Fall 2021

Employee Churn Prediction using Logistic Regression and Support Vector Machine

Rajendra Maharjan
San Jose State University

Employee Churn Prediction using Logistic Regression and Support Vector Machine

A Project

Presented to

The Faculty of the Department of Computer Science

San José State University

In Partial Fulfillment

of the Requirement for the Degree

Master of Science

by

Rajendra Maharjan

December 2021

The Designated Project Committee Approves the Project Titled

Employee Churn Prediction using Logistic Regression and Support Vector Machine

by

Rajendra Maharjan

APPROVED FOR THE DEPARTMENT OF COMPUTER SCIENCE

SAN JOSÉ STATE UNIVERSITY

DECEMBER 2021

Dr. Robert Chun	Department of Computer Science
Prof. Christopher Pollett	Department of Computer Science
Ms. Stuti Patel	Software Engineer (Cisco)

ABSTRACT

It is a challenge for Human Resource (HR) team to retain their existing employees than to hire a new one. For any company, losing their valuable employees is a loss in terms of time, money, productivity, and trust, etc. This loss could be possibly minimized if HR could beforehand find out their potential employees who are planning to quit their job hence, we investigated solving the employee churn problem through the machine learning perspective. We have designed machine learning models using supervised and classification-based algorithms like Logistic Regression and Support Vector Machine (SVM). The models are trained with the IBM HR employee dataset retrieved from https://kaggle.com and later fine-tuned to boost the performance of the models. Metrics such as precision, recall, confusion matrix, AUC, ROC curve were used to compare the performance of the models. The Logistic Regression model recorded an accuracy of 0.67, Sensitivity of 0.65, Specificity of 0.70, Type I Error of 0.30, Type II Error of 0.35, and AUC score of 0.73 where as SVM achieved an accuracy of 0.93 with Sensitivity of 0.98, Specificity of 0.88, Type I Error of 0.12, Type II Error of 0.01 and AUC score of 0.96.

Index terms – **Algorithm, AUC, Classification-based, Churn, Confusion matrix, Machine learning Models, Logistic Regression, Precision, Recall, ROC curve, Sensitivity, Specificity, Support Vector Machine, Supervised.**

ACKNOWLEDGEMENTS

First of all, I am grateful to my advisor Dr. Robert Chun. His continuous support and guidance have made this project better. I am thankful to Dr. Chun for giving me the freedom to choose the project topic of my interest. I would like to thank my committee members Prof. Christopher Pollett and Ms. Stuti Patel for their valuable feedbacks and suggestions. A heartful thanks go to my coworkers whose tremendous support helped me balance my office workload during this project. Last but not least, I am always thankful to my friends and family members who motivated me throughout my Master’s Degree journey.

TABLE OF CONTENTS

LIST OF FIGURES

I. INTRODUCTION

Employee churn is defined as a voluntary decision made by an employee to resign or to retire from their job requiring the position to be filled up by another candidate. Feeling lack of coaching and feedback, lack of growth, commute time, unsatisfied pay scale, feeling devalued, work stress, work-life imbalance, and lack of trust from the supervisor, etc. are some of the common reasons why people quit their jobs [1]. In 2018 a report was published by LinkedIn Talent Solutions that showed the technology industry is on the top with the highest employee turnover of 13.2% [2]. To solve the employee churn problem, we designed two supervised models one using the Logistic Regression and the other using SVM models to predict whether an employee will churn or not. An exploratory data analysis was performed to explore and clean the IBM HR dataset that we retrieved from www.kaagle.com. Techniques such as heatmap, χ^2, Extra Tree classifier, and Information Gain were implemented as a part of the feature selection process. Stratified K-Fold cross-validation, under sampling, over sampling, and SMOTETOMEK methods were implemented to balance the IBM HR dataset, and metrics such as Confusion matrix, Classification report, Sensitivity, Specificity, Type I Error, Type II Error, ROC, and AUC score were used to monitor the performance of the models. To optimize the performance of the model, the GridSearchCV method was used for the Logistic Regression, and parameters such as C, kernel, and gamma were used for the SVM model.

Employees are always considered valuable assets of the company. Especially those employees who have worked for a longer period gained years of experience and the top performers are considered special employees. Companies have to bear more loss when special employees decide to resign compared to regular employees. A leading and experienced employee’s resignation can have a psychological effect on the team, lowering the morale of the team members ultimately leading to the loss of productivity. Having an experienced employee resigning from the job can bring insecurity to the HR and Management team thinking that their former employee may apply the acquired intellectual knowledge to their rival companies making the competition fiercer. Companies having a larger number of employee turnover can potentially damage their brand value and also start losing the trust of their valuable clients. Filling up the vacant positions sometimes needs to coordinate with the third-party staffing agency which could be extra expenses to the company. A newly recruited staff needs to be sent for various training so that he/she can perform his/her duties in a decent way and sending the employees to the training cost additional money to the company. The newly recruited employees require some time to adjust to the new teammates and workplace before they can start performing. The HR team has to go through all those steps every time there is a search for a replacement resource. Hence, to minimize that situation the HR team focuses on retaining their existing employees.

Various strategies have been approached by the managers and HR team to retain their top talents. Compensation and Benefits packages are one of those strategies where the HR team work on setting up a proper compensation structure across organizational levels, assuring fair pay and equity, offering better benefits packages such as vesting, stock options, and cash bonuses, etc. [3]. Company perks such as extended PTO and flex time have been an effective way of retaining the employees [4]. Establishing a staff/employee recognition program can make employees feel their work is being valued and hence that may encourage them to stay with the company for a longer period [5]. Conducting surveys is another good approach to collect employees' feedback which helps to get an insight into what the employees feel towards their employer. Some HR conduct exit interviews to know the reason why an employee decided to quit the job. Exit interviews are in-person interviews so there can be a situation where an employee may not feel comfortable sharing the actual reason for leaving. Once an employee makes a final decision of leaving the company and sends a resignation letter then it is almost impossible for HR to convince the employee to change their decision. However, the case is different if HR discovers beforehand that their staff is on a plan of resigning and they could take all preliminary countermeasures which could potentially help to change the employees' minds of resigning. Multiple efforts have been made in the past to solve employee or customer churn problems using various machine learning algorithms such as Naïve Bayes, KNN, Decision Trees, Reinforcement learning and Neural Networks, etc. Past research work has recorded the highest accuracy of 0.92, Recall of 0.75, F1-measure of 0.815, and AUC of 0.88.

There are two main objectives to conduct this research work. The first objective is to perform a deep analysis of the employee dataset, mine some patterns or useful information, and visualize that information so that HR can get some insights. The second objective is to build two predictive models using the Logistic regression and SVM that predict whether an employee will churn or not. The goal is to obtain a robust and accurate model through fine-tuning so that the HR has high confidence in using our designed models. We are hopeful that the employee churn prediction made by our designed models will help HR in some way or the other in taking necessary actions to retain their employees.

The rest of the paper is organized as follows: Section II, explains the prior works that have been conducted related to customer or employee churn problems, Section III discusses machine learning topics and Section IV explains algorithm selection. Section V and Section VI covers the implementation details, conclusion, and future works respectively.

II. RELATED WORK

There has been a lot of study and research work conducted by various groups and individuals on the topic related to churn, attrition, and turnover. Zhang et al. proposed a solution predicting the customer churn of a leading mobile telecommunication service provider [6]. In the paper, the authors tried to prove that the network attributes or interpersonal influence have a better contribution to improving the prediction accuracy compared to that of traditional attributes. The authors approached the classification and the propagation models implementing decision trees, logistic regression, and neural network machine learning algorithms. The results showed that the propagation model outperformed classification models on prediction accuracy.

In [7] Yadav and his team have applied data mining techniques in conjunction with machine learning algorithms to predict the early attrition of an employee. The brute-force approach has been implemented to convert a long list of categorical data into two numerical values 0 and 1. In addition to that, the one-hot encoding technique has also been implemented to convert categorical variables into numerical values. Each feature available in the dataset has a different level of contribution to the performance of the model. So, to collect only the meaningful features, the authors implemented a feature selection method called Recursive Feature Elimination with Cross-Validation (RFECV). This method recursively selects an optimal feature from smaller sets of features.

Another interesting paper was conducted by Raman and Bhattacharya [8] where they approached a solution predicting whether a teaching faculty member will leave the business school or not. Using the archival email dataset, the authors tried to study the email patterns, extracted the sentiments expressed on the emails, and discovered the features that have a significant contribution to the model's prediction. Using the R program, correlation analysis was conducted to discover the rich set of features. Similarly, word count analysis and sentiment analysis were also performed using data logging tools. An email pattern was discovered where the faculty members leaving the business school had more external email communication compared to internal. It was also discovered that the negative sentiments were less for the faculty members who left the school compared to those who stayed with the school. Researching the root cause of expressed sentiments and the decision of leaving the school was beyond the scope of the study.

A case study prepared by Saradhi and Palshikar gives two distinct models, the predictive model and the value model [9]. The predictive model is designed to predict whether a customer or an employee will churn or not, whereas the value model is used to identify how many of the churned employees or customers were valuable. Under the predictive model, the authors performed a comparison of various machine learning algorithms. Two of the value models, the customer lifetime value model, and the employee value model were briefly covered in the paper. To further elaborate the employee model, the authors proposed a simple employee value model that identified whether a churn employee was a valuable employee or not. The paper also defined the differences between an employee churn vs customer churn and the valuable customer vs valuable employee.

A deep learning model using a feed-forward neural network has been proposed to predict employee attrition by Dutta and Bandyopadhyay [10]. The model is trained on 1470 sample records received from https://kaggle.com. Features such as environment satisfaction, performance rating, distance from home, monthly income, stock market option, and work-life balance, etc. were taken into consideration to make the prediction. The proposed model comprised three layers of neural network with 32, 16, and 1 number of nodes respectively. To perform diverse computation, relu and sigmoid activation functions were applied to each of the three layers. To evaluate the designed predictive model, the authors implemented a 10-fold cross-validation method where the entire dataset got split into 10 groups and on each iteration, there would be 1 test data and the remaining 9 groups as training data. The proposed model reported an accuracy of 87.01% and 0.1299 of Mean Square Error (MESE) which outperformed the other 6 classifiers SVM, Naïve Bayes, KNN, Decision Tree, AdaBoost, and Random Forest.

Rodan and et al. conducted a study on customer churn prediction of a telecommunication company [11]. The paper talked about how a negative correlation learning method can be used to resolve the issue of target class/label imbalance. The authors claimed that the trained ensemble multilayer perceptron (MLP) using the negative correlation learning method has better customer churn prediction accuracy compared to that without NCL and other data mining techniques. The optimal model parameters such as hidden layers of N = 10, learning rate = 0.3, use of Sigmoid logistic activation function, 5-fold cross-validation, and penalty factor in range [0,1] were discovered as a valuable finding of the study.

Gao and the team implemented a weighted quadratic random forest (WQRF) algorithm to build a model which predicted employee turnover [12]. The model is an improved version of the random forest algorithm which uses the F-measure of each decision tree and the weighted voting mechanism to solve the data imbalance issue. The dataset was extracted from a Chinese communication company which consists of 2000 employee records and 32 features where only 13.5% of the total population were churned. As a part of feature selection, the attributes were ordered in descending order based on their importance score and only the top 15 attributes were selected. A benchmark comparison of the WQRF algorithm with C4.5, logistic regression, bp, and the random forest was conducted where the proposed algorithm outperformed the rest with the reported accuracy of 92.80 %, 0.653 of Recall, 0.711 of F-measure, and 0.881 of ROC area.

Models based on Random Forest and Naïve Bayes classifiers were proposed by Valle and Ruz to predict the turnover of the sales agent in a call center [13]. As a part of the experiment, a sample of 3543 sales activity records from 2407 sales agents was taken into consideration. The proposed models were trained with only 6 attributes such as logged hours, talked hours, effective contacts, number of approved sales, number of finished records, and approved production. The authors implemented 10-fold cross-validation to evaluate the performance of the models. In addition to that, the mean standard deviation of performance measure metrics such as accuracy, precision, recall, and area under the curve (AUC) was computed and compared for the Random Forest and Naïve Bayes classifier.

A customer churn prediction model based on textual customer interactions is proposed by Andrea and Tronscoso [14]. The textual comments used by the customers to interact with the service provider were considered as the main dataset. 23,195 interactions from 14,531 customers of Chilean bank was considered to build the model. Text mining techniques such as part of speech tagging and linguistic pattern analysis procedures were adapted to extract the churn determinant. Models like logistic regression, decision tree, multi-layer perceptron neural network, support vector machine, random forest, adboost, and kNN were designed. Out of all, SVM was reported as the best performing model with an overall accuracy of 58.3%, precision = 58.5%, and recall = 58.4%.

Using reinforcement learning, a customer churn prediction model was proposed by Panjasuchat and Limpiyakorn [15]. The original telecommunication dataset extracted from kaggle contained 100,000 sample records and 99 attributes. From the original dataset, 3 versions of the dataset were prepared, dataset1 which is the preprocessed version, dataset2 is the shuffled version and dataset3 is the combination of dataset1 and dataset2. Applying reinforcement learning and using the Deep Q Network (DQN) algorithm, the model used the concept of reward with a positive integer value for correct prediction and a negative integer as a penalty for an incorrect prediction. The proposed neural network consists of 4 connected layers and 2 hidden layers where each layer contains 256 neurons. Each layer of the neural network used ReLU as the activation function, 0.001 as a parameter for the learning rate, and Adam was the optimizer algorithm. The proposed Deep Q Network (DQN) algorithm outperformed other algorithms XGBoost, random forest, and kNN with the highest accuracy of 65.26%, Precision = 63.71%, Recall = 65.81%, and F1 score = 64.74%.

Madushanka and et al. presented a new concept of cognitive learning technique to predict customer churn behavior [16]. The model is designed into two phases, dataset clustering phase, and model training phase. During the clustering phase, Kernelized Growing Self Organizing Map (KGSOM) and Growing Self Organizing Map (GSOM) techniques were implemented to cluster the data evenly. Later the pruning technique is applied to reduce the number of nodes and obtain faster and better clusters. Models such as SVM, random forests, and logistic regression were trained with KGSOM preprocessed data. It was observed that KGSOM has a good contribution to improving the prediction accuracy of the model.

Multiple supervised classification algorithms like support vector machine, random forest, Naïve Bayes, kNN, decision tree, and logistic regression are implemented to help the HR team with employee turnover prediction [17], [18], [19]. Various performance measure metrics like accuracy, confusion matrix, precision, recall, specificity are used to compare the performance of each model.

Shang introduced a new idea of predicting employee turnover using survival analysis [20]. Survival analysis is a time-dependent and events occurrence statistical technique that computes the probability of occurrence of an event at a given point of time. The author implemented CoxRF

(CoX proportional hazards model with Random Forest) algorithm in conjunction with survival analysis to build the predictive model. The probability of occurrence of an event is computed which is considered as a feature to train the model on predicting employee turnover. Using the Kaplan-Meier method it was discovered that factors such as gender (female employees), external environmental factors (GDP growth), and industry type (IT) have a great influence on employee turnover. It was proved that the CoxRF method outperformed other algorithms like SVM, naïve Bayes, logistic regression, decision tree, XGBoost, and random forest with a reported accuracy of 0.85, recall = 0.757, F1-measure = 0.815 and AUC = 0.842.

All the related works that were discussed earlier focused on the performance metrics of their models and had no specific information on whether the used dataset was balanced or imbalanced. To cover this research gap, we decided to use an imbalanced IBM HR dataset and experimented with various balancing techniques such as Stratified K-Fold cross-validation, Over and Under Sampling, and SMOTETOMEK. We also compared the performance of the Logistic Regression and SVM model before and after the balancing techniques were implemented. In our project, we experimented with some of the new feature selection techniques such as the χ^2 test, ExtraTree classifier, and Information Gain which were not discussed in any of the earlier studies.

IV. MACHINE LEARNING

Machine learning is a statistical learning framework that falls under the branch of artificial intelligence. It has the self-learning ability to detect hidden patterns from the data and can make predictions and decisions based on its learning. Fig 1. shows the hierarchical relationship of machine learning with other studies of computer science. Machine learning is composed of three-step processes Data, Model, and Action [21]. Data is split into two parts training and testing data. Training data is used during the learning phase where the model tries to learn the patterns and gain empirical information and knowledge. After the training phase, the testing data is used to check how well the trained model performed. Finally, the designed model will decide to solve the problem. In our project, we have chosen a data split ratio of 70:30 where 70% of the entire dataset is considered as training data and the rest 20% as testing data. Various performance metrics like accuracy, precision, recall, sensitivity, confusion matrix, ROC (Receiver Operating Characteristic) , and AUC (Area under the curve, etc. are used to evaluate the performance of the model. With an iterative training process and hyperparameter tuning, the model gets better than its earlier version. One of the salient features of machine learning technology is that it does not require any direct input command from humans.

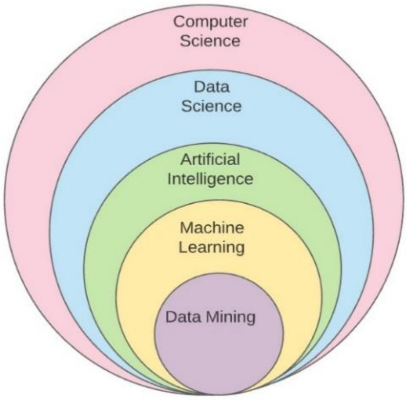


Fig. 1 Hierarchical level of machine learning

A very commonly raised question is when do we need machine learning? In today's world, machine learning is used almost everywhere. The tasks that humans are doing every day such as driving, speech recognition, etc. can be implemented via machine learning. Other complex tasks that are beyond human capabilities such as weather prediction, analysis of astronomical data, etc. are made possible by machine learning techniques. Since machine learning requires no input commands, tasks or problems which require adaptivity on the input data can also be solved through machine learning [22]. Machine learning is broadly divided into 3 categories: supervised learning, unsupervised learning, and reinforcement learning. Each of the learning methodologies has its kind of problems solving capabilities and application scope. Fig 2. Shows the different types of machine learning algorithms.

a. Supervised learning

In supervised learning, we have a dataset with input variables (X) as well as labeled output variables (Y). $X^T= (X_1, X_2,\ldots\ldots\ldots.., X_p)$ represents the vector of input features or the independent variable and Y represents the response or a dependent variable. Using the past input and output pairs, a supervised learning algorithm discovers a true function or a rule that gives the best prediction of Y on the given values of X. The derived rule or function will map X->Y. Classification and regression are the two main problems that are solved by supervised learning.

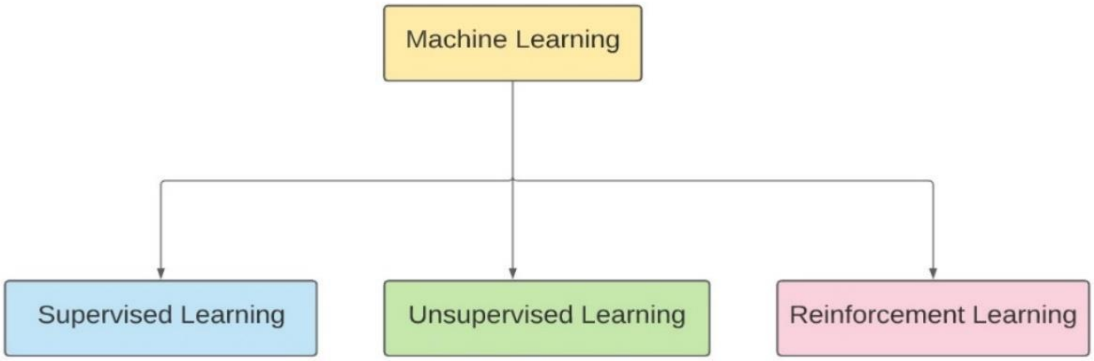


Fig. 2 Types of Machine learning

A classification problem is solved by the model predicting some qualitative, discrete, or categorical values such as predicting a male or female, detecting cancer or not, and whether a patient survives or not, etc. [23]. Classification problems can be further divided into binary and multi-class classification. Fig.3 shows an example of binary and multi-class classification. Binary classification is a classification that contains binary or only two class labels. Churn or not and buy or not etc. are some examples of binary classification whereas, in multi-class classification, it contains more than two class labels. Human face classification and DNA classification are some examples of multi-class classification [24]. The employee churn problem that we are trying to solve in our project is considered as a binary classification problem where class label 0 represents churn and 1 as non-churn. Two supervised machine learning algorithms, Logistic Regression, and SVM are used to solve our classification problem. On the other hand, the regression problem is solved by predicting continuous or quantitative variables such as predicting an age, salary or price, etc. Naïve Bayes, Decision Tree, Random Forest, SVM, Linear Regression, Logistic Regression, and k-nearest neighbor, etc. are some of the most commonly used supervised learning algorithms.

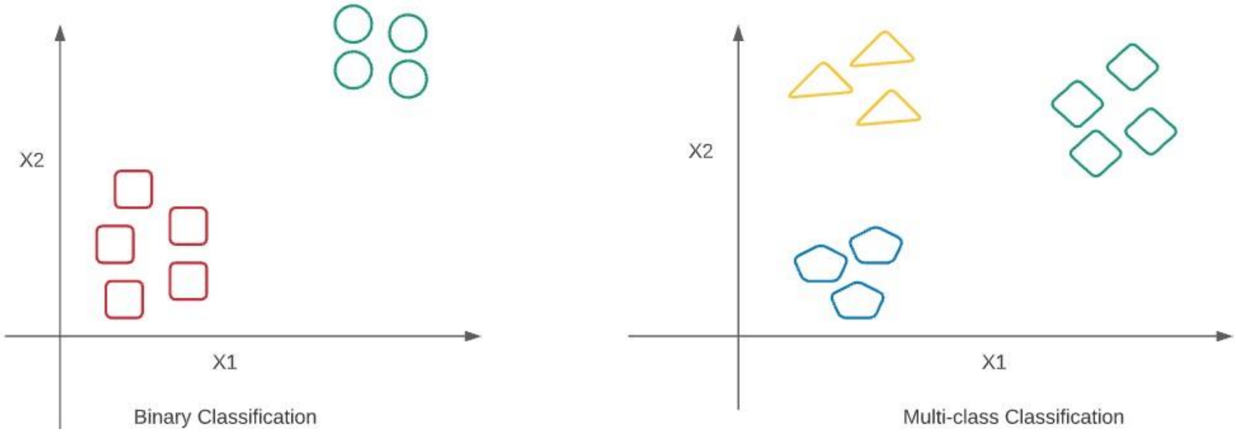


Fig. 3 Binary and Multi-class Classification

b. Unsupervised learning

In unsupervised learning, we have the input variables X but do not have any output variables or labels Y. In other words, the model is trained only with the input data such that it tries to detect hidden patterns or define a rule out of them. Generally, the descriptive models are built using an unsupervised learning algorithm where the model learns through clustering by forming a cluster of data points with similar characteristics or through the associate rule which discovers the rule that describes the data [25]. K-means clustering, DBSCAN, and association rule are some of the unsupervised learning algorithms. An unsupervised learning algorithm is widely used in anomaly detection such as detection of fraud transactions, malware, fake customer reviews, etc.

c. Reinforcement learning

Unlike supervised algorithms, reinforcement learning does not rely on labeled data; rather, it uses a unique concept of reward and penalty to train the model. A reward is granted when the model performs the task correctly and a penalty is issued for any mistakes performed by the model. A positive score value is represented as a reward and a penalty is denoted by a negative score. The positive and negative score is used as feedback to iteratively improve the performance of the model; i.e., the model is in a continuous phase of learning, leveraging the feedback received from the previous iteration. Reinforcement learning can be related to a real-life example of a kid trying to learn to play a computer game [21]. Since the game is completely new to the kid, so, in the beginning, the kid makes multiple mistakes, but as he moves on, he learns from his previous mistakes and using his learning he keeps on improving his gaming skills in the subsequent play. Finally, the kid will complete the game. Q-Learning, Temporal Difference (TD), and Deep Adversarial Networks are some of the common reinforcement learning algorithms. Fig 4. Shows the block diagram of reinforcement learning.

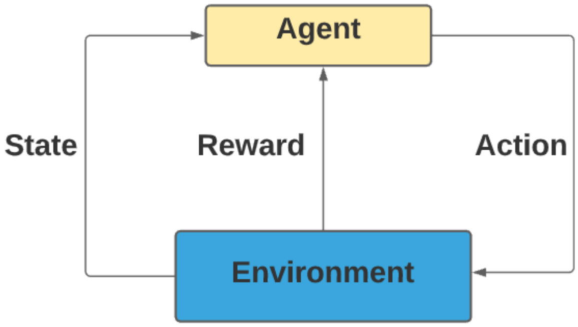


Fig. 4 Block diagram of Reinforcement Learning

V. ALGORITHM SELECTION

In machine learning, there is a famous theorem called “No Free Lunch”. According to that theorem, no such single model or algorithm exists that can perfectly solve every kind of problem [22]. Algorithm selection is one of the preliminary and the most important decision that we need to make before we start jumping on solving the problem. Failure on choosing an appropriate algorithm can result in the poor performance of the model thus receiving an unsatisfactory result. In our project, factors such as problem type, dataset size, computation time, resources, feature correlation, target class types (qualitative or quantitative) were considered while selecting the Logistic Regression and SVM model as our machine learning algorithm.

In this paper, we are trying to solve an employee churn classification problem with a predictive model through a supervised learning algorithm, so the pros and cons of various supervised learning algorithms are summarized as follows [26], [27]. Naïve Bayes is a simple and faster-to-implement algorithm that supports a larger dataset. The algorithm is less susceptible to irrelevant features and can be used to predict multi-label classification problems. As a downside, the algorithm makes a Bayesian assumption that the variables of the input data are independent, which may not be true all the time. The algorithm also fails to perform well if the dataset has an unequal distribution of labeled classes. Since Logistic Regression is a simple and effective algorithm that suits best to solve the binary classification problem, therefore we chose it as one of our algorithms in our project. Maximum likelihood estimates and stochastic gradient descent methods can be well implemented to best fit the model. Since the algorithm is a linear model, it may not perform well with the non-linear data. The performance of the model may not be good when trained with irrelevant or highly correlated data. In the case of a decision tree classifier, the algorithm has a minimal impact on missing values or irrelevant features present in the dataset.

Normalization or scaling of data is not required with the decision tree algorithm. However, the algorithm is very prone to a common overfitting issue and it takes a longer training time compared to the rest of the other classifiers. kNN or K nearest neighbor makes it simple and easy to implement an algorithm. It makes no prior assumption about the data but the algorithm is sensitive to the outliers and it slows down while processing the larger dataset. In order words, the algorithm performs poorly with higher dimension attributes. Random Forest is an ensemble of decision trees that performs well even with the data having an unequal distribution of labeled classes. The algorithm is not being impacted by the outliers and is free from overfitting issues but the model relies heavily on the selected features hence, we need to be cautious while selecting the features. Support vector machine is another popular classifier that suits best for solving a binary classification problem. The algorithm has the capability of working with linear as well as nonlinear data. One of the big advantages of the kernel SVM algorithm is that it makes use of various kernel methods which can transform lower dimension data to a higher dimension. During the hyperparameter turning of the SVM model, we used Radial Basis Function (rbf) as one of the parameters for the kernel method. SVM is a memory-intensive algorithm and it performs slow with a larger dataset having overlapped classes. In addition to that, choosing appropriate hyperparameters is a bit challenging with the SVM algorithm. After comparing the pros and cons of supervised classification-based algorithms we decided to choose Logistic Regression and Support Vector Machine as our primary algorithms.

a. Logistic Regression

Logistic Regression is a linear supervised classification algorithm that is mainly used to solve binary classification problems. Instead of making the direct prediction, the model outputs the probability of a point that falls on one of the two sides of the plane. Let X be the predictor variable and Y be its response variable. Then the relationship between X and Y can be mathematically represented as follows

$$Y \approx \beta 0 + \beta 1X \qquad \text{............ equation 1}$$

Mathematically, β0 and β1 are considered as the y-intercept and slope of a line; whereas, in machine learning models, it is considered as the coefficients/parameters or the weights [23]. Specifically, β0 is known as the bias of the model. Considers as a binary classification of the problem, there are two class labels 0 and 1 (default class). According to posterior probabilities, the probability of having 1 as the class label for a given X is provided by the following equation [23]

$$p(X) = \Pr(Y = 1|X) \quad \text{............ equation 2}$$

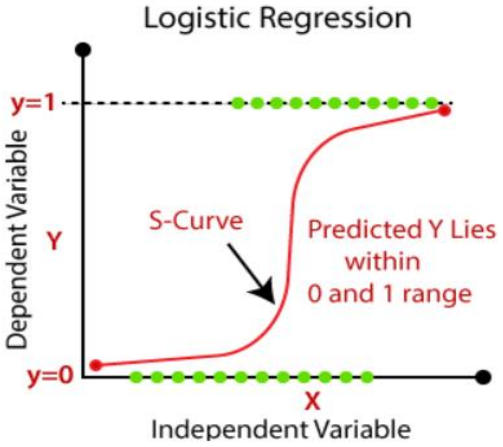


Fig. 5 Sigmoid/Logistic function

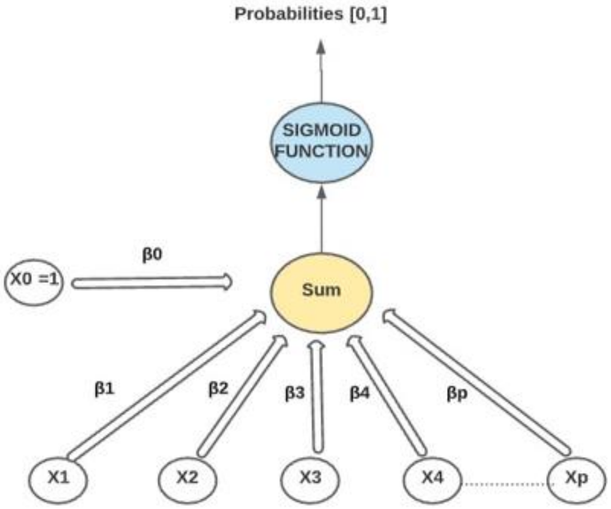


Fig. 6 Logistic Regression model

From the linear regression model, we have

$$p(X) = \beta_0 + \beta_1 X. \quad \text{............ equation 3}$$

Approaching with linear regression, the probabilities obtained p(X) may not fall within the range of [0,1] so, to overcome this issue we introduce a special function called logistic or sigmoid function. The sigmoid function produces an S-shaped curve line as shown in fig. 5. Fig. 6 shows the diagram of a logistic regression model in which the sigmoid function takes the sum of the predicted values and its coefficients, maps the predicted values to corresponding probabilities. The output of the sigmoid function will always remain in the boundary range of [0,1]. Mathematically it can be represented by $0 \leq \sigma(x) \leq 1$. The sigmoid function can be represented with the following equation

$$\sigma(x) = 1 / (1 + e^{-X})$$

$$= e^{X} / (1 + e^{X}) \quad \text{............ equation 4}$$

Applying a sigmoid function to equation 3,

$$p(X) = \frac{e^{\beta_0+\beta_1 X}}{1 + e^{\beta_0+\beta_1 X}}$$

$$\frac{p(X)}{1 - p(X)} = e^{\beta_0+\beta_1 X}$$

The ratio of p(X) / 1-p(X) is called the odds and can take the value from 0 to ∞. Taking the logarithm base 10 on both sides of the equation, we get the following equation:

$$\log\left(\frac{p(X)}{1 - p(X)}\right) = \beta_0 + \beta_1 X. \quad \text{............. equation 5}$$

The ratio of log (p(X) / 1-p(X)) is called log-odds or logit. To fit the logistic regression model, we need to best estimate the coefficient values (β0 and β1) such that, using those coefficient values, the model p(X) yields a number that is as close to 1 for a default class and as close to 0 for the non-default class. It is an iterative process of estimating the coefficient values until and unless the accuracy of the model reaches an acceptable point. Methods such as least squares, maximum likelihood, and stochastic gradient descent are available to estimate the coefficient values but since maximum likelihood is a general method that can fit both the linear and nonlinear model so we decided to go with the maximum likelihood method to fit the logistic regression model. Mathematically, the maximum likelihood function can be represented by the equation provided below.

$$\ell(\beta_0, \beta_1) = \prod_{i:y_i=1} p(x_i) \prod_{i':y_{i'}=0} (1 - p(x_{i'})).$$

The overall goal is to best estimate the values of β0, β1which can maximize the likelihood function.

b. Support Vector Machine

Support vector machine is a supervised learning algorithm that can be used to solve

both classifications as well as regression problems. Particularly it serves best for solving a binary classification problem which is one of the reasons we selected SVM as our algorithm to predict employee churn. The goal of the SVM algorithm is to find the optimal hyperplane that can classify the data points accurately. A hyperplane is a flat plane of p-1 dimension in a p dimensional space. E.g. in a two-dimensional space, a hyperplane would be one dimension which means a simple line. In our project, the hyperplane acts as a separating plane that will separate the churn employees from non-churn. Mathematically it can be represented by the following equation:

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 = 0$$

In a two-dimensional space, the hyperplane will divide the space into two halves and it will act as a separating plane for classifying the data points. Let X = (X1, X2, ……., Xp) be a vector of features in p-dimensional space, then the equation $\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p > 0$

indicates that X lies on the one side of the plane representing the class label with positive value +1, and the equation $\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p < 0$ indicates the other side of the plane with a negative value -1. So, the model will classify the observations into one of the two class labels {-1, +1}. As shown in Fig. 7 [23], there could exist multiple hyperplanes that still classifies the data points correctly but among all those hyperplanes, the optimal hyperplane contains two parallel margins that passes through one or more of the nearest data points of both the classes or support vectors while keeping the maximum marginal distance. So, to obtain that optimal hyperplane, a classifier called maximal margin classifier can be used. Fig. 8 [23] shows various components of a maximal margin classifier. The maximal margin classifier is based on the maximum margin distance rule. While following the rule, the optimal hyperplane may continuously shift its position just to accommodate one or two newly added outliers. So, to avoid that situation a classifier called soft margin or support vector classifier can be implemented. For the soft margin classifier, it is completely acceptable to have a few observations be misclassified. In other words, it is ok to classify some observations on the incorrect side of the margin or even on the incorrect side of the planes.

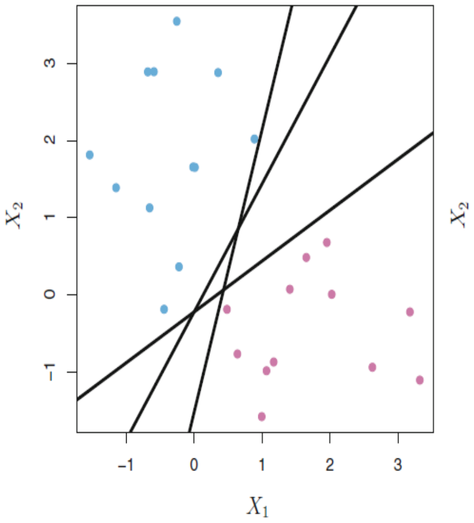


Fig. 7 Multiple hyperplanes

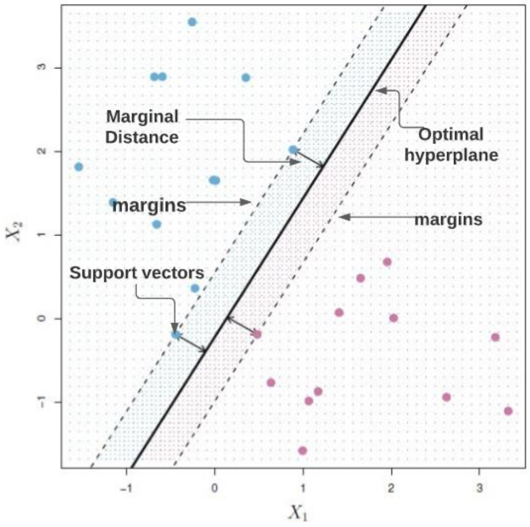
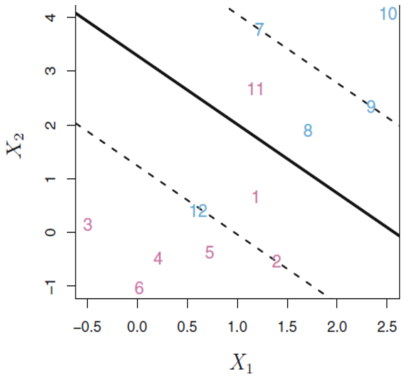
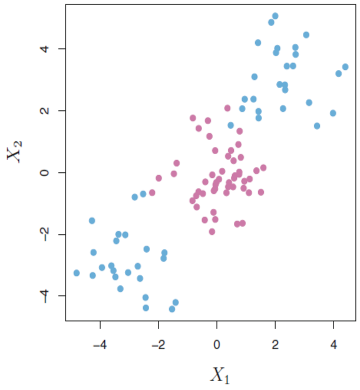


Fig. 8 Maximal margin classifier

Fig. 9 [23] shows a few observations that are classified on the wrong side of the margin and the hyperplane. The observations that lie on the margins or the wrong side of the margin and hyperplane are known as the support vectors and they do affect the performance of the classifier. In the case of the support vector classifier, it is up to the user to decide the number of misclassifications that the model can accept and this is done through configuring a tuning parameter C. In the real world, there can be a dataset with an overlapping class or a group of nonlinear observations where even the maximal margin classifier fails to come up with a separating hyperplane. In other words, a linear decision boundary will not work classifying the observation correctly so in that situation, a method called a support vector machine can help mitigate the issue. Fig. 10 a [23] shows a group of non-linearly separable observations and fig. 10 b [23] shows the linear boundary created by the support vector classifier that has poorly classified the observations. Support vector machine is an extended version of support vector classifier which uses the concept of kernel methods. Using the kernel tricks, the algorithm can transform a lower dimension data into a higher dimension so that a decision boundary can be created to classify the non-separable observations easily. Polynomial, RBF, and Sigmoid kernels are some of the most common kernel methods that are used in the support vector machine algorithm to classify linear and nonlinear





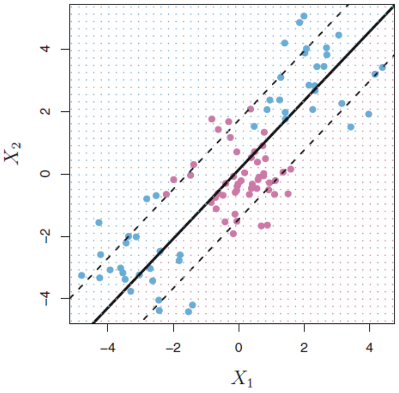


Fig. 9. Misclassified observations Fig.10 a. Nonlinear observations Fig. 10 b. Poor classifications

overlapping observations. In our project, we used rbf as a kernel method parameter which helped to classify our churn and non-churn employee with higher accuracy. Fig. 11 [21] shows how a two-dimension data space is transformed into three-dimensions and how a hyperplane is used to classify the observation.

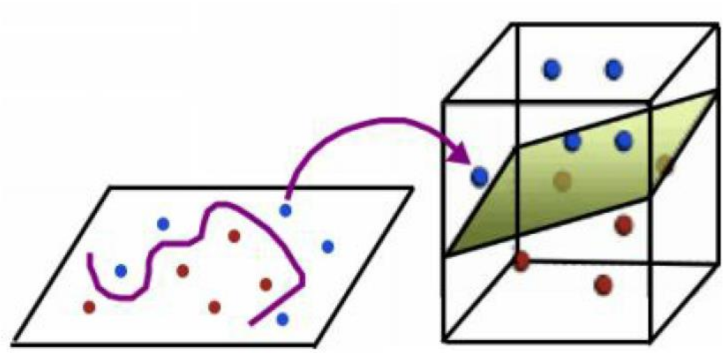


Fig. 11 Transforming two-dimension into three-dimension space using kernel method

VI. IMPLEMENTATION

As per the implementation of this project, we chose Python as our primary programming language and Visual Studio Code as the integrated development environment (IDE). All the Python scripts are coded on Python Version 3.9. numpy, pandas, array, collections, seaborn, matplotlib, sklearn, imblearn, textwrap, time and random, etc. are a few of the libraries that are being referenced in the project. The implementation of the entire project is divided into 5 different phases and they are as follows:

a. Exploratory Data Analysis

b. Feature Selection

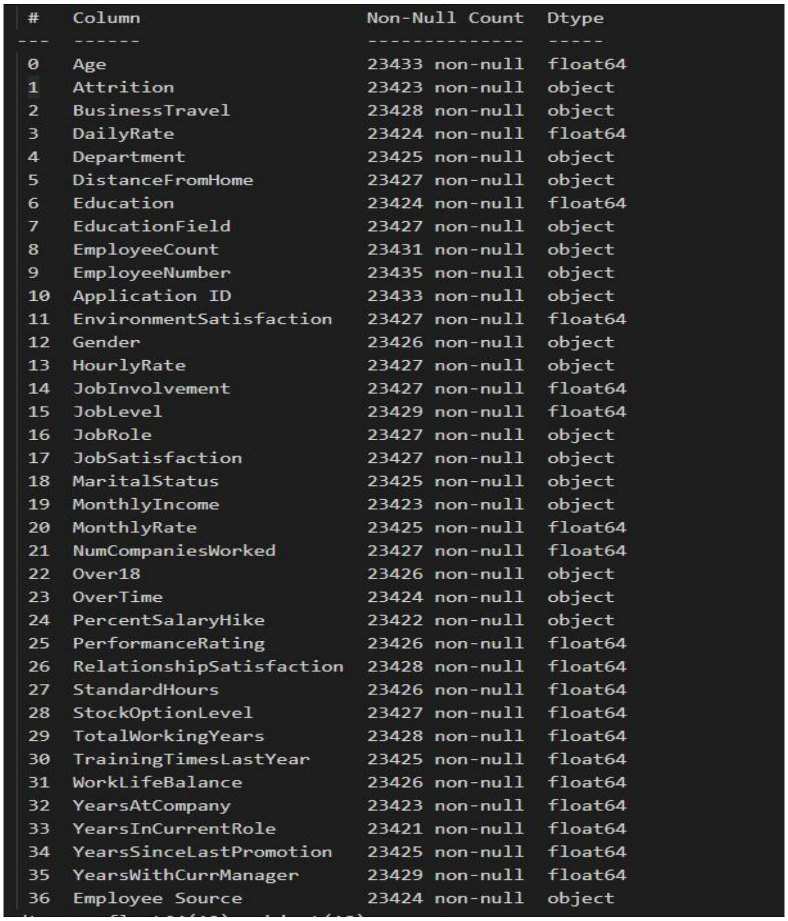
c. Model Building

d. Hyperparameter Tuning

e. Performance Evaluation

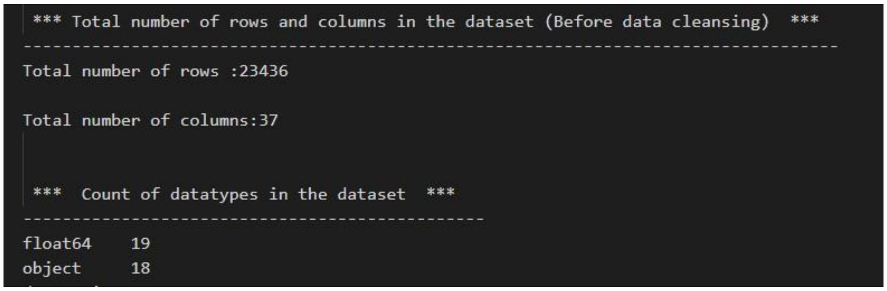
a. Exploratory Data Analysis

Exploratory Data Analysis (EDA) started with the data collection process. For this project, we extracted an IBM HR analytics dataset of 4.3 MB (approx.) that was made publicly available in https://kaggle.com. The dataset contained the records of 23,436 employees who had either left the company or were still being employed. There were 37 different attributes to hold the various information of the employees. Fig. 12 shows the list of all the attributes present in the dataset and Fig. 13 shows the count of all the records and the various data types. Performing a quick data quality check, we identified that the dataset contained some dirty data such as data mismatch and null values. Fig. 14 and Fig. 15 show the count of dirty values and null values in the dataset. Since data is the key element in this project and knowing that the dataset is not 100% clean



```
 #   Column                    Non-Null Count  Dtype  
---  ------                    --------------  -----  
 0   Age                       23433 non-null  float64
 1   Attrition                 23423 non-null  object 
 2   BusinessTravel            23428 non-null  object 
 3   DailyRate                 23424 non-null  float64
 4   Department                23425 non-null  object 
 5   DistanceFromHome          23427 non-null  object 
 6   Education                 23424 non-null  float64
 7   EducationField            23427 non-null  object 
 8   EmployeeCount             23431 non-null  object 
 9   EmployeeNumber            23435 non-null  object 
 10  Application ID            23433 non-null  object 
 11  EnvironmentSatisfaction   23427 non-null  float64
 12  Gender                    23426 non-null  object 
 13  HourlyRate                23427 non-null  object 
 14  JobInvolvement            23427 non-null  float64
 15  JobLevel                  23429 non-null  float64
 16  JobRole                   23427 non-null  object 
 17  JobSatisfaction           23427 non-null  object 
 18  MaritalStatus             23425 non-null  object 
 19  MonthlyIncome             23423 non-null  object 
 20  MonthlyRate               23425 non-null  float64
 21  NumCompaniesWorked        23427 non-null  float64
 22  Over18                    23426 non-null  object 
 23  OverTime                  23424 non-null  object 
 24  PercentSalaryHike         23422 non-null  object 
 25  PerformanceRating         23426 non-null  float64
 26  RelationshipSatisfaction  23428 non-null  float64
 27  StandardHours             23426 non-null  float64
 28  StockOptionLevel          23427 non-null  float64
 29  TotalWorkingYears         23428 non-null  float64
 30  TrainingTimesLastYear     23425 non-null  float64
 31  WorkLifeBalance           23426 non-null  float64
 32  YearsAtCompany            23423 non-null  float64
 33  YearsInCurrentRole        23421 non-null  float64
 34  YearsSinceLastPromotion   23425 non-null  float64
 35  YearsWithCurrManager      23429 non-null  float64
 36  Employee Source           23424 non-null  object 
```

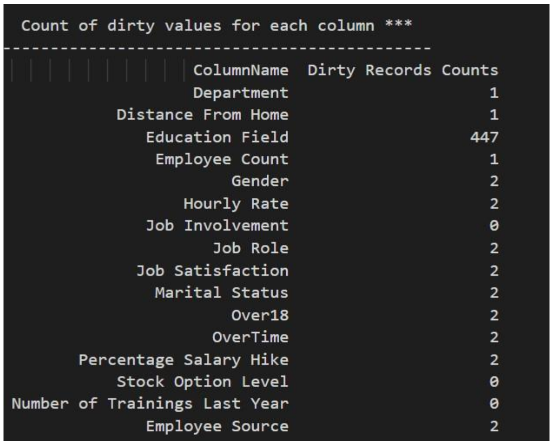
Fig. 12 List of columns with its data types and non-null counts



```
 *** Total number of rows and columns in the dataset (Before data cleansing)  ***
---------------------------------------------------------------------------------
Total number of rows :23436

Total number of columns:37


 ***  Count of datatypes in the dataset  ***
----------------------------------------------
float64    19
object     18
```

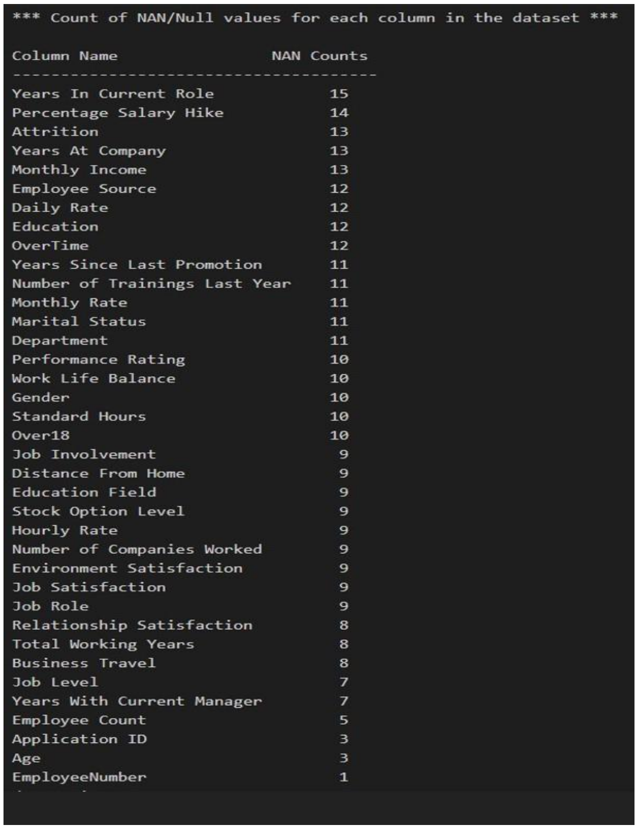
Fig. 13 Count of rows, columns, and data types in the dataset

so, we performed data cleansing to obtain valid, consistent, and accurate data. Based on the type of data that the attributes were holding off; various data cleansing methods were implemented. For the categorical data, we applied a random sampling method were for all the columns with dirty data, a randomly picked item from the clean subset were used to update the corresponding dirty data. Whereas for the continuous data, a median value was computed for a column and then the computed value was used to update the corresponding dirty data. Unlike the mean, the median is robust and less impacted by the outliers; hence, we decided to use the median value. For the non-relevant attributes such as Employee Number, Application ID, and Employee Count, if they hold any dirty data, we just deleted those corresponding records from the dataset. Once the data cleansing task was completed, we validated the data by checking the counts of Null values and the total records in the cleaned dataset. Fig 16. shows that there are 0 Null values and Fig 17. shows the total number of rows and columns in the cleaned dataset. Exploring the cleaned dataset, we observed 3 different data types as Nominal, Ordinal and Numerical data. Attributes such as Employee Source, Education Field and Department, etc. have no inherited order, so these



```
 Count of dirty values for each column ***
--------------------------------------------
                  ColumnName  Dirty Records Counts
                  Department                     1
          Distance From Home                     1
             Education Field                   447
              Employee Count                     1
                      Gender                     2
                 Hourly Rate                     2
              Job Involvement                     0
                    Job Role                     2
            Job Satisfaction                     2
              Marital Status                     2
                      Over18                     2
                    OverTime                     2
      Percentage Salary Hike                     2
          Stock Option Level                     0
Number of Trainings Last Year                     0
             Employee Source                     2
```

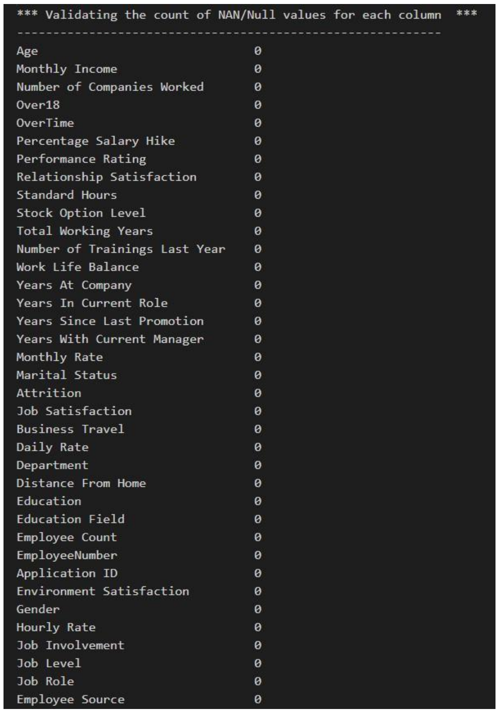
Fig. 14 Count of dirty data



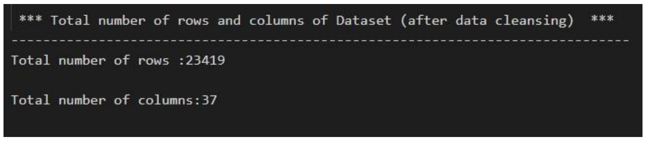
```
*** Count of NAN/Null values for each column in the dataset ***

Column Name                NAN Counts
-------------------------------------
Years In Current Role          15
Percentage Salary Hike         14
Attrition                      13
Years At Company               13
Monthly Income                 13
Employee Source                12
Daily Rate                     12
Education                      12
OverTime                       12
Years Since Last Promotion     11
Number of Trainings Last Year  11
Monthly Rate                   11
Marital Status                 11
Department                     11
Performance Rating             10
Work Life Balance              10
Gender                         10
Standard Hours                 10
Over18                         10
Job Involvement                 9
Distance From Home              9
Education Field                 9
Stock Option Level              9
Hourly Rate                     9
Number of Companies Worked      9
Environment Satisfaction        9
Job Satisfaction                9
Job Role                        9
Relationship Satisfaction       8
Total Working Years             8
Business Travel                 8
Job Level                       7
Years With Current Manager      7
Employee Count                  5
Application ID                  3
Age                             3
EmployeeNumber                  1
```

Fig. 15 Count of Null values

attributes are categorized as nominal data; Whereas, Percent Salary Hike, Job Level, and Job Satisfaction, etc. have some inherent order and they are considered as ordinal data. Performing statistical analysis on nominal and ordinal data is not meaningful and not possible; hence, we did not carry out any statistical analysis for those attributes. In contrast, attributes such Age, Monthly Income and Distance from Home, etc. are the numerical data so we performed some simple statistical analysis such as min, max, median, range, and standard deviation. Fig. 18 shows the statistical analysis of attributes with the numerical data types.



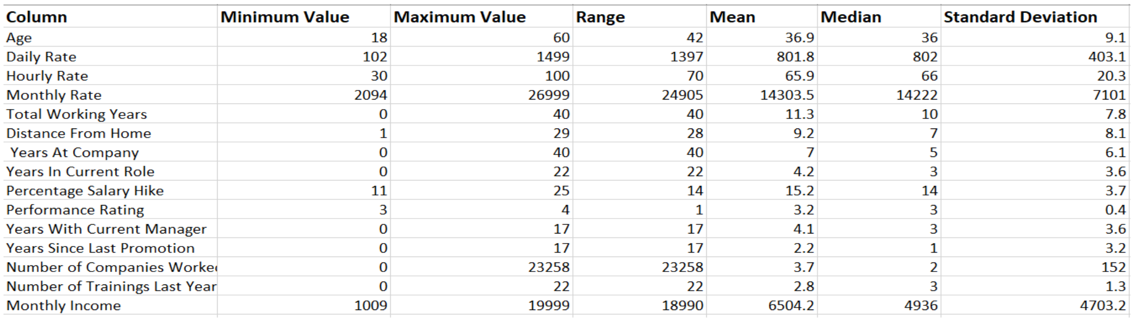
```
*** Validating the count of NAN/Null values for each column  ***
----------------------------------------------------------
Age                              0
Monthly Income                   0
Number of Companies Worked       0
Over18                           0
OverTime                         0
Percentage Salary Hike           0
Performance Rating               0
Relationship Satisfaction        0
Standard Hours                   0
Stock Option Level               0
Total Working Years              0
Number of Trainings Last Year    0
Work Life Balance                0
Years At Company                 0
Years In Current Role            0
Years Since Last Promotion       0
Years With Current Manager       0
Monthly Rate                     0
Marital Status                   0
Attrition                        0
Job Satisfaction                 0
Business Travel                  0
Daily Rate                       0
Department                       0
Distance From Home               0
Education                        0
Education Field                  0
Employee Count                   0
EmployeeNumber                   0
Application ID                   0
Environment Satisfaction         0
Gender                           0
Hourly Rate                      0
Job Involvement                  0
Job Level                        0
Job Role                         0
Employee Source                  0
```



```
*** Total number of rows and columns of Dataset (after data cleansing)  ***
--------------------------------------------------------------------------
Total number of rows :23419

Total number of columns:37
```

Fig. 16 Validating count of null values

Fig. 17 Count of records after data cleansing



Column	Minimum Value	Maximum Value	Range	Mean	Median	Standard Deviation
Age	18	60	42	36.9	36	9.1
Daily Rate	102	1499	1397	801.8	802	403.1
Hourly Rate	30	100	70	65.9	66	20.3
Monthly Rate	2094	26999	24905	14303.5	14222	7101
Total Working Years	0	40	40	11.3	10	7.8
Distance From Home	1	29	28	9.2	7	8.1
Years At Company	0	40	40	7	5	6.1
Years In Current Role	0	22	22	4.2	3	3.6
Percentage Salary Hike	11	25	14	15.2	14	3.7
Performance Rating	3	4	1	3.2	3	0.4
Years With Current Manager	0	17	17	4.1	3	3.6
Years Since Last Promotion	0	17	17	2.2	1	3.2
Number of Companies Worke	0	23258	23258	3.7	2	152
Number of Trainings Last Year	0	22	22	2.8	3	1.3
Monthly Income	1009	19999	18990	6504.2	4936	4703.2

Fig. 18 Basic Statistical analysis of continuous data

To understand the data in-depth and get more insights, we used matplotlib and seaborn libraries from Python to visualize the data. The dataset contained a column named Attrition which stored binary values 0 and 1. 0 represent the employees who had left the company and 1 indicated the current employees. To add clarity, we did a label mapping where 0 was mapped to Attrition and 1 to Non-Attrition respectively. Fig. 19 shows the percentage distribution of the employees based on attrition and non-attrition.

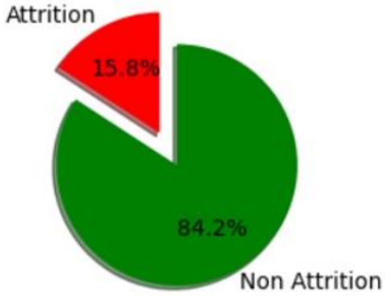


Fig. 19 Pie chart distribution of attrition vs non-attrition employees

The dataset did contain some attributes such as Work-life balance, Job level, and Stock option level, etc. which were categorical but still held numerical values. So, to add better clarity, we mapped those numerical values with a meaningful label. Fig. 20-28 shows the cascaded bar chart of various categorical attributes with respect to the Attrition attribute.

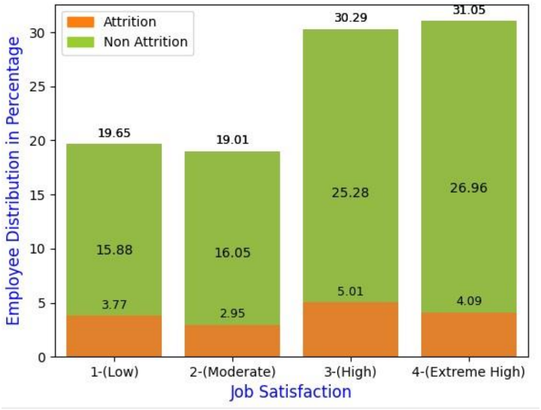


Fig. 20 Distribution of Attrition by Job Satisfaction

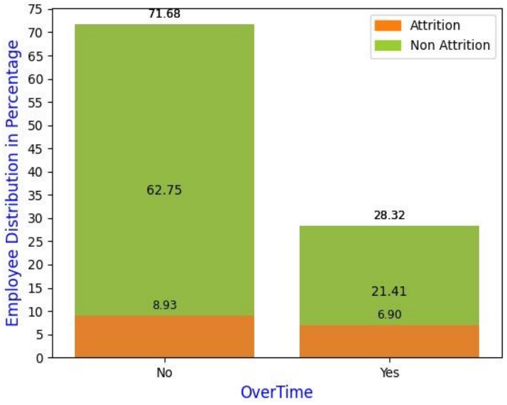


Fig. 21 Distribution of Attrition by Overtime

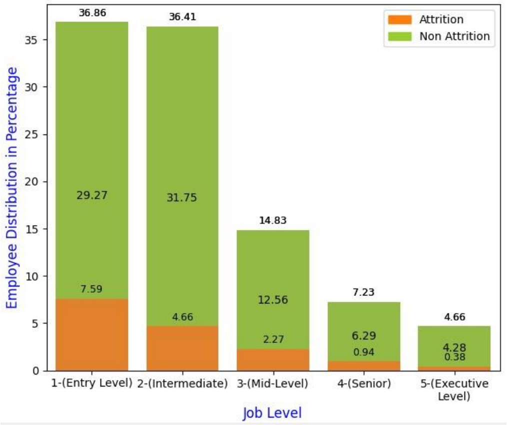


Fig. 22 Distribution of Attrition by Job Level

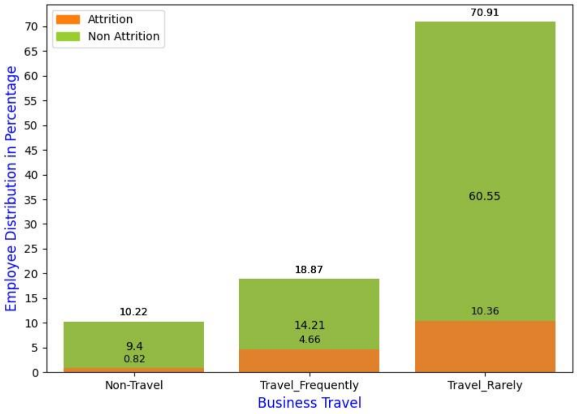
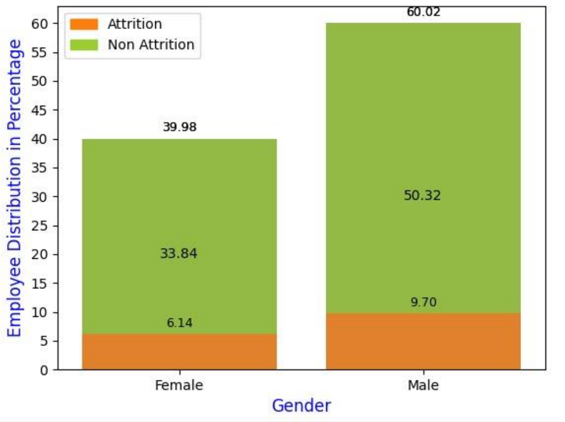


Fig. 23 Distribution of Attrition by Business Travel



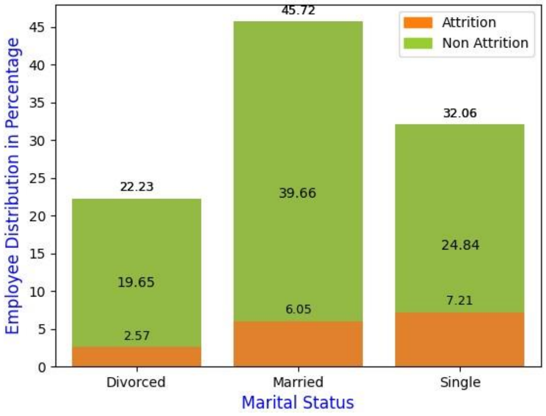


Fig. 24 Distribution of Attrition by Gender

Fig. 25 Distribution of Attrition by Marital Status

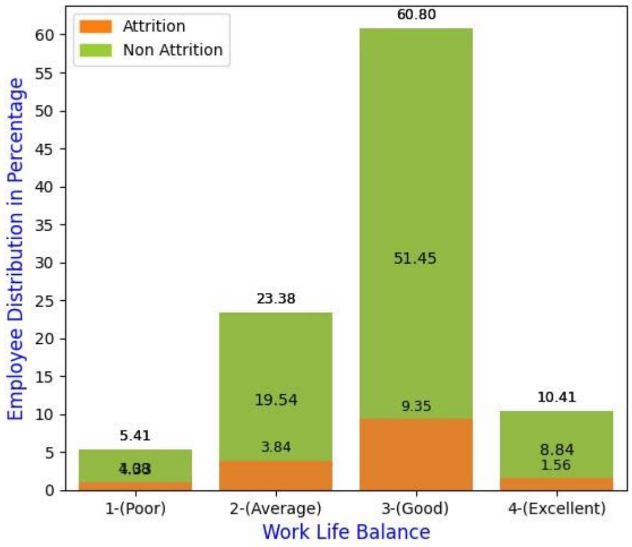


Fig. 26 Distribution of Attrition by Work-Life Balance

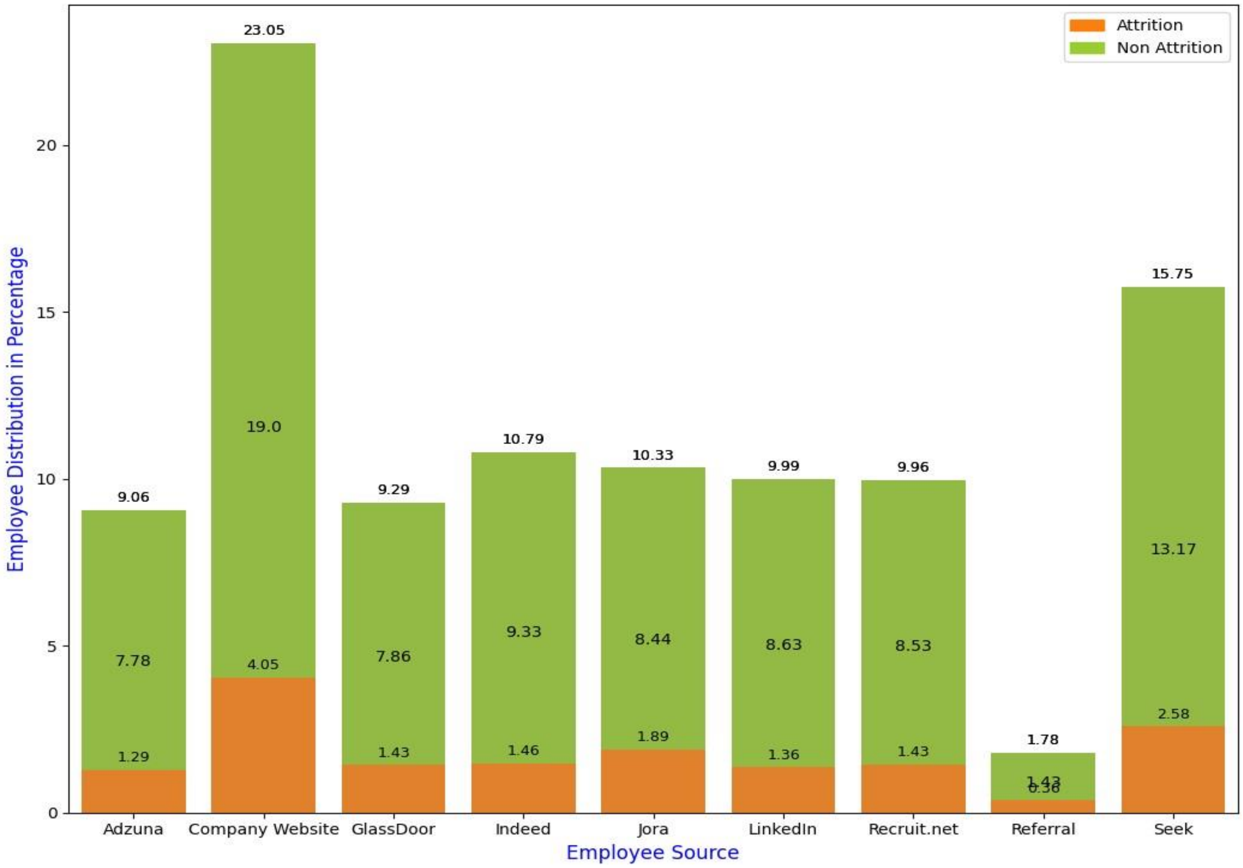


Fig. 27 Distribution of Attrition by Employee Source

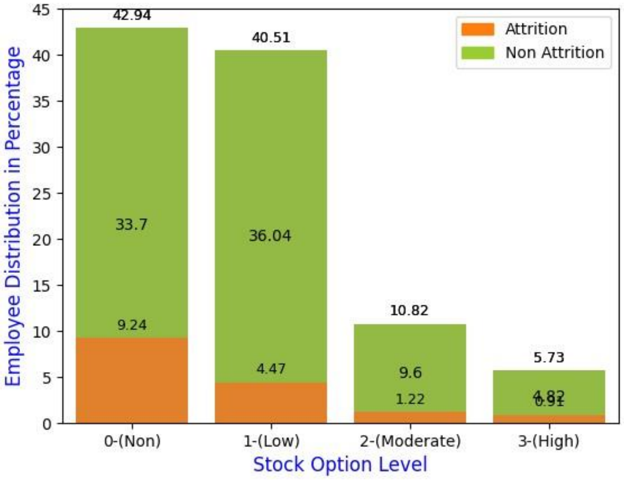
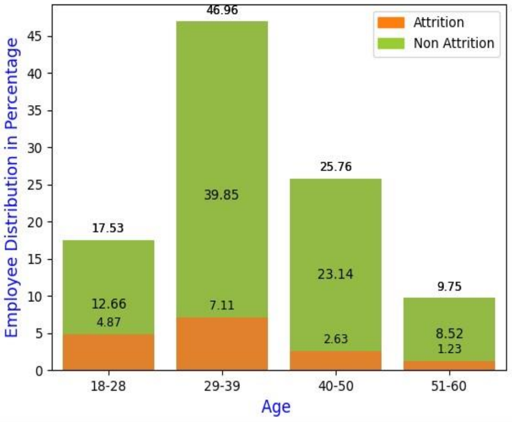


Fig. 28 Distribution of Attrition by Stock Option Level

To graphically represent the continuous attributes, we applied binning. Based on the distribution of the data and their corresponding ranges, the continuous data variables were mapped into bins of appropriate intervals. Fig. 29-34 shows the cascaded bar chart of various continuous attributes with respect to Attrition. Furthermore, to observe the variations in the distribution of continuous data variables, we plotted a distribution plot. In a distribution plot, the data points were represented by a combination of a histogram and a line. Fig. 35 shows the histogram distribution plot for the various continuous attributes.



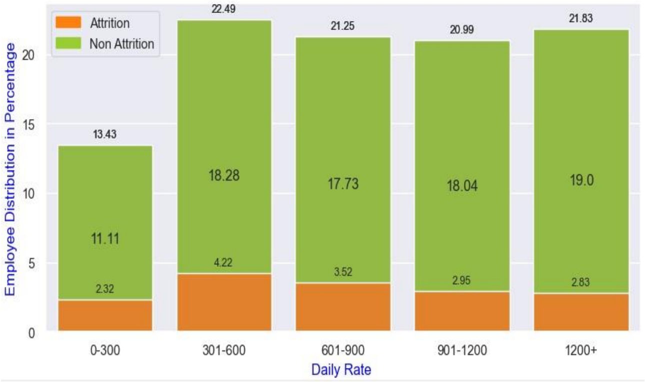


Fig. 29 Distribution of Attrition by Age

Fig. 30 Distribution of Attrition by Daily Rate

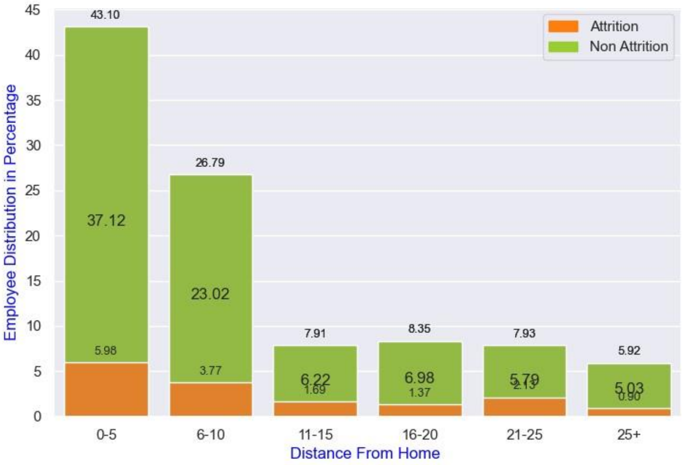


Fig. 31 Distribution of Attrition by Distance from Home

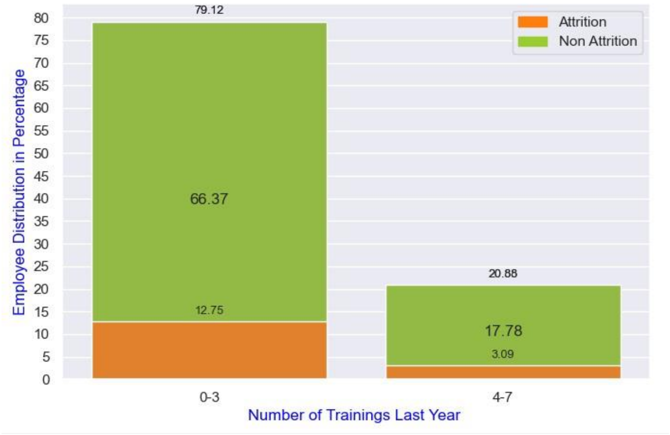


Fig. 32 Distribution of Attrition by Number of Trainings Last Year

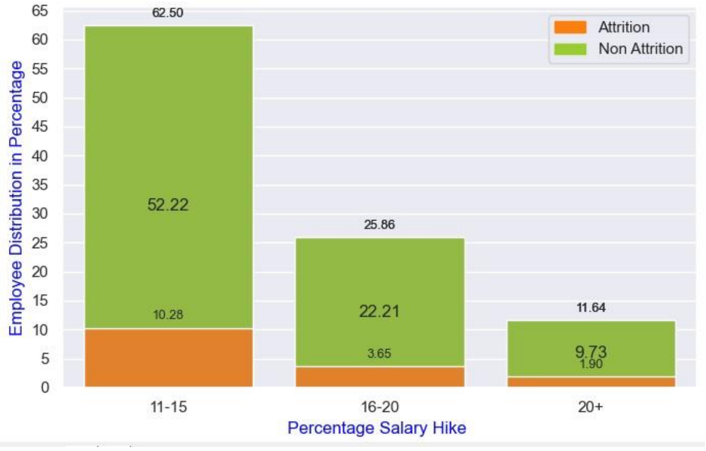


Fig. 33 Distribution of Attrition by Percentage Salary Hike

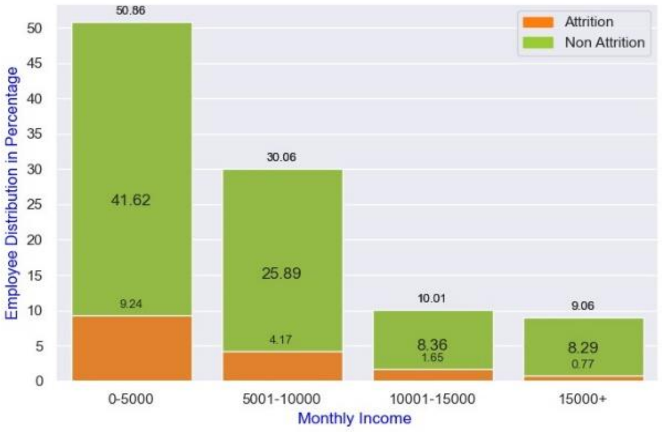


Fig. 34 Distribution of Attrition by Monthly Income

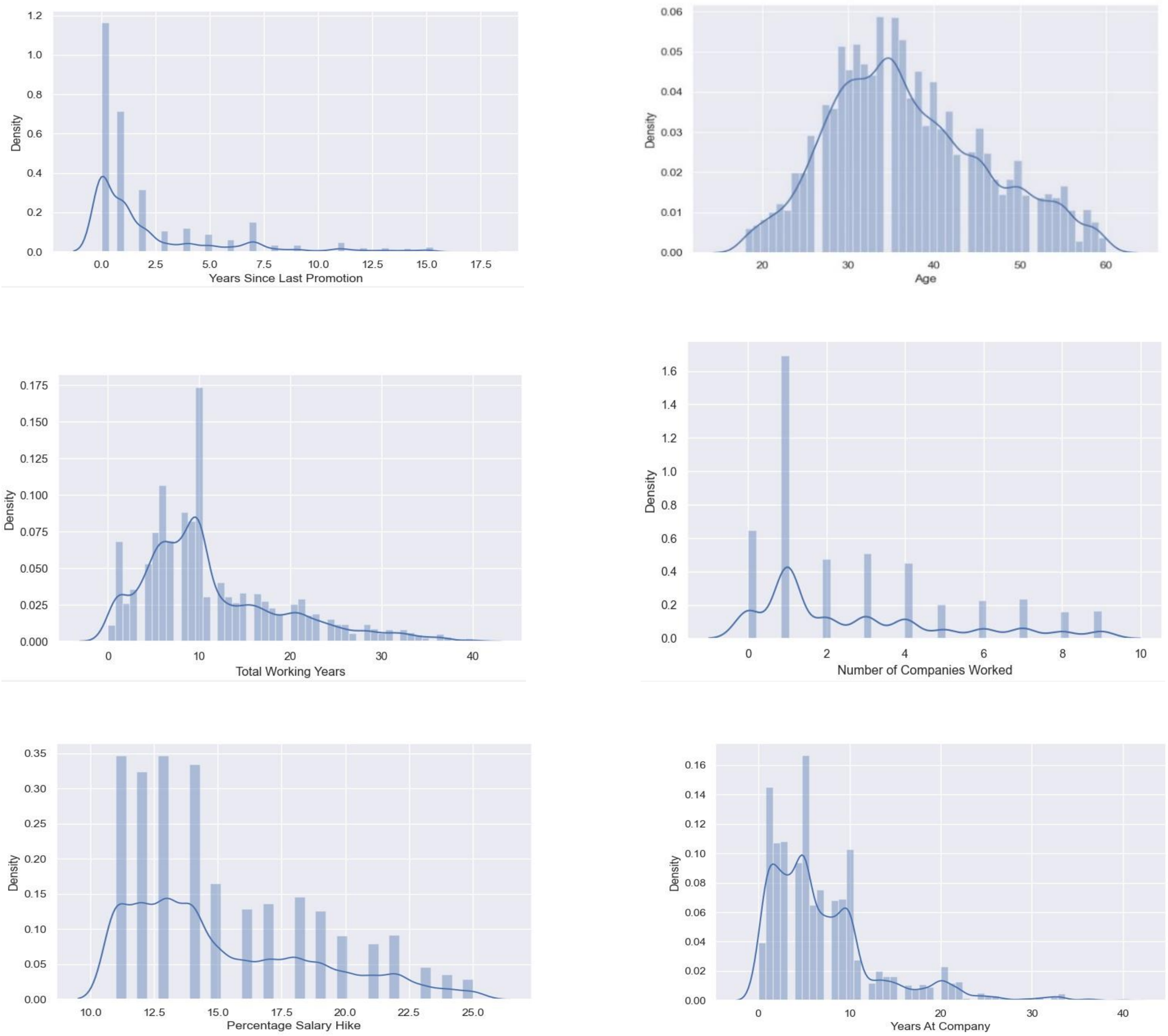


Fig. 35 Histogram distribution of continuous variables

In addition, we also plotted a bar plot for continuous attributes to check if there exist any outliers in our dataset. Fig. 36 shows bar plots for some sample continuous attributes. In some box plots, it is observed that some data points are plotted beyond the maximum whisker boundary and they are considered as an outlier. However, after logical analysis, we considered those observations as valid data points and did not filter them out as an outlier.

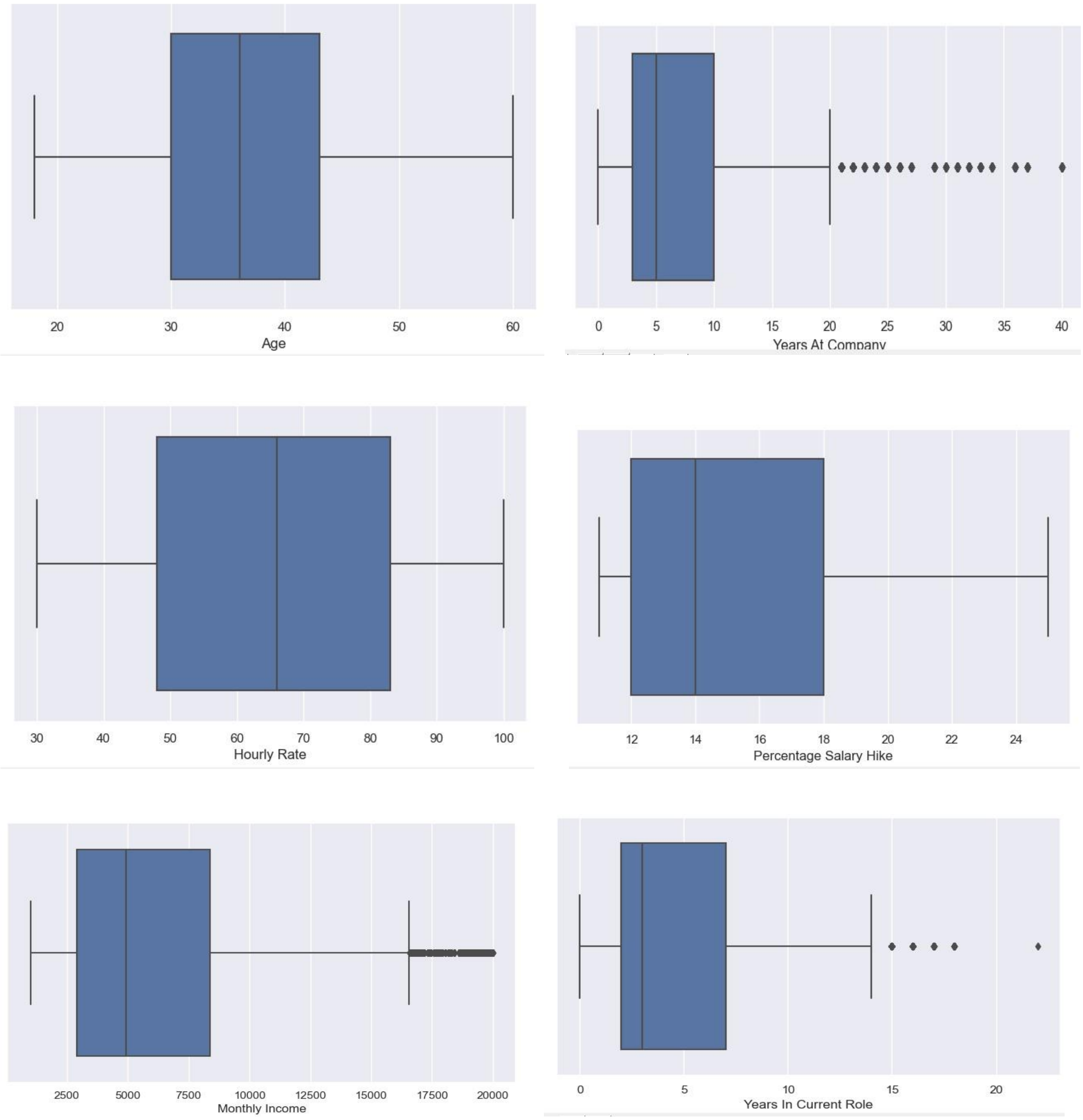


Fig. 36 Box plot of continuous attributes

In the dataset, except the Attrition column, all the other columns are considered as the features or independent variables. Since the Attrition column is dependent upon the independent variables, it is called a dependent variable. Since the Attrition column contained binary values 0 and 1and these values did not provide any specific meanings, we used a label encoder method in

Python to map those values to some meaningful labels. Hence,0 was mapped to attrition and 1 to non-attrition labels respectively. The independent variables or features are used by the machine learning models to make the predictions of the dependent variable. Since the machine learning models understand numbers better than words and may not perform well with the categorical attributes, we used a dummy variable from pandas to map those categorical data into numerical. With a dummy variable technique, a categorical column with a k number of distinct values will result in a k-1 number of additional columns. Since this technique increases the number of columns, the column that has a large number of distinct values can result in very high dimensions resulting in a situation known as the curse of dimensionality.

b. Feature Selection

The dataset contains 1 dependent variable and 36 independent variables or features and not all the features are equally important and relevant. Training the machine learning models without filtering the irrelevant features costs an overhead of computational resources, time and also results in the poor performance of the models. The higher the number of features or dimensions, the more resources are required for the computation, and the longer the models will take time for training and making predictions. This condition is well known as the curse of dimensionality. Feature selection is a technique that helps to pick only the important and relevant features which heavily contribute to determining or predicting the dependent variables. In this project, we have implemented a series of procedures to pick the most important and relevant features from the dataset.

We started the feature selection process by plotting two different heatmaps. One for the continuous and the other for the categorical variables. Each cell of the heatmap indicates the correlation values between the attributes listed on the x and y-axis. A correlation threshold value of 0.05 was set. In other words, any independent variable that holds the correlation values of greater than or equal to 0.05 with respect to the dependent variable is considered as highly correlated, and that particular independent variable is chosen for further validation. At the same time, if the correlation values between two independent variables are greater than or equal to 0.05, then in that case, only one independent variable is chosen for the next step. Fig. 37 and 38 show the heatmap of the Attrition attribute with categorical and continuous variables.

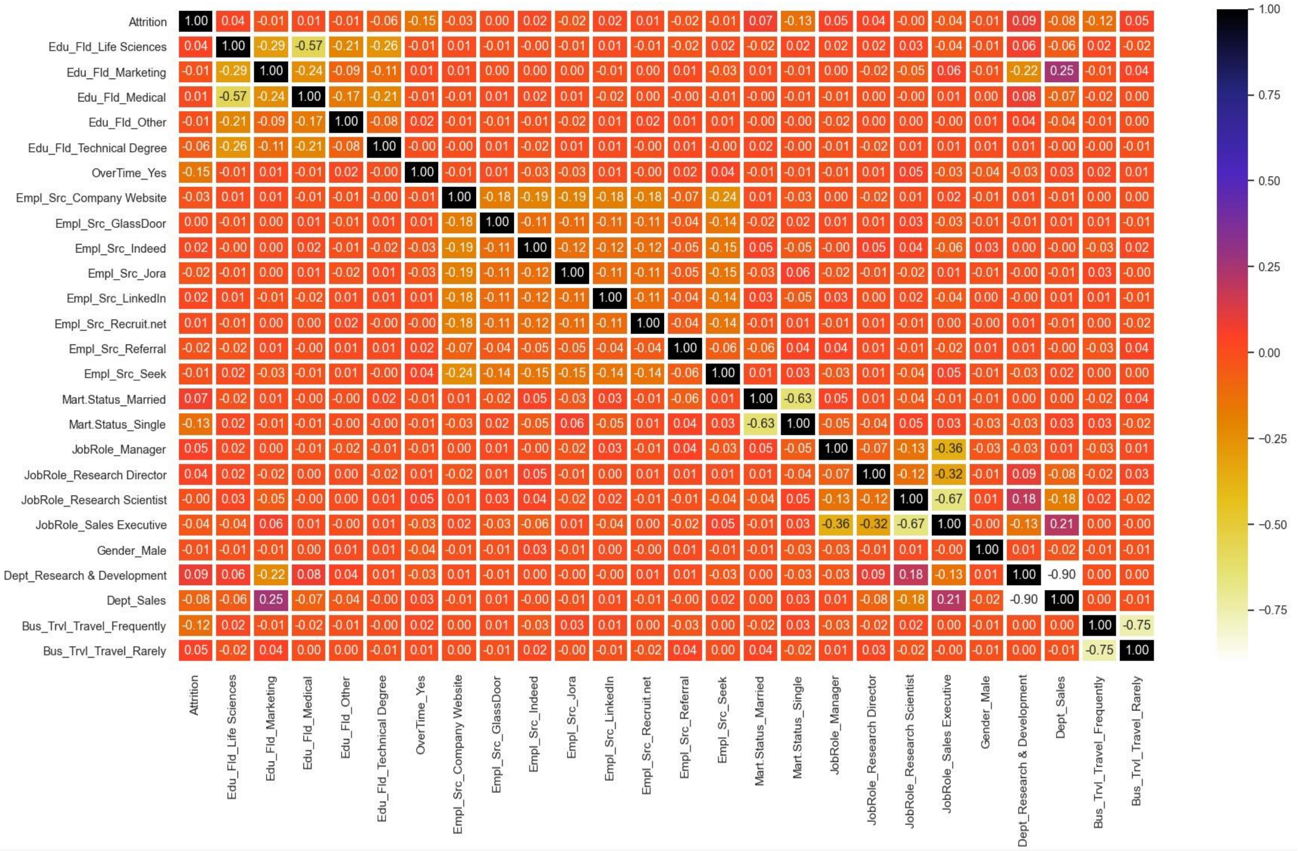


Fig. 37 Heatmap of Attrition attribute vs categorical variables

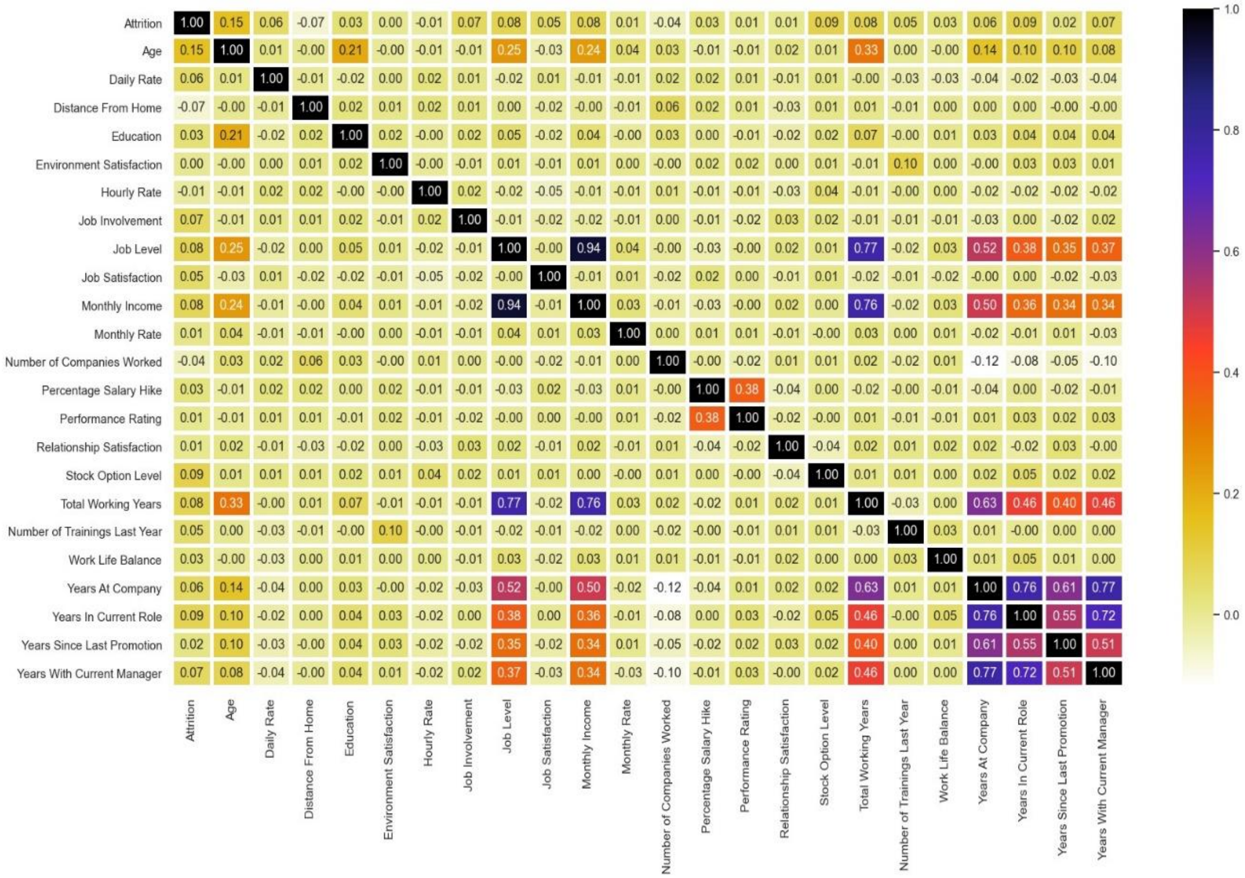
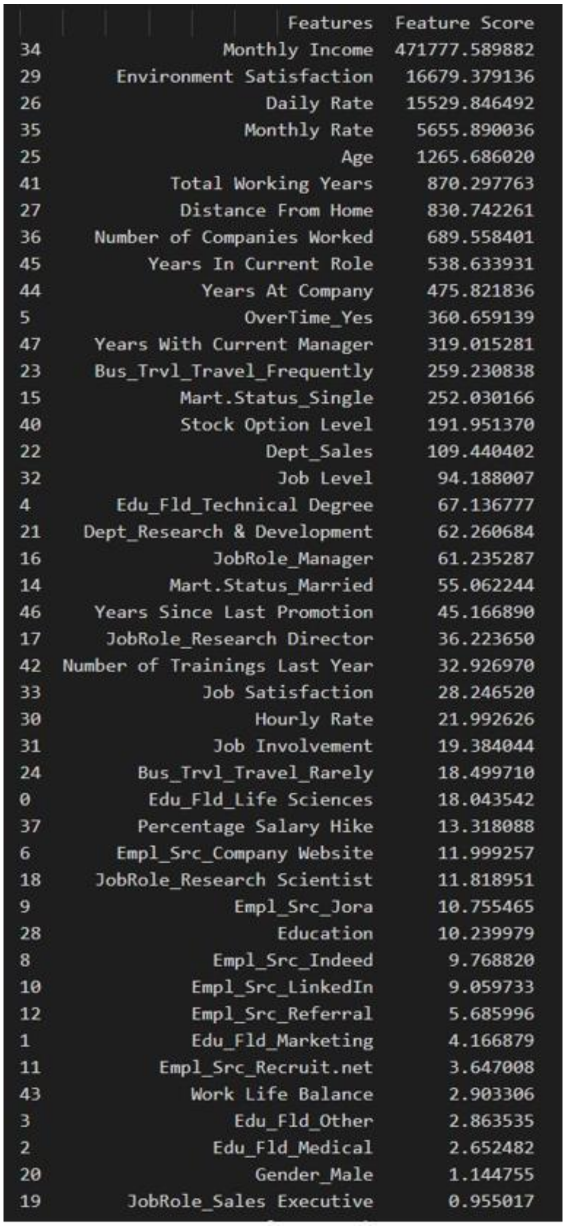


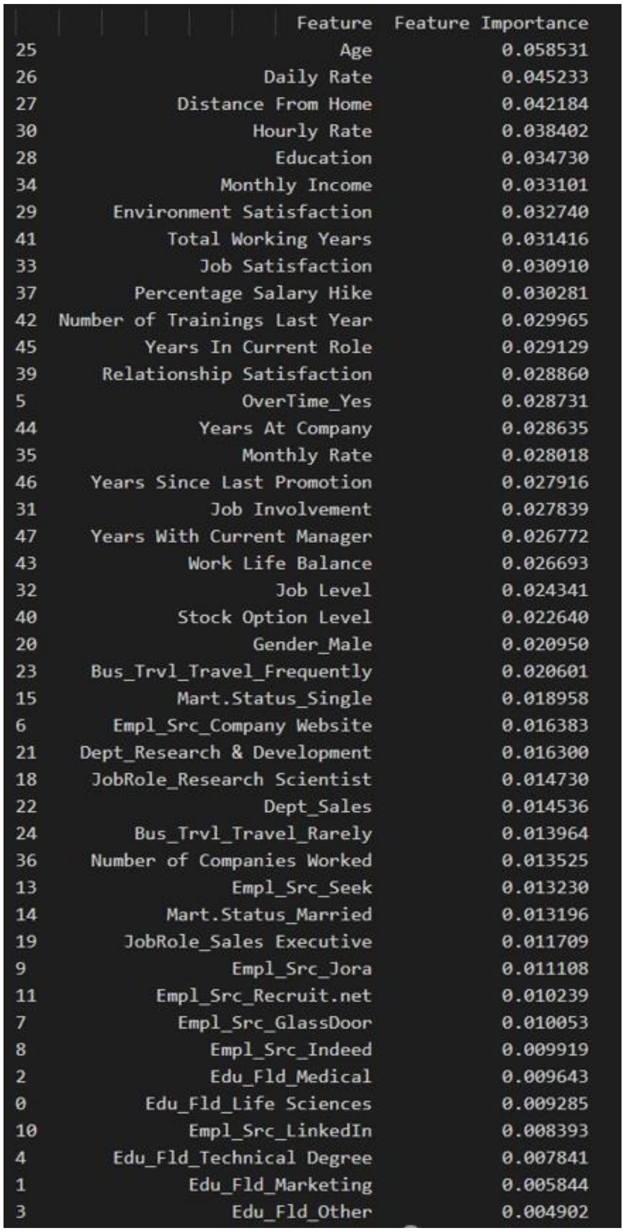
Fig. 38 Heatmap of Attrition attribute vs continuous variables

The χ^2 test was the next task that was performed as a part of the feature selection process. This task helped to determine the correction between two features, and based on the level of correlation, a score was generated. The higher the score value, the more the features are correlated. In conjunction with χ^2, we implemented the SelectKBest algorithm to select the top K best features based on the score obtained from the χ^2 test. Among all the features, Monthly Income recorded the highest score of 471777 approx. and was marked as the most correlated attribute with respect to Attrition. Environment Satisfaction, Daily Rate, Monthly Rate, and Age were the other features that followed the list subsequently. Fig. 39 shows the list of features and their corresponding feature scores. We also used an ExtraTree Classifier from the sklearn package to calculate the importance score of all the features. Higher the score value, the more relevant the features are. According to the ExtraTreeClassifier, Age recorded the highest feature importance score of 0.058531 standing as the most relevant feature. Along with Age, Daily Rate, Distance From Home, and Hourly Rate followed the list subsequently. Fig. 40 shows the list of all the features along with their feature importance score.



```
                           Features    Feature Score
34                   Monthly Income  471777.589882
29        Environment Satisfaction   16679.379136
26                       Daily Rate   15529.846492
35                     Monthly Rate    5655.890036
25                              Age    1265.686020
41              Total Working Years     870.297763
27               Distance From Home     830.742261
36       Number of Companies Worked     689.558401
45            Years In Current Role     538.633931
44                 Years At Company     475.821836
5                      OverTime_Yes     360.659139
47       Years With Current Manager     319.015281
23       Bus_Trvl_Travel_Frequently     259.230838
15              Mart.Status_Single      252.030166
40               Stock Option Level     191.951370
22                       Dept_Sales     109.440402
32                        Job Level      94.188007
4         Edu_Fld_Technical Degree       67.136777
21    Dept_Research & Development        62.260684
16                  JobRole_Manager      61.235287
14             Mart.Status_Married       55.062244
46       Years Since Last Promotion      45.166890
17        JobRole_Research Director      36.223650
42  Number of Trainings Last Year        32.926970
33                 Job Satisfaction      28.246520
30                      Hourly Rate      21.992626
31                  Job Involvement      19.384044
24          Bus_Trvl_Travel_Rarely       18.499710
0           Edu_Fld_Life Sciences        18.043542
37          Percentage Salary Hike       13.318088
6         Empl_Src_Company Website       11.999257
18       JobRole_Research Scientist      11.818951
9                     Empl_Src_Jora      10.755465
28                        Education      10.239979
8                   Empl_Src_Indeed       9.768820
10                Empl_Src_LinkedIn       9.059733
12                Empl_Src_Referral       5.685996
1                 Edu_Fld_Marketing       4.166879
11             Empl_Src_Recruit.net       3.647008
43                Work Life Balance       2.903306
3                     Edu_Fld_Other       2.863535
2                   Edu_Fld_Medical       2.652482
20                      Gender_Male       1.144755
19          JobRole_Sales Executive       0.955017
```

Fig. 39 Result of χ^2 Test

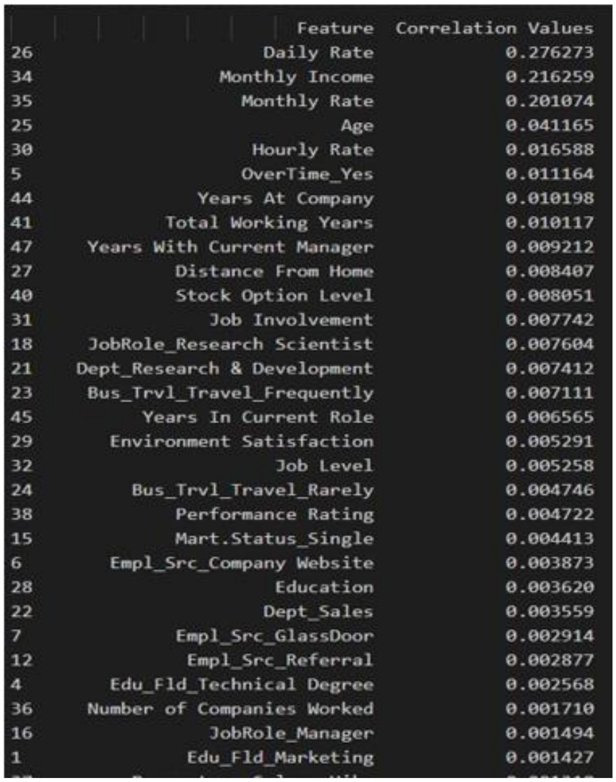


```
                            Feature  Feature Importance
25                              Age            0.058531
26                       Daily Rate            0.045233
27               Distance From Home            0.042184
30                      Hourly Rate            0.038402
28                        Education            0.034730
34                   Monthly Income            0.033101
29         Environment Satisfaction            0.032740
41              Total Working Years            0.031416
33                 Job Satisfaction            0.030910
37           Percentage Salary Hike            0.030281
42    Number of Trainings Last Year            0.029965
45            Years In Current Role            0.029129
39        Relationship Satisfaction            0.028860
5                      OverTime_Yes            0.028731
44                 Years At Company            0.028635
35                     Monthly Rate            0.028018
46       Years Since Last Promotion            0.027916
31                  Job Involvement            0.027839
47       Years With Current Manager            0.026772
43                Work Life Balance            0.026693
32                        Job Level            0.024341
40               Stock Option Level            0.022640
20                      Gender_Male            0.020950
23       Bus_Trvl_Travel_Frequently            0.020601
15               Mart.Status_Single            0.018958
6          Empl_Src_Company Website            0.016383
21      Dept_Research & Development            0.016300
18       JobRole_Research Scientist            0.014730
22                       Dept_Sales            0.014536
24           Bus_Trvl_Travel_Rarely            0.013964
36       Number of Companies Worked            0.013525
13                    Empl_Src_Seek            0.013230
14              Mart.Status_Married            0.013196
19          JobRole_Sales Executive            0.011709
9                     Empl_Src_Jora            0.011108
11             Empl_Src_Recruit.net            0.010239
7                Empl_Src_GlassDoor            0.010053
8                   Empl_Src_Indeed            0.009919
2                   Edu_Fld_Medical            0.009643
0             Edu_Fld_Life Sciences            0.009285
10                Empl_Src_LinkedIn            0.008393
4          Edu_Fld_Technical Degree            0.007841
1                 Edu_Fld_Marketing            0.005844
3                     Edu_Fld_Other            0.004902
```

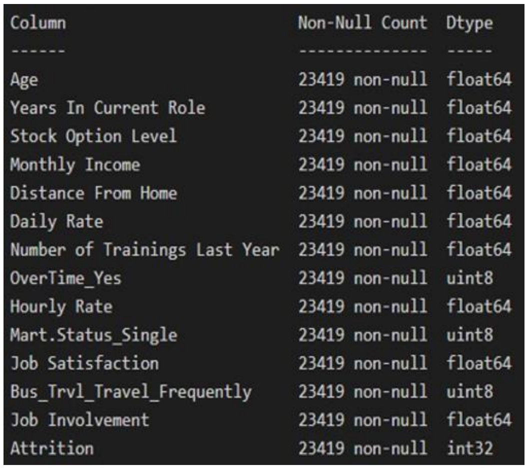
Fig. 40 Result of ExtraTree Classifier

The last technique used in the feature selection was Information Gain. In this technique, we used a mutual info classifier to compute the correlation values between the features and the dependent variable. High correlation values indicated that the features are more relevant with the dependent variable and have a better contribution to the model predicting the target class label. The mutual info classifier recorded Daily Rate as the most correlated feature with the highest score of 0.276273. Fig. 41 shows the list of features with their corresponding correlation values.

We experimented with heat map, χ^2 Test, ExtraTree Classifier, and Information Gain to select the most relevant features for our models. Since each method has its own mechanism to determine the relevant features, we decided to analyze and compare the experimental results of all the techniques and prepare a final list of features. The selected features shown in Fig. 42 are simply listed without any rank or order because we are aware that in the training phase, the model uses its own internal mechanism to rank those features and will not rely on the user computed rank.



```
                             Feature  Correlation Values
26                        Daily Rate            0.276273
34                    Monthly Income            0.216259
35                      Monthly Rate            0.201074
25                               Age            0.041165
30                       Hourly Rate            0.016588
5                       OverTime_Yes            0.011164
44                  Years At Company            0.010198
41               Total Working Years            0.010117
47        Years With Current Manager            0.009212
27                Distance From Home            0.008407
40                Stock Option Level            0.008051
31                   Job Involvement            0.007742
18        JobRole_Research Scientist            0.007604
21      Dept_Research & Development             0.007412
23        Bus_Trvl_Travel_Frequently            0.007111
45             Years In Current Role            0.006565
29          Environment Satisfaction            0.005291
32                         Job Level            0.005258
24            Bus_Trvl_Travel_Rarely            0.004746
38                Performance Rating            0.004722
15                Mart.Status_Single            0.004413
6           Empl_Src_Company Website            0.003873
28                         Education            0.003620
22                        Dept_Sales            0.003559
7                 Empl_Src_GlassDoor            0.002914
12                 Empl_Src_Referral            0.002877
4           Edu_Fld_Technical Degree            0.002568
36         Number of Companies Worked           0.001710
16                   JobRole_Manager            0.001494
1                  Edu_Fld_Marketing            0.001427
```



```
Column                          Non-Null Count  Dtype
------                          --------------  -----
Age                             23419 non-null  float64
Years In Current Role           23419 non-null  float64
Stock Option Level              23419 non-null  float64
Monthly Income                  23419 non-null  float64
Distance From Home              23419 non-null  float64
Daily Rate                      23419 non-null  float64
Number of Trainings Last Year   23419 non-null  float64
OverTime_Yes                    23419 non-null  uint8
Hourly Rate                     23419 non-null  float64
Mart.Status_Single              23419 non-null  uint8
Job Satisfaction                23419 non-null  float64
Bus_Trvl_Travel_Frequently      23419 non-null  uint8
Job Involvement                 23419 non-null  float64
Attrition                       23419 non-null  int32
```

Fig. 41 Result of Information Gain

Fig. 42 List of selected features

c. Model Building

After our study and research, we decided to use the Logistic Regression and SVM model to predict employee churn. We started building the model by following 3 steps: training the model, predicting the outcome using the trained model, and finally evaluating the performance of the model. Before training the model, we plotted a scatter plot as shown in Fig. 43 to check the proportion of the class label. Observing the scatter plot, we analyzed that there existed a 1:5 uneven data distribution between attrition and non-attrition employees. In order words, our dataset is imbalanced where the population of attrition employees is significantly less compared to non-attrition employees. Even though we were aware of the imbalance dataset problem, still for our learning purposes, we decided to build the model using the imbalanced dataset. To avoid the overfitting situation, the entire dataset was split into train and test set with the split ratio of 70:30.70% of the entire data was considered as a training dataset and the remaining 30% as a testing dataset. Fig 44 shows the bar chart of attrition and non-attrition employee counts for the train and test dataset.

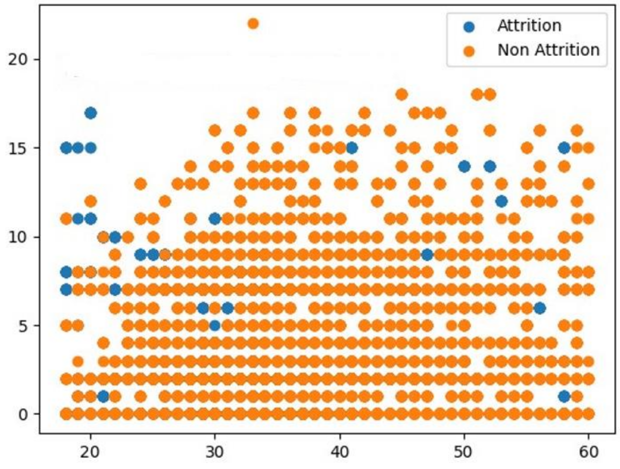


Fig. 43 Scatter plot of attrition and non-attrition employees

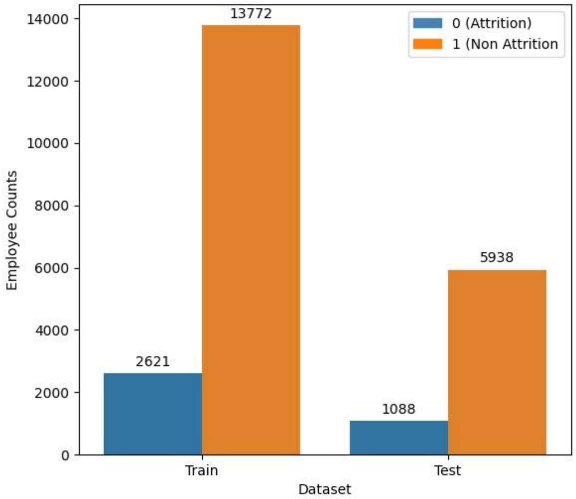
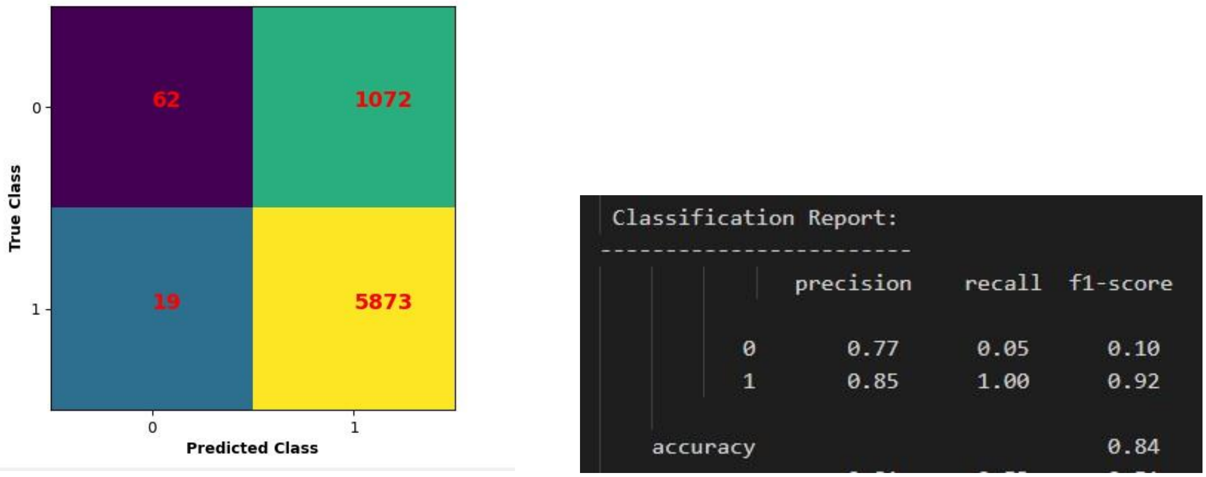


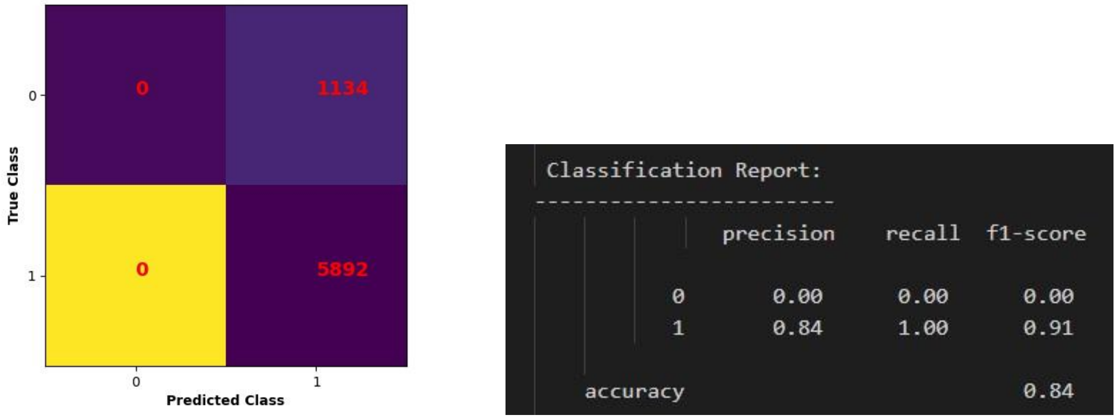
Fig. 44 Distribution of attrition and non-attrition employee in the train and test dataset

Using the training and testing dataset, Logistic Regression and SVM models were trained and tested respectively. To evaluate the performance of the models, we plotted a confusion matrix and generated classification reports. Fig. 45-46 shows the confusion matrix and classification report of Logistic Regression and SVM model respectively. In the confusion matrix, class 0 is considered as positive and 1 as negative. Since the proportion of 1 is significantly higher than 0 in the dataset, 1 is considered a majority and 0 as a minority class. Both the models recorded significantly low true positive compared to the true negative. With this observation, we understood that both the models seem biased towards the majority class and did not perform well on predicting the minority class even though both the models recorded an accuracy of 84%. However, other metrics such as precision, recall, and f1-score for the minority class were recorded as being significantly low compared to that of the majority class. Since the imbalanced dataset directly impacted the performance of both the Logistic Regression and SVM model, we decided to work on balancing the data set before we start training our models.



```
Classification Report:
------------------------
              precision    recall  f1-score

           0       0.77      0.05      0.10
           1       0.85      1.00      0.92

    accuracy                           0.84
```

Fig. 45 Confusion matrix and Classification report of Logistic Regression model

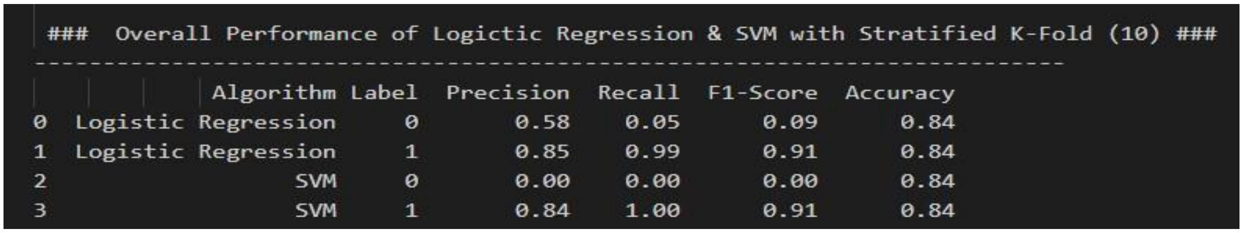


```
Classification Report:
------------------------
              precision    recall  f1-score

           0       0.00      0.00      0.00
           1       0.84      1.00      0.91

    accuracy                           0.84
```

Fig. 46 Confusion matrix and Classification report of SVM

To fix the imbalanced dataset issue, we experimented with 4 different techniques. The first technique is known as Stratified K-fold cross-validation. In this technique, first, we defined the value of K, and based on that value the entire dataset will get split into that number of folds. The algorithm gets executed for K iterations and on each iteration, one-fold will be picked as a testing dataset and the remaining K-1 folds will be used as the training dataset. The models then get trained and tested with the dataset. At the end of all the iterations, an average accuracy, precision, recall, and f1-score are computed and considered as the final performance metrics of the model. The

the beauty of Stratified K-fold cross-validation is that it tries to make sure that there is a uniform distribution of the majority and minority classes across training and testing datasets, and also avoids overfitting. We implemented Stratified K- fold cross-validation for K = 10 on the Logistic Regression and SVM algorithm but did not observe any improvements in the performance of the models. Fig. 47 shows the performance metrics of both Logistic Regression and SVM using Stratified K-Fold cross-validation.

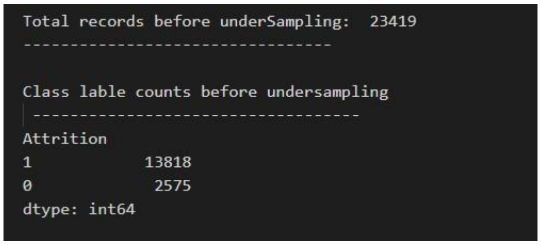


```
 ###  Overall Performance of Logictic Regression & SVM with Stratified K-Fold (10) ###
--------------------------------------------------------------------------------
          Algorithm Label  Precision  Recall  F1-Score  Accuracy
0  Logistic Regression     0       0.58    0.05      0.09      0.84
1  Logistic Regression     1       0.85    0.99      0.91      0.84
2                  SVM     0       0.00    0.00      0.00      0.84
3                  SVM     1       0.84    1.00      0.91      0.84
```

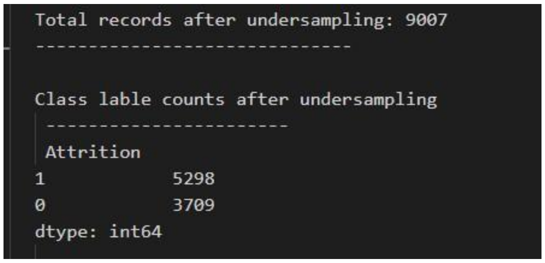
Fig. 47 Results of Stratified K-Fold Cross-Validation

The second technique that we implemented is called Under Sampling. This technique uses the NearMiss method which accepts an argument called sampling strategy. Based on the sampling strategy value, the size of the majority class will be scaled down to match approximately the size of the minority class. Scaling down the dataset results in loss of data which is one of the major drawbacks of this technique. Fig. 48 shows the count of the majority and minority classes before and after Under Sampling. With Under Sampling, the total number of records was reduced from 23419 to 9007. The Logistic Regression and SVM recorded a count of 530 and 277 as True Positive which were higher but the counts of True Negative were recorded of 1354 and 1449 which were lower in comparison to the results of the unbalanced dataset. In addition, the overall accuracy recorded for the Logistic Regression was 0.7 and that of SVM was 0.64 which was still not considered acceptable. However, there was a significant improvement in the precision, recall, and F1-score for the minority class which shows that both the classes were fairly treated by the models.

Fig. 49 and Fig. 50 shows the confusion matrix and classification report for Logistic Regression and SVM model with an under-sampled dataset.

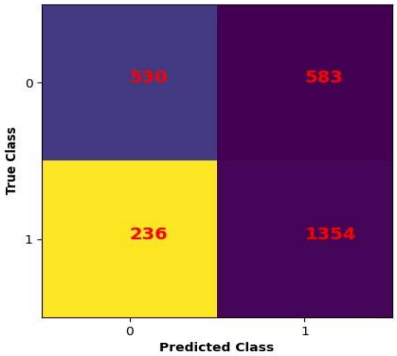


```
Total records before underSampling:  23419
--------------------------------

Class lable counts before undersampling
 ----------------------------------
Attrition
1            13818
0             2575
dtype: int64
```

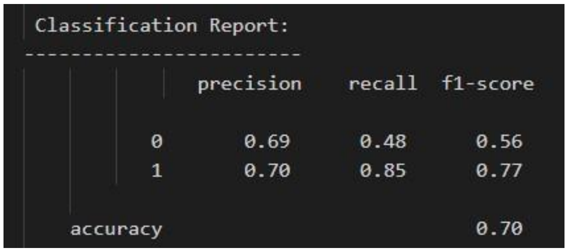


```
Total records after undersampling: 9007
------------------------------

Class lable counts after undersampling
 -----------------------
 Attrition
1            5298
0            3709
dtype: int64
```

Fig. 48 Total records and class label counts of before and after Under Sampling

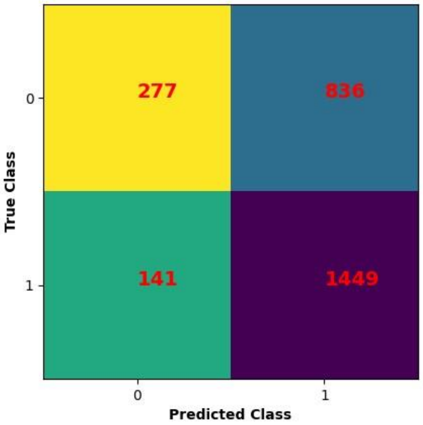


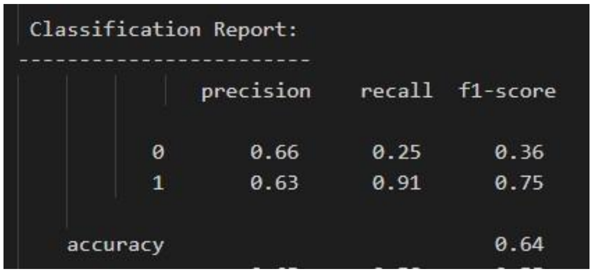
.



```
 Classification Report:
------------------------
              precision    recall  f1-score

           0       0.69      0.48      0.56
           1       0.70      0.85      0.77

    accuracy                           0.70
```

Fig. 49 Confusion matrix and Classification report of Logistic Regression using Under Sampling

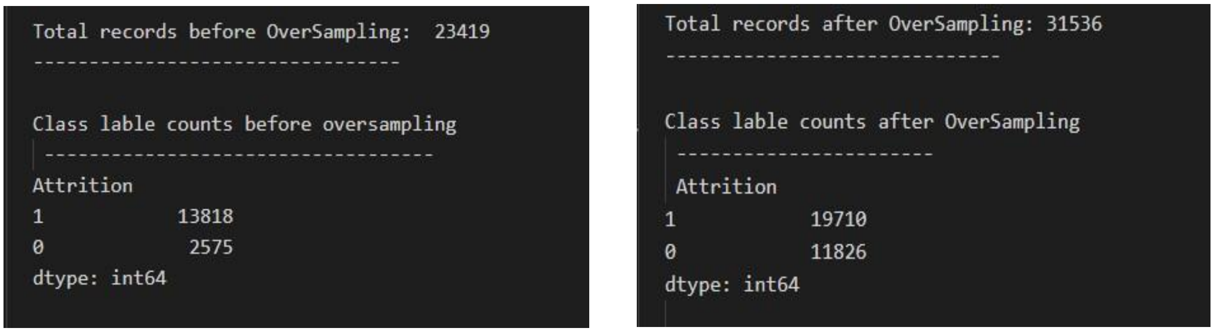




```
 Classification Report:
------------------------
              precision    recall  f1-score

           0       0.66      0.25      0.36
           1       0.63      0.91      0.75

    accuracy                           0.64
```

Fig. 50 Confusion matrix and Classification report of SVM using Under Sampling

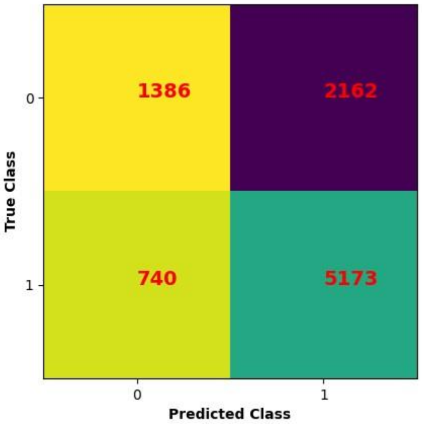
The third technique that we implemented is called Over Sampling. This technique uses the RandomOverSampler method which accepts an argument value for sampling strategy. Based on the argument value, the RandomOverSampler method will upscale the size of the minority class by adding more replicas of its own class, thus approximately matching with the size of the majority class. Since the method adds the exact replicas of the same class, this can lead to the situation of overfitting which is the major drawback of this technique. Fig. 51 shows the count of the majority and minority classes before and after oversampling. With the Over Sampling technique, the total number of records increased from 23419 to 31536. The Logistic Regression model recorded 1386 as True Positive,5173 as True Negative, 740 as False Positive, and 2162 as False Negative which were higher compared to the results of Under Sampling. However, the SVM model performed very poorly with oversampled data recording 0 as True and False Positive, indicating that the model did not perform well on predicting with the minority class. Fig. 52-53 shows the confusion matrix and classification report for Logistic Regression and SVM with an oversampled dataset.

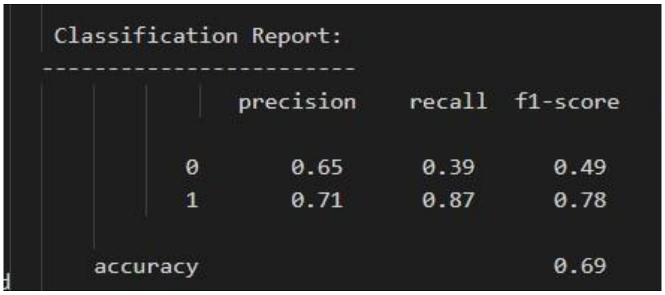


```
Total records before OverSampling:  23419
--------------------------------

Class lable counts before oversampling
 ----------------------------------
Attrition
1           13818
0            2575
dtype: int64
```

```
Total records after OverSampling: 31536
------------------------------

Class lable counts after OverSampling
 ----------------------
 Attrition
1           19710
0           11826
dtype: int64
```

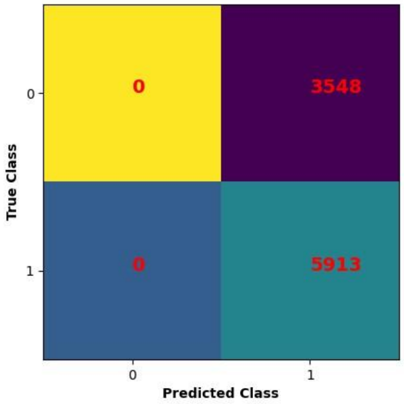
Fig. 51 Total records and counts of the class label before and after Over Sampling

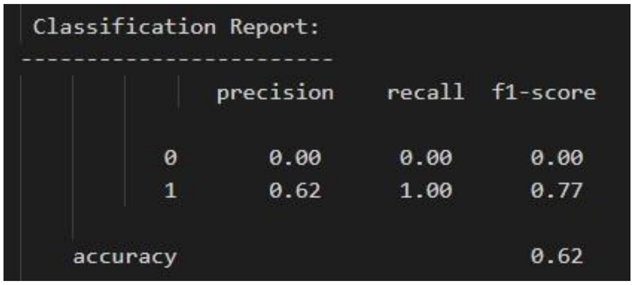




```
Classification Report:
-----------------------
              precision    recall  f1-score

           0       0.65      0.39      0.49
           1       0.71      0.87      0.78

    accuracy                           0.69
```

Fig. 52 Confusion matrix and Classification report of Logistic Regression with Over Sampling





```
Classification Report:
-----------------------
              precision    recall  f1-score

           0       0.00      0.00      0.00
           1       0.62      1.00      0.77

    accuracy                           0.62
```

Fig. 53 Confusion matrix and Classification report of SVM with Over Sampling

SMOTETOMEK is another dataset balancing mechanism that is considered as the combination of the Under and Over Sampling technique and overcomes the issue of data loss and overfitting. SMOTETOMEK method accepts an argument value for sampling strategy, and based on that argument value, the method will either upscale the minority class or downscale the majority class. This technique randomly picks a data point, and using the K-Nearest Neighbor algorithm, it will add more data points around its proximity. As shown in Fig. 54, the SMOTETOMEK method increased the total number of attrition employee records from 3709 to 19707 and non-attrition records from 19710 to 19707 respectively, thus making the number of attrition and non-attrition records equal. With SMOTETOMEK, True Positive counts were recorded as 3707 for the Logistic Regression and 3780 for the SVM model whereas, in the case of True Negative, the Logistic Regression recoded 4127 which was almost double compared to 2892 of SVM. For the False Positive, the Logistic Regression recorded 1786 which was almost half compared to 3021 of SVM. A False Negative count of 2205 was recorded for the Logistic Regression model and 2132 for the SVM respectively. The Logistic Regression model recorded an accuracy of 0.66 and 0.56 by SVM. From all the above observations, we made a statement that with the SMOTETOMEK technique, the Logistic Regression model performed better compared to the SVM model. Fig 55-56 shows the confusion matrix and classification report of the Logistic Regression and SVM model. On Comparing all the data balancing techniques, SMOTE TOMEK turns out to be the most effective technique and hence we decided to use this method to balance our dataset.

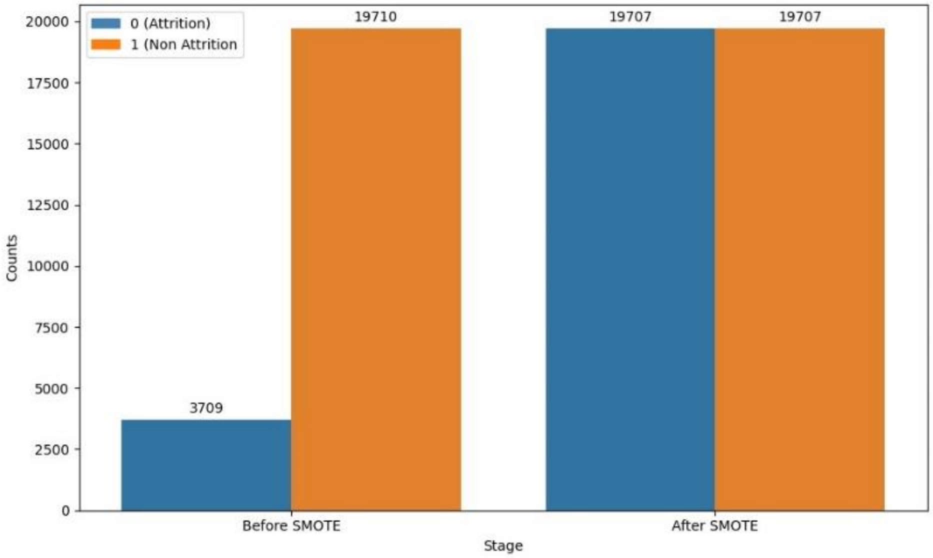
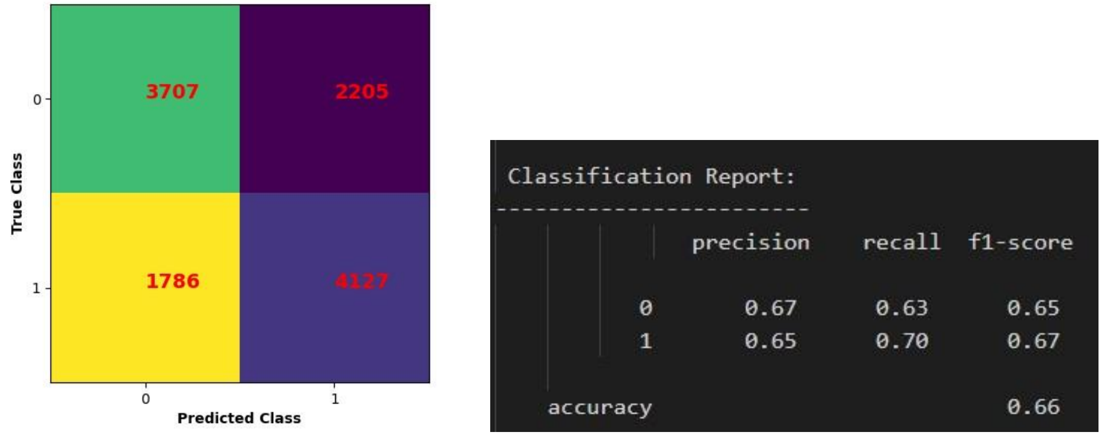
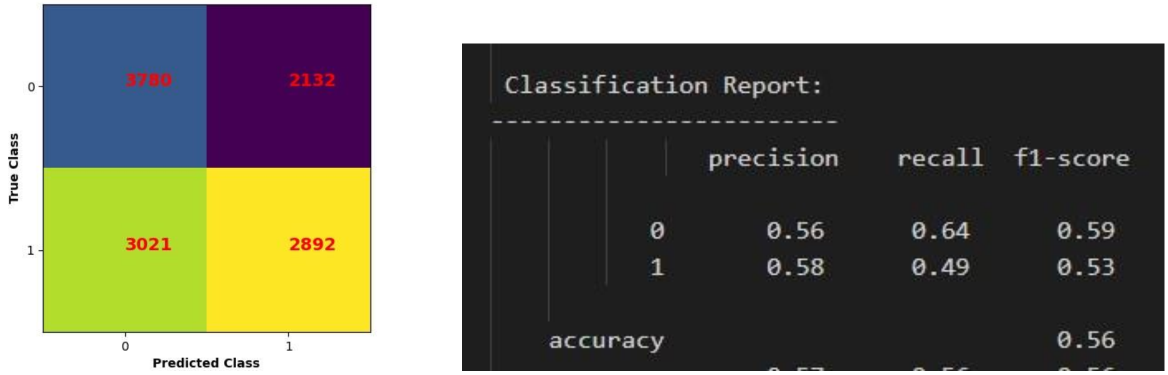


Fig. 54 Class label counts of before and after SMOTETOMEK process



```
Classification Report:
------------------------
                     precision    recall  f1-score

                0         0.67      0.63      0.65
                1         0.65      0.70      0.67

    accuracy                                  0.66
```

Fig. 55 Confusion matrix and Classification report of Logistic Regression with SMOTETOMEK



```
Classification Report:
------------------------
                     precision    recall  f1-score

                0         0.56      0.64      0.59
                1         0.58      0.49      0.53

    accuracy                                  0.56
```

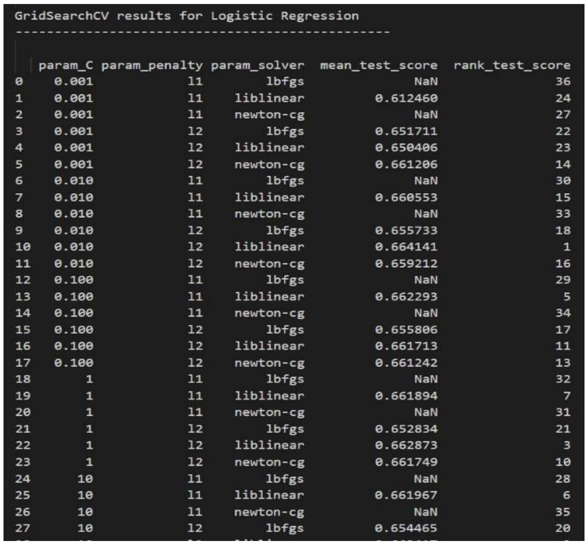
Fig. 56 Confusion matrix and Classification report of SVM with SMOTETOMEK

d. Hyperparameter Tuning

So far, the Logistic Regression and SVM models were trained with the default parameters and we analyzed the results obtained from both models. However, in machine learning, most of the algorithms provide the flexibility of tuning certain parameters called hyperparameters tuning, thus optimizing the performance and making the models robust and accurate. Each model has its own set of hyperparameters to be configured and tuning these parameters can significantly change the behavior of the algorithms so, care must be exercised when adjusting the hyperparameters. In the Logistic Regression model, there are 3 standard parameters C, penalty, and solver that are available for hyperparameter tuning. C parameter is a floating number that controls the penalty strength. The lower the value of c, the lower is the penalty for misclassification and vice versa. The next parameter is a penalty, and also called regularization, and it controls the model's overfitting condition by reducing the variance. The penalty parameter accepts 4 different values none, l1, l2, and elasticnet. The last parameter is the solver which specifies the algorithm that will be used to optimize the performance of the model and it accepts 5 different values lbfgs, loglinear, newton-cg, sag, and saga. After referencing the sklearn documentation for the Logistic Regression model, we understood the compatibility between the penalty and solver. Meaning, not all the values of the penalty parameter are compatible with the solver parameters. So, after reviewing the compatibility chart and researching the standard set of parameters that are commonly being used, we decided to use l1, l2 as the penalty parameters and lbfgs, loglinear, and newton-cg as the solver parameters.

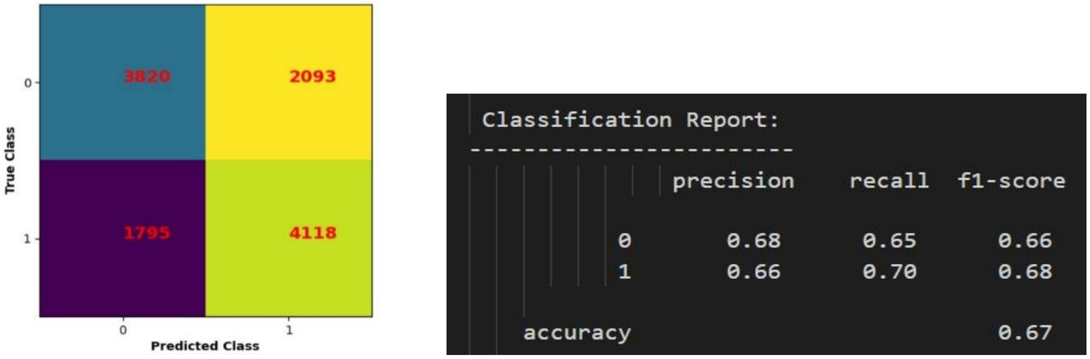
The SVM model has 3 standard parameters C, kernel, and gamma available for the hyperparameter tuning. A kernel is a function that is very helpful for creating the decision surface when the data points are not linearly separable. Using the kernel trick, the kernel function computes the decision boundaries in terms of similarity measures in a high dimensional space without actually transforming the data points into higher dimensions. Kernel parameters accept linear, Radial Basis Function (rbf), sigmoid, and poly. Experimenting with poly and sigmoid parameters, we experienced our system getting extremely slow and unresponsive for hours, so we decided to choose rbf as our kernel parameter. Gamma is the next parameter that controls the distance of influence of a single training point A lower value of gamma increases the similarity radius, meaning data points that are far apart from each other are also considered similar and get classified as the same class. In contrast, a higher value of gamma reduces the similarity radius, so the data points need to be close enough to each other to get classified under the same class. Higher values of gamma can cause the overfitting situation; whereas the lower values create a generalized decision boundary, thus causing high misclassifications. Maintaining the balance between the lower and higher gamma values and reviewing the commonly used gamma values, we decided to go with the following gamma values 0.0001, 0.0001, 0.001, 0.01,0.1, and 1.

During hyperparameter tuning, trying out every possible set of parameters and checking its results manually is an iterative and inefficient process. So, to avoid this, we implemented a GridSearchCV method which works on every combination and permutation of the parameters and provides the best set of parameters for that model. Fig. 57 shows the results of GridSearchCV for the Logistic Regression. The best parameters suggested by GridSearchCV for the Logistic Regression was C= 0.01, penalty = l2 and solver = liblinear and for SVM it was C=100, kernel =rbf and gamma =0.00001. Both the models were again trained and tested with their corresponding hyperparameters and the results are shown in Fig 58-59. For the Logistic Regression model, the counts of False Positive got reduced to 2093, but besides that, no other significant improvements were recorded. However, for SVM, the hyperparameters drastically changed the overall performance of the model. SVM model recorded the highest counts of 5818 as True Positive and 5197 as True Negative and also the lowest counts of 95 as False Positive and 726 as False Negative respectively. In addition, the precision, recall, and f1-score for both the majority and the minority classes were also recorded high and the accuracy of the model was boosted to 0.93. The rbf function that was chosen as kernel parameter created the most appropriate decision boundary and surface to separate the two classes distinctly; whereas the Logistic Regression model created a generic decision boundary resulting in higher misclassification errors compared to the SVM. Along with that, the suggested optimum value of gamma = 0.00001 contributed on grouping the two classes correctly. We believed that these could be the potential reasons for the SVM model to perform better than the Logistic Regression with the hyperparameter tuning.



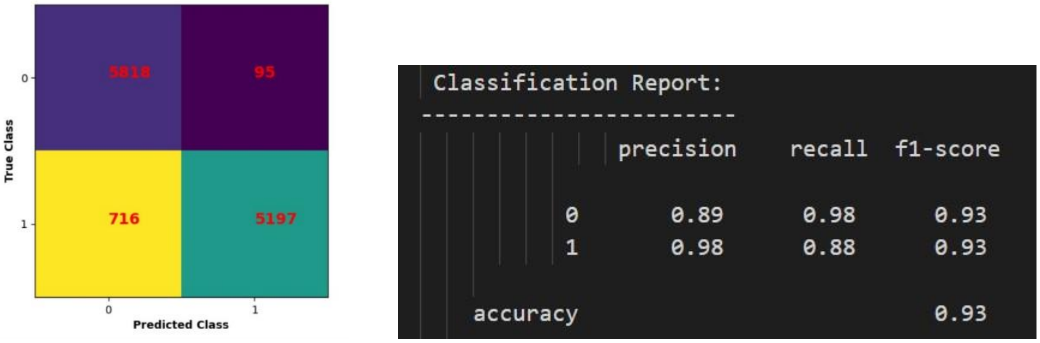
```
GridSearchCV results for Logistic Regression
---------------------------------------------

    param_C param_penalty param_solver  mean_test_score  rank_test_score
0     0.001            l1        lbfgs              NaN               36
1     0.001            l1    liblinear         0.612460               24
2     0.001            l1    newton-cg              NaN               27
3     0.001            l2        lbfgs         0.651711               22
4     0.001            l2    liblinear         0.650406               23
5     0.001            l2    newton-cg         0.661206               14
6     0.010            l1        lbfgs              NaN               30
7     0.010            l1    liblinear         0.660553               15
8     0.010            l1    newton-cg              NaN               33
9     0.010            l2        lbfgs         0.655733               18
10    0.010            l2    liblinear         0.664141                1
11    0.010            l2    newton-cg         0.659212               16
12    0.100            l1        lbfgs              NaN               29
13    0.100            l1    liblinear         0.662293                5
14    0.100            l1    newton-cg              NaN               34
15    0.100            l2        lbfgs         0.655806               17
16    0.100            l2    liblinear         0.661713               11
17    0.100            l2    newton-cg         0.661242               13
18        1            l1        lbfgs              NaN               32
19        1            l1    liblinear         0.661894                7
20        1            l1    newton-cg              NaN               31
21        1            l2        lbfgs         0.652834               21
22        1            l2    liblinear         0.662873                3
23        1            l2    newton-cg         0.661749               10
24       10            l1        lbfgs              NaN               28
25       10            l1    liblinear         0.661967                6
26       10            l1    newton-cg              NaN               35
27       10            l2        lbfgs         0.654465               20
```

Fig. 57 Results of GridSearchCV for Logistic Regression model



```
Classification Report:
-----------------------
              precision    recall  f1-score

       0         0.68       0.65      0.66
       1         0.66       0.70      0.68

   accuracy                           0.67
```

Fig. 58 Confusion matrix and Classification report of Logistic Regression with hyperparameters



```
Classification Report:
-----------------------
              precision    recall  f1-score

       0         0.89       0.98      0.93
       1         0.98       0.88      0.93

   accuracy                           0.93
```

Fig. 59 Confusion matrix and Classification report of SVM with hyperparameters

e. Performance Evaluation

We evaluated the performance of the Logistic Regression and SVM model based on the following metrics

1. **True Positive Rate (Recall/ Sensitivity)**
2. **True Negative Rate (Specificity)**
3. **False Positive Rate (Type I Error)**
4. **False Negative Rate (Type II Error)**
5. **Accuracy**
6. **Receiver Operating Characteristics (ROC) curve**
7. **Area Under Curve (AUC) score**

1. True Positive Rate (Recall/ Sensitivity):

True Positive Rate specifies the proportion of actual positive classes that got correctly classified by the model. True Positive Rate is also known as recall or sensitivity and is calculated by the formula

$$\text{True Positive Rate} = \frac{\text{True Positive}}{\text{True Positive + False Negative}}$$

The logistic Regression model recorded the True Positive Rate of 0.65 and 0.93 for SVM respectively. In other words, SVM classified the positive class correctly with an accuracy of 93% compared to 65% of that of the Logistic Regression model.

2. True Negative Rate (Specificity)

True Negative Rate is defined as the proportion of actual negative classes that got correctly classified by the model. It is also known as Specificity and is calculated by the formula

$$\text{True Negative Rate} = \frac{\text{True Negative}}{\text{True Negative + False Positive}}$$

True Negative Rate of 0.65 was observed for the Logistic Regression model and 0.98 for SVM respectively. In other words, SVM classified the negative class correctly with an accuracy of 98% compared to 65% of that of the Logistic Regression model.

3. False Positive Rate (Type I error)

False Positive Rate is defined as the proportion of actual negative class entries that were incorrectly classified as being in the positive class. It is also known as Type I error and can be calculated by the following formula

$$\text{False Positive Rate} = \frac{\text{False Positive}}{\text{False Positive + True Negative}}$$

$$\text{False Positive Rate} = 1 - \text{Specificity}$$

The False Positive Rate of 0.12 was recorded for SVM and 0.30 for the Logistic Regression model respectively. In other words, for the negative class, the SVM model has a 12% misclassification error compared to 30% that of the Logistic Regression model.

4. False Negative Rate (Type II Error):

False Negative Rate is defined as the proportion of actual positive class entries that were incorrectly classified as being in the negative class. It is also known as Type II error and can be calculated by the following formula

$$\text{False Positive Rate} = \frac{\text{False Negative}}{\text{False Negative + True Positive}}$$

A False Negative Rate of 0.01 was recorded for the SVM model and 0.35 for the Logistic Regression respectively. In other words, SVM correctly classified the positive class with an accuracy of 99%, and only 1% were misclassified as the negative class. Whereas for the Logistic Regression model, only 65% of the positive class were classified correctly, and the rest 35% of the positive class were misclassified as the negative class.

5. Accuracy:

Accuracy of the model is defined as the proportion of total actual class predictions to the total number of observations and it is calculated by the following formula

$$\text{Accuracy} = \frac{\text{True Positive} + \text{True Negative}}{\text{True Positive} + \text{False Positive} + \text{True Negative} + \text{False Negative}}$$

Logistic Regression recorded an accuracy of 0.67 whereas SVM recorded 0.93. It shows that SVM classified both the positive and negative classes correctly with an accuracy of 93% compared to that of 67% of the Logistic Regression model.

6. Receiver Operating Characteristics (ROC) curve:

Based on the prediction probabilities, the True Positive Rate and False Positive Rate for the different threshold values were calculated. Using those calculated values, a probability curve is plotted with True Positive Rate on the y-axis and False Positive Rate on the x-axis, and the curve is known as the ROC curve. A ROC curve is a tool that helps to measure how well the model performed on classifying the observations. Fig. 60 shows the ROC curve for the Logistic Regression and SVM model. A blue dotted diagonal line seen in fig 60 is considered as the reference line and any curve that falls above it is considered a good model and any curve below is a bad model. The closer and higher the curve towards the y-axis of the plot, the more accurate the model. The curve for the SVM model climbs up steeply to the top left of the plot whereas the curve for the Logistic Regression is above the reference line but still below the SVM curve. Therefore, from this observation, we can say that the SVM model performed better in predicting the classes than the Logistic Regression model.

7. Area Under Curve (AUC):

Area Under Curve is the summary of the ROC curve. It is a score that ranges between the values 0 and 1. A model with an AUC score of 0 is considered a bad model and classified randomly whereas an AUC score of 1 is considered as an ideal classifier that classifies all the observations perfectly with no misclassification error. In general, the model within the range of 0.5 and 1 is acceptable. In our experiment, SVM recorded a high AUC score of 0.96 compared to 0.73 of Logistic Regression. Fig. 61-62 shows the bar chart with all the performance metrics for the Logistic Regression and SVM model. As shown in Fig. 63, we plotted a bar chart comparing the

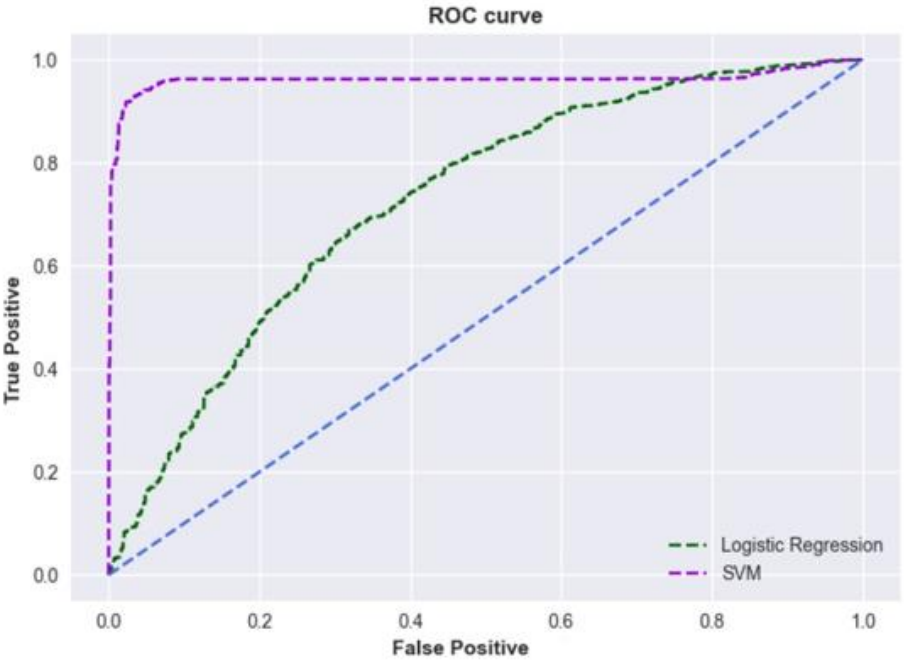


Fig. 60 ROC curve of Logistics Regression vs SVM

training time for each of the models and observed that SVM took 413 seconds (approx.) for training whereas the Logistic Regression model got quickly trained in 0.28 seconds. We always demand a model with higher accuracy, True Positive Rate, and True Negative Rate and lower False Negative Rate and False Positive Rate, so observing and comparing all the performance metrics for both the models on the given dataset, we concluded that SVM performed much better on predicting an employee churn than the Logistic Regression model.

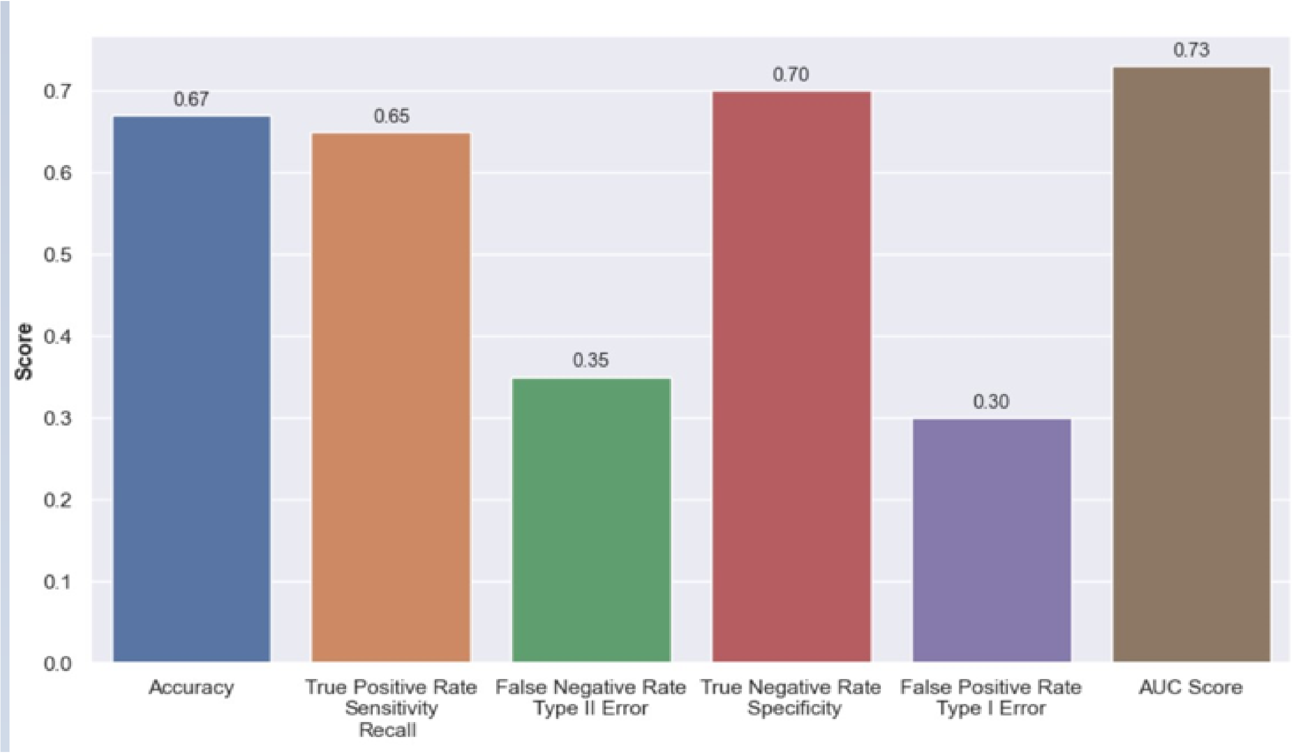


Fig. 61 Performance metrics of the Logistic Regression model

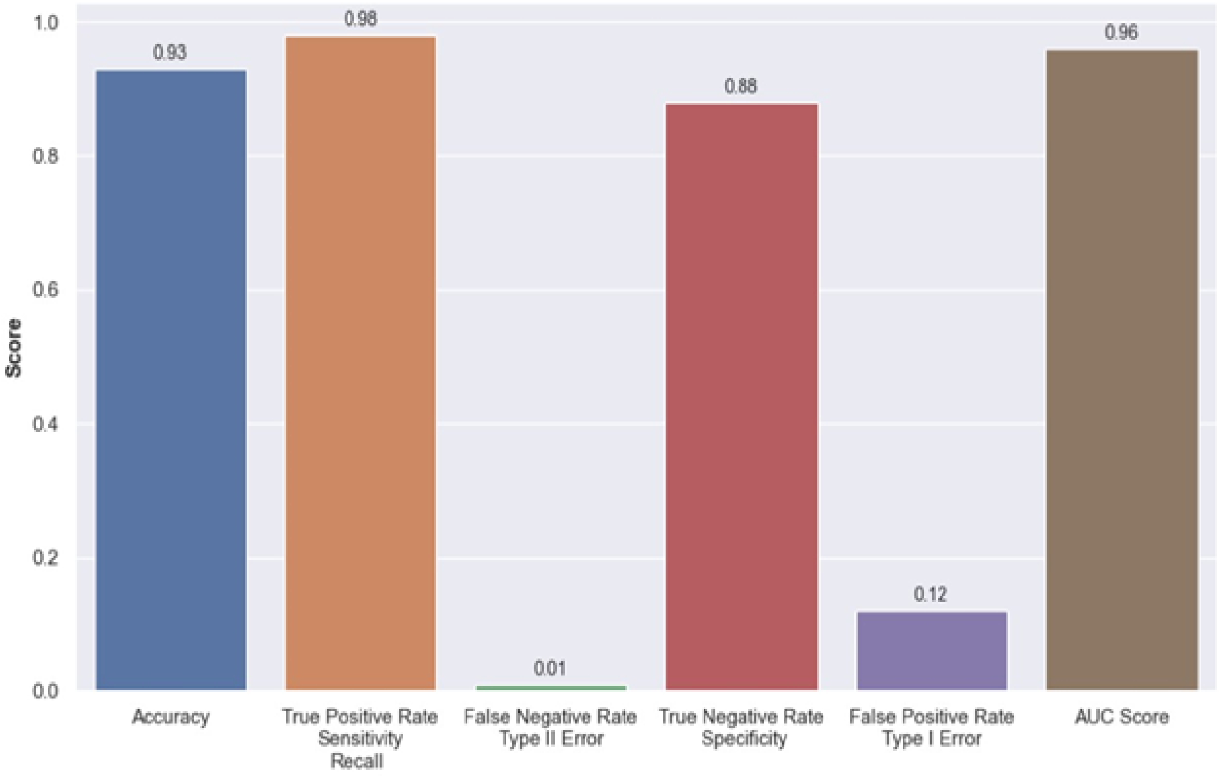


Fig. 62 Performance metrics of the SVM model

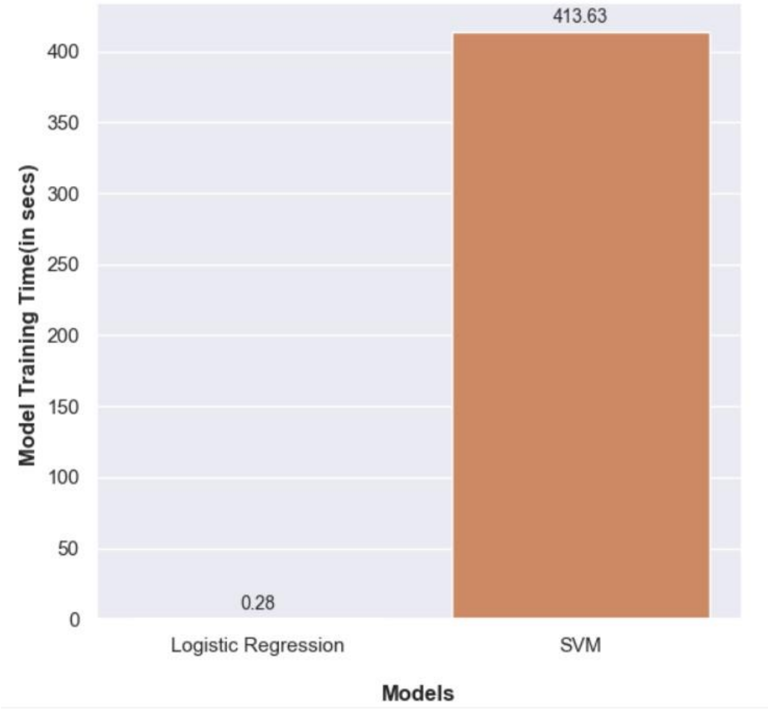


Fig.63 Training time of Logistic Regression vs SVM model

VII. CONCLUSION AND FUTURE WORKS

Using the balanced dataset obtained from SMOTETOMEK, the Logistic Regression model achieved an accuracy of 0.66 with 3707 as True Positive, 4127 as True Negative, 1786 as False Positive, and 2205 as False Negative whereas the SVM model recorded an accuracy of 0.56 with 3780 as True Positive, 2892 as True Negative, 3021 as False Positive and 2132 as False Negative respectively. With the hyperparameter tuning, the Logistic Regression model achieved an accuracy of 0.67 with counts of 3820 True Positive, 4118 True Negative, 1795 False Positive, and 2093 False Negative whereas the SVM model recorded an accuracy of 0.93 with counts of 5818 True Positive, 5197 True Negative, 716 False Positive and 95 False Negative respectively. The Logistic Regression model also recorded Sensitivity of 0.65, Specificity of 0.70, Type I Error of 0.30, Type II Error of 0.35, and AUC score of 0.73 whereas the SVM model recorded Sensitivity of 0.98, Specificity of 0.88, Type I Error of 0.12, Type II Error of 0.01 and AUC score of 0.96 respectively. From all this observation, we concluded that with the model's default configurations the Logistic Regression model predicted employee churn better than the SVM whereas with hyperparameter turning, the SVM performed better than the Logistic Regression model.

In the future, we would like to extend this project by hosting our trained SVM model on the cloud platform and allowing the end-user to use our model via a web interface. We would also like to explore Neural networks to train our dataset and observe its predictions. Last but not least, we would like to explore another HR dataset that has the records of employees who had resigned before and during COVID-19, and using that dataset, discover the correlation and patterns of employee churn.

REFERENCES

[1] L. Branham, The 7 hidden reasons employees leave: How to recognize the subtle signs and act before it's too late, 2005. [Online]. Available: https://ebookcentral.proquest.com/lib/sjsu/reader.action?docID=242984.

[2] https://business.linkedin.com/talent-solutions/blog/trends-and-research/2018/the-3- Industries- with-the-highest-turnover-rates.

[3] P. C. Bryant and D. G. Allen, "Compensation, Benefits and Employee Turnover: HR Strategies For Retaining Top Talent," Compensation and benefits review, vol. 45, issue 3, pp. 171-175, May 2013.

[4] A. Robison, "Effective Employee Retention Strategies Use HR Self-Service," kioskMarketplace.com, Nov. 2018. [Online]. Available: https://link.gale.com/apps/doc/A562171993/ITOF?u=csusj&sid=ITOF&xid=cd0dc0cc.

[5] Abrams and M. N," Employee retention strategies: lessons from the best," Healthcare executive, vol. 19, issue 4, pp. 18-22, 2004.

[6] X. Zhang, J. Zhu, S. Xu, and Y. Wan, "Predicting customer churn through interpersonal influence," Knowledge-based Systems, vol. 28, pp. 97-104, April. 2012.

[7] S. Yadav, A. Jain, and D. Singh, "Early prediction of Employee Attrition using Data Mining Techniques," in *Proc*. 2018 IEEE 8th International Advance Computing Conference (IACC), Dec. 2018, pp.349-354.

[8] R. Ramakrishnan, S. Bhattacharya, and P. Dhanya, "Predict employee attrition by using predictive analytics," Benchmarking: An International Journal, vol. 26, issue 1, pp. 2-18, 2018.

[9] V. V. Saradhi and G. K. Palshikhar, "Employee churn prediction," Expert systems with applications, vol. 38, issue 3, pp. 1999-2006, 2011.

[10] S. Dutta and S. K. Bandyopadhyay, “Employee attrition prediction using neural network cross validation method,” International J. of Commerce and Management Research, vol. 6, issue 3, pp. 80-85, 2020.

[11] A. Rodan, A. Fayyoumi, H. Faris, J. Alsakran and O. Al-kadi, “Negative correlation Learning for customer churn prediction: a comparison study,” The Scientific World Journal, vol. 15, 2015, doi: http://dx.doi.org.libaccess.sjlibrary.org/10.1155/2015/473283

[12] X. Gao, J. Wen, and C. Zhang, “An Improved Random Forest Algorithm for Predicting Employee Turnover,” Mathematical Problems in Engineering, vol. 2019, pp. 1-12, 2019. [Online]. Available: https://doi.org/10.1155/2019/4140707

[13] M. A. Valle and G. A. Ruz, “Turnover Prediction in a Call Center: Behavioral Evidence of Loss Aversion Using Random Forest and Naïve Bayes Algorithm,” Applied Artificial Intelligence, vol. 29, issue 9, pp. 923-942, 2015. doi: 10.1080/08839514.2015.1082282.

[14] C. A. M. Troncoso, “Predicting Customer Churn using Voice of the Customer. A Text Mining Approach,” ph.D. thesis, Business, and Management, University of Manchester, Manchester, The United Kingdom, 2018.

[15] M. Panjasuchat and Y. Limpiyakorn, “Applying Reinforcement Learning for Customer Churn Prediction,” Journal of Physics. Conference Series, vol. 1619, issue 1, pp. 12015, 2020.

[16] D. Senanayake, L. Muthugama, L. Mendis and T. Madushanka, “Customer Churn Prediction: A Cognitive Approach,” Internation Journal of Computer, Electrical, Automation, Control and Information Engineering, vol. 9, no. 3, 2015.

[17] S. N. Khera and Divya, “Predictive Modelling of Employee Turnover in Indian IT Industry Using Machine Learning Techniques,” Vision, vol. 23, issue 1, pp. 12-21, 2018. doi: 10.1177/0972262918821221.

[18] N. Bandyopadhyay and A. Jadhav, “Churn Prediction of Employees Using Machine Learning Techniques,” Tehnički glasnik, vol. 15, issue 1, pp. 51-59, 2021. [Online]. Available: https://doi.org/10.31803/tg-20210204181812.

[19] R. S. Shankar, J. Rajanikanth, V. V. Sivaramaraju, and K. V. Murthy, “Prediction of Employee Attrition using Datamining,” in proc.2018 IEEE International Conference on System, Computation, Automation, and Networking, ICSCA 2018, [Online]. Available: https://doi.org/10.1109/ICSCAN.2018.8541242.

[20] Q. Zhu, J. Shang, and X. Cai, “CoxRF: Employee Turnover Prediction based on Survival Analysis,” 2019 IEEE Smart World, Ubiquitous Intelligence, and Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People and Smart City Innovation, 2019. doi: 10.1109/SmartWorld-UIC-ATC-SCALCOM-IOP-SCI.2019.00212.

[21] O. Thebald, Machine Learning for Absolute Beginners, 2nd edition, Independently published , 2018.

[22] S. Shalev-Shwartz and S. Ben-David, Understanding Machine Learning from Theory to Algorithms, NY, USA: Cambridge University Press, 2014.

[23] G. James, D. Witten, and T. Hastie, An Introduction to Statistical Learning with Application in R, NY, USA: Springer Science+Business Media, 2017.

[24] J. Brownless, “4 Types of Classification Tasks in Machine Learning,” Machine Learning Mastery, Aug. 2020. [Online]. Available: https://machinelearningmastery.com/types-of-classification-in-machine-learning/.

[25] G. Edwards, “Machine Learning: An Introduction,” Medium, Towards Data Science, Jan. 2020. [Online]. Available: https://towardsdatascience.com/machine-learning-an-introduction-23b84d51e6d0.

[26] S. Gupta, “Pros and Cons of Various Classification ML Algorithms,” Medium, Towards Data Science, June 2020. [Online]. Available: https://towardsdatascience.com/pros-and-cons-of-various-classification-ml-algorithms-3b5bfb3c87d6.

[27] EliteDataScience.com, ” Modern Machine Learning Algorithms: Strengths and Weaknesses,” EliteDataScience, June. 2020. [Online]. Available: https://elitedatascience.com/machine-learning-algorithms.