Chapter 1

Introduction

1.1 Purpose

- An intrusion detection system (IDS) or Network intrusion detection system (NIDS) has been developed that is capable of detecting all types of network anomalies or attacks in the available environments.
- Naïve Byes classifier and classifier is proposed for intrusion detection. In the proposed model, a multi-layer Hybrid Classifier is adopted to estimate whether the action is an attack or normal data.
- The purpose is to reduce the false alerts in order to reduce the work of security analyst.

1.2 Scope

- The main objective is to reduce the false alerts and improve the performance of Intrusion Detection System.
- It has been developed that is capable of detecting all types of network anomalies or attacks in the available environments.
- Machine Learning can be used to implement an Intrusion Detection System to detect if an alert is truly malicious or not.

1.3 Literature Survey

Field writing survey demonstrates that Network security is one of the major issues and different types of developed systems are being implemented. Principally two types on network intrusion detection can be classified:

- SIGNATURE BASED: Compares the signature of an incoming packet with the database of existing signatures.If the signature matches the signatures in the database it is marked malicious. It is very effective in detecting known type of attacks, but is unable to detect new attacks.

- ANOMALY BASED: This is used to model a behaviour and if it deviates too much from normal it is marked malicious.Its a statistical based approach that is used to define normal behaviour. It works well with new attacks. But even if a new attack is just different from the normal it is marked malicious.

SUPPORT VECTOR MACHINE: It's a supervised machine learning algorithm. It provides a powerful method for classiffcation. The basic approach in svm : The main goal seems to maximize the hyper plane. Hyper plane are the best separate between two classes and can Find by measuring the hyper plane margin and Find maximal data. Margin is the distance between the hyper plane with the closest data from every class. The closest data is nothing but the support vector. In purposed framework NSL-KDD dataset is ranked using IGR and later feature subset selection is done using K-mean algorithm.

GENETIC ALGORITHMS: Based on identification of complex anomalous behaviours mainly focused on TCP/IP network protocol. The basic procedure: Selection of chromosomes, represent the solutions. Different position of each chromosome are encoded as bits also called genes. Fitness function / Evaluation Function: Used to evaluate the goodness of each chromosome. Two basic operators: Crossover and Mutation Based on rule of ”Survival Of the Fittest - Darwin Theory”. It performs a match between the current n/w and rules in IDS such as src/dest IP address indicating probability of intrusion. Action: Report to Admin, Stop the connection.

HYBRID CLASSIFIERS: Collects every single incoming network requests The basic procedure: A decision tree algorithm is used to train the Misuse Detection model from the available normal and attack training data. Info Gain is calculated for all attributes to understand the contribution of the attributes towards the classification. After training, its checked against the DT Model to classify if it's an attack or not, if there is no response it is sent to Naive bayes classifier - anomaly detection.

1.4 Existing Systems

The existing systems implemented using various algorithms like SVM,Hybrid, Genetic algo, Decision tree,and have a high false positive rate. Thereby our aim is to improve the accuracy and reduce the false positive rate.

1.5 Proposed System

A new algorithm is proposed using hybrid classifier which is a combination of C4.5 classifier and Naive baye's classifier to detect if its a normal packet or malicious. The system can be used to detect the type of attack in order to reduce false rate and improve accuracy.

1.6 Statement of the problem

We are implementing a machine learning framework for ids to detect if a alert is truly malicious or not thereby reducing the false errors and reducing the work of a security analyst.

1.7 Summary

This chapter gave a brief introduction on what exactly the proposed system is. It also covered the future scope and the demand of this product in the market. Also how this product can reduce or minimize the noise caused by feature differences and improve the performance of Intrusion Detection System. It even marked the main features of the device which helps in overcoming the drawbacks of the existing system.

Chapter 2

System Requirements Specifications

2.1 Software Requirements Specifications

Requirement specification is the movement of interpreting the data assembled amid investigation into prerequisite report.
Software requirements specifications are the detailed enlisting of all necessary requirements that arise in the project. The aim of having these requirements is to gain an idea of how the project is to be implemented and what is to be expected as a result of the project. The sections in this chapter deal with the various kinds of software, hardware and other functional and non functional requirements of the project. A brief description of the various users of the system is also mentioned.

2.1.1 Operating Environment

This section gives a brief about the hardware and software prerequisites for the project.

Hardware Requirements

- **Processor**: Intel Core i5 or higher
- **RAM**: 8.0GB
- **Storage**: 1TB hard disk
- **Monitor**: LCD
- Other general hardwares such as a mouse and keyboard for inputs.

Software Requirements

- **Operating system**: Ubuntu 14.04 and above
- **Programming languages**: Python, HTML5, CSS
- **Documentation**: Overleaf

2.1.2 Functional Requirements

Functional requirements are a formal way of expressing the expected services of a project. We have identified the functional requirements for our project as follows:

- The system should be able to explain the integration of the models.
- The system should be able to discuss the properties of the proposed hybrid intrusion detection methods.
- The system should model the traffic and differentiate between a normal packet and an anomaly.
- The system should be able to have the capacity to decide the contribution of each attribute towards the decision made by the predictor.

2.1.3 Non-Functional Requirements

Non functional requirements are the various capabilities offered by the system. These have nothing to do with the expected results, but focus on how well the results are achieved.

- Usability: The Prediction system is highly user-friendly and conveniently usable because of the easy-to-use graphical user interface.
- Reliability :The prediction subsystem should give accurate results.
- Security : Including bug tracking the system must provide necessary security and must secure the whole process from crashing.
- Performance : The software system will be hosted on a web server with a single application and should be able to detect normal and abnormal anomalous behavior of the network.
- Portability : This is required when the web server, which is facilitating the framework stalls out because of a few issues, which requires their system to be taken to another system.
- Re-usability : The degree to which existing applications can be reused in new application. The predicted output could be reused in many fields.

2.1.4 User Characteristics

There is only one type of user associated with the system:

- User of a system is a person supervising the network.

2.1.5 Applications

The intrusion detection system framework can be used by the soc (security operation center) analyst to detect if a particular alert is actually malicious thereby minimizing the false positive. Since cyber security is area of great concern , this framework can be used there. It reduces the work of the network analyst by giving accurate result. It can be used to assure cyber security of a enterprise like SIEM.

2.1.6 Summary

This chapter discussed the basic software and hardware requirements. More importantly it discusses the functional and non-functional requirements.

Chapter 3

High Level Design

This section mainly covers the design technique of the entire system which involves the implementation of 2 modules which are as follows:

- Training the machine learning model and save
- Using the trained model as a web service for prediction

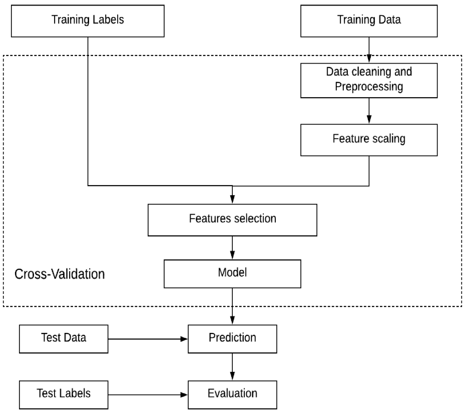


Figure 3.1: High Level Design

3.1 Design Approach

Here are two methodologies for software designing:

- Top-down Design:It takes the entire programming framework as one entity and after that disintegrates it to accomplish in excess of one subsystem or some components based on few attributes.

- Bottom-up Design: The model begins with most particular and essential components. It accedes with making more elevated amount out of subsystems by utilizing essential or lower level components.

As mentioned above the project requires two main modules to be implemented. Each module has its own components to be developed. We use bottom-up design strategy in this product design phase as we start designing the basic components in each module and finally we interlink both the modules to get the final product.

3.2 System Architecture

The architecture diagram design outline gives a review of a whole framework, distinguishing the primary segments that would be created for the item and their interfaces.

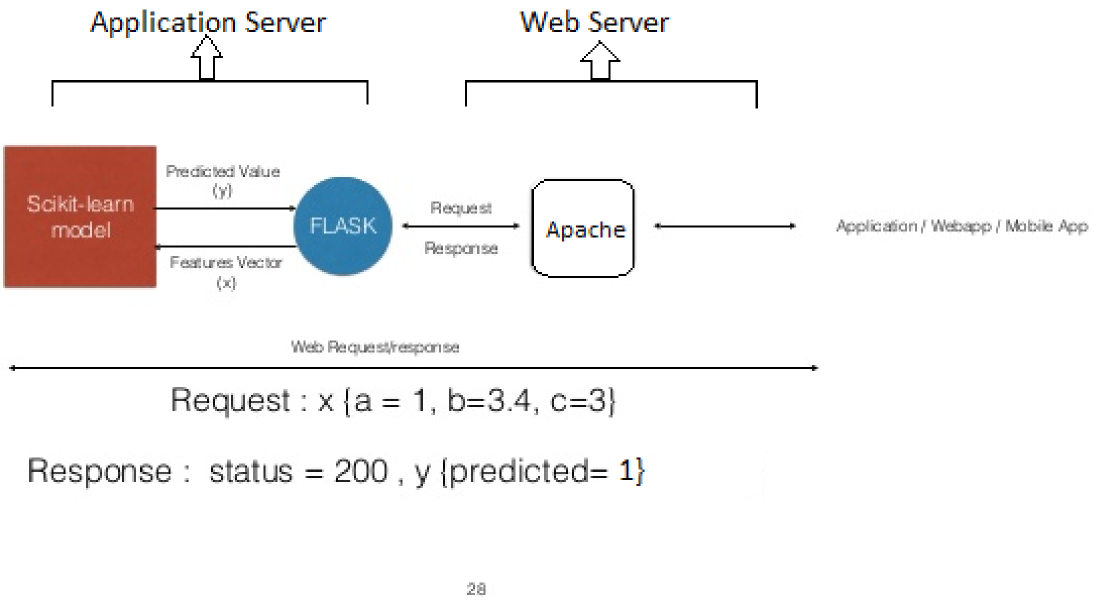


Figure 3.2: System Architecture

3.3 Data Flow Diagram

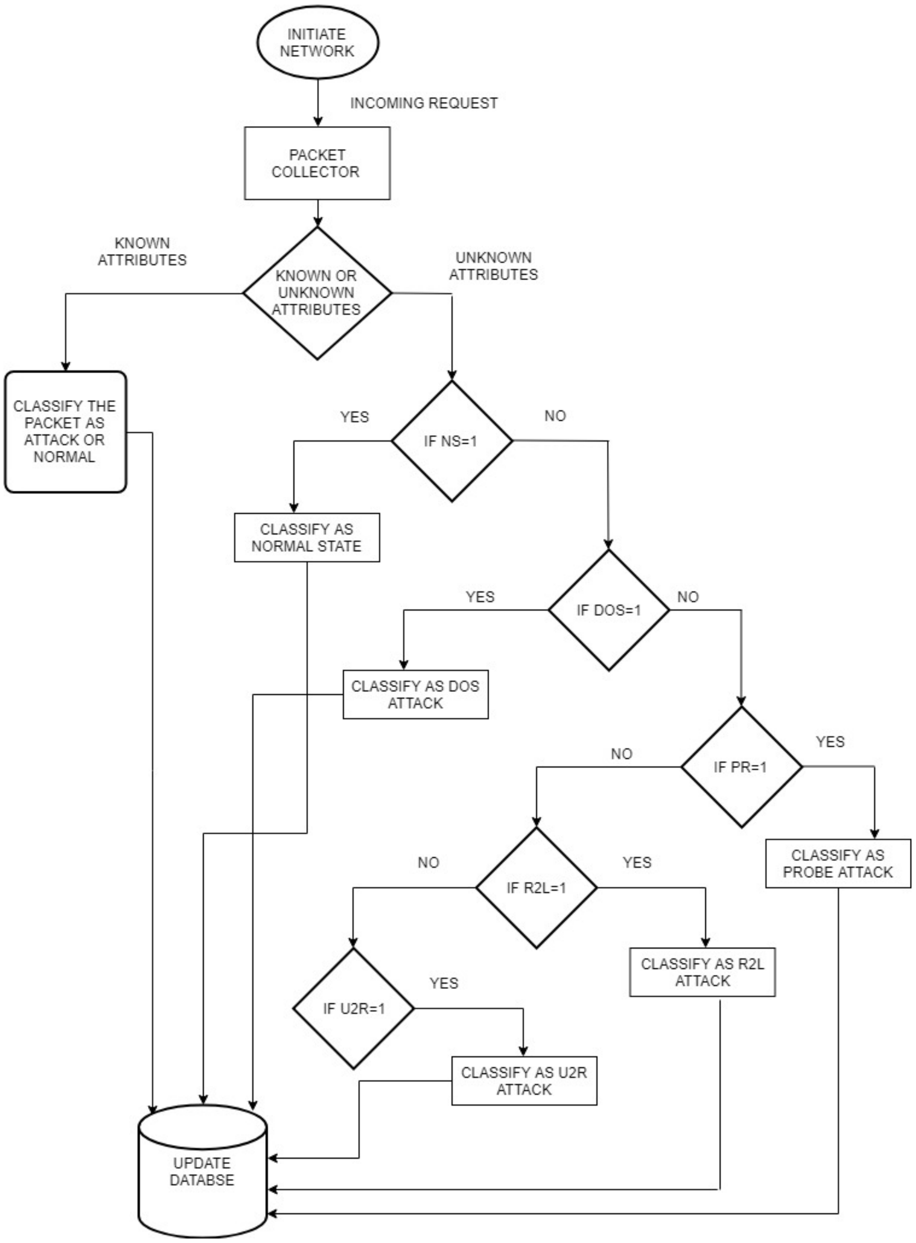


Figure 3.3: Data Flow Diagram

3.4 Sequence Diagram

A sequence diagram is the representation of interactions of components among each other in order. It shows the interactions between the objects in the system with the time order that particular interaction takes place. Since it shows the time order of the interactions it is called as sequence of events and hence the sequence diagram.

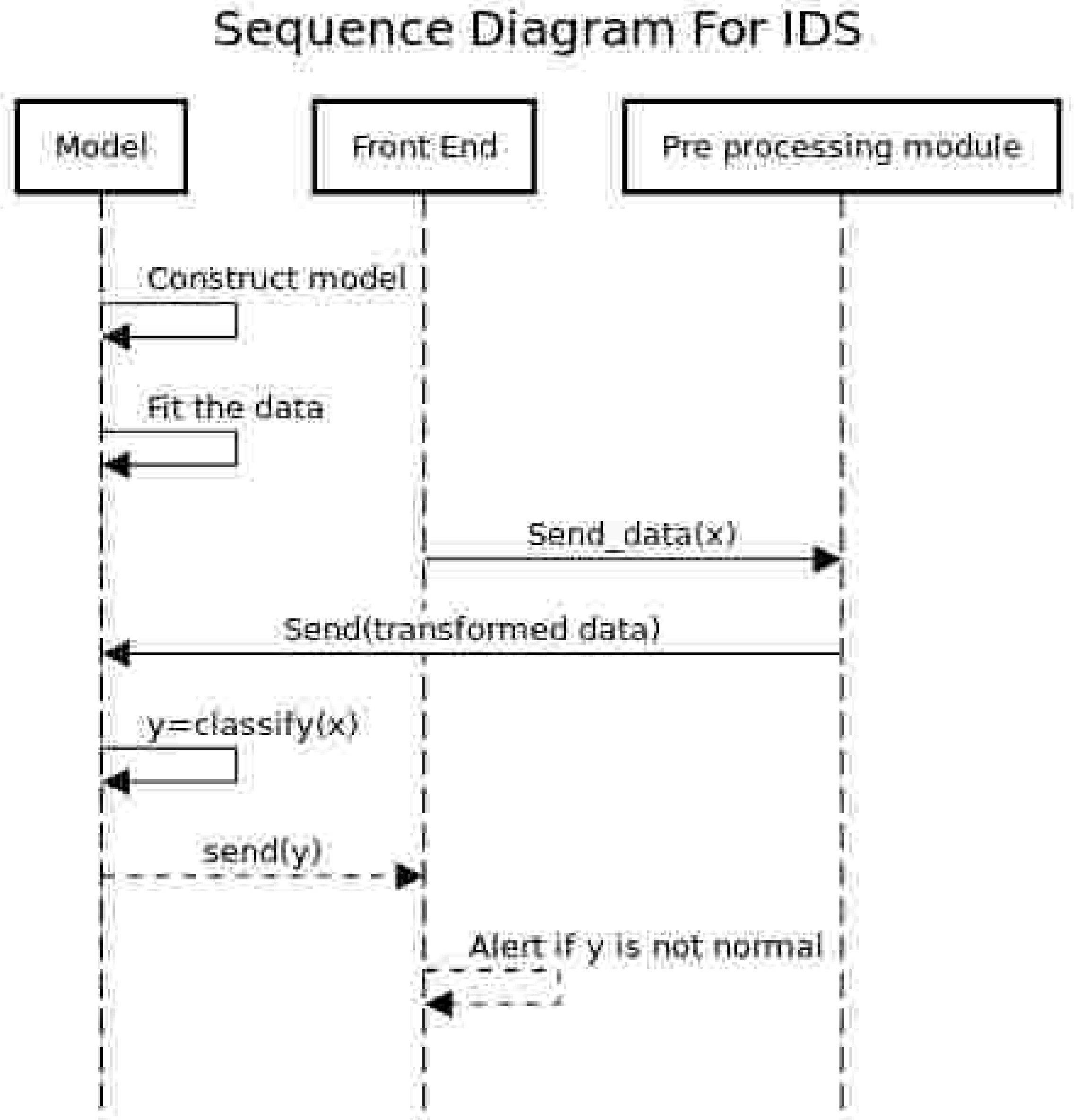


Figure 3.4: Sequence Diagram

3.5 Summary

In this chapter we discussed the different design patterns which can be used in any product development cycle. For our project mainly sequence diagram is used. We also discussed the flowchart which shows the data flow between various components. This chapter even described the high level design.

Chapter 4

Detailed Design

4.1 Purpose

The purpose of the detailed design is to plan our system to meet the requirements specified at the start. In the detailed design we see what is the input data for each model, how the model implementation is carried out and how the output is interpreted. The basic purpose of the project is to find the malicious packets in a network.

4.2 Module 1: Prediction

Consider the following flowchart. The dataset we used is obtained from KDD Dataset. In the preprocessing phase the missing values are handled and one-hot encoding is done on specific features. This obtained dataset is split into training and test data. The model is trained with the help of train data and the obtained model which is then tested against the test data achieved an accuracy of 98%.The model predicts the class label which is attrition and the same is displayed to the user via the web interface. This ends the prediction phase.

4.3 Classification

Classification is the process of grouping of entities with the similar attribute under one class label. This is necessary because it validates our hypothesis on the efficiency of our model. Classification in our case is a binary classification showing the overall attrition of the company by predicting which class the employee belongs to, that is whether he is going to leave the company or going to stay in the company. Type 1 represents the samples in which the employee is going to leave the company and Type 2 represents the remaining samples in which employee will not leave the company.

4.4 Employee Dataset

The Employee Dataset contains several features of the employees, such as Satisfaction,Evaluation, Number of projects,Years at the company,Gender, Work accident and Salary etc.All the features which are not required with the prediction of Employee Attrition have been removed.Only the features which

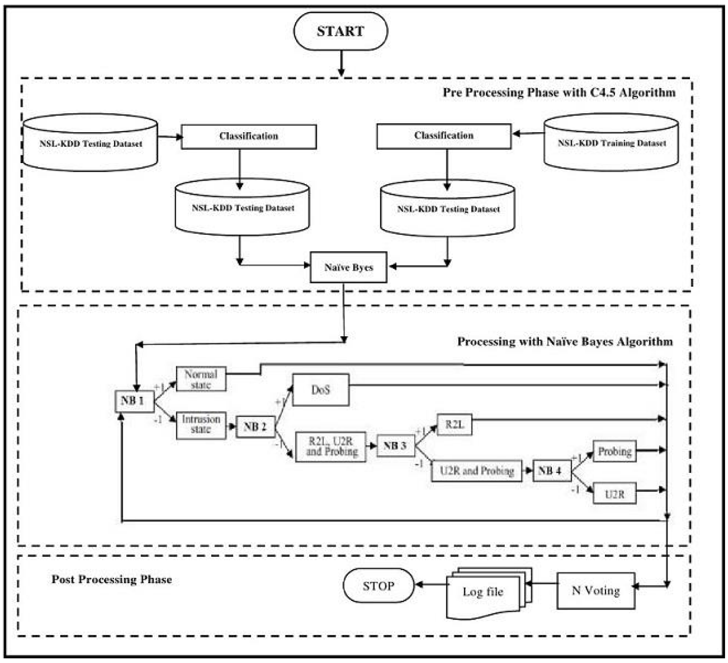


Figure 4.1: Detailed Design

are required have been isolated.These features are mentioned below in the figure 4.3.The approach of this project was from a hypothetical perspective so data preprocessing has been done, that is we removed the data which has any blank features or information vacancy.This was done to provide the classifier with the highest quality of data also can be used to test maximum amount of real time data. This dataset can be used as a training grounds to best type of fit for all the further incoming data of similar nature.As mentioned before, the isolation of features as required by Employee Prediction is necessary to initiate the right model.

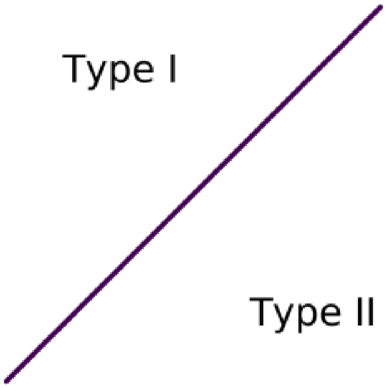
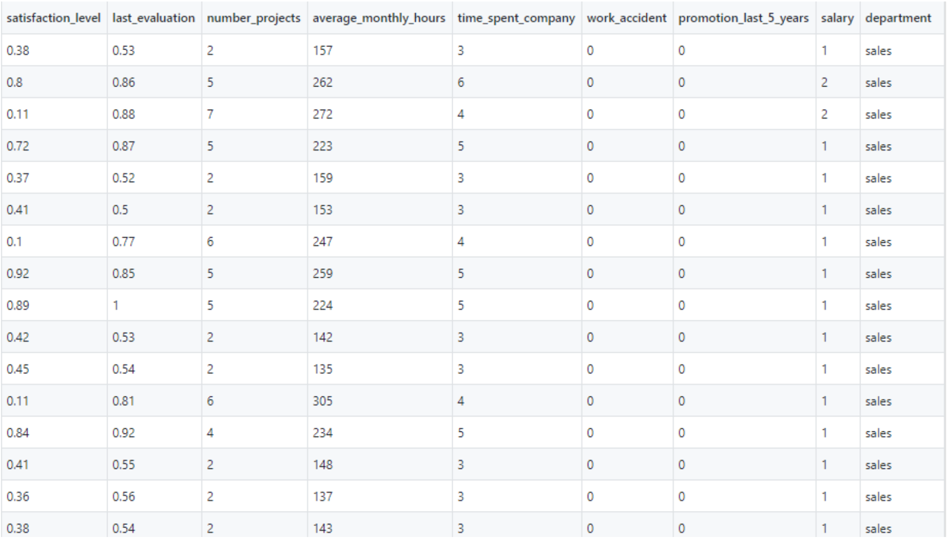


Figure 4.2: Binary Classification



satisfaction_level	last_evaluation	number_projects	average_monthly_hours	time_spent_company	work_accident	promotion_last_5_years	salary	department
0.38	0.53	2	157	3	0	0	1	sales
0.8	0.86	5	262	6	0	0	2	sales
0.11	0.88	7	272	4	0	0	2	sales
0.72	0.87	5	223	5	0	0	1	sales
0.37	0.52	2	159	3	0	0	1	sales
0.41	0.5	2	153	3	0	0	1	sales
0.1	0.77	6	247	4	0	0	1	sales
0.92	0.85	5	259	5	0	0	1	sales
0.89	1	5	224	5	0	0	1	sales
0.42	0.53	2	142	3	0	0	1	sales
0.45	0.54	2	135	3	0	0	1	sales
0.11	0.81	6	305	4	0	0	1	sales
0.84	0.92	4	234	5	0	0	1	sales
0.41	0.55	2	148	3	0	0	1	sales
0.36	0.56	2	137	3	0	0	1	sales
0.38	0.54	2	143	3	0	0	1	sales

Figure 4.3: Employee Dataset

4.5 Data Distribution

The below figure 4.4 is a bi-modal distribution for those who had a turnover employees with low evaluation tend to leave the organization more, employees with superior likewise also tend to leave the organization more.High density of the employees that stayed is within the organization have the evaluation range between 0.6-0.8.

The below figure 4.5 is a tri-modal distribution for employees that has left the organization the employees who had really low satisfaction levels (0.2 or less) left the organization and the employees

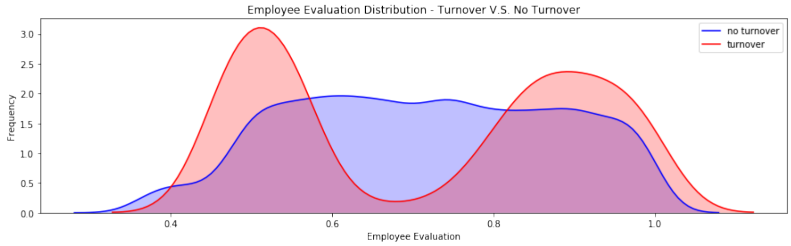


Figure 4.4: Bi-Modal Distribution - Turnover V.S. No Turnover

who had low satisfaction levels (0.3 0.5) left the organization and also the employees who had really high satisfaction levels (0.7 or more) too left the organization in most significant cases.

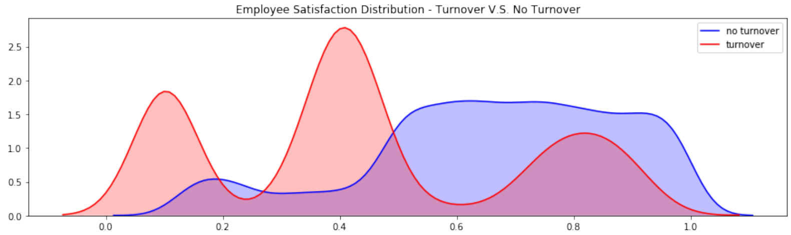


Figure 4.5: Tri-Modal Distribution - Turnover V.S. No Turnover

There are three unique clusters for employees who left the organization

Cluster 1 (Hard-working and Sad Employee): Satisfaction beneath 0.2 and evaluations more prominent than 0.75, Which could be a decent sign that employees who left the organization were great specialists but however felt ghastly at their activity.

Cluster 2 (Bad and Sad Employee): Satisfaction between 0.35-0.45 and evaluations underneath -0.58. This could be viewed as employees who were severely assessed and felt awful at work.

Cluster 3 (Hard-working and Happy Employee): Satisfaction between 0.7-1.0 and evaluation more prominent than 0.8. Which could imply that employees in this bunch were ”perfect”. They cherished their work and were assessed exceptionally for their execution.

4.6 Random Forest Classifier

After the Data Assessment is done in the previous section it now falls heavily on us to choose the classifier.There are multiple ways we can approach this problem such as SVM,Decision Trees et cetera.But as we tried and tested we got the best scores for the Ensemble Method Random Forests. Random

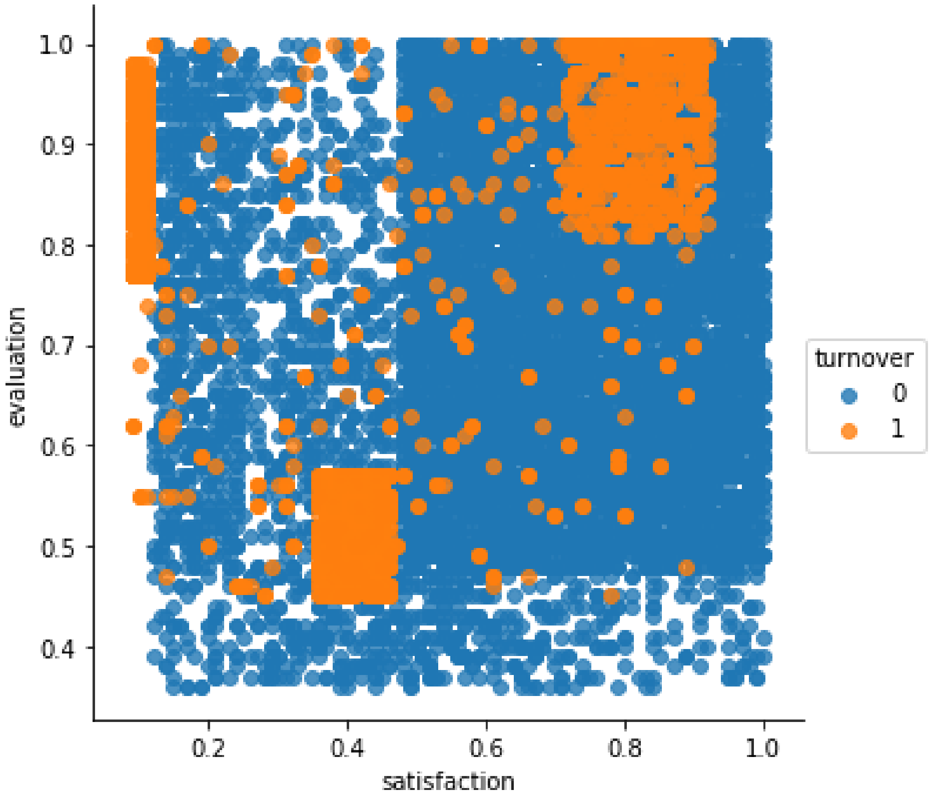


Figure 4.6: Clusters in the Dataset

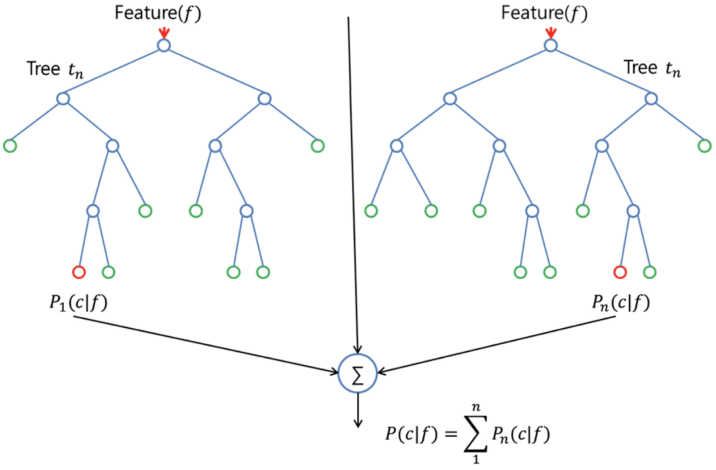


Figure 4.7: Visualization approach using Random Forest

forests is one of the most used supervised learning algorithm because of it’s simplicity and the fact that it can be used for both classification and regression tasks. It basically is an ensemble of decision trees trained with bagging method.

$$\hat{f} = \frac{1}{B} \sum_{b=1}^{B} f_b(x')$$

Figure 4.8: Equation

The training algorithm for random forests applies the general strategy of bootstrap aggregating, or bagging, to tree learners. Given a training set X = x_1, ..., x_n with responses Y = y_1, ..., y_n, bagging over and over (B times) chooses an arbitrary sample with replacement of the training set and fits trees to these samples:

For b = 1, ..., B:

- Sample, with replacement, n training examples from X, Y; call these X_b, Y_b.
- Train a classification or regression tree f_b on X_b, Y_b.

So to say in simple words,Random forest creates multiple decision trees and merges them together to get a more accurate and stable prediction.This method ensures that the results are usually better or equal to decision trees for most cases.

Chapter 5

Implementation

Implementation is the phase of the undertaking when the hypothetical plan is transformed out into a working framework. Subsequently it can be thought to be the most basic stage in accomplishing fruitful new framework and in giving the client, certainty that the new framework will work and be efficient.

5.1 Programming Language Selection

This project is implemented in Python,JavaScript and Flask as they are:

- Simple: The languages was designed to be easy for the professional programmer to learn and use effectively. They inherits the C/C++ style and many object oriented features of C++.
- Object-oriented: The object model in Flask is simple and easy to extend, while primitive types are kept as non-objects for performance reasons.
- Robust: Python frees you from worrying about a portion of the basic programming mistakes. It is an entirely composed dialect and checks the code both at arrange time and run time.
- Multi threaded: JavaScript was intended to meet this present reality prerequisite of making intelligent system programs. To accomplish this, it supports server-side programming on the client side.
- Interpreted and High-performance: Web development enables the creation of cross-platform programs by running on the browser.

5.2 Platform Selection

5.2.1 Linux(Ubuntu 16.10)

Linux is one of famous form of UNIX working System. It is open source as its source code is uninhibitedly accessible. It is allowed to utilize. Linux was composed considering UNIX compat-ibility. Its usefulness list is very like that of UNIX.Virtual conditions have the preferred standpoint that they never introduce the required dependencies system wide so we have a superior control over the environment in which our application is running. We can choose only to install the required libraries and packages and keep the environment clean.

Virtual environment is also important as we may have multiple applications on one system with conflicting requirements.

5.2.2 Flask

Flask is a Python framework, based on Werkzeug, Jinja2 and inspired by Sinatra Ruby framework, available under BSD license. Some of the important features of Flask are:

- built-in development server and fast debugger
- integrated support for unit testing
- RESTful request dispatching
- Jinja2 template
- support for secure cookies (client side sessions)
- WSGI 1.0 compliant
- Unicode based

5.2.3 Python

Python is a translated, object oriented programming language like PERL, that has picked up prevalence in light of its reasonable syntax and comprehensibility. Python is said to be relatively simple to learn and convenient, which means its statements can be translated in various working frameworks, including UNIX-based frameworks, Mac OS, MS-DOS, OS/2, and different renditions of Microsoft Windows 98. Python was made by Guido van Rossum, a previous occupant of the Netherlands, whose most loved satire gathering at the time was Monty Python's Flying Circus. The source code is unreservedly accessible and 'open for modification and reuse. Python has a critical number of clients. A remarkable highlight of Python is its indenting of source explanations to make the code less demanding to peruse. Python offers dynamic information compose, instant class, and interfaces to numerous framework calls and libraries. It can be broadened, utilizing the C or C++ dialect.

5.3 Libraries Required

- **scikit-learn**: scikit learn is a wide python library which practices machine learning, cross validation of data, and preprocessing of data. It is built on NumPy and SciPy.
- **NumPy**: NumPy's main object is an multidimensional array, it holds an array data structure. NumPy is a core python library which contains a collection of tools and techniques. One of these tools is multi dimensional object.
- **Maplotlib**: It is a library extensively used for visualizing the data.
- **SciPy**: It is a library which contains functions to perform more complex calculations, particularly scientific computations.
- **Pandas**:Pandas is an open-source, BSD-authorized Python library giving high execution, simple to-utilize information structures and information examination devices for the Python programming dialect.

5.4 Summary

This chapter dealt with the various techniques used in the development of the project,starting with the language and platform selection to finally explain the entire process of implementation steps.

Chapter 6

Testing

6.1 Software Testing

To deliver evidence about the excellence of the product or system under test software testing used.From an unbiased view, it tries to establish the quality of the product.

The resolution of software testing is to find the bugs or errors or in the program.Because of being an iterative process,a bug identification can lead to illumination of another bug.Because of the countless number of possible examinations for any simple software component, all software components use same strategy to select test that are feasible for the available time and resources.

6.1.1 White Box Testing

Basically white box testing is a method of software testing that tests the working of the program or an application rather than functionality.Path testing is an obvious example.

It is basically used in spaces where a black box testing cannot reach.For redundancy that occurs in regression of our project, white box testing is used.Since errors can occur anywhere in classification code ,redundancy tests are done.

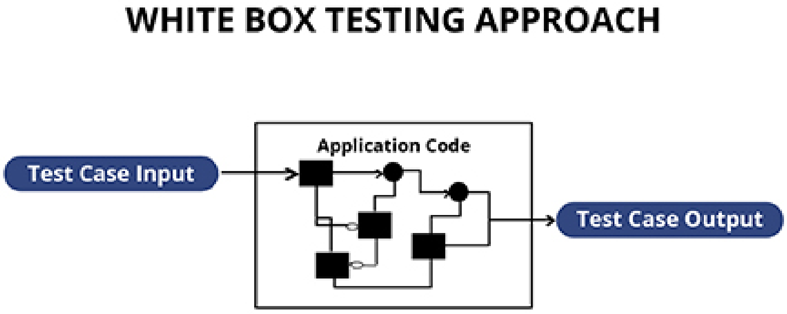


Figure 6.1: White box testing

6.1.2 Black Box Testing

In black box testing the function of the black box is understood completely in terms of inputs and outputs in-spite of not knowing the content or implementation of black box . Specification testing is additionally called as functionalfunctional testing in light of the fact that the program is considered as a capacity that maps esteems from its info area to values in its yield extend.

Test cases of Specification based testing as 2 specific points of interest:

- They are not needy of how the product is actualized. On the off chance that if the implementation changes , the experiments are as yet valuable.
- Reduction in the overall project development interval because both the test case development and implementation can occur in parallel.

6.2 Levels of Testing

There are normally 3 levels of testing:unit testing,system testing and integration testing.

From the point of view of customers there are 2 different levels:low-level testing,high-level testing.

- low level testing is a set of tests for different level components of software application.
- High level testing is a set of tests for the application as a whole.

6.2.1 Unit Testing

In the method individual units of source code,program module associated control data,other procedures are tested to find their fit for use.Specific functionality of the developed subunits are tested.The more appropriate testing at unit testing is structural testing.This testing is followed by integration testing.Integration testing basically clubs all the modules tested for unit testing and perform the tests defined in integration test plan to them.This is then followed by system testing.

6.2.2 Integration Testing

Integration testing is the second level of testing.As mentioned before this combines all the modules of unit testing and tests them.Its main goal is to test whether there are any problems in the integration of different modules.

6.2.3 System Testing

System testing basically evaluates whether a system meets the requirements specified before.For example , a system testing may include inputing values in the input fields,printing results ,format of the results etc.

It also tests the behaviour of system as specified by the customer.It not only tests the requirements in software/hardware specification but also tests beyond that.This is then followed by acceptance testing.

6.2.4 Acceptance Testing

As the name itself suggests acceptance testing is used to test the system compliance for the requirements.Hence basically the system is tested for acceptability.It is the last test that is performed before making the system.

6.3 Unit Testing of Module

Execution of the prediction system is tested for various conditions and the test cases are tabulated as follows:

Test Case ID	Unite Test Case P 1
Description	To test the predictor with input values
Input	0.68,0.46,4,143,3,0,0,0,7
Expected Output	0
Actual Output	0
Remarks	The system performed as expected.

Unit Test Case 1

Test Case ID	Unite Test Case P 2
Description	To test the predictor with input values
Input	0.41,0.47,2,138,3,0,0,0,7
Expected Output	1
Actual Output	1
Remarks	The system performed as expected.

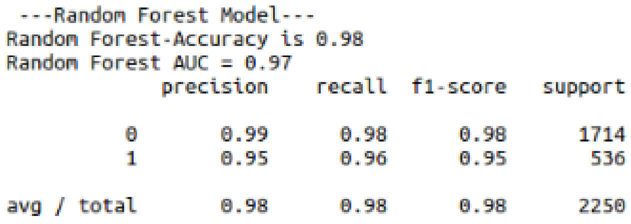
Unit Test Case 2

Test Case ID	Unite Test Case P 3
Description	To test the predictor with input values
Input	0.86,0.87,5,156,4,0,0,0,7
Expected Output	0
Actual Output	0
Remarks	The system performed as expected.

Unit Test Case 3

Test Case ID	Unite Test Case P 4
Description	To test the predictor with input values
Input	0.60,0.86,6,272,4,0,0,0,9
Expected Output	0
Actual Output	0
Remarks	The system performed as expected.

Unit Test Case 4

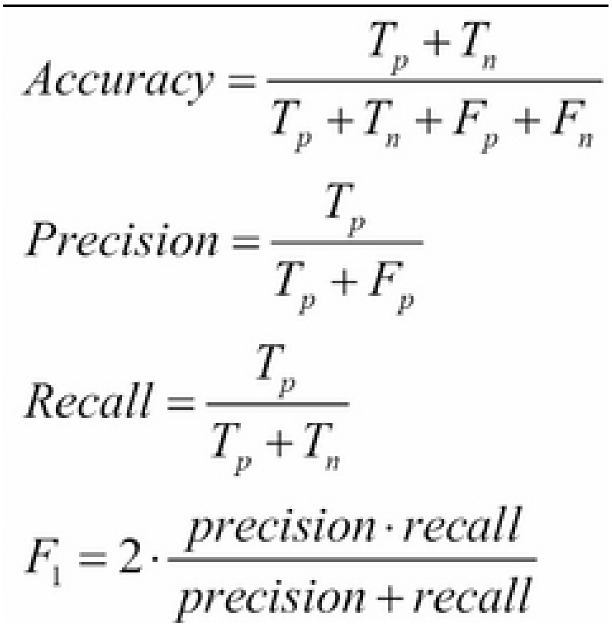


```
 ---Random Forest Model---
Random Forest-Accuracy is 0.98
Random Forest AUC = 0.97
             precision    recall  f1-score   support

          0       0.99      0.98      0.98      1714
          1       0.95      0.96      0.95       536

avg / total       0.98      0.98      0.98      2250
```

Figure 6.2: Result



$$Accuracy = \frac{T_p + T_n}{T_p + T_n + F_p + F_n}$$

$$Precision = \frac{T_p}{T_p + F_p}$$

$$Recall = \frac{T_p}{T_p + T_n}$$

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

Figure 6.3: Formulae

6.4 Summary

Software testing is a process to show the customers the quality of the product.Unit testing is a method used to test individual modules.Integration testing aggregates all modules that are unit tested and tests them using Integration testing methods.System testing basically tests the system to check whether the system meets all the specified requirements.

Chapter 7

User Interface

7.1 Graphical User Interface

A web based user interface was implemented for this project.The Web UI is a HTML-based application used to design and deal with the server apparatus from a remote client.The Website provides a clear description of all the work done in the project.Users will feed the data to the trained model and the results of the prediction are presented on the webpage.

Following are the technologies used for implementing the UI:

- Bootstrap
- HTML & CSS
- Javascript
- Flask

Bootstrap is used for front-end development, it contains HTML and CSS based plan layouts for typography, frames, catches, route and other interface components,as well as Javascript extensions.

Following are the advantages of using Bootstrap:

- It already has predefined design templates and classes,which saves a lot of time.
- All bootstrap components share a consistent design throughout.
- It is easy to use and compatible with all browsers.

Flask is a miniaturized scale web structure written in Python and in light of the Werkzeug toolbox and jinja2 layout engine.Flask is considered and used for the following advantages:

- It's easy to set up
- It's well documented
- It's very simple and minimalistic, and doesn't include anything you won't use
- It's flexible enough that you add extensions,if you need more functionality

Jinja is a layout engine for the python programming dialect and is authorized under a BSD License. It gives Python like articulations while guaranteeing that the layouts are assessed in a sandbox. It is a content based format dialect and along these lines can be utilized to produce any increase and in addition source Code.

As the below figure 7.1 would be the index page of the project and the form input validation has been taken care using Jquery library.Figure 7.2 is showing the valid forms values that are entered.

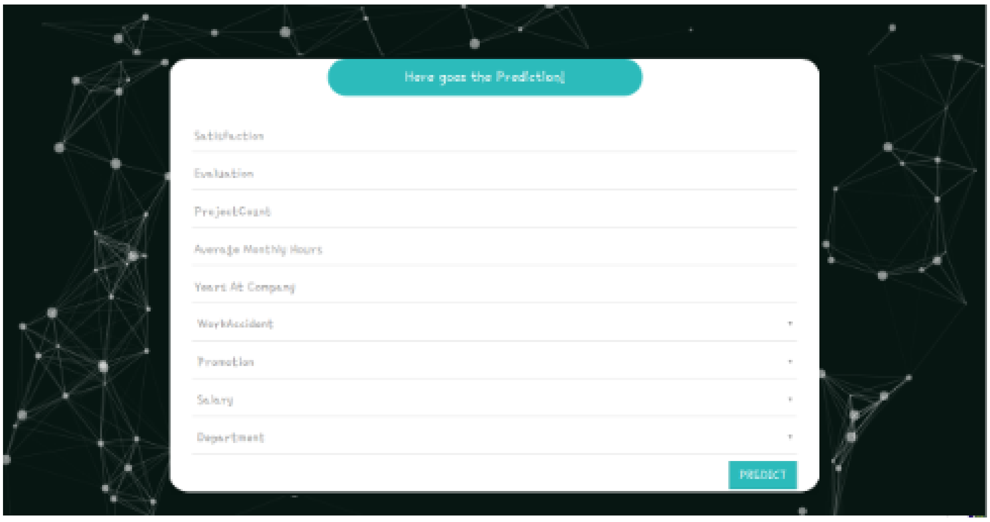
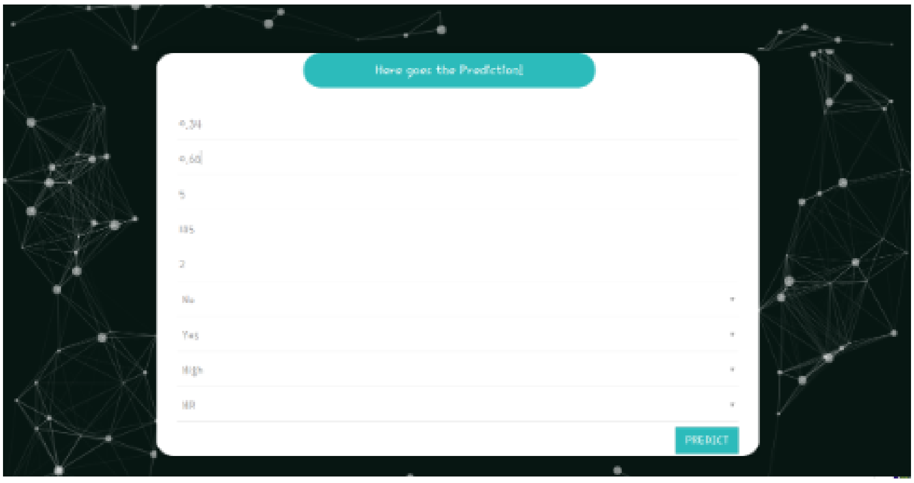


Figure 7.1: Snapshot

Inputting the appropriate and acceptable form values would redirect the page to the below figure 7.3



0.34

0.68

5

185

2

No

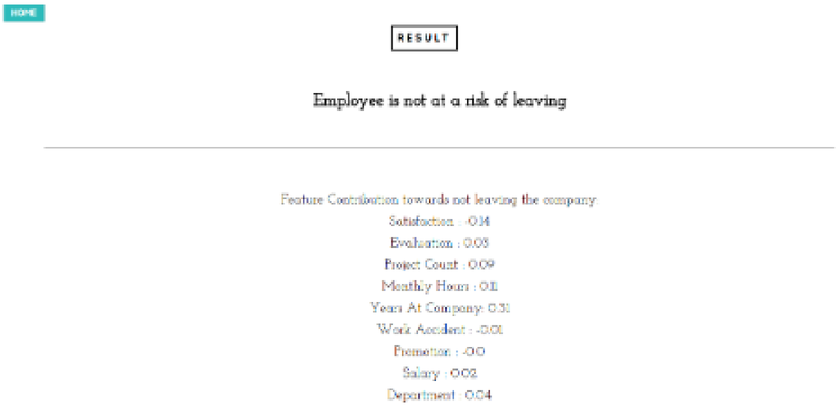
Yes

High

HR

Figure 7.2: Snapshot

Figure 7.3 as shown below include the employee prediction according to the details entered by the user and it also includes the contributions of each feature that leads to that particular prediction this score changes according the details entered by the user.



RESULT

Employee is not at a risk of leaving

Feature Contribution towards not leaving the company

Satisfaction : -0.14

Evaluation : 0.03

Project Count : 0.09

Monthly Hours : 0.1

Years At Company: 0.31

Work Accident : -0.01

Promotion : -0.0

Salary : 0.02

Department : 0.04

Figure 7.3: Snapshot

Chapter 8

Analysis & Conclusion

8.1 Analysis

As a part of the prediction stage in our system, we initially split the dataset obtained from kaggle repository to training and test data following the 0.7:0.3 ratio i.e.,70% of the information was utilized as preparing information and the last 30% was used as test data. In the ensemble classifier we fed the data into the random forest algorithm and modeled the classifier. To this modeled classifier we fed in the test information and the class marks for the test information are anticipated .We obtained an accuracy of 98%.The Summary of our analysis as shown below:

- Representatives by and large left when they are under worked (under 150 hour for each month or 6 hour per month)
- Representatives by and large left when they are overworked(more than 250 hour for every month or 10 hour for each day)
- Employees with either extremely high or low assessment ought to be contemplated for high turnover rate.
- Employees with low to medium pay rates are the people who most probably leave the organization
- Employee fulfillment is the most elevated pointer for worker turnover
- Worker that has the esteem 4 and 5years At Company value ought to be taken in to thought for high turnover rate
- Employee Satisfaction,Years At Company, Evaluation are the greatest factors in deciding the turnover

8.2 Conclusion

This project outlines the different analysis done on the employee dataset and the usage of a random forest model to make predictive insights on the probability of an employee to turnover. This model can be applied throughout the various departments of the company and be used as an aid to help make better decisions in employee retention.The model should be updated periodically and include more additional features for it to make more accurate predictions.

8.3 Limitations

- The results of the study were limited to the area in which research data was collected.
- This research takes into consideration only those predictors of employee turnover that are in the range of our study.

8.4 Future Work

- Future researchers can examine the same correlations by carrying out longitudinal research study i.e, doing research on data which is gathered over a long span of time.
- Future work can also consider including more factors from the database that could have more effect on deciding representative turnover, for example, remove from home,gender,etc.

8.5 Summary

In this project we see the analysis of the project which we created.The analysis is done by taking in the Kaggle data set and splitting it into 70:30 ratio as training and test data respectively. This data is taken and Random Forests algorithm is applied. The thus modeled classifier gives an accuracy of 98%. In the next section precision, recall, F-score, accuracy are shown.

The limitations of the project are mentioned in the next section and future scope shows what development can be made to get better results.

Bibliography

[1] Ladelsky Limor Kessler *, the effect of organization culture on IT employees turnover intention in ISRAEL .*

[2] Zheng WeiBo1*, Sharan Kaur 2and Tao Zhi 3 *, A critical review of employee turnover model (1938-2009) and development in perspective of performance, African Journal of Business Management*. Vol. 4(19), pp. 4146-4158, December Special Review, 2010.

[3] W. Stanley Siebert,Nikolay Zubanov,Arnaud Chevalier,Tarja Viitanen *, Labour Turnover and Labour Productivity in a Retail Organization,* . IZA Discussion Paper No. 2322 September 2006.

[4] Jacob Rubæk Holm *, The effects on performance of voluntary and involuntary labour turnover in an evolutionary signaling model .*