PREDICTING EMPLOYEE ATTRITION USING MACHINE LEARNING

A PROJECT REPORT

Submitted by

ABINAYA.M	**813814104001**
AISHWARYALAKSHMI.S	**813814104003**
AKSSHAYA.K	**813814104005**
GANESH.V	**813814104018**

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

SARANATHAN COLLEGE OF ENGINEERING, TRICHY

ANNA UNIVERSITY: CHENNAI 600 025

APRIL 2018

ANNA UNIVERSITY: CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report "**PREDICTING EMPLOYEE ATTRITION USING MACHINE LEARNING**" is the bonafide work of

ABINAYA.M	**813814104001**
AISHWARYALAKSHMI.S	**813814104003**
AKSSHAYA.K	**813814104005**
GANESH.V	**813814104018**

who carried out the project work under my supervision.

SIGNATURE
Dr. S.A.Sahaaya Arul Mary
HEAD OF THE DEPARTMENT
Computer Science and Engineering,
Saranathan College of Engineering,
Panjappur,
Tiruchirappalli – 620 012

SIGNATURE
Mr. M.Anbazhagan, ME.,
ASSISTANT PROFESSOR
Computer Science and Engineering,
Saranathan College of Engineering,
Panjappur,
Tiruchirappalli – 620 012

VIVA – VOCE EXAMINATION

PREDICTING EMPLOYEE ATTRITION USING MACHINE LEARNING

Submitted by

ABINAYA.M	**813814104001**
AISHWARYALAKSHMI.S	**813814104003**
AKSSHAYA.K	**813814104005**
GANESH.V	**813814104018**

The Viva – Voce Examination of this project work done as a part of B.E. Computer Science and Engineering was held on ___________.

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

Gratitude is not only the greatest of virtues, but the parent of all others. We take this opportunity to thank all the encouragers and supporters of this project.

First and foremost, we thank the Almighty who has been an unfailing source of strength and comfort for showering his blessings throughout this study.

We sincerely thank our beloved Director late **Prof. V.Nagarajan, B.E.,** and **Dr. D.Valavan,** Principal of Saranathan College of Engineering, for providing all the necessary facilities to do our work.

We would like to extend our deep sense of gratitude to **Dr. S.A.Sahaaya Arul Mary,** Head of the department, Computer Science and Engineering, and **Ms. R Thillaikarasi, M.Tech.,** Project Coordinator, for supporting us throughout our venture.

We are obliged to our supervisor **Mr. M.Anbazhagan, M.E.,** Assistant Professor, for being our internal guide and facilitating us with his valuable support and guidance.

We are thankful to all the teaching and supporting staff members of Computer Science department for the help rendered by them in completion of this project. We are also thankful to our parents and friends who have been encouraging and morally supportive.

ABSTRACT

Bill Gates was once quoted as saying,

“You take away our top 20 employees and we [Microsoft] become a mediocre company”.

His statement cuts to the core of a major problem: Employee Attrition. It is a major cost to an organization. Some costs are tangible such as training expenses and the time it takes from when an employee starts to when they become a productive member. However, the most important costs are intangible, such as new product ideas, great project management, or customer relationships.

To reduce the cost of attrition, organizations need to ensure that employees’ aspirations are met. Employee attrition control is critical to the long term health and success of any organization. An organization is only as good as its employees, and these people are the true source of its competitive advantage.

Accurate predictions enable organizations to take action for the retention of employees. This project aims to use different supervised classifiers to make predictions, and chooses the most accurate one.

TABLE OF CONTENTS

APPENDICES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ABBREVIATION	EXPANSION
LR	Logistic Regression
kNN	k Nearest Neighbor
RF	Random Forest
SKB	SelectKBest
RFE	Recursive Feature Elimination
ANOVA	Analysis of Variance
SPSS	Statistical Package for the Social Sciences
SNR	Signal to Noise Ratio
ROC-AUC	Area under Receiver Operating Characteristic Curve
SAS	Statistical Analysis Software

CHAPTER 1

INTRODUCTION

1.1. MACHINE LEARNING

Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. Machine learning focuses on the development of computer programs that can access data and use it learn for themselves.

The process of learning begins with observations or data in order to look for patterns in data and make better decisions in the future based on the examples that we provide. The primary aim is to allow the computers learn automatically without human intervention or assistance and adjust actions accordingly.

Machine learning enables analysis of massive quantities of data. While it generally delivers faster, more accurate results in order to identify profitable opportunities or dangerous risks, it may also require additional time and resources to train it properly. Combining machine learning with AI and cognitive technologies can make it even more effective in processing large volumes of information.

Within the field of data analytics, machine learning is a method used to devise complex models and algorithms that lend themselves to prediction. These analytical models produce reliable, repeatable decisions and results and uncover hidden insights through learning from historical relationships and trends in the data.

1.1.1. Supervised Learning

Supervised learning is the machine learning task of learning a function that maps an input to an output based on example input-output pairs. Each example is a pair consisting of an input object (typically a vector) and a desired output value (also called the supervisory signal).

A supervised learning algorithm analyzes the training data and produces an inferred function, which can be used for mapping new examples. An optimal scenario will allow for the algorithm to correctly determine the class labels for unseen instances. This requires the learning algorithm to generalize from the training data to unseen situations in a "reasonable" way.

It is called supervised learning because the process of an algorithm learning from the training dataset can be thought of as a teacher supervising the learning process. Learning stops when the algorithm achieves an acceptable level of performance.

Supervised learning problems can be further grouped into regression and classification problems.

- **Classification**: A classification problem is when the output variable is a category, such as “red” or “blue” or “disease” and “no disease”. It approximates a mapping function (f) from input variables (X) to discrete output variables (y).
- **Regression**: A regression problem is when the output variable is a real value, such as “dollars” or “weight. It approximates a mapping function (f) from input variables (X) to a continuous output variable (y).

1.1.2. Unsupervised Learning

Unsupervised machine learning is the machine learning task of inferring a function to describe hidden structure from "unlabeled" data (a classification or categorization is not included in the observations). We only have input data and no corresponding output variables. The goal for unsupervised learning is to model the underlying structure or distribution in the data in order to learn more about the data.

Since the examples given to the learner are unlabeled, there is no evaluation of the accuracy of the structure that is output by the relevant algorithm - which is one way of distinguishing unsupervised learning from supervised learning.

These are called unsupervised learning because unlike supervised learning above there is no correct answers and there is no teacher. Algorithms are left to their own devises to discover and present the interesting structure in the data.

Unsupervised learning problems can be further grouped into clustering and association problems.

- **Clustering**: In clustering, the inherent groupings in the data are discovered. It groups a set of objects in such a manner that objects in the same group are more similar than to those object belonging to other groups.
- **Association**: Rules that describe large portions of the data, such as people that buy X also tend to buy Y, are discovered. Association is about finding associations amongst items within large commercial databases.

1.1.3. Semi Supervised Learning

Semi-supervised learning is a class of supervised learning tasks and techniques that also make use of unlabeled data for training – typically a small amount of labeled data with a large amount of unlabeled data. Semi-supervised learning falls between unsupervised learning (without any labeled training data) and supervised learning (with completely labeled training data).

Many machine-learning researchers have found that unlabeled data, when used in conjunction with a small amount of labeled data, can produce considerable improvement in learning accuracy. The acquisition of labeled data for a learning problem often requires a skilled human or a physical experiment. The cost associated with the labeling process thus may render a fully labeled training set infeasible, whereas acquisition of unlabeled data is relatively inexpensive. In such situations, semi-supervised learning can be of great practical value.

We can use unsupervised learning techniques to discover and learn the structure in the input variables. We can also use supervised learning techniques to make best guess predictions for the unlabeled data, feed that data back into the supervised learning algorithm as training data and use the model to make predictions on new unseen data.

In comparison with a supervised algorithm that uses only labeled data, one hope to have a more accurate prediction by taking into account the unlabeled points. For semi-supervised learning to work, certain assumptions will have to hold.

1.2. EMPLOYEE ATTRITION - AN OVERVIEW

Human resource is the most important asset for a company to be competitive. Thanks to liberalization on the labor market, it has become possible for an employee to leave his job. However, having excess employees leave their jobs will influence the morale of the companies.

The loss of good employees can diminish a company’s competitive advantage and furthermore lead to a reduction in output and quality. High employee attrition has a significant negative effect on an organization by virtue of lost productivity, increased training and recruitment costs.

Employees voluntarily leave an organization for various reasons, such as new opportunities, limited or no professional growth in current position, unhappiness with compensation, personal reasons, etc. By taking proactive action to retain its top employees, a company can thus reap substantial benefits, thereby increasing its top and bottom line.

1.3. METHODS

This project discusses different classification algorithms of supervised learning and feature selection algorithms. This section gives a summary of each of these machine learning algorithms.

1.3.1. Logistic Regression

Logistic regression is a statistical method for evaluating a dataset which consists of one or more independent variables that determine an outcome. The outcome is measured with a variable which takes on one of only two possible outcomes. The goal of logistic regression is to find the best fitting model that describes the relationship between a set of independent (predictor or explanatory) variables and the dichotomous characteristic of interest (dependent variable = response or outcome variable). Logistic regression generates the coefficients of a formula that predicts a logit transformation of the probability of the presence of the characteristic of interest. The logit transformation is defined as the logged odds:

$$\frac{\mathrm{p}}{1-\mathrm{p}} = \frac{\text{probability of presence of characteristic}}{\text{probability of absence of characteristic}}$$

where p is the probability of presence of the characteristic of interest.

And logit(p) = $\ln\left(\frac{\mathrm{p}}{1-\mathrm{p}}\right)$

Logistic regression can be binomial, ordinal or multinomial. Binomial logistic regression deals with situations in which the observed outcome for a dependent variable can have only two possible types, "0" and "1". Multinomial logistic regression deals with situations where the outcome can have three or more possible types that are not ordered. Ordinal logistic regression deals with dependent variables that are ordered.

1.3.2. k-Nearest Neighbors

K-nearest neighbors algorithm is a non-parametric method used for classification, where the input consists of the k closest training examples in the feature space and the output is a class membership. An object is classified by a majority vote of its neighbors, with the object being assigned to the class most common among its k nearest neighbors measured by a distance function (k is a positive integer, typically small). If k = 1, then the object is simply assigned to the class of that single nearest neighbor. The distance function used is Hamming distance, given by:

$$D_H = \sum_{i=1}^{k} |x_i - y_i|$$

k can be kept as an odd number so that a clear majority can be calculated in the case where only two groups are possible. With increasing K, we get smoother, more defined boundaries across different classifications. Also, the accuracy of the classifier increases as we increase the number of data points in the training set.

kNN is a type of instance-based learning, or lazy learning, where the function is only approximated locally and all computation is deferred until classification. A useful technique is used to assign weight to the contributions of the neighbors, so that the nearer neighbors contribute more to the average than the more distant ones.

For example, a common weighting scheme consists in giving each neighbor a weight of 1/d, where d is the distance to the neighbor. The neighbors are taken from a set of objects for which the class is known. This can be thought of as the training set for the algorithm, though no explicit training step is required.

1.3.3. Random Forest

Random forest is an ensemble learning method that gives an improved performance using divide and conquer technique. The main principle behind ensemble methods is that a group of "weak learners" can come together to form a "strong learner".

As the name suggest, this algorithm creates the forest with a number of trees. In general, the more trees in the forest the more robust the forest looks like. In the same way in the random forest classifier, the higher the number of trees in the forest gives the high accuracy results. It operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes of the individual trees. When the training set for the current tree is drawn by sampling with replacement, about one-third of the cases are left out of the sample. This oob (out-of-bag) data is used to get a running unbiased estimate of the classification error as trees are added to the forest.

After each tree is built, all of the data are run down the tree, and proximities are computed for each pair of cases. If two cases occupy the same terminal node, their proximity is increased by one. At the end of the run, the proximities are normalized by dividing by the number of trees. Proximities are used in replacing missing data, locating outliers, and producing illuminating low-dimensional views of the data.

Random forest gives much more accurate predictions. These cases generally have high number of predictive variables and huge sample size. This is because it captures the variance of several input variables at the same time and enables high number of observations to participate in the prediction.

1.3.4. SelectKBest

SelectKBest is a feature selection algorithm that scores the features of a dataset using a score function and then removes all but the k-highest scoring features. It takes as a parameter the score function, which must be applicable to a pair of data from the training set (X) and test set (y). The score function returns an array of scores, and SelectKBest simply retains the first k features of training set with the highest scores.

The score function used is Chi-Square (chi2). It measures how well the observed distribution of data fits with the distribution that is expected if the variables are independent. The test compares the observed data to a model that distributes the data according to the expectation that the variables are independent.

A Chi-square test is designed to analyze categorical data. It can tell information based on how the data is divided. However, it cannot tell whether the constructed categories are meaningful. The Chi-square test is only meant to test the probability of independence of a distribution of data, and not tell details about the relationship between them.

SelectKBest computes the chi2 statistic between each feature of X and y (assumed to be class labels). A small value will mean the feature is independent of y. A large value will mean the feature is non-randomly related to y, and so likely to provide important information. The formula for the chi2 statistic is given by:

$$\chi^2 = = \sum_{i=1}^{n} \frac{(O_i - E_i)^2}{E_i}$$

where O_i is the observed value and E_i is the expected value.

1.3.5. Recursive Feature Elimination

Using an external estimator that assigns weights to features (e.g., the coefficients of a linear model), Recursive Feature Elimination selects features by considering smaller sets of features. The estimator used here is the linear model of Logistic Regression.

First, the estimator is trained on the initial set of features and the importance of each feature is obtained either through a coef_ attribute or through a feature_importances_ attribute of the estimator. Then the least important features are pruned from current set of features. Until the desired number of features is selected, this procedure is recursively repeated on the pruned set. The stability of RFE depends heavily on the type of model that is used for feature ranking at each iteration.

SelectKBest	**RFE**
Age	Department
Daily Rate	Environment Satisfaction
Distance From Home	Gender
Monthly Income	Job Involvement
Monthly Rate	Job Level
OverTime	Job Satisfaction
Total Working Years	Marital Status
Years At Company	Overtime
Years In Current Role	Stock Option Level
Years With Current Manager	Work Life Balance

Comparison of features selected by SelectKBest and Recursive Feature Elimination

1.3.6. XGBoost

XGBoost stands for eXtreme Gradient Boosting. This algorithm goes by lots of different names such as gradient boosting, multiple additive regression trees, stochastic gradient boosting or gradient boosting machines.

Boosting is an ensemble technique where new models are added to correct the errors made by existing models. Models are added sequentially until no further improvements can be made. Gradient boosting is an approach where new models are created that predict the residuals or errors of prior models and then added together to make the final prediction. It is called gradient boosting because it uses a gradient descent algorithm to minimize the loss when adding new models.

XGBoost is an implementation of gradient boosting machines, engineered for efficiency of compute time and memory resources. The features of XGBoost include regularization, parallel processing, high flexibility, handling missing values, tree pruning, built-in cross validation, etc.

Every parameter in XGBoost has a significant role to play in the model's performance. Its parameters can be divided into three categories:

- General Parameters - Controls the booster type in the model which eventually drives overall functioning
- Booster Parameters - Controls the performance of the selected booster
- Learning Task Parameters - Sets and evaluates the learning process of the booster from the given data

CHAPTER 2

LITERATURE SURVEY

2.1. APPLICATION OF XGBOOST

The approach is to explore the application of extreme gradient boosting as an improvement on the traditional algorithms, specifically in its ability to generalize on noise-ridden data which is prevalent in this domain. This is done by using data from the HRIS of a global retailer and treating the attrition problem as a classification task and modeling it using supervised techniques. The conclusion is reached by contrasting the superior accuracy of the XGBoost classifier against other techniques and explaining the reason for its superior performance. [1]

2.2. ANALYSIS USING CHI2 AND CORRELATION COEFFICIENT

The purpose is to assess the causes of attrition and its remedies. The main aim is to ensure that the required data are collected objectively and accurately. Data regarding the causes of attrition and its remedies was collected directly by interacting with the employees of the organization by a structured questionnaire. The secondary data was collected from the magazines, journals and the internet. Data regarding perception towards employee attrition had been collected from 100 employees working in different software Industries.

Puposive sampling method and collected quantitative data was used, the data collected from primary source were analyzed by using simple statistical tools like tabulation, percentage chi squared test & correlation coefficient etc.

The chi-squared distribution is used in the chi squared test for goodness of fit of an observed distribution to a theoretical one, the independence of two criteria of classification of qualitative data, and in confidence interval estimation for a population standard deviation of a normal distribution from a sample standard deviation.

A correlation coefficient is a numerical measure of some type of correlation, meaning a statistical relationship between two variables. The variables may be two columns of a given data set of observations or two components of a multivariate random variable with a known distribution. [2]

2.3. RANDOM SAMPLING TECHNIQUE AND STATISTICAL MEASUREMENTS

Primary research consists of a collection of original primary data collected by the researcher. It is often undertaken after the researcher has gained some insight into the issue by reviewing secondary research or by analyzing previously collected primary data. It can be accomplished through various methods including questionnaires, telephone interviews in market research, etc and direct observations in the physical sciences, amongst others.

A questionnaire is a research instrument consisting of a series of questions and other prompts for the purpose of gathering information from respondents. Although they are often designed for statistical analysis of the responses, it can contain both open ended and closed ended questions. First open ended questions are formulated, samples are collected and then closed ended questions are formulated from those.

A simple random sampling technique is used here, with the principle that every object has the same probability of being chosen. A simple random sample is a subset of individuals (a sample) chosen from a larger set (a population). Each individual is chosen randomly and entirely by chance, such that each individual has the same probability of being chosen at any stage during the sampling process, and each subset of k individuals has the same probability of being chosen for the sample as any other subset of k individuals.

The statistical measurements used are

- Descriptive statistics: It aims to summarize a sample, rather than use the data to learn about the population that the sample of data is thought to represent. This generally means that descriptive statistics, unlike inferential statistics, are not developed on the basis of probability theory. Even when a data analysis draws its main conclusions using inferential statistics, descriptive statistics are generally also presented.
- Anova: It is a collection of statistical models used to analyze the differences between group means and their associated procedures (such as "variation" among and between groups). In its simplest form, Anova provides a statistical test of whether or not the means of several groups are equal, and therefore generalizes t-test to more than two groups. Anova is useful in comparing (testing) three or more means (groups or variables) for statistical significance.
- Factor Analysis: It is a statistical method used to describe variability among observed, correlated variables in terms of a potentially lower number of unobserved variables called factors. The observed variables are modeled as linear combinations of the potential factors, plus "error" terms. The observed variables are modeled as linear combinations of the potential factors, plus "error" terms. The information gained about the interdependencies between observed variables can be used later to reduce the set of variables in a dataset [3]

2.4. FRAMEWORK FOR PROACTIVE RETENTION

An analytics based framework is described for tackling voluntary employee attrition using a one-time proactive salary increase as the retention lever. The approach involves the identification of individual employees at a high risk of voluntary resignation, and optimally choosing that set of such employees for retention action for whom the total cost of retention action (by virtue of a salary increase) is the least compared to the total cost of replacing them if they depart.

While the framework described is focused solely on salary increases as the sole retention lever, the framework can be easily extended to include other retention actions as long as the cost of such actions can be quantified. The proposed framework consists of a number of steps:

- Understanding reasons for attrition and identifying potential attriters: First, a company has to understand the reasons for voluntary attrition. Mining historical employee data can help build models to understand factors that affect voluntary attrition as well as identify employees likely to attrit in the future based on such factors. A very important consideration in such a mining exercise is that the models must be easily interpretable and understandable so that reasons for identifying employees as potential attriters can be explained and supported by fact. By carefully constructing features around such actions and mining historical data to build attrition models, it is possible to understand how such actions may affect attrition.
- Understanding the cost of attrition: The salary premium for an employee is an important factor in determining whether it is financially viable for a company to try to retain an employee or not by way of

financial retention actions. The higher the salary premium, the more expensive it is to replace the employee and probably cheaper to try to retain instead. In addition to salary premiums, another potential cost which may be incurred by an employee if a valued employee departs is the cost of hiring a replacement. Based on the tightness of the labor market, the kinds of skills needed and the availability of such skills, economic conditions, etc., these costs may vary widely.

While salary premiums and hiring costs are tangible costs (the company has to spend a measurable amount of money), an intangible but significant cost of employee attrition is productivity loss. This is especially true in services where deals may fall through or contractual obligations may be missed or delayed due to unexpected departures. Moreover, new hires often have little or no productivity for a significant amount of time after joining as they get up to speed in terms of various products and processes of their new employer.

- Determining optimal compensation investments: In addition to costs, a company also has to decide appropriate investment levels (what sizes of raises to give and to whom) and the population to target. This includes deciding on whether to target employees already being paid above or near market (or those who would reach that level after getting a raise) if it will be beneficial to try to retain such employees, or to focus on those employees who are significantly underpaid and where the impact of a salary raise will perhaps be most strongly felt. Similarly, a decision has to be made regarding the skills, performance levels, job roles etc. that are most critical for the company so that retention actions can be focused in that direction.

- Choosing employees for proactive retention action: Once attrition reasons have been understood and employees at risk of voluntary attrition identified, costs have been quantified and compensation decisions made, a subset of those employees have to be chosen for retention action such that the maximum possible savings are generated subject to the constraints imposed by financial limits. [4]

2.5. T-TEST AND DUNCAN’S POST HOC TEST

The purpose is to determine what and how job-related and demographic variables are associated with employee satisfaction of the BPO employees. In this approach, data collected from 500 middle level BPO employees is analyzed. T-tests and Duncan’s post hoc tests were done to compare the various dimensions of employee satisfaction across selected demographic variables such as gender, marital status, education, age and tenure.

Correlation was done to find out the relationship between employee satisfaction and various job characteristics as well as demographic variables and finally, regression was done to find out the actual determinants of employee satisfaction.

The t-tests are a type of hypothesis test that allows you to compare means. They are called t-tests because each t-test boils your sample data down to one number, the t-value. The t-value is a ratio between the difference between two groups and the difference within the groups. The larger the t-value, the more difference there is between groups. The smaller the t-value, the more similarity there is between groups. The ANOVA won’t pinpoint the pairs of means that are different. Duncan’s Post Hoc Test will identify the pairs of means (from at least three) that differ. [5]

2.6. USING SPSS

The methodology deployed consists of primary research, with insights being captured through questionnaires and face -to- face discussions with senior management of participant organizations. Inputs from potential customers to this industry have also been collated, with respect to their key concerns while considering outsourcing to India. The analysis was further supplemented by PwC knowledge-bases and published data to validate trends and best practices, emerging from primary sources. SPSS was religiously used for the statistical analyses.

- Primary Data Analysis: A questionnaire was designed to tap the factors responsible for attrition, the factors that are expected to be present in a specific job for retention. The instrument was divided into 4 parts. Part I gathered information about the personal profile of the respondents. Part II consisted of questions about their reasons for change or probable change in their jobs. Part III aimed at knowing what accorind to the respondents is important for their sustenance in an organization. Part IV was about their overall perception of the work which included their level of satisfaction, level of motivation, level of involvement and level of life interest and work compatibility.
- Secondary Data Analysis: Major causal factors identified for high attrition in Indian BPO industry were based on qualitative research using secondary data. In order to gain a deeper understanding about the phenomenon of high attrition, and identification of the factors behind it, a lot of literature on BPO was studied in detail. These were compared with causal factors for attrition identified through personal interview with a number of BPO employees. [6]

2.7. SAMPLING AND ANALYSIS

Primary data was gathered by filling up of the questionnaires by the respondents. Survey research method has been adopted to collect the data from current employees. Mail questionnaire method has been adopted to collect the data from ex-employees. Two separate well-structured questionnaires were designed for the collection of primary data from both ex-employees and current employees. Convenience sampling method has been adopted for the collection of data from the Ex-Employees. Stratified Random sampling divides the population into mutually exclusive groups and random samples are drawn from each group.

Convenience sampling is a non-probability sampling technique where subjects are selected because of their convenient accessibility and proximity to the researcher. The subjects are selected because they are easy to recruit for the study and the researcher does not consider selecting subjects that are representative of the entire population. This method allows the researcher to obtain basic data and trends regarding his study without the complications of using a randomized sample. This technique is also useful in documenting that a particular quality of a substance or phenomenon occurs within a given sample.

Stratified random sampling is a method of sampling that involves the division of a population into homogeneous groups known as strata. The strata are formed based on members' shared attributes or characteristics. A random sample from each stratum is taken in a number proportional to the stratum's size when compared to the population. These subsets of the strata are then pooled to form a random sample. [7]

2.8. TAGUCHI METHOD & NEAREST NEIGHBOR CLASSIFICATION RULES

As a well-known robust experimental design approach, the Taguchi method uses two principal tools, the orthogonal array and the signal to noise ratio, for the purpose of evaluation and improvement. Consider that a specific object domain contains q design parameters (or factors). Orthogonal arrays are primarily used to reduce the experimental efforts regarding these q different design factors.

An orthogonal array can be viewed as a fractional factorial matrix that provides a systematic and balanced comparison of different levels of each design factor and interactions among all design factors. In this two-dimensional matrix, each column specifies a particular design factor and each row represents a trial with a specific combination of different levels regarding all design factors. In the proposed method, the well-known two-level orthogonal array is adopted for feature subset selection. The Taguchi method offers many advantages for robust experimental design. First, the number of experimental runs can be substantially reduced. Meanwhile, the significance of each design parameter regarding a particular object domain can be analyzed precisely.

The nearest neighborhood rule and leave-one-out method see each sample as a tested sample and the others as the relative samples. Therefore, the nearest neighborhood rule would run m times (sample size). Subsequently, the average accuracy should be calculated to evaluate the efficiency at the level of each feature and the best SNR value are related to the efficiency. Thus, the higher efficiency will be used to calculate SNR value. [8]

2.9. APPLICATION OF kNN

The KNN algorithm classifies new data based on the class of the k nearest neighbors. The distance from neighbors can be calculated using various distance metrics, such as Euclidean distance, Manhattan distance, Minkowski distance, etc. The class of the new data may be decided by majority vote or by an inverse proportion to the distance computed. KNN is a non-generalizing method, since the algorithm keeps all of its training data in memory, possibly transformed into a fast indexing structure such as a ball tree.

The choice of model validation technique is the area under the ROC. The AUC of a classifier is equal to the probability that the classifier will rank a randomly chosen positive example higher than a randomly chosen negative value. When used with its optimal configuration, it is a robust method that delivers accurate results in spite of the noise in the dataset, which is a major challenge for machine learning algorithms.

The ROC curve shows the general predictiveness of the classifier. It measures the probability that the classifier ranks a randomly chosen positive instance higher than a randomly chosen negative instance. The ROC curve is created by plotting the true positive rate against the false positive rate at various threshold settings. The ROC curve’s strength and weakness is the observation that classifier score is itself a nuisance parameter.

The accuracy and F1 scores of the classifier is also used to compare the results of the models. The accuracy score computes subset accuracy i.e., the set of labels predicted for a sample must exactly match the corresponding set of labels in training data. The F1 score is the harmonic average of the precision and recall. These two are important because they clearly show how suitable the model is for use in an application. [9]

2.10. USING LOGISTIC REGRESSION

This approach uses two modeling techniques to predict overall attrition risk. Behavioral risk modeling is based on the online survey data and attrition risk modeling is based on the demographic data. Logistic regression modeling is based on the attrition data only and rest of the dataset is used for other qualitative analysis. It uses stratified random sampling technique and significant testing tool to select the sample.

Stratified random sampling is relevant because sub-groups within the population are heterogeneous. Through stratification, grouping of members of the population is done to get them into relatively homogeneous subgroups before sampling. The SAS is used to run the logistic regression model. The analysis is done to find out the probability of occurrence of an event (probability of leaving or not leaving the organization) by fitting the data into a logistic curve.

At the outset, Logistic Regression model included all demographic variables and subsequently eliminated insignificant variables through an iterative process. Wald Chi-Square test and Maximum Likelihood Estimates were used to identify coefficients for significance for inclusion or elimination from the model. The fitment of the model was tested after each round of elimination. The analysis was concluded when no more variables needed to be eliminated from the model and the model converged. [10]

CHAPTER 3

PROPOSED SYSTEM

A fictional dataset created by IBM data scientists is used for analysis. It has 35 features and 1470 observations. There are two class labels for the feature “Attrition” – Yes and No. The dataset includes various important features such as Age, DailyRate, DistanceFromHome, Overtime, EnvironmentSatisfaction, JobLevel, JobSatisfaction, MonthlyIncome, WorkLifeBalance, etc. There are 34 independent features and 1 dependent feature (Attrition). Out of the thirty four features, six are categorical and the remaining are numeric.

3.1. DATA PREPROCESSING

The missing data from the dataset is handled by Interpolation, which is a mathematical technique to estimate the missing values in some interval, when a number of observed values are available within that interval.

All the categorical values in each column are converted to numerical values using LabelEncoder. It is used to assign ordinal levels to categorical data. It encodes the labels with values between 0 and (number of classes - 1). For example, the feature MaritalStatus with labels “Divorced”, “Married” and “Single” is encoded as 0, 1 and 2 respectively.

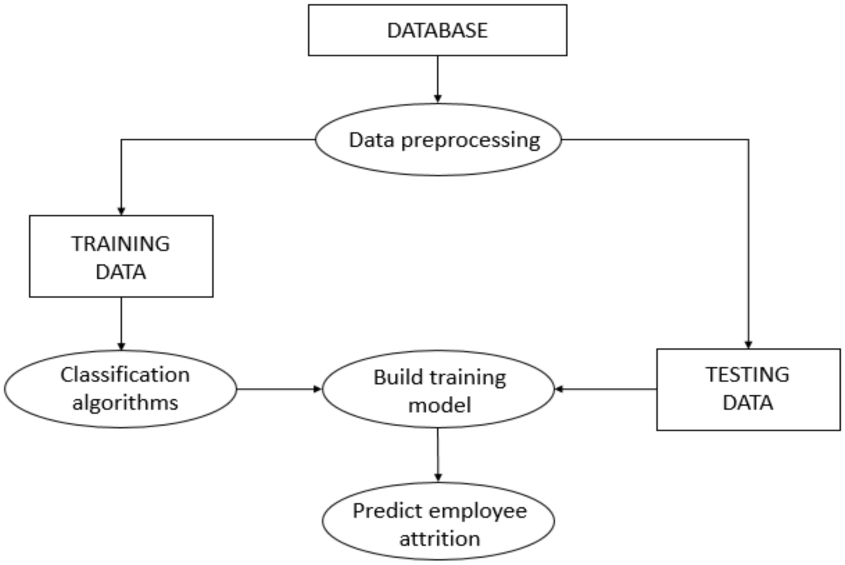
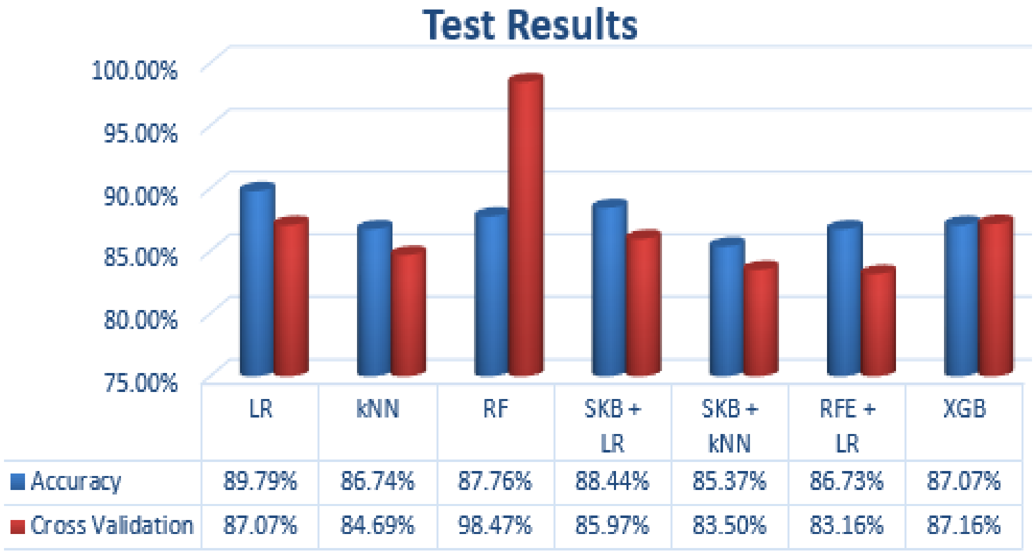


Figure 2.1 – Flow diagram

3.2. CLASSIFICATION

The dataset is split into training and test data in the ratio 80:20. Different classification algorithms are used to train the training set. The trained model is used to make predictions on the test data, and is also applied on the training data for cross validation. The model with the highest accuracy is used to make the final prediction.



	LR	kNN	RF	SKB + LR	SKB + kNN	RFE + LR	XGB
🟦 Accuracy	89.79%	86.74%	87.76%	88.44%	85.37%	86.73%	87.07%
🟥 Cross Validation	87.07%	84.69%	98.47%	85.97%	83.50%	83.16%	87.16%

Figure 2.2 – Test Results

CHAPTER 4

SOFTWARE REQUIREMENTS SPECIFICATION

4.1. INTRODUCTION

4.1.1. Purpose

Employee turnover reflects an organization’s internal strengths and weaknesses. Organizations face difficulties in retaining the employees as well as attracting potential employees. All this has a significant impact on the strength of a company in managing their business in a competitive environment. Hence, it has become critical for the companies to satisfy their employees in order to retain them.

4.1.2. Project Scope

This project can help an organization identify the employees who are vulnerable to quitting their jobs.

4.2. SYSTEM FEATURES

4.2.1. Use Case

Name: Predict attrition

Input: Employee dataset

Output: Attrition rate

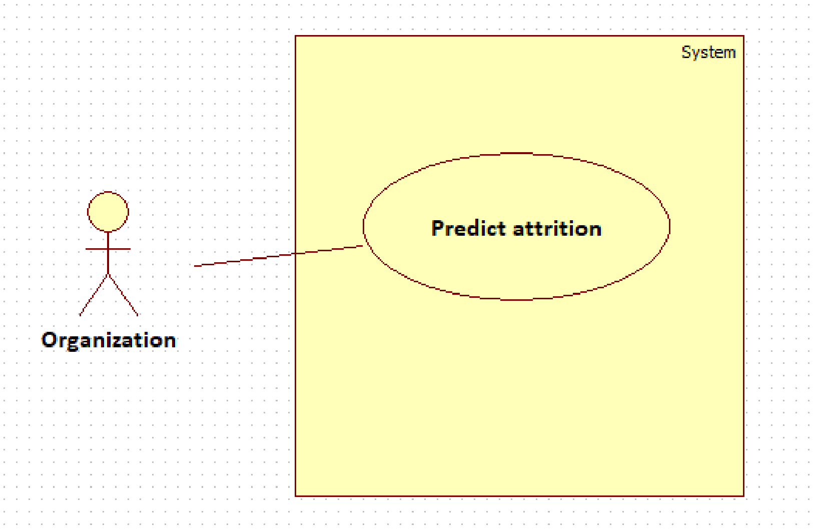


Figure 3.1 – Use case diagram

Pre-condition: An employee dataset with necessary attributes

Steps:

1. Preprocessing:

   The missing data in the dataset is fixed.

   The non-numeric attributes are encoded to contain numeric values.

2. Classification:

   Various classification algorithms of supervised learning are used to find the number of employees who would leave an organization.

Post-condition: The employee attrition rate for the given dataset is displayed

4.3. SYSTEM REQUIREMENTS

4.3.1. Software Requirements

- IDE: Jupyter Notebook (Anaconda)
- PACKAGES: NumPy, Pandas, Scikit-learn, XGBoost (Python)

4.4. SOFTWARE DESCRIPTION

4.4.1. 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. Its uses include: data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much more.

It excels in a form of programming called “literate programming”. Literate programming is a software development style pioneered by Stanford computer scientist, Donald Knuth. This type of programming emphasizes a prose first approach where exposition with human-friendly text is punctuated with code blocks. It excels at demonstration, research, and teaching objectives especially for science.

The Jupyter Notebooks web-based technology were created because of the clear advantages of literate programming and improved web browser technologies (e.g., HTML5). They are a series of “cells” containing executable code, or markdown, the popular HTML markup language for prose descriptions. Moreover, because these notebook environments are for writing and developing code, they offer many niceties available in typical Interactive Development Environments (IDEs) such as code completion and easy access to help.

Anaconda distribution is used to install Jupyter. It is a free open source distribution for large-scale data processing, predictive analytics, and scientific computing, that aims to simplify package management and deployment. Package versions are managed by the package management system conda.

4.4.2. NumPy

NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these. It is the fundamental package for scientific computing with Python. It contains among other things:

- a powerful N-dimensional array object
- sophisticated (broadcasting) functions
- tools for integrating C/C++ and Fortran code
- useful linear algebra, Fourier transform, and random number capabilities

Besides its obvious scientific uses, NumPy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined. This allows NumPy to seamlessly and speedily integrate with a wide variety of databases.

NumPy provides multidimensional arrays, functions and operators that operate efficiently on arrays. The core functionality of NumPy is its "ndarray", for n-dimensional array, data structure. These arrays are strided views on memory. In contrast to Python's built-in list data structure (which, despite the name, is a dynamic array), these arrays are homogeneously typed: all elements of a single array must be of the same type.

4.4.3. Pandas

Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. It offers data structures and operations for manipulating numerical tables and time series. Prior to Pandas, Python was majorly used for data munging and preparation. It had very little contribution towards data analysis. Pandas solved this problem. Using Pandas, we can accomplish five typical steps in the processing and analysis of data, regardless of the origin of data — load, prepare, manipulate, model, and analyze. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc. Its key features include:

- Easy handling of missing data (represented as NaN) in floating point as well as non-floating point data
- Columns can be inserted and deleted from DataFrame and higher dimensional objects
- The user can simply ignore the labels and let Series, DataFrame, etc. automatically align the data for you in computations
- Powerful, flexible group-by functionality to perform split-apply-combine operations on data sets

4.4.4. Scikit-learn

Scikit-learn is a free software machine learning library for the Python programming language. It consists of simple and efficient tools for data mining and analysis, accessible to everybody and reusable in various contexts.

It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy. Scikit-learn is largely written in Python, with some core algorithms written in Cython to achieve performance. Some popular groups of models provided by scikit-learn include:

- Clustering: for grouping unlabeled data such as KMeans.

- Cross Validation: for estimating the performance of supervised models on unseen data.
- Dimensionality Reduction: for reducing the number of attributes in data for summarization, visualization and feature selection such as Principal component analysis.
- Ensemble methods: for combining the predictions of multiple supervised models.
- Feature selection: for identifying meaningful attributes from which to create supervised models.
- Parameter Tuning: for getting the most out of supervised models
- Supervised Models: a vast array not limited to generalized linear models, discriminate analysis, naive bayes, lazy methods, neural networks, support vector machines and decision trees.

4.4.5. XGBoost (Python)

XGBoost is an open-source software library which provides the gradient boosting framework for C++, Java, Python, R, etc. It aims to provide a scalable, portable and distributed Gradient Boosting library. It can be downloaded and installed on our machine, then accessed from a variety of interfaces. Specifically, XGBoost supports the following main interfaces:

- Command Line Interface (CLI).
- C++ (the language in which the library is written).

- Python interface as well as a model in scikit-learn.
- Java and JVM languages like Scala and platforms like Hadoop.

The library is laser focused on computational speed and model performance, and offers a number of advanced features.

- Parallelization of tree construction using all of our CPU cores during training.
- Distributed Computing for training very large models using a cluster of machines.
- Out-of-Core Computing for very large datasets that don’t fit into memory.
- Cache Optimization of data structures and algorithm to make best use of hardware.

CHAPTER 5

SOFTWARE TESTING

Software Testing is the process of evaluating a system or its component(s) with the intent to find whether it satisfies the specified requirements or not. It can also be stated as the process of executing a program or application with the intent of finding the software errors. The job of testing is an iterative process as when one bug is fixed, it can illuminate other, deeper bugs, or can even create new ones. Software testing can provide objective, independent information about the quality of software and risk of its failure to users or sponsors.

5.1. TESTING METHODS

5.1.1. White Box Testing

This tests the internal structures or workings of a program, as opposed to the functionality exposed to the end-user. In white-box testing, an internal perspective of the system, as well as programming skills, are used to design test cases. The tester chooses inputs to exercise paths through the code and determine the appropriate outputs. While this method can be applied at the unit, integration and system levels of the software testing process, it is usually done at the unit level.

5.1.2. Black Box Testing

Black-box testing treats the software as a "black box", examining functionality without any knowledge of internal implementation, without seeing the source code. The testers are only aware of what the software is supposed to do, not how it does it. This method can be applied to unit, integration, system and acceptance levels of software testing.

5.2. TESTING LEVELS

The testing levels can typically be broken down between functional and non-functional testing.

Functional testing involves testing the application against the business requirements. It incorporates all test types designed to guarantee each part of a piece of software behaves as expected by using uses cases provided by the design team or business analyst. These testing methods are usually conducted in order and include:

5.2.1. Unit Testing

Unit testing is the first level of testing and is often performed by the developers themselves. It is the process of ensuring individual components of a piece of software at the code level are functional and work as they were designed to. Developers in a test-driven environment will typically write and run the tests prior to the software or feature being passed over to the test team. Unit testing can be conducted manually, but automating the process will speed up delivery cycles and expand test coverage.

5.2.2. Integration Testing

After each unit is thoroughly tested, it is integrated with other units to create modules or components that are designed to perform specific tasks or activities. These are then tested as group through integration testing to ensure whole segments of an application behave as expected (i.e., the interactions between units are seamless). These tests are often framed by user scenarios, such as logging into an application or opening files. Integrated tests can be conducted by either developers or independent testers and are usually comprised of a combination of automated functional and manual tests.

5.2.3. System Testing

System testing is a black box testing method used to evaluate the completed and integrated system, as a whole, to ensure it meets specified requirements. The functionality of the software is tested from end-to-end and is typically conducted by a separate testing team than the development team before the product is pushed into production.

5.2.4. Acceptance Testing

Acceptance testing is the last phase of functional testing and is used to assess whether or not the final piece of software is ready for delivery. It involves ensuring that the product is in compliance with all of the original business criteria and that it meets the end user’s needs. This requires the product be tested both internally and externally. Beta testing is key to getting real feedback from potential customers and can address any final usability concerns.

Non-functional testing methods incorporate all test types focused on the operational aspects of a piece of software. These include:

5.2.5. Performance Testing

Performance testing is a non-functional testing technique used to determine how an application will behave under various conditions. The goal is to test its responsiveness and stability in real user situations. Performance testing can be broken down into four types:

- Load testing is the process of putting increasing amounts of simulated demand on your software, application, or website to verify whether or not it can handle what it’s designed to handle.

- Stress testing takes this a step further and is used to gauge how your software will respond at or beyond its peak load. The goal of stress testing is to overload the application on purpose until it breaks by applying both realistic and unrealistic load scenarios. With stress testing, you’ll be able to find the failure point of your piece of software.
- Endurance testing, also known as soak testing, is used to analyze the behavior of an application under a specific amount of simulated load over longer amounts of time. The goal is to understand how your system will behave under sustained use, making it a longer process than load or stress testing (which are designed to end after a few hours). A critical piece of endurance testing is that it helps uncover memory leaks.
- Spike testing is a type of load test used to determine how your software will respond to substantially larger bursts of concurrent user or system activity over varying amounts of time. Ideally, this will help you understand what will happen when the load is suddenly and drastically increased.

5.2.6. Security Testing

With the rise of cloud-based testing platforms and cyber-attacks, there is a growing concern and need for the security of data being used and stored in software. Security testing is a non-functional software testing technique used to determine if the information and data in a system is protected. The goal is to purposefully find loopholes and security risks in the system that could result in unauthorized access to or the loss of information by probing the application for weaknesses. There are multiple types of this testing method, each of which aimed at verifying six basic principles of security: Integrity, Confidentiality, Authentication, Authorization, Availability, and Non-repudiation.

5.2.7. Usability Testing

Usability testing is a testing method that measures an application’s ease-of-use from the end-user perspective and is often performed during the system or acceptance testing stages. The goal is to determine whether or not the visible design and aesthetics of an application meet the intended workflow for various processes, such as logging into an application. Usability testing is a great way for teams to review separate functions, or the system as a whole, is intuitive to use.

5.2.8. Compatibility Testing

Compatibility testing is used to gauge how an application or piece of software will work in different environments. It is used to check that your product is compatible with multiple operating systems, platforms, browsers, or resolution configurations. The goal is to ensure that your software’s functionality is consistently supported across any environment you expect your end users to be using.

5.3. TEST CASES

The first five rows of the feature “Attrition” have been taken to demonstrate the output. It can take one of the two values: 0, 1. Here, 0 represents “No” and 1 represents “Yes”.

Test ID	Scenario	Input	Expected Output	Actual Output	Result
1	Prediction using Logistic Regression	Fictitious dataset by IBM Data Scientists	[1, 1, 0, 1, 1]	[1, 0, 0, 1, 1] (Accuracy: 89.79%)	
2	Prediction using kNN	Fictitious dataset by IBM Data Scientists	[1, 1, 0, 1, 1]	[0, 0, 0, 0, 0] (Accuracy: 86.73%)	
3	Prediction using Random Forest	Fictitious dataset by IBM Data Scientists	[1, 1, 0, 1, 1]	[1, 0, 0, 0, 0] (Accuracy: 85.71%)	
4	Prediction using Recursive Feature Elimination	Fictitious dataset by IBM Data Scientists	[1, 1, 0, 1, 1]	[0, 0, 0, 0, 0] (Accuracy: 86.73%)	
5	Prediction using SelectKBest+LR	Fictitious dataset by IBM Data Scientists	[1, 1, 0, 1, 1]	[0, 0, 0, 0, 1] (Accuracy: 87.07%)	
6	Prediction using SelectKBest+kNN	Fictitious dataset by IBM Data Scientists	[1, 1, 0, 1, 1]	[0, 0, 0, 0, 0] (Accuracy: 86.73%)	
7	Prediction using XGBoost	Fictitious dataset by IBM Data Scientists	[1, 1, 0, 1, 1]	[1, 0, 0, 1, 0] (Accuracy: 88.77%)	

CHAPTER 6

CONCLUSION AND FUTURE WORK

This paper outlined the various classification algorithms used to predict employee attrition. The accuracy was checked using different metrics like accuracy_score, confusion matrix, etc. The error of each algorithm was calculated using mean squared error and r-squared error. The cross validation score was also calculated for each model to check for underfitting or overfitting. For the dataset used in this project, the results showed the superiority of Logistic Regression in terms of accuracy and false positive count (from confusion matrix). The combination of SelectKBest feature selection and Logistic Regression exhibited the least amount of error.

The future work of this project is to identify the most important attributes that contribute to the attrition of the employees. It can also be extended to identify the most vulnerable employees and the specific features that have higher importance in their attrition.

APPENDIX 1

SOURCE CODE

Main.ipynb

```
import pandas as pd
import numpy as np
from sklearn import preprocessing
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.pipeline import make_pipeline
df = pd.read_csv("WA_Fn-UseC_-HR-Employee-Attrition.csv")
df.columns[df.isnull().any()]
df.interpolate()
accuracy_arr=[]
cross_validation_scores=[]
feature_names = list(df.select_dtypes(['object']).columns.values)
le = preprocessing.LabelEncoder()
def encode_col(col_name):
        encodes = le.fit(df[col_name])
        new_col_name = "e."+col_name
```

```
    df[new_col_name] = df[col_name].map(lambda x: encodes.transform([x]))
    df[new_col_name] = df[new_col_name].map(lambda x:x[0])
    return
for i in feature_names :
    encode_col(i)
y = df["e.Attrition"]
X = df.select_dtypes(exclude=[object]).drop(["e.Attrition"], axis=1)
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size = 0.2,
random_state = 42)
def add_metrics(accuracy,cross_validation,algorithm,one_indices) :
    accuracy_arr.append([accuracy,algorithm,one_indices])
    cross_validation_scores.append(cross_validation)
%run LogisticRegression.ipynb
add_metrics(accuracy,cross_validation,'LogisticRegression',one_indices)
%run KNN.ipynb
add_metrics(accuracy,cross_validation,'KNN',one_indices)
%run RandomForest.ipynb
add_metrics(accuracy,cross_validation,'RandomForest',one_indices)
%run RecursiveFeatureElimination.ipynb
add_metrics(accuracy,cross_validation,'RecursiveFeatureElimination',one_indices)
```

```
%run SelectKBest+LR.ipynb

add_metrics(accuracy,cross_validation,'SelectKBest+LR',one_indices)

%run SelectKBest+kNN.ipynb

add_metrics(accuracy,cross_validation,'SelectKBest+kNN',one_indices)

%run XGBoost.ipynb

add_metrics(accuracy,cross_validation,'XGBoost',one_indices)

max_accuracy = max(accuracy_arr)

print(max_accuracy)

print("EMPLOYEE NUMBERS OF EMPLOYEES WHO ARE LIKELY TO LEAVE :")

for i in max_accuracy[2] :

        print(X_test.iloc[i,5].to_string(index=False))
```

Metrics_Evaluation.ipynb

```
%run Analysis.ipynb

def get_metrics(y_pred,clf) :

        accuracy=get_accuracy(y_pred)

        cross_validation=get_cross_validation(clf)

        mean_square_error=get_mean_square_error(y_pred)

        r2_error=get_r2_error(y_pred)

        confusion_matrix=get_confusion_matrix(y_pred)

        report=get_classification_report(y_pred)

        return accuracy, cross_validation, mean_square_error, r2_error,
        confusion_matrix, report
```

Analysis.ipynb

```
from sklearn.metrics import accuracy_score

from sklearn.metrics import mean_squared_error

from sklearn.metrics import r2_score

from sklearn.metrics import confusion_matrix

from sklearn.metrics import classification_report

from sklearn.model_selection import cross_val_score

def get_accuracy(y_pred) :
        accuracy=accuracy_score(y_test,y_pred)
        return accuracy
def get_cross_validation(clf) :
        cross_validation = cross_val_score(clf, X_train, y_train)
        return cross_validation.mean()
def get_mean_square_error(y_pred) :
        m_error=mean_squared_error(y_test,y_pred)
        return m_error
def get_r2_error(y_pred) :
        r2_error=r2_score(y_test,y_pred)
        return r2_error
def get_confusion_matrix(y_pred) :
        matrix=confusion_matrix(y_test,y_pred)
        return matrix
def get_classification_report(y_pred) :
        report = classification_report(y_test,y_pred,labels=np.unique(y_pred))
        return report
```

LogisticRegression.ipynb

```
import sklearn

from sklearn import linear_model

clf = linear_model.LogisticRegression()

trained_model=clf.fit(X_train,y_train)

y_pred =clf.predict(X_test)

one_indices = np.where(y_pred==1)

%run Metrics_Evaluation.ipynb

accuracy,cross_validation,mean_square_error,r2_error,confusion_matrix,report=get_metrics(y_pred,clf)
```

KNN.ipynb

```
from sklearn.neighbors import KNeighborsClassifier

record_length = len(df.index)

clf = KNeighborsClassifier(n_neighbors=int(math.sqrt(record_length)))

clf.fit(X_train, y_train)

y_pred = clf.predict(X_test)

one_indices = np.where(y_pred==1)

%run Metrics_Evaluation.ipynb

accuracy,cross_validation,mean_square_error,r2_error,confusion_matrix,report=get_metrics(y_pred,clf)
```

RandomForest.ipynb

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.pipeline import Pipeline
from sklearn.feature_selection import SelectFromModel
from sklearn.svm import LinearSVC
from sklearn import linear_model
clf = Pipeline([
  ('feature_selection', SelectFromModel(LinearSVC(random_state=42))),
  ('classification', RandomForestClassifier(random_state=42))
])
trained_model=clf.fit(X_train, y_train)
print(trained_model)
y_pred=trained_model.predict(X_test)
one_indices = np.where(y_pred==1)
%run Metrics_Evaluation.ipynb
accuracy,cross_validation,mean_square_error,r2_error,confusion_matrix,report=get_metrics(y_pred,clf)
```

RecursiveFeatureElimination.ipynb

```
from numpy import array

from sklearn import linear_model

from sklearn.feature_selection import RFE

log_reg = linear_model.LogisticRegression(random_state=42)

clf = RFE(log_reg, n_features_to_select=1)

trained_model=clf.fit(X_train,y_train)

y_pred = clf.predict(X_test)

one_indices = np.where(y_pred==1)

rank = clf.ranking_

feature_names = list(X.columns.values)

features = array(feature_names)

con = np.column_stack((rank,features))

%run Metrics_Evaluation.ipynb

accuracy,cross_validation,mean_square_error,r2_error,confusion_matrix,report=get_metrics(y_pred,clf)
```

SelectKBest+LR.ipynb

```
from sklearn.pipeline import make_pipeline

from sklearn.feature_selection import SelectKBest

from sklearn.feature_selection import chi2

from sklearn import linear_model

clf = make_pipeline(

    SelectKBest(chi2,k=20),

    linear_model.LogisticRegression()

)

clf.fit(X_train, y_train)

y_pred = clf.predict(X_test)

one_indices = np.where(y_pred==1)

trained_model = clf.fit(X_train,y_train)

%run Metrics_Evaluation.ipynb

accuracy,cross_validation,mean_square_error,r2_error,confusion_matrix,report=get_metrics(y_pred,clf)
```

SelectKBest+kNN.ipynb

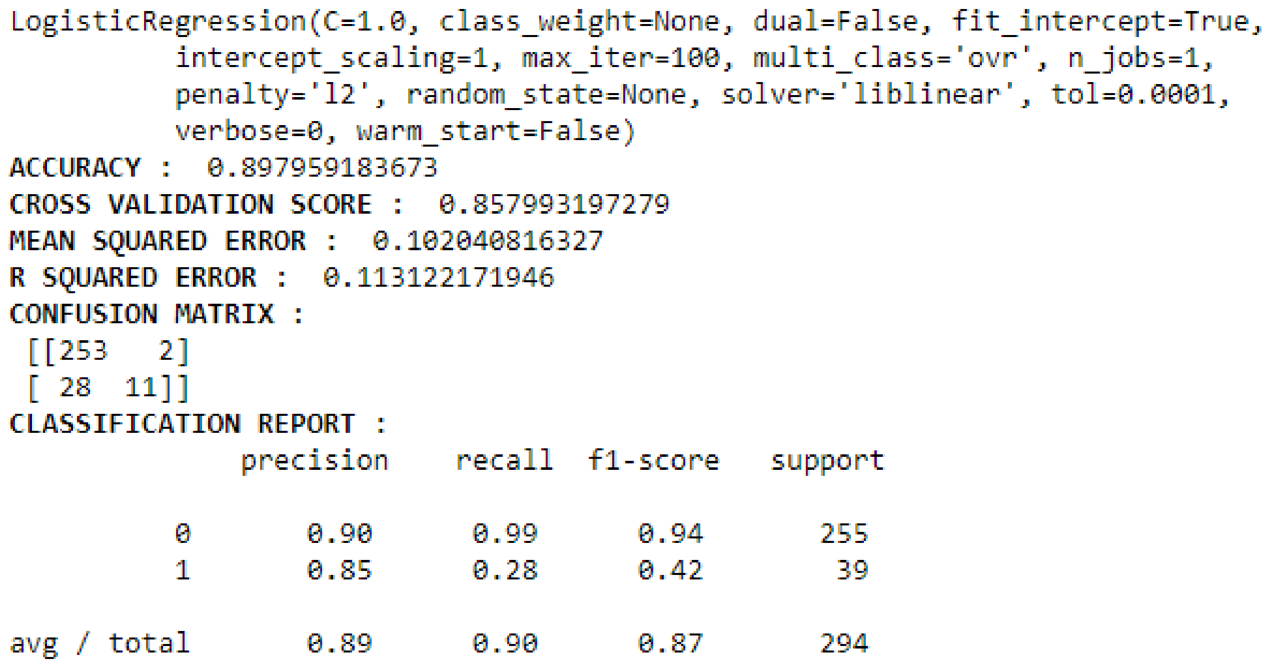
```
from sklearn.pipeline import make_pipeline
from sklearn.feature_selection import SelectKBest
from sklearn.feature_selection import chi2
from sklearn.neighbors import KNeighborsClassifier
record_length = len(df.index)
print (record_length)
clf = make_pipeline(
    SelectKBest(chi2,k=20),
    KNeighborsClassifier(n_neighbors=int(math.sqrt(record_length)))
)
clf.fit(X_train, y_train)
y_pred = clf.predict(X_test)
one_indices = np.where(y_pred==1)
trained_model = clf.fit(X_train,y_train)
%run Metrics_Evaluation.ipynb
accuracy,cross_validation,mean_square_error,r2_error,confusion_matrix,report=get_metrics(y_pred,clf)
```

XGBoost.ipynb

```
from xgboost.sklearn import XGBClassifier

clf = XGBClassifier()

trained_model=clf.fit(X_train,y_train)

print(trained_model)

y_pred = clf.predict(X_test)

one_indices = np.where(y_pred==1)

%run Metrics_Evaluation.ipynb

accuracy,cross_validation,mean_square_error,r2_error,confusion_matrix,report=get_metrics(y_pred,clf)
```

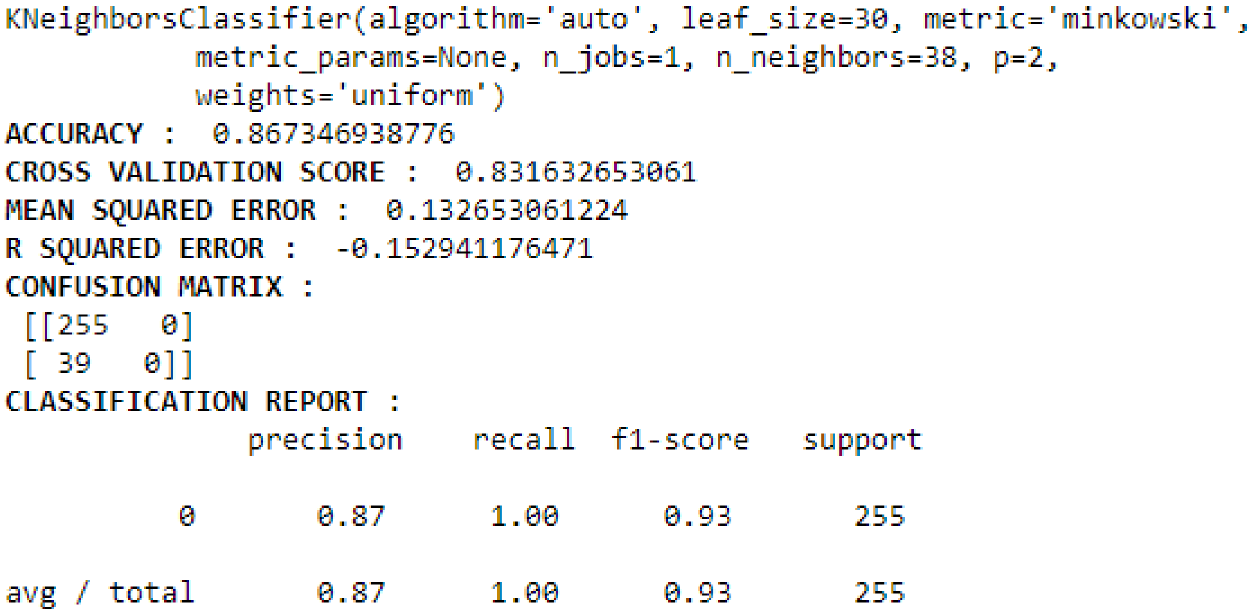
APPENDIX 2

SCREENSHOTS



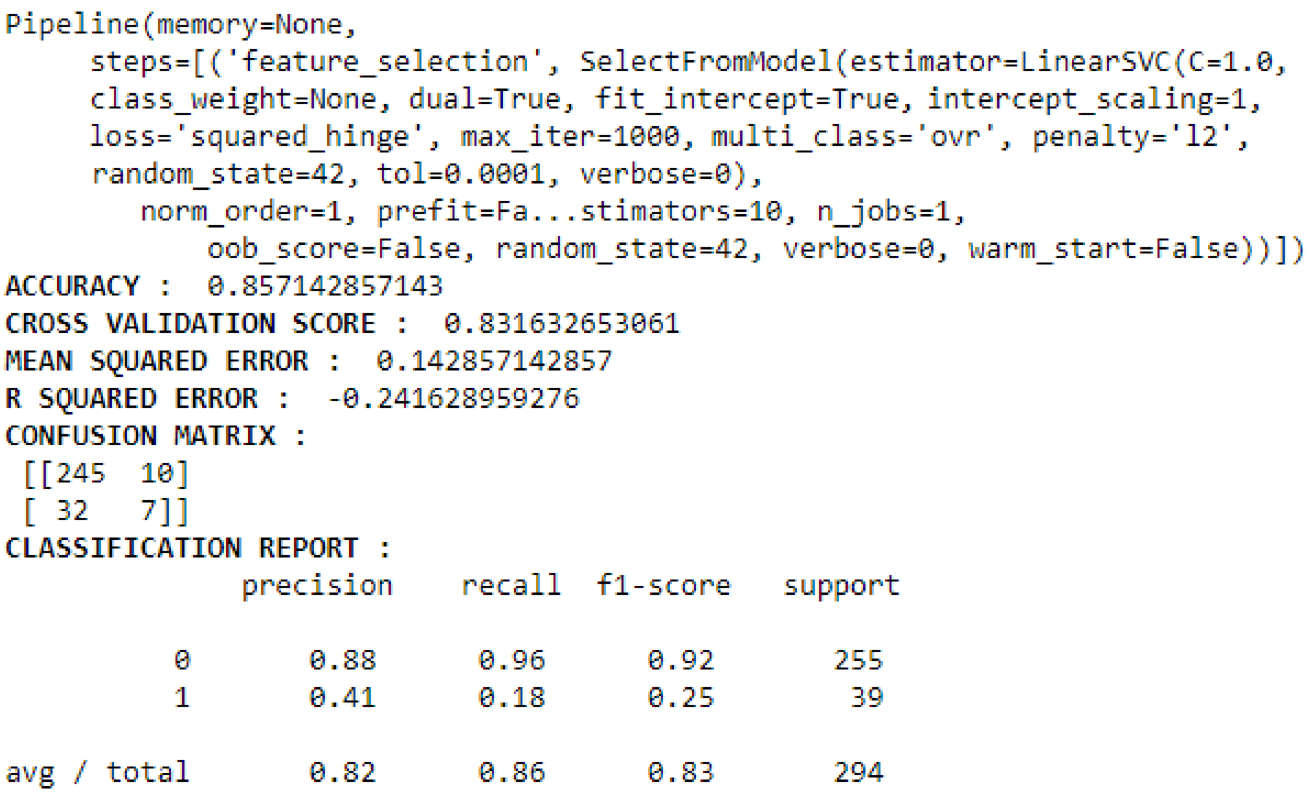
```
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
          intercept_scaling=1, max_iter=100, multi_class='ovr', n_jobs=1,
          penalty='l2', random_state=None, solver='liblinear', tol=0.0001,
          verbose=0, warm_start=False)
ACCURACY :  0.897959183673
CROSS VALIDATION SCORE :  0.857993197279
MEAN SQUARED ERROR :  0.102040816327
R SQUARED ERROR :  0.113122171946
CONFUSION MATRIX :
 [[253   2]
 [ 28  11]]
CLASSIFICATION REPORT :
              precision    recall  f1-score   support

          0       0.90      0.99      0.94       255
          1       0.85      0.28      0.42        39

avg / total       0.89      0.90      0.87       294
```

Figure 6.1 – Logistic Regression results



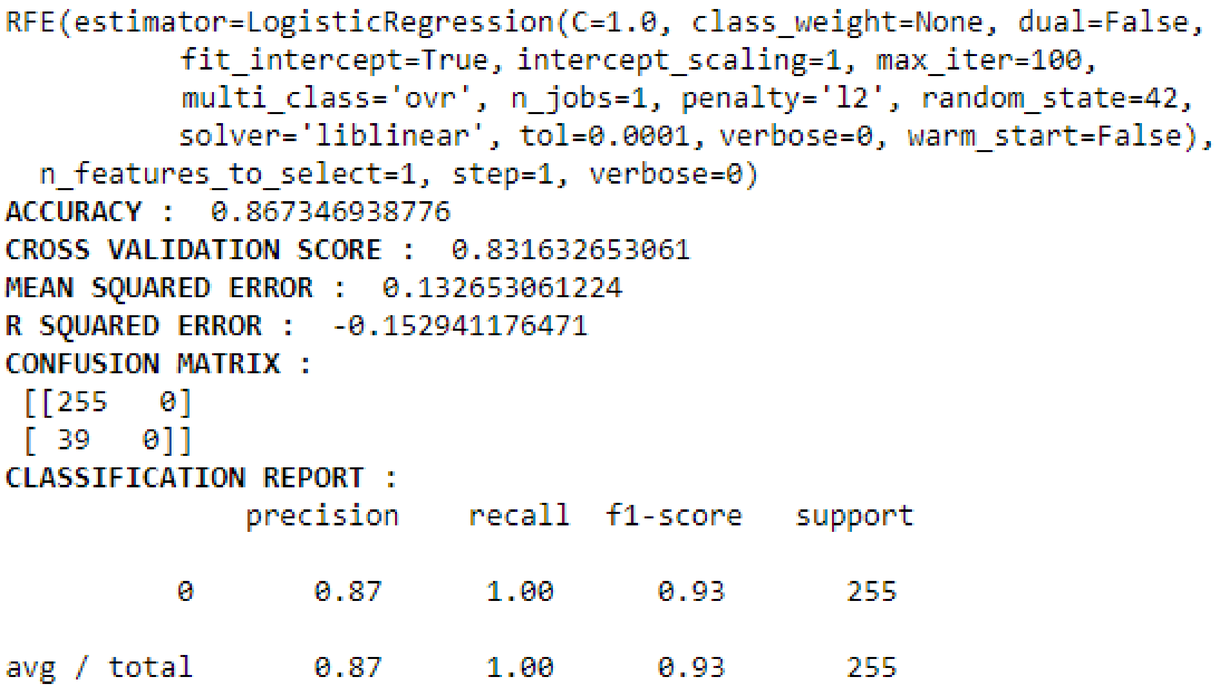
```
KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
           metric_params=None, n_jobs=1, n_neighbors=38, p=2,
           weights='uniform')
ACCURACY :  0.867346938776
CROSS VALIDATION SCORE :  0.831632653061
MEAN SQUARED ERROR :  0.132653061224
R SQUARED ERROR :  -0.152941176471
CONFUSION MATRIX :
 [[255   0]
 [ 39   0]]
CLASSIFICATION REPORT :
              precision    recall  f1-score   support

          0       0.87      1.00      0.93       255

avg / total       0.87      1.00      0.93       255
```

Figure 6.2 – kNN results



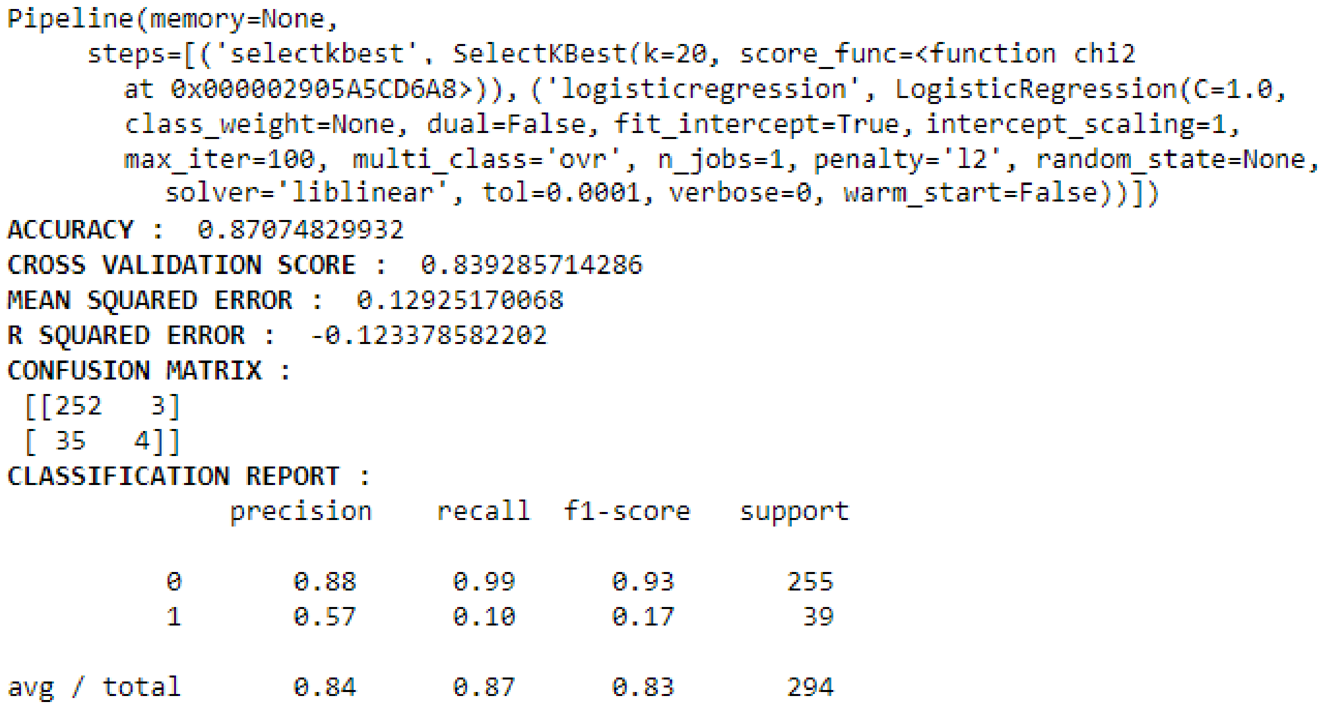
```
Pipeline(memory=None,
     steps=[('feature_selection', SelectFromModel(estimator=LinearSVC(C=1.0,
     class_weight=None, dual=True, fit_intercept=True, intercept_scaling=1,
     loss='squared_hinge', max_iter=1000, multi_class='ovr', penalty='l2',
     random_state=42, tol=0.0001, verbose=0),
        norm_order=1, prefit=Fa...stimators=10, n_jobs=1,
            oob_score=False, random_state=42, verbose=0, warm_start=False))])
ACCURACY :  0.857142857143
CROSS VALIDATION SCORE :  0.831632653061
MEAN SQUARED ERROR :  0.142857142857
R SQUARED ERROR :  -0.241628959276
CONFUSION MATRIX :
 [[245  10]
 [ 32   7]]
CLASSIFICATION REPORT :
              precision    recall  f1-score   support

          0       0.88      0.96      0.92       255
          1       0.41      0.18      0.25        39

avg / total       0.82      0.86      0.83       294
```

Figure 6.3 – Random Forest results



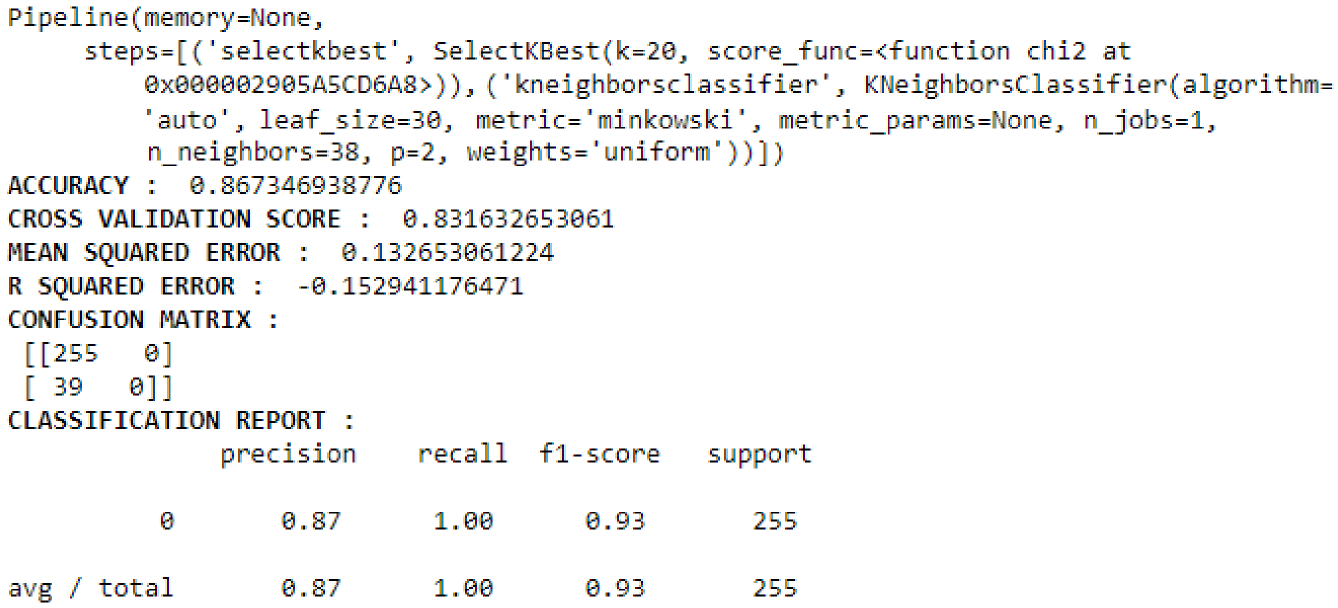
```
RFE(estimator=LogisticRegression(C=1.0, class_weight=None, dual=False,
          fit_intercept=True, intercept_scaling=1, max_iter=100,
          multi_class='ovr', n_jobs=1, penalty='l2', random_state=42,
          solver='liblinear', tol=0.0001, verbose=0, warm_start=False),
  n_features_to_select=1, step=1, verbose=0)
ACCURACY :  0.867346938776
CROSS VALIDATION SCORE :  0.831632653061
MEAN SQUARED ERROR :  0.132653061224
R SQUARED ERROR :  -0.152941176471
CONFUSION MATRIX :
 [[255   0]
 [ 39   0]]
CLASSIFICATION REPORT :
              precision    recall  f1-score   support

          0       0.87      1.00      0.93       255

avg / total       0.87      1.00      0.93       255
```

Figure 6.4 – RecursiveFeatureElimination results



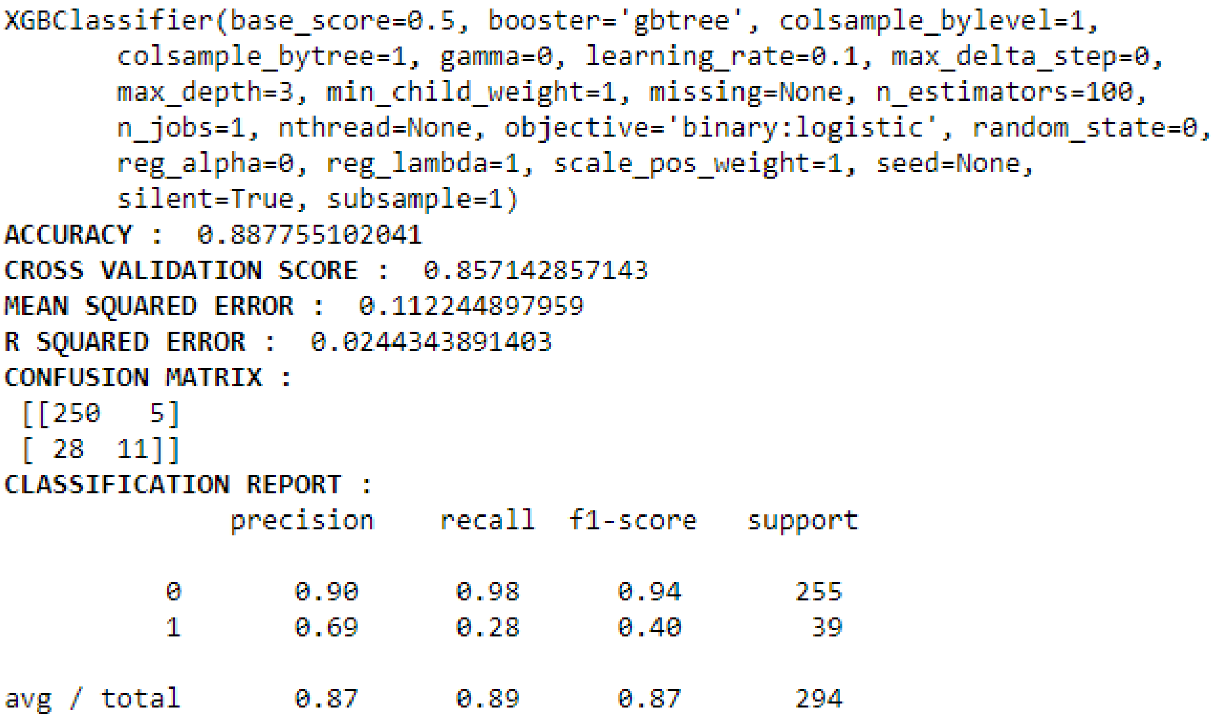
```
Pipeline(memory=None,
     steps=[('selectkbest', SelectKBest(k=20, score_func=<function chi2
        at 0x000002905A5CD6A8>)), ('logisticregression', LogisticRegression(C=1.0,
        class_weight=None, dual=False, fit_intercept=True, intercept_scaling=1,
        max_iter=100, multi_class='ovr', n_jobs=1, penalty='l2', random_state=None,
          solver='liblinear', tol=0.0001, verbose=0, warm_start=False))])
ACCURACY :  0.87074829932
CROSS VALIDATION SCORE :  0.839285714286
MEAN SQUARED ERROR :  0.12925170068
R SQUARED ERROR :  -0.123378582202
CONFUSION MATRIX :
 [[252   3]
 [ 35   4]]
CLASSIFICATION REPORT :
             precision    recall  f1-score   support

          0       0.88      0.99      0.93       255
          1       0.57      0.10      0.17        39

avg / total       0.84      0.87      0.83       294
```

Figure 6.5 – SKB+LR results



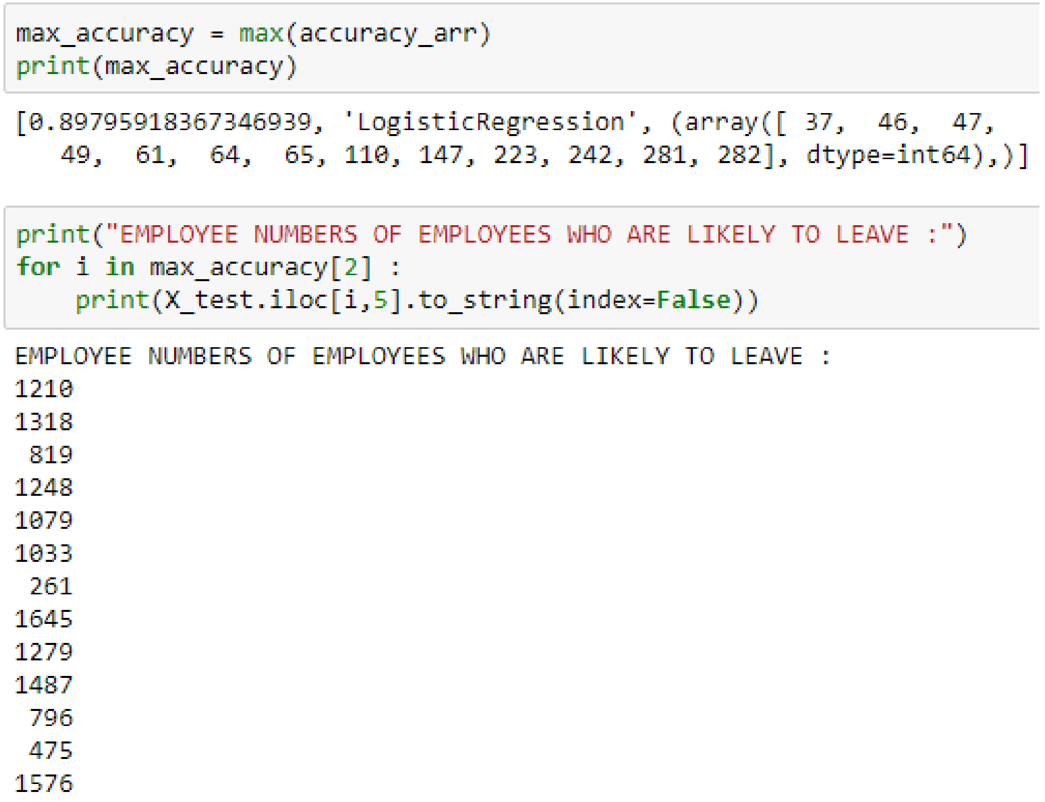
```
Pipeline(memory=None,
     steps=[('selectkbest', SelectKBest(k=20, score_func=<function chi2 at
         0x000002905A5CD6A8>)), ('kneighborsclassifier', KNeighborsClassifier(algorithm=
         'auto', leaf_size=30, metric='minkowski', metric_params=None, n_jobs=1,
         n_neighbors=38, p=2, weights='uniform'))])
ACCURACY :  0.867346938776
CROSS VALIDATION SCORE :  0.831632653061
MEAN SQUARED ERROR :  0.132653061224
R SQUARED ERROR :  -0.152941176471
CONFUSION MATRIX :
 [[255   0]
 [ 39   0]]
CLASSIFICATION REPORT :
             precision    recall  f1-score   support

          0       0.87      1.00      0.93       255

avg / total       0.87      1.00      0.93       255
```

Figure 6.6 – SKB+kNN results



```
XGBClassifier(base_score=0.5, booster='gbtree', colsample_bylevel=1,
       colsample_bytree=1, gamma=0, learning_rate=0.1, max_delta_step=0,
       max_depth=3, min_child_weight=1, missing=None, n_estimators=100,
       n_jobs=1, nthread=None, objective='binary:logistic', random_state=0,
       reg_alpha=0, reg_lambda=1, scale_pos_weight=1, seed=None,
       silent=True, subsample=1)
ACCURACY :  0.887755102041
CROSS VALIDATION SCORE :  0.857142857143
MEAN SQUARED ERROR :  0.112244897959
R SQUARED ERROR :  0.0244343891403
CONFUSION MATRIX :
 [[250   5]
 [ 28  11]]
CLASSIFICATION REPORT :
              precision    recall  f1-score   support

          0       0.90      0.98      0.94       255
          1       0.69      0.28      0.40        39

avg / total       0.87      0.89      0.87       294
```

Figure 6.7 – XGBoost results



```
max_accuracy = max(accuracy_arr)
print(max_accuracy)
```

```
[0.89795918367346939, 'LogisticRegression', (array([ 37,  46,  47,
   49,  61,  64,  65, 110, 147, 223, 242, 281, 282], dtype=int64),)]
```

```
print("EMPLOYEE NUMBERS OF EMPLOYEES WHO ARE LIKELY TO LEAVE :")
for i in max_accuracy[2] :
    print(X_test.iloc[i,5].to_string(index=False))
```

```
EMPLOYEE NUMBERS OF EMPLOYEES WHO ARE LIKELY TO LEAVE :
1210
1318
 819
1248
1079
1033
 261
1645
1279
1487
 796
 475
1576
```

Figure 6.8 – Final Results

REFERENCES

1. Rohit Punnoose, Pankaj Ajit, “Prediction of Employee Turnover in Organizations using Machine Learning Algorithms (A case for Extreme Gradient Boosting)”, (IJARAI) International Journal of Advanced Research in Artificial Intelligence, Vol. 5, No. 9, 2016

2. Vidya Sunil Kadam, H.M.Thakar, “A Study of Attrition in IT Industries in Pune”, International Journal of Advanced Research (2014), Volume 2, Issue 3, 650-656, ISSN: 2320-5407

3. Dr. Sunil Kumar Dhal, Amaresh C Nayak, “A Study on Employee Attrition in BPO Industries in India”, International Journal of Science and Research (IJSR), ISSN: 2319-7064

4. Moninder Singh, Kush R. Varshney, Jun Wang, Aleksandra Mojsilovic, “An Analytics Approach for Proactively Combating Voluntary Attrition of Employees”, IEEE 12th International Conference on Data Mining Workshops (2012), 317-323

5. Santoshi Sengupta, “An exploratory study on job and demographic attributes affecting employee satisfaction in the Indian BPO industry", Strategic Outsourcing: An International Journal (2011), Volume 4, Issue 3, 248- 273

6. Ankita Srivastava, Yogesh Tiwari, Hradesh Kumar, “Attrition and Retention of employees in BPO sector”, International Journal of Computer Technology and Applications, Volume 2, 3056-3065, ISSN: 2229-6093

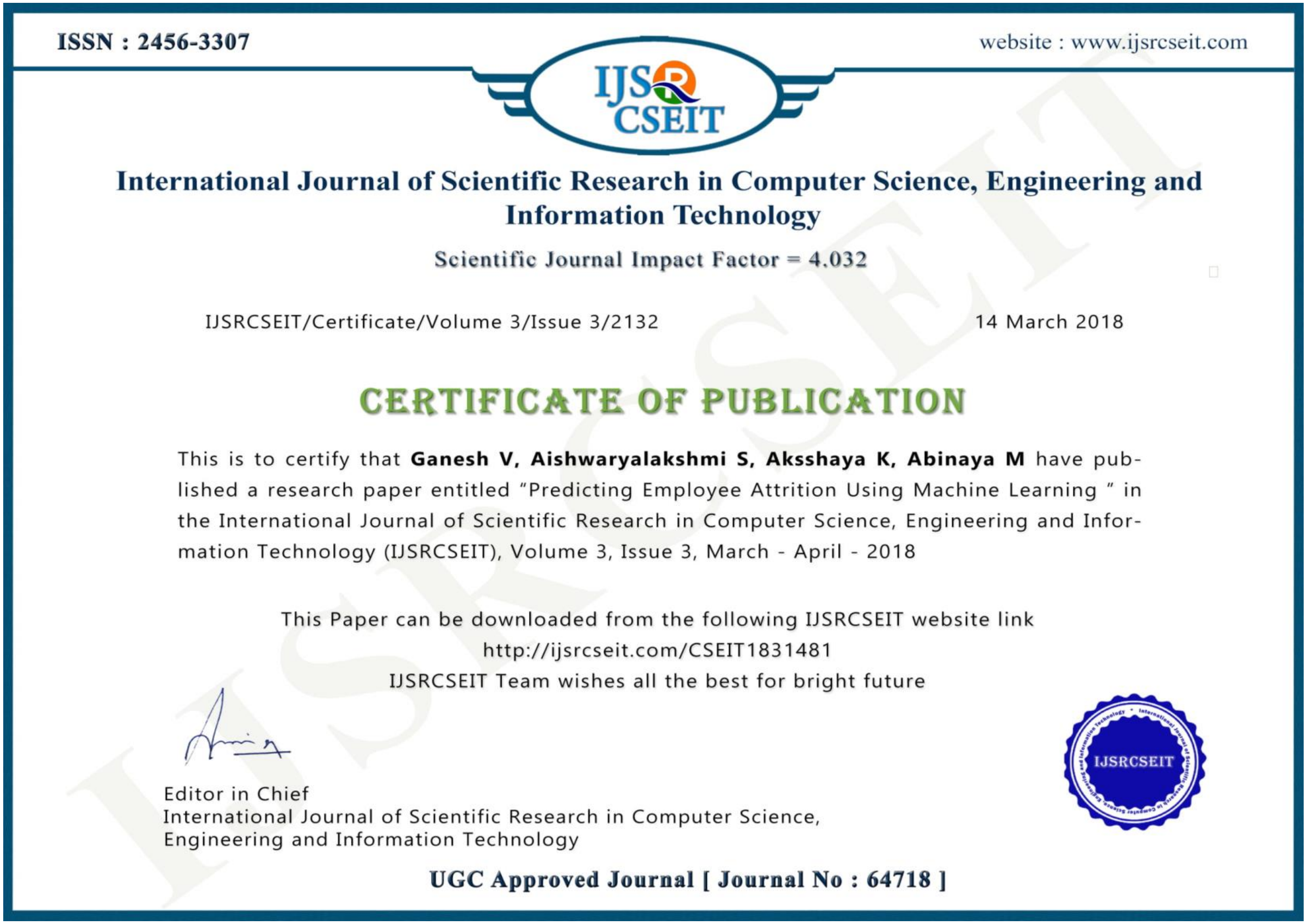
7. V. Vijay Anand, R. Saravanasudhan, R. Vijesh, “Employee Attrition - A pragmatic study with reference to BPO Industry”, IEEE - International Conference on Advances In Engineering, Science And Management (2012), 769-775

8. Hsin-Yun Chang, “Employee Turnover: A Novel Prediction Solution with Effective Feature Selection”, WSEAS Transactions on Information Science and Applications (2009), Issue 3, Volume 6, 417-426

9. Rahul Yedida, Rahul Reddy, Rakshit Vahi, Rahul J, Abhilash, Deepti Kulkarni, “Employee Attrition Prediction”, International Journal of Science and Research (2017), ISSN: 2319-7064

10. Rupesh Khare, Dimple Kaloya, Chandan Kumar Choudhary, Gauri Gupta, “Employee Attrition Risk Assessment using Logistic Regression Analysis”, 2nd IIMA International Conference on Advanced Data Analysis, Business Analytics and Intelligence (2011)

CERTIFICATE OF PUBLICATION



ISSN : 2456-3307

website : www.ijsrcseit.com

International Journal of Scientific Research in Computer Science, Engineering and Information Technology

Scientific Journal Impact Factor = 4.032

IJSRCSEIT/Certificate/Volume 3/Issue 3/2132

14 March 2018

CERTIFICATE OF PUBLICATION

This is to certify that **Ganesh V, Aishwaryalakshmi S, Aksshaya K, Abinaya M** have published a research paper entitled “Predicting Employee Attrition Using Machine Learning ” in the International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), Volume 3, Issue 3, March - April - 2018

This Paper can be downloaded from the following IJSRCSEIT website link
http://ijsrcseit.com/CSEIT1831481
IJSRCSEIT Team wishes all the best for bright future

Editor in Chief
International Journal of Scientific Research in Computer Science,
Engineering and Information Technology

IJSRCSEIT

UGC Approved Journal [Journal No : 64718]