Data mining on HR Employee Attrition dataset:

In problem statement we have HR employee attrition data to be analyzed and also have definition of data. In a data, we have “**Attrition**” as dependent variable and **34** independent x variables which impacts the attrition rate. Through analysis we have to find out of 34 variables how many variables are impacting the attrition rate. There are total **2940** observations present which have both records i.e. employees who have left the company and employees who have not.

Attrition “Yes” denotes employees who have left the company and “No” represents employees who are still with the company. We have to build a model that will help the organization in predicting whether particular employee will stay with the company or not. Also, these models will help company to change the strategy to reduce the attrition rate.

When we start analyzing the given data, we found that out of 2940 observations, 2466 observations are of employees who are still with the company means over 83% data is having attrition “No” records. This clearly shows data is skewed and we have to do oversampling in order to balanced out the data.

|  |
| --- |
| **0: Employees who are still with the company (2466)**  **1: Employees who have left the company (474)** |

Post oversampling of “Yes” records. We have total **4932** observations and data is balanced out i.e. 50 % records are of “Yes” and remaining 50% of “No”.

|  |
| --- |
|  |

When we observed the type of data, we realized that most of the variables are not in factors which supposed to be. So, we converted below variables in factors:

* Education
* EmployeeCount
* EnvironmentSatisfaction
* JobInvolvement
* JobLevel
* JobSatisfaction
* RelationshipSatisfaction
* StockOptionLevel
* WorkLifeBalance

|  |
| --- |
|  |

**Identify columns which are of no use. drop those columns**

By looking at the data, we removed some of the unwanted variables.

|  |
| --- |
| ##Factors having single value  hrData$EmployeeCount = NULL  hrData$Over18= NULL  ### Not required for analysis  hrData$EmployeeNumber = NULL |

When we performed logistic regression on the given data set and identify the significance of important variables using anova technique, we found out that 4 variables are insignificant hence we removed those variables and rebuild the model.

|  |
| --- |
|  |

**Cart Technique:**

CART technique basically provides the decision tree as an output which depicts how data is classified and grouped.

To start with the CART technique, we need to split the data into training and test set in 70:30 ratio as given in the problem statement.

|  |
| --- |
| **#Split training and test data set as 70:30**  set.seed(1)  split = sample.split(hrData$Attrition, SplitRatio = 0.70)  **# Create training and testing sets**  HRDATA.train = subset(hrData, split == TRUE)  dim(HRDATA.train)  HRDATA.test = subset(hrData, split == FALSE)  dim(HRDATA.test)  **# Training and test data set records count**  dim(HRDATA.train)  [1] 3452 28  > HRDATA.test = subset(hrData, split == FALSE)  > dim(HRDATA.test)  [1] 1480 28 |

When we build the CART model using **rpart**, we found the below node split details:

**Primary splits**:

JobSatisfaction splits as RRLL, improve=20.66251, (0 missing)

Age < 25.5 to the left, improve=12.25319, (0 missing)

PercentSalaryHike < 14.5 to the left, improve=11.61538, (0 missing)

TotalWorkingYears < 4.5 to the left, improve=10.95102, (0 missing)

DailyRate < 1285 to the right, improve=10.31177, (0 missing)

**Surrogate splits**:

DistanceFromHome < 2.5 to the left, agree=0.699, adj=0.389, (0 split)

TotalWorkingYears < 5.5 to the left, agree=0.694, adj=0.378, (0 split)

Age < 31 to the left, agree=0.683, adj=0.356, (0 split)

WorkLifeBalance splits as RLRR, agree=0.661, adj=0.311, (0 split)

JobRole splits as --R---LRL, agree=0.656, adj=0.300, (0 split)

|  |
| --- |
|  |

By plotting the graph, it is clear that **Overtime** is the decisive node that has “Yes” and “No” values. Root node got further split into **YearsWithCurrentManager** >= 1 and having **Job role** as 1,4,5 and 6.

**YearsWithCurrentManager** >= 1 got further split into employees who are having **stock options** 1 and 2 and second split is **Age** >33.

Likewise, decision tree got constructed in which there are 5 primary splits and 5 surrogate splits that explains the data.

|  |
| --- |
| **#Predict performed on test data**  predictionCart <- predict(Model.CART, newdata=HRDATA.test, type="class")  predictionCartTrain <- predict(Model.CART, newdata=HRDATA.train, type="class")  #CART Accuracy and Confusion matrix  tCart <- table(HRDATA.test$Attrition, predictionCart)  tCart  #CART model accuracy  (tCart[1]+tCart[4])/(nrow(HRDATA.test)) |

By performing the prediction on test data set we build the confusion matrix as below:

|  |
| --- |
| predictionCart  No Yes  No 593 147  Yes 176 564  > #CART model accuracy  > (tCart[1]+tCart[4])/(nrow(HRDATA.test))  0.7817568 |

CART model giving us the accuracy of 78%. CART model is good at identifying employees who will stay with the company as compared to employees who would leave the company though it is not that bad in identifying employees who would leave the company. Difference between these two is quite less so we can conclude that this model will performs well in identifying both the categories of employee.

**Neural net technique:**

To start with neural net, we have to scaled the data since neural net does not scale in on its own.

Below numeric variables need o be scaled so that it would not suppress the effect of other variables on dependent variable.

* Dailyrate
* HourlyRate
* MonthlyIncome
* PercentSalaryHike
* TotalWorkingYears
* TrainingTimesLastYear
* Age

We scaled the numeric data using dplyr library.

|  |
| --- |
| ################################# Scaling data ####################  library(dplyr)  numericHRDATA <- select\_if(hrData, is.numeric)  ################################# Scaling data #################### |

We build the neural net by setting hidden layer as 3, threshold as 0.013 and maximum permissible count as 40,000.

|  |
| --- |
| nn1 <- neuralnet(formula = AttritionYes ~ . ,  data = scaledDATA.train,  hidden = c(3),  #err.fct = "sse",  linear.output = FALSE,  lifesign = "full",  lifesign.step = 10,  threshold = 0.013,  stepmax = 40000,  #startweights = startweightsObj ## Comment this as well if load is commented  ) |

Below is the neural net plot, which clearly depicts the number of hidden layers involved and how variables are contributing to calculate the associated weights.

|  |
| --- |
|  |

|  |
| --- |
|  |

We build the neural net model on scaled training data set by spitting it into 70:30 ratio. After that we predicted the result on scaled test data set to measure the accuracy of the model.

From the histogram, it is clear that model is almost equal in identifying both Attrition “Yes” and “No” cases with over 90% accuracy.

* discreteValuesEnsemble<-ifelse(ENSPrediction>0.5,1,0)

Here we prepare the discrete value set by having 1 with probability > 0.5 and rest of the data is 0.

**Confusion matrix and model accuracy**

|  |
| --- |
| discreteValues  0 1  0 671 69  1 68 672  > (t2[1]+t2[4])/(nrow(newHRDATA.test))  [1] 0.9074324 |

From the output of both the techniques it is quite evident that neural net has outperformed CART model since accuracy of the NN model is high.

We can conclude that neural net model will be the good option for the company to control the attrition rate since it is equally good at identifying the employees who would leave the company or not.

**Ensemble technique:**

In ensemble technique, we basically prepare one test data set from the training dataset. Our data set contains one dependent variable y which is actual output i.e. Attrition column and two independent variables which are basically output of CART and NN models.

This complete data we sliced into test data set further to perform the liner regression on it.

|  |
| --- |
| CART O/P    Linear regression  Model  Neural net O/P |

|  |
| --- |
| ENSModel <- lm(formula = Expected\_Output ~., data = ENSDataSet)  summary(ENSModel)  ENSPrediction <- predict(ENSModel,newdata = ENSDataSetTest)  discreteValuesEnsemble<-ifelse(ENSPrediction>0.5,1,0) |

We performed the logistic regression on training data set and predicted the output of it by passing the test data set.

|  |
| --- |
| discreteValuesEnsemble  0 1  0 671 69  1 68 672  > #Ensemble model accuracy  > (tEnsemble[1]+tEnsemble[4])/(nrow(newHRDATA.test))  [1] 0.9074324 |

From the discrete values it is clear that ensemble technique is giving the exact same result of neural net model with same accuracy.

So, we can say that ensemble technique did not outperformed NN but CART technique.