**HR Analytics**

**Problem Statement:**

Every year a lot of companies invest in their employees through HR Analytics. The aim of these programs is to increase the effectiveness of their employees. But where does HR Analytics fit in this? and is it just about improving the performance of employees or are there other benefits as well?

HR analytics aims to provide insight into each process by gathering data and then using it to make relevant decisions about how to improve these processes. HR analytics does not just deal with gathering data on employee efficiency - it also aims to help organizations get a better return on investment.

Attrition in human resources refers to the gradual loss of employees overtime. HR professionals assume a leadership role in designing company compensation programs, work culture, and motivation systems.

A major problem in high employee attrition is its cost to an organization. Job postings, hiring processes, paperwork, and new hire training are some of the common expenses of losing employees and replacing them. Additionally, regular employee turnover prohibits your organization from increasing its collective knowledge base and experience over time. This is especially concerning if your business is customer-facing, as customers often prefer to interact with familiar people. Errors and issues are more likely if you constantly have new workers.

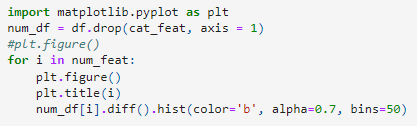
**Data Analysis and EDA:**

The data is collected from 1470 employees for 35 different features. Those features are

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

By inspection, we came to know that the columns StandardHours and EmployeeCount has only single value. This type of features won’t contribute in the model so we can remove those features.

We have plotted histogram for each of the features to take a look at the distribution of the data for that feature. Following code was used to create histogram.



If the distribution is normal then it is good for the model designing. But if the data is skewed then we have to remove or reduce the skewness. By measuring the skewness of the features, it is found that the following features have large skewness.

"DistanceFromHome", "JobLevel", "MonthlyIncome", "NumCompaniesWorked", "PercentSalaryHike", "StockOptionLevel", "YearsAtCompany", "YearsInCurrentRole","YearsWithCurrManager"

A snapshot of the code and skewness is shown below.

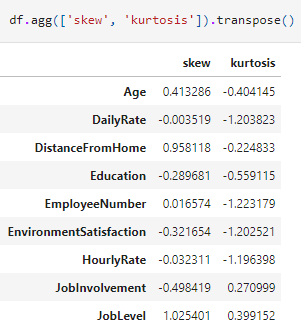


Figure with skewness

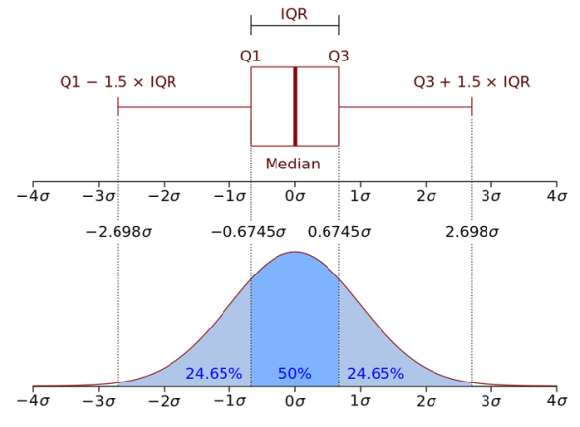


Figure reduced skewness

Figure 2 Reduced skewness

Figure 1 Skewness

Since the features have large skewness, have positive skewness so we can take root of the instances to reduce the skewness. In the above two figures, it is showing the skewness and reduced skewness of some of the features.

The real data is not cleaned and it may have missing values or outliers. Outlier is an instance whose value is extreme. For example in the “Age” column the usual values ranges from 18 yrs to 70 yrs but if the value in the “Age” column is 200 yrs then it is an outlier and we have to handle such outliers for each of the column. We will plot the data in the form of boxplot. The schematic structure of the boxplot is shown below.

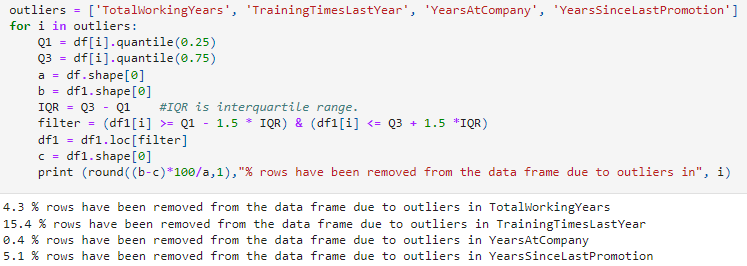
Q1- First quartile

Q3- Third quartile

IQR- Interquartile range

Any instance lying beyond and will be considered as an outlier and we will remove that instance. The following columns contains outliers.

'TotalWorkingYears', 'TrainingTimesLastYear', 'YearsAtCompany', 'YearsSinceLastPromotion'

We will execute the following code to remove the outliers and its output is also shown below.

The data is expensive and we can’t lose huge amount of data while handling the outliers. The outliers in TrainingTimesLastYear are 15.4% of the total data. We can’t remove all of those instances so we will avoid removing the outliers for that column for now.

**Preprocessing:**

Outliers belong to the numerical features. But some of the features are categorical features. The machine learning models do not accept the strings but numbers. So we have to convert those strings into numbers. Categorical features may have multiple unique categories but if any of the features has single category then it is not going to contribute to the model. We can remove those features having single category.

To convert categorical features into numbers encodes can be used. There are several encodes available in python. The encoder simply creates a mapping between the categories and numbers. For example, by using label encoder on Gender features, “male” will be converted to “1” and “female” will be converted to “0”. In this project we will be using helmert encoder. The working of helmert encoder is similar to the dummies variables but the way it creates the variables is a bit different. For example, the column “Gender”, a new column will be created in a new dataframe with a “Gender” column. The “Female” will be converted to “-1” and male will be converted to “1” after applying the helmert encoder to the dataframe. This encoder gives best result when the categorical variables are in the ordered form. But it also performs well otherwise.

After handling the categorical variables, the data has to be divided into features and targets. Following code will split the data into features(X) and targets(y).



Our target variable is “Attrition” and it has two values “Yes” or “No”. The first line of the code is used to create a mapping such that “Yes” change to “1” and “No” change to “0”. This is done because the machine learning models do not accept the strings but numbers. Some fraction of the data should be used to train the model and rest of the data should be used to test the model. In this project we will keep this ratio as 80/20. 80% of the data kept for the training and the 20% for the testing.



It is advised to shuffle the data before splitting so that if any order exists between the instances should be randomized.

**Machine Learning Models:**

1. **Logistic Regression**

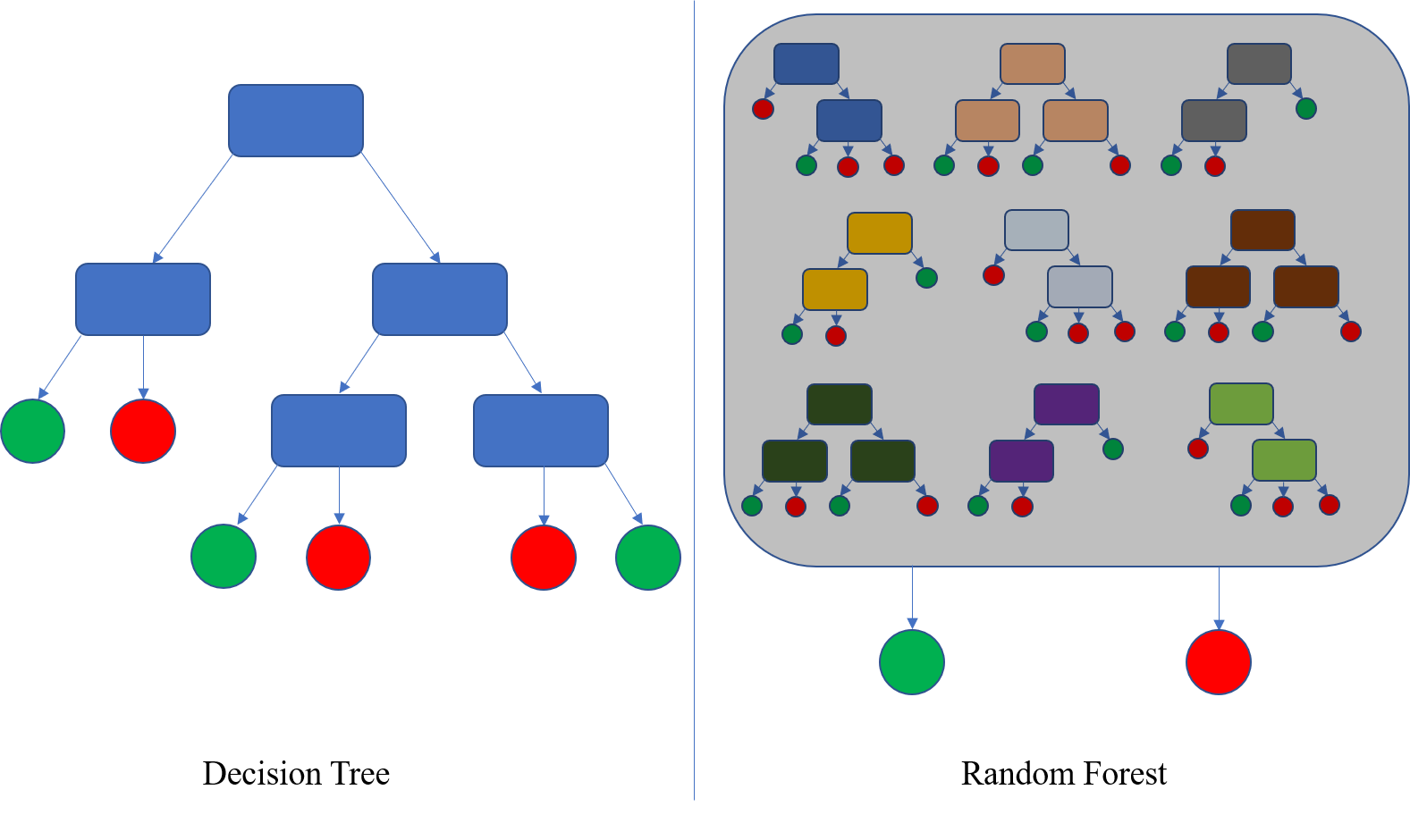
The logistic regression is modification of the linear regression. It is not a regression model but a classification model. The model will try to fit the data with respect to the following function.

Where P stands for the probability and establishes a linear equation.



It is found that the accuracy of the logistic regression is 86% for training data and 90% for the test data. Since the accuracy for the test set is more for the training data, there is no overfitting.

1. **Decision Tree Classifier**

The decision tree classifier classifies the data by generating some questions by itself. The last leaf red and green shows whether attrition happens or not.

The following code will train the decision tree model on given data.



The decision tree is trained so well that it can predict 100% of the test data correctly. In case decision tree fails to perform so well, it can be improved by changing the threshold of decision tree.

1. **Random Forest Classifier**

Random forest is an ensemble based model. It consists of number of decision trees. It accumulates the prediction of the several trees and returns a single prediction.



As decision tree, random forest is also able to predict all of the test data correctly. We don’t need to fine tune the random forest. In case accuracy is low, it can be improve by grid search CV or random forest. Furthermore, ROC and AUC can be used to set the threshold value of the prediction.

1. **Stochastic Gradient Descent Classifier**

It is an optimization algorithm and following code can be used to fit SDG on given data.



The SDG only able to perform with 83% accuracy. This accuracy is much less than the decision tree and random forest.

**Conclusion:**

Out of the four models, decision tree and random forest able to perform with 100% accuracy. We will prefer decision tree as our final model because it simple model and computation wise less expensive. On the other hand random forest needs more calculations than the decision tree. The following code will save the model as pickle file and this pickle file can be reloaded as a decision tree model.