**CS 513 KDD Final Report**

**Neil Gupte**

**10445674**

* Final Project titled the Data Analysis of employee attrition dataset utilizes most of the classification algorithms covered in the lectures and the model are built using libraries in **R**. The purpose of the project is to build a reliable model for Employee Retention using the given dataset. Many organizations treat their employees like they’re a dime a dozen. These companies think that even their best workers can easily be replaced, particularly in today’s tricky economy where there is no shortage of folks looking for work. They're wrong. Great businesses understand the importance of keeping their employees for a long time.
* Following underscores, the importance of employee retention
* You Cannot Build a Business Without Consistency: retention is great for ROI.
* Turnover Crushes the Bottom Line: According to the Center for American Progress, it costs roughly 20% of an employee’s salary to replace that individual. So when your employees are constantly leaving for greener pastures, you’re forced to spend the equivalent of over two months’ worth of salary just to find someone to replace them.
* You Lose Talent and Ideas:
* It is Difficult to Establish Camaraderie with a Transient Workforce
* Customers Notice When Names Change: Your clients have been interacting with employees for some time. When the people they are used to dealing with leave, it can put a bad taste in their mouth, leaving your future working relationship with that organization in doubt.
* Job Seekers Notice Too: Thanks to sites like Glassdoor, it’s easier than ever to get a glimpse into what it’s like to work at an organization — even if you just found out that organization exists in the first place. Almost 50% of candidates use Glassdoor during their job search, according to Software Advice.
* Constantly Training New Employees Is a Waste of Resources: Those managers you hired? You almost certainly did not hire them to spend their time training new hires over and over again. You hired them to help employees develop and to grow your business.
* Your Competitors Could Benefit Directly: When you lose a talented employee, maybe that person decided to go back to school, move across the country, or switch industries altogether. Or maybe that individual sought employment at one of your competitors — the one that seems to be most respected and offers its employees the best perks. Worse yet, maybe that organization went on the offensive and approached your employee first.

The dataset (attrition\_data.csv) has 9612 rows and 27 columns. The dataset contains various parameters related to the employee's personal information and information related to his work performance. The objective is to build a classifier using the given data which can help in employee retention.

The first step in the process is to carry out Exploratory Data Analysis and Preprocessing. **Exploratory Data Analysis (EDA)**, also known as Data Exploration, is a step in the Data Analysis Process, where several techniques are used to better understand the dataset being used. Using summary function available in R we get a gist of all necessary important stats about the data. While doing it we find out there are 5394 missing values in TERMINATION\_YEAR column. Which we decide to replace with a zero. There are various algorithms which we will be using further which do not allow missing values in the data. R provides with library called **ggplot2 ,** which provides aesthetically pleasing functions to visualize the data. We then create various frequency distribution graphs, scatterplots, histograms, and bar charts to find certain pattern in the given data and to visualize them.

A classification algorithm, in general, is a function that weighs the input features so that the output separates one class into positive values and the other into negative values. The following are the algorithms used to classify our dataset: KNN (weighted and unweighted), Naïve Bayes Classifier, Dtree Classifer, Random Forest and Neural Networks. We go on building models using each algorithm to figure which gives the best results.

The first model is built using K-nearest neighbors. KNN can be used for both classification and regression predictive problems. However, it is more widely used in classification problems in the industry. To evaluate any technique, we generally look at 3 important aspects:

1. Ease to interpret output

2. Calculation time

3. Predictive Power

KNN is a non-parametric. When we say a technique is non-parametric, it means that it does not make any assumptions on the underlying data distribution. In other words, the model structure is determined from the data. After doing the min max normalization, we created unweighted knn model using the **kknn** library provided in R and parameter of kernel=“rectangular”. We get accuracy of 61.06103%.Next we do weighted knn using the parameter of kernel =“triangular”. The weighted knn produces accuracy of 60.26352% which is slightly lower than unweighted knn.

The next classifier we use is the Naïve Bayes Classifier. A Naive Bayes classifier is a probabilistic machine learning model that is used for classification task. The crux of the classifier is based on the Bayes theorem.

We use the **naiveBayes()** function in R to build the model. We get an accuracy of 61.1997% from the model which is similar to knn.

The model we use next is the decision tree classifier. A Decision Tree is a simple representation for classifying examples. It is a Supervised Machine Learning where the data is continuously split according to a certain parameter. We use the **rpart** library to build the model and **rattle** to visualize the model. From the dtree plot we get to see that which parameter does the algorithm pick for classification. We get to see that if the prev 5 year rating is greater than or equal to 0.5 then he has 79.21 % probability of being active. It means that there is a pattern where employees who have worked of more than 5 years seem to be loyal to the company. Thus, we can read the whole tree and obtain needed information. We get an accuracy of 62.55201% from the model.

The next classifier in line is the Random Tree Classifier. Random forest, like its name implies, consists of a large number of individual decision trees that operate as an [ensemble](https://en.wikipedia.org/wiki/Ensemble_learning). Each individual tree in the random forest spits out a class prediction and the class with the most votes becomes our model’s prediction.

We have the parameter importance graph wherein we have Mean Decrease in Accuracy and Mean Gini Decrease.

Mean Decrease in Accuracy the number or proportion of observations that are incorrectly classified by removing the feature (or values from the feature) in question from the model.

GINI importance measures the average gain of purity by splits of a given variable. If the variable is useful, it tends to split mixed labeled nodes into pure single class nodes. Splitting by a permuted variable tend neither to increase nor decrease node purities. Thus, we get the important parameters to consider in our classifier using this which we plot next. We use **randomForest()** function available inside the **randomForest** library to build the model. We get an accuracy of 63.34951% from the model.

We finally come to our last model using the Neural Network Classifier. Neural networks are a set of algorithms, modeled loosely after the human brain, that are designed to recognize patterns. They interpret sensory data through a kind of machine perception, labeling or clustering raw input. The patterns they recognize are numerical, contained in vectors, into which all real-world data, be it images, sound, text or time series, must be translated. Here we use the **neuralnet()** function from **neuralnet** library to build the model. We build the model with 5 neurons in the hidden layer and threshold value 0.05. We get an accuracy of 56.90014% from the model.

After building all the models and comparing their accuracies we found out that Random Forest Classifier produces the best results for the given dataset.We get an accuracy of 63.34951%. Random Forest is especially good because of the following reasons:

**1)Random Forest is Versatile: I**t can handle binary features, categorical features, and numerical features. There is very little pre-processing that needs to be done. The data does not need to be rescaled or transformed.

**2)Parallelizable**: They are parallelizable, meaning that we can split the process to multiple machines to run. This results in faster computation time.

**3) Great with High dimensionality**

**4) Quick Prediction/Training Speed**

**5) Robust to Outliers and Non-linear Data**

**6) Low Bias, Moderate Variance**

**Most of the algorithms did not perform well on the given dataset that means there is high variance between the columns and the parameters are not clearly separable. There is still a lot of room to improve the accuracy of the model, hyper tuning some of the parameters can produce better results than what we have acquired.**