IBM Employee HR Attrition

3X Data Mining

Mohammed Topiwalla, Patricia Londono, Sanchita Kumari, Valerio Trotta, Igor Pedevani, Millicent Ottchere



# Problem Statement

**How can we reduce IBM company's attrition rate by predicting if a candidate will exit in India within the year?**

* **S**pecific :- To Indian geography in IBM
* **M**easurable:- To reduce attrition rate(By at least 5%)
* **A**ction oriented:- Reduce employee attrition & suggest employee engagement & satisfaction programs
* **R**elevant:- Direct impact on company's top and bottom line
* **T**ime bound :- 12 months

Our client is IBM a leading firm and in the IT sector. It is recently facing a steep increase in its employee attrition . Employee attrition has gone up from 14% to 25% in the last 1 year . We are asked to prepare a strategy to immediately tackle this issue such that the firm’s business is not hampered and also to propose an efficient employee satisfaction program for the long run. Currently, no such program  is in place . Further salary hikes are not an option.

The attrition problem is not only unique to ibm but to other IT companies such as Infosys, India’s second largest IT services company, that is also battling high attrition, with a peak attrition of 20.4 % in the October-December quarter of FY15.

## **Methodology to solve the problem**

## **Data cleansing and Smote**

Our data set had no missing values therefore no special treatment was required, however since we are running a classification algorithm

1. We had many categorical variables which we needed to convert into dummy variables or ordered integer variables
2. There was a massive class imbalance to the proportion of 83:16 , in order to tackle this we understood that even without any algorithm if we predicted that the employee wouldn’t leave we would still be right 83% of the time. So any model we make should be able to provide a better accuracy than this or else it would not be worth it.
3. The second solution was to balance the classes (A paper by Chawla led us the way - > <https://www.jair.org/media/953/live-953-2037-jair.pdf>)
4. With the help of smote and code help from stack overflow we achieved Smote
   1. What smote does is simple, it first of all takes the class with low proportion and artificially boosts its values to increase the number of records, it tries to generate new rows by replicating them in the bases of a range of values already present making various random combinations
   2. It takes the overpopulated proportion and performs sampling to pick up those rows which are the most representative of the population and then it uses them to reduce this proportion
   3. This is repeated till the classes are nearly balanced
5. Two techniques of smote from 2 different packages were used therefore we have 2 smote data sets

Who made this – Mohammed

Who reviewed this – Sanchita , Igor

## **Classification**

In order to create an early warning system, it was essential to perform a supervised classification.

We had 3 sets of data

1. The original data (With one hot encoding)
2. Data generated from SMOTE (Set1)
3. Data generated from SMOTE (Set 2)

Before running any model it was ensured that the best parameters were selected on the basis of validation set or cross validation set. The splits were always on the basis of 50:25:25 (Training:Validation:Testing). All the sampling is stratified sampling in order to ensure perfect class balance

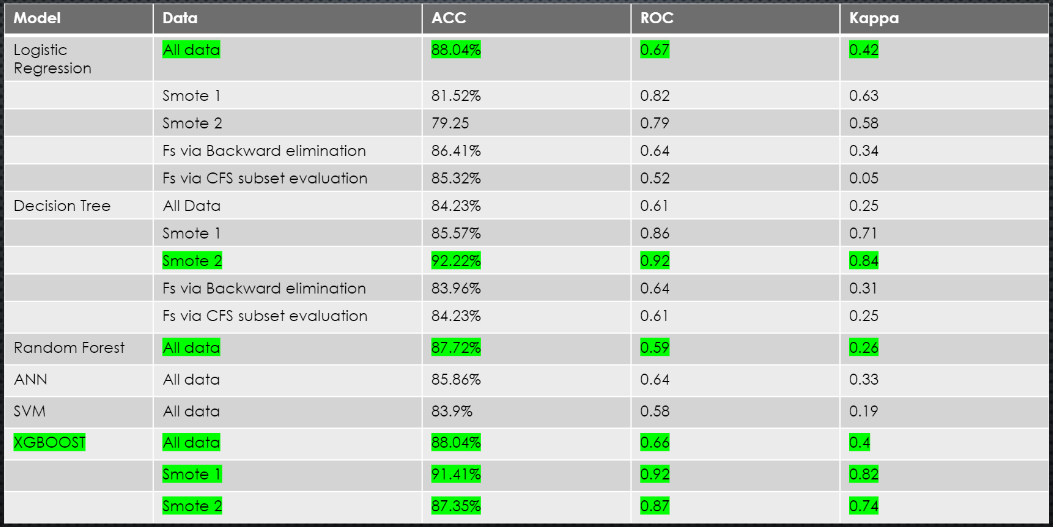
The approach was to run simple models like

1. Decision tree
2. Logistic Regression

On both these models we would train each of the 3 data sets

After the process was completed we trained data on complex models live

1. Support vector machines
2. Artificial Neural Network
3. Random forest
4. Extreme Gradient Boosting

The output for running all those models are 

The interpretation is simple SMOTE data is the winner and decision tree stole the show!

But what are we missing at times it may be necessary to explain the HR our results, why our model believes our person is planning to leave, in that case decision tree or logistic regression are the best options.

However, if why decision made was not required we can simply use more powerful models like XGBoost, because as data increases the accuracy of XGBOOST will be higher than a decision tree because it’s a powerful ensemble algorithm.

Who made this – Mohammed

Who reviewed this – Patricia , Valerio