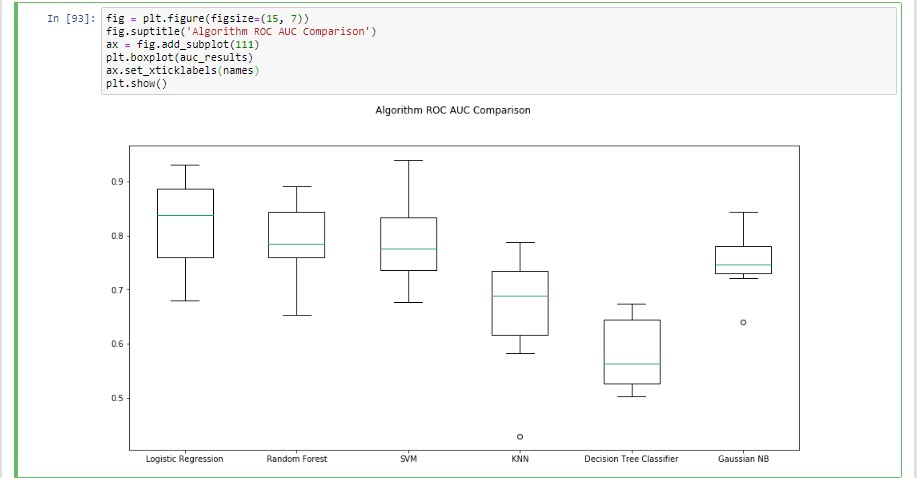


Table 5.3.2 Comparison between Algorithms based on ROC and AUC values





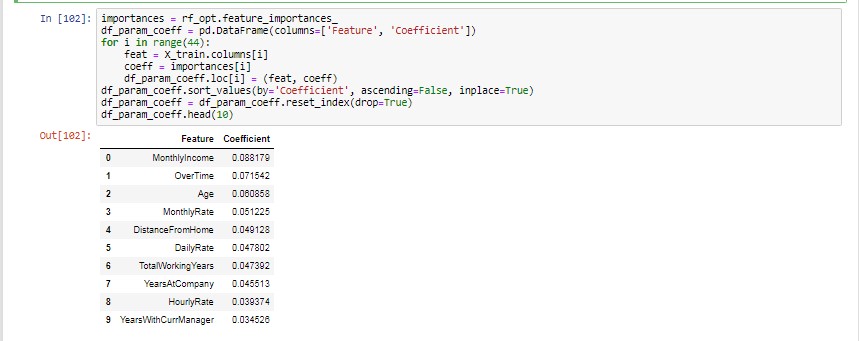
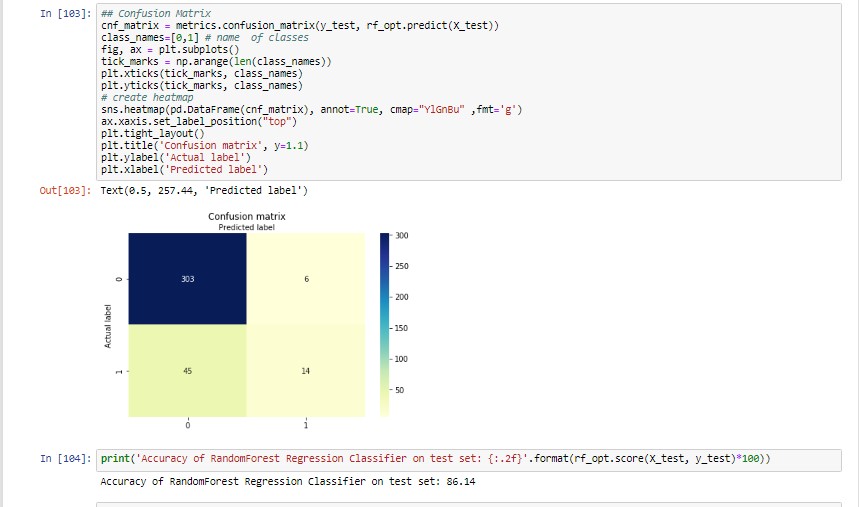


Table 5.3.3 Estimating the Feature Coefficients

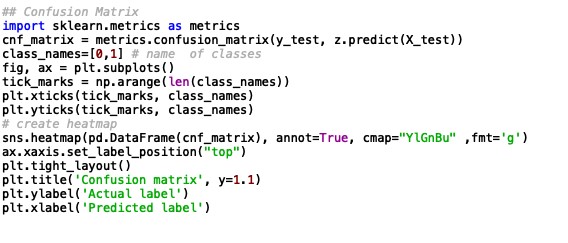


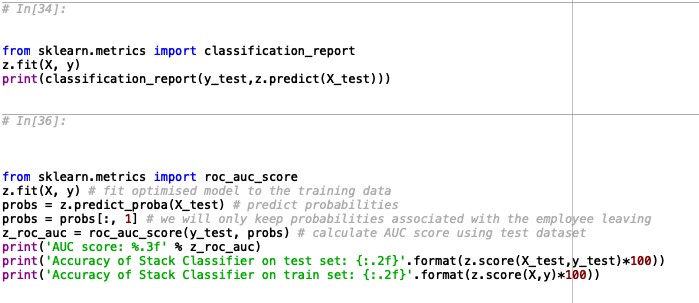
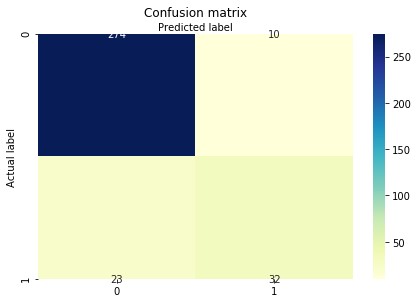
**Code Description:** The following code is executed on the Jupyter notebook or Spyder notebook. This code finds the accuracy of various algorithms individually and helps us to choose which of the algorithms can be used stacked together to get a better accuracy.

#### 5.3.2 Stackupdated.py



**Code Description:** The following code is used to stack the weak classifiers together using the linear combination to get a final classifier which is a strong classifier. Here we are using Adaptive Boosting and Support Vector Machine are the weak classifiers and combining with a meta classifier that is Decision Tree and thus the result for the feature score is 0.88.

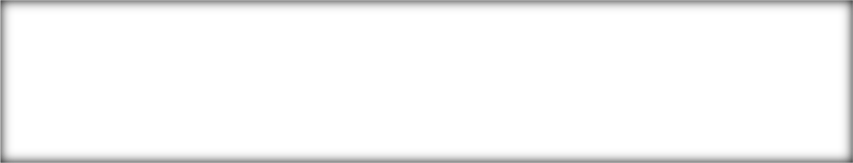
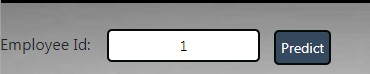




**Code Description:** The following code is used to find the confusion matrix using which we compare the True and False Positive values and also the True and False Negative Values. Using the predict function we get the accuracy of the stacking classifier together on the training and the testing dataset as 90.10%.

# 6. TESTING

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test case id** | **Test cases** | **Preconditions** | **Input test data** | **Steps to be executed** | **Expected result** | **Actual result** | **Pass/ Fail** |
| 1 | Using Employee  id | Performs  Predictive  Analysis | Reading data from the input .csv file | Choosing any id from the given list | If  employee  id exists shows  attrition rate | Checks whether  attrition occurs or not | Pass |



## Figure.6.1. Using Employee ID

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test case id** | **Test cases** | **Preconditions** | **Input test data** | **Steps to be executed** | **Expected result** | **Actual result** | **Pass/ Fail** |
| 2 | Confidence Level | Performs  Predictive Analysis denotes confidence of the algorithm | Based on  all the  features shows how accurate the prediction  is | Choosing any id from the given list and examining the features | To have maximum confidenc  e level for each employee record | Estimating with  maximum  confidence | Pass |

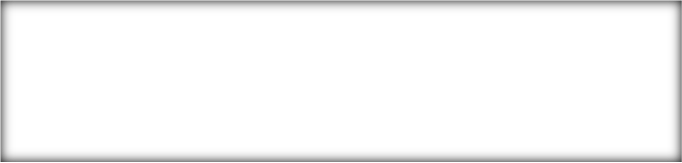
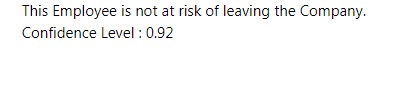


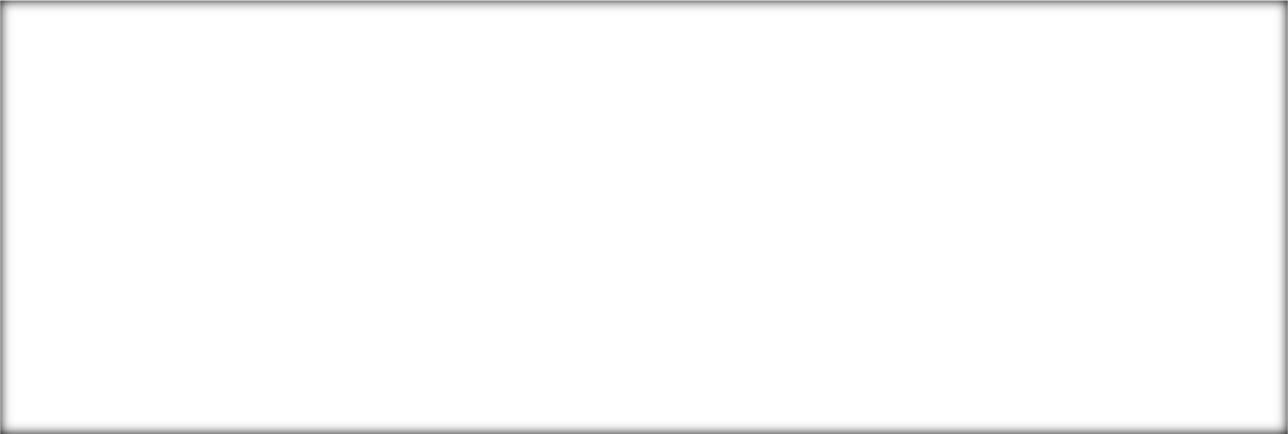
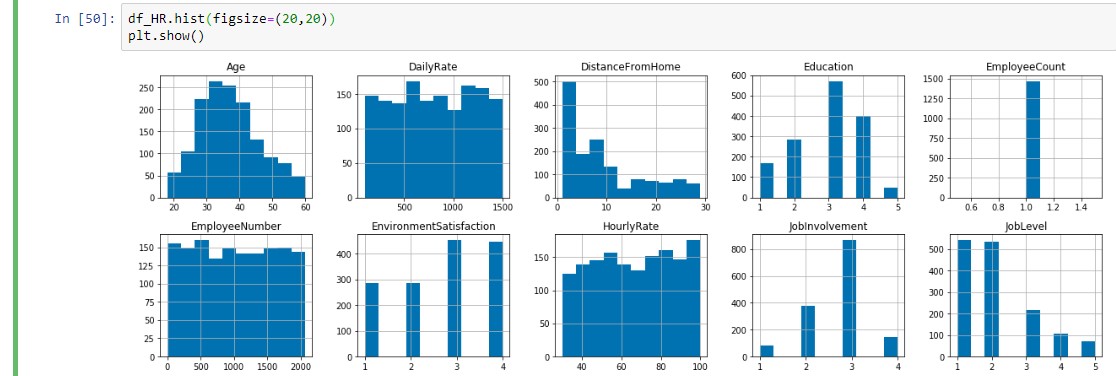
Figure.6.2. Determining the Confidence Level

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test case id** | **Test cases** | **Preconditions** | **Input test data** | **Steps to be executed** | **Expected result** | **Actual result** | **Pass/ Fail** |
| 3 | Descriptive Analysis | Determines the visual representation of the dataset for all features | The dataset provided containing features of the employees | Examining the changes in the features | Graphic representation of the data | Graphic representation of the data involving different features | Pass |

Figure

.6.3

. Graphical Representation of the data



|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test case id** | **Test cases** | **Preconditions** | **Input test data** | **Steps to be executed** | **Expected result** | **Actual result** | **Pass/ Fail** |
| 4 | Predictive Analysis | Predicts if the employee leaves the organisation or not using various algorithms | Uses Features of the  employees from the dataset | Applying algorithms  on the employee  features to predict turnover | Accurately predict  the employee  attrition rate | Prediction of the  employee  attrition | Pass |

Figure

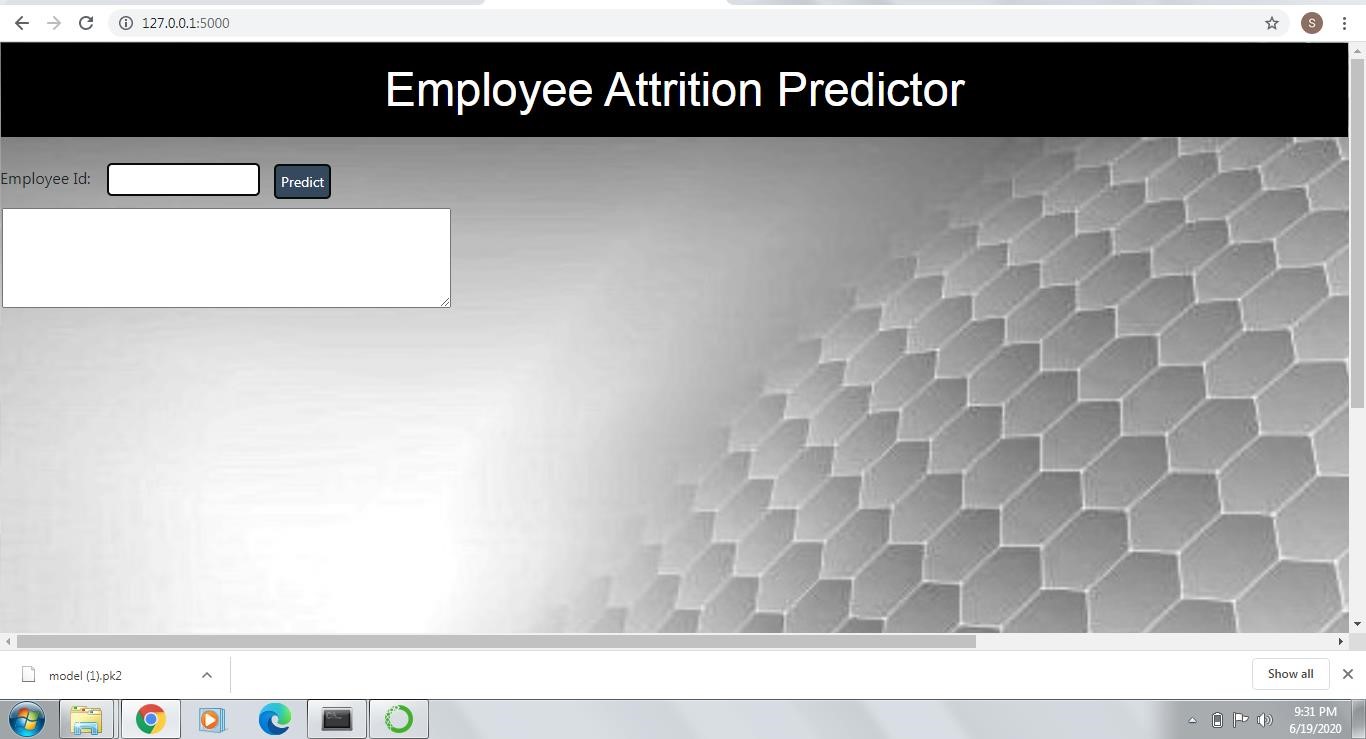
.6.4

. Prediction of Attrition



# 7. OUTPUT SCREENS

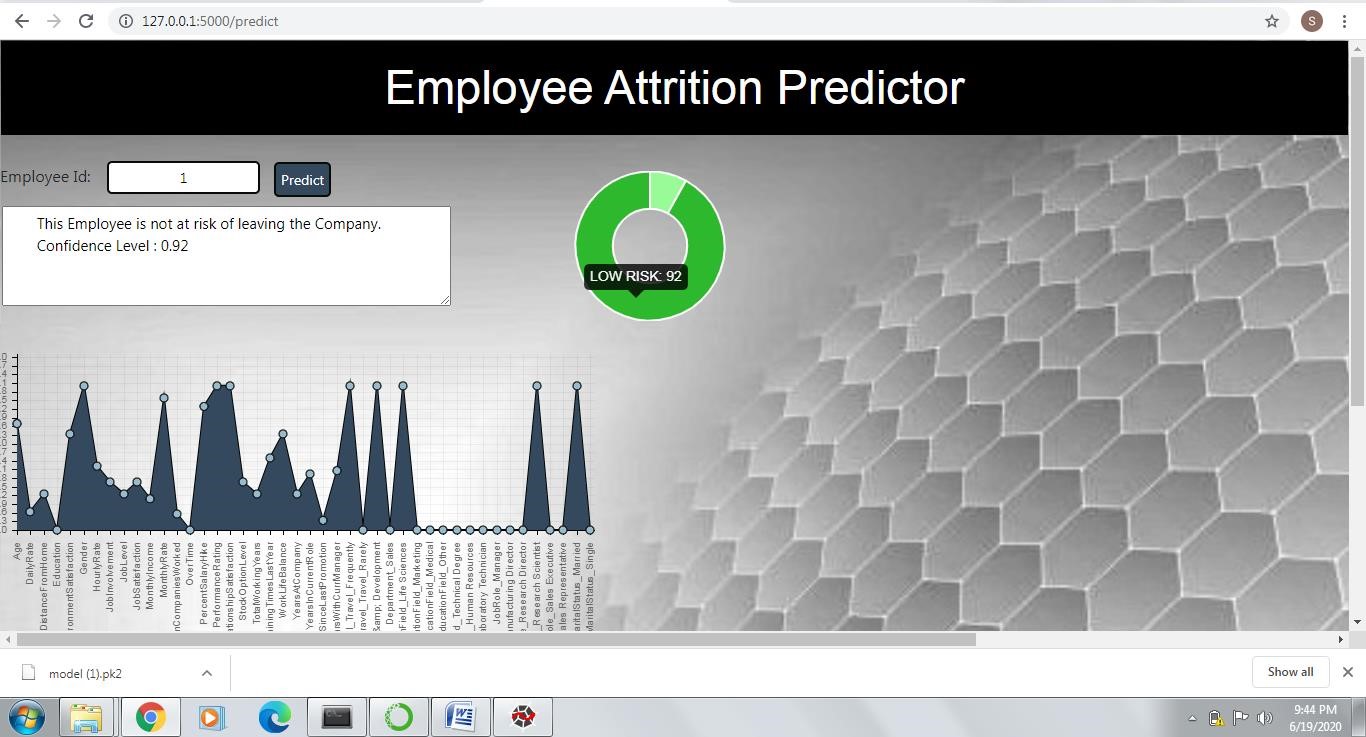
## 7.1 Home Page of the Employee Attrition Predictor



### Figure.7.1. Opening page of the Employee Attrition Predictor

The above output screen indicates the home page of the Employee Attrition Predictor. On executing the commands for setting up a Flask Environment using FLASK\_APP and set development command, we run flask using flask run command. The execution starts and a link is visible which has to be copied on the browser to open the predictor. The link is **https://127.0.0.1:5000/**

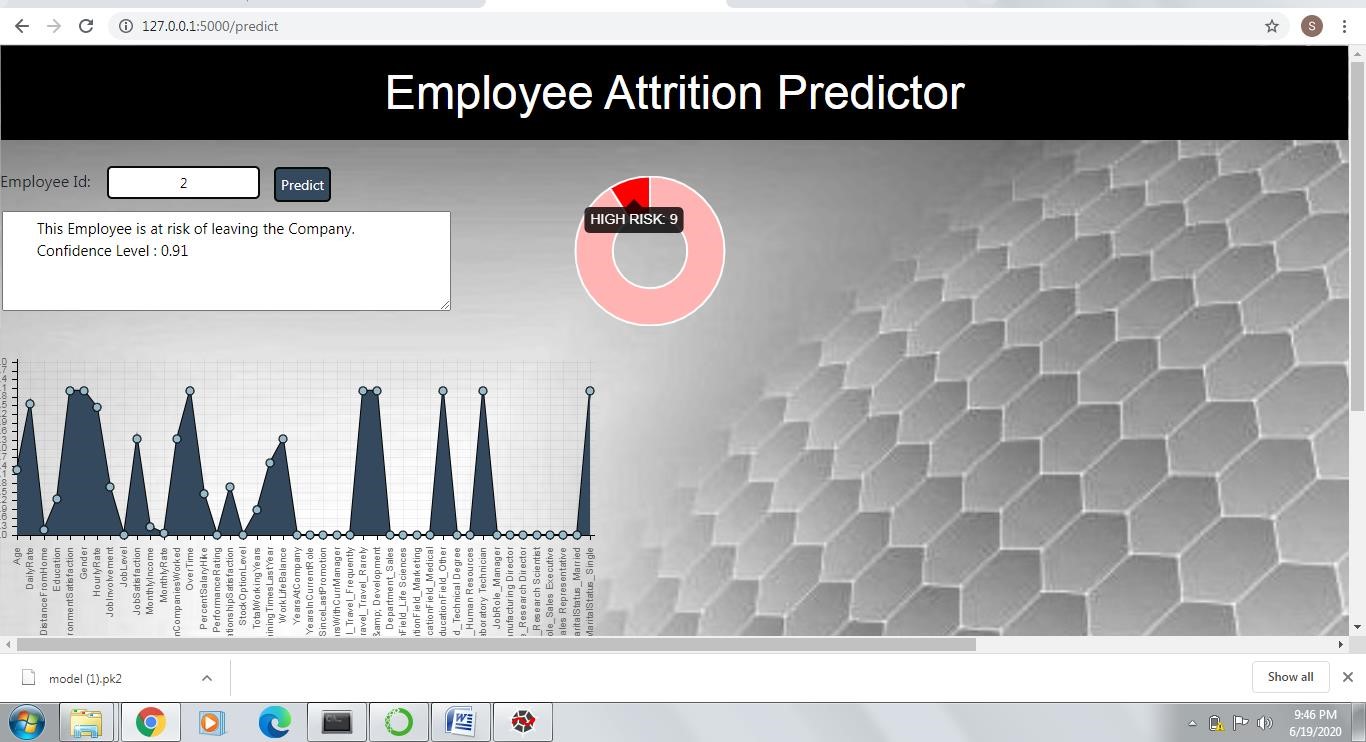
## 7.2 Employee with Low risk of leaving the company



### Figure.7.2. Employee with Low risk of leaving the company

The above output shows the low risk of an employee leaving the company or the organisation. Green colour indicates low risk and its confidence level is also displayed. This means that the algorithms are 92% confident that the employee with id 1 has a low risk of leaving the company. The graph in the output indicates the changes of the features of the employee.

## 7.3 Employee with High risk of leaving the company



### Figure.7.3. Employee with High risk of leaving the company

The above output screen indicates high risk of an employee whose leaving the company. The red colour indicates high risk of leaving or attrition. The algorithms are 91% confident about the attrition of the employee with id 2 from the dataset used. The graph in the output indicates the changes of the features of the employee and describes how drastic the changes are.

# CONCLUSION

After performing various empirical tests on the IBM HR Analytics Employee Attrition & Performance dataset from Kaggle with 35 features for employee attrition prediction in an organization, the results from the table in the above section depict that the Stacked Classifier provides the most optimal solution with a score accuracy of >90%. Hence, it can be stated that multiple trained models learned on different classification algorithms can be cross validated, averaged and passed into a Meta Classifier to improve the accuracies of the individual classification models. An employee’s attrition value prediction could benefit the organization by knowing where they are going wrong and with which employees. The organization could accordingly start looking for people to fill the positions which are predicted to be empty by using this stacked classifier model.

## Future Enhancement

Currently, the dataset size is 1470 rows. If this size is increased, the accuracy can be further improved by the stacked classifier. The logic here is that if there is more data, it will enable training more exhaustively and one could also implement deep learning algorithms. Furthermore, feature engineering will not be required as principal component analysis could work on the 35 features of the dataset and use the important ones which contribute to the attrition value.

Further research can be conducted by adding a couple of facets to it. The study can be replicated in other sectors where attrition has become a common problem. IT industry has close similarities with that of a BPO industry with an equally alarming rate of attrition. Similar study can also be done in sectors like hospitality, infrastructure, retail, education etc. as intrinsic motivation becomes a pillar for performance, loyalty and eventual sustenance almost everywhere.

## References

1. IBM HR Analytics Employee Attrition & Performance (2018, 28, August) Kaggle Inc. [Online]Available: https://www.kaggle.com/pavansubhasht/ibm-hr-analytics-attrition-dataset/
2. 7 Types of Classification Algorithms (2018, 6, September) Analytics India Magazine [Online] Available:analyticsindiamag.com/7-types-classification-algorithms/
3. Stacking Classifier (2018, 10, September) Github [Online] Available:

https://rasbt.github.io/mlxtend/user\_guide/classifier/StackingClassifier/

1. Ensemble methods - Adaptive Boosting (2018, 14, September) Scikit Learn [Online] Available: http://scikit-learn.org/stable/modules/ensemble.html#adaboost
2. Decision Trees (2018, 15, September) Scikit Learn [Online] Available: http://scikitlearn.org/stable/modules/tree.html#tree [6] R. Anbu Ranjith Kumar and Dr. V. Antony Joe Raja
3. Understanding Support Vector Machines (2018, 16, September) Analytics Vidhya [Online] Available:https://www.analyticsvidhya.com/blog/2017/09/understanding-support-vector machine-example-code/
4. B. C. Holtom, D. R. Smith, D. R. Lindsay, and J. P. Burton,“The relative strength of job attitudes and job embeddedness in predicting turnover in a U.S. military academy,” *Military* *Psychology*, vol. 26, no. 5-6, pp. 397–408, 2017.
5. Xiang Gao , JunhaoWen , and Cheng Zhang“Methodolgy” *An Improved Random Forest Algorithm for Predicting Employee Turnover*, 17 April 2019.
6. https://en.wikipedia.org/wiki/Anaconda\_(Python\_distribution)

# APPENDIX 1

**RELEVANCE OF PROJECT TO POs / PSOs**

|  |  |
| --- | --- |
| **Title of Project** | EMPLOYEE ATTRITION SYSTEM:  Evaluation of Staff Performance |
| **Implementation Details** | PYTHON |
| **Cost (hardware or software cost)** | -NA- |
| **Type (Application,**  **Product, Research, Review, etc.)** | APPLICATION |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Mapping with POs and PSOs with Justification** | | | | | | | | | | | | |
| **Relevance** | PO 1 | PO  2 | PO  3 | PO  4 | PO  5 | PO  6 | PO  7 | PO  8 | PO  9 | PO 10 | PO 11 | PSO  1 | PSO  2 |
| 1 | 2 | 3 | - | 2 | - | - | 1 | 2 | 3 | - | 1 | 1 |
| **Program**  **Outcomes**  **Justification** | PO1: Engineering Knowledge: SDLC phases are followed in the execution of the project.  PO2: Problem Analysis: The different steps involved in Problem Analysis for formulation of the solution i.e. literature survey and use of fundamental subject knowledge has been followed.  PO3: Design/Development of solutions – Existing strategy has been enhanced using the design principles.  PO5: Modern Tool Usage: Jupyter and Spyder Notebook are used. PO8: Ethics: Students have followed professional ethics during the various stages of Project completion.  PO9: Individual and Team Work: Students have worked both in individual as well as team capacity during the various stages of project work.  PO10: Communication: Effective communication with team members and during project reviews, project seminar and viva-voce has been exhibited. PO12: Lifelong Learning. The project carried out gives the students scope to continue the work in the Computer Vision area in future. | | | | | | | | | | | | |
| **Program**  **Specific**  **Outcomes**  **Justification** | PSO1: Use of open source software Jupyter and Spyder Notebook (Anaconda Navigator).  PSO2: Web browser is used to display the predictor tool to perform predictions on employee attrition. | | | | | | | | | | | | |

# APPENDIX 2 GANTT CHART

## EMPLOYEE ATTRITION SYSTEM: Evaluation of Staff Performance

PLAN OF WORK

