International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2015): 6.391

Attrition Prediction Using Machine Learning to Help in Astute Decision

Reshad Abdullah[1], Sachin Bojewar[2]

[1]Scholar, Masters Degree Program in Information Technology, Vidyalankar Institute of Technology (VIT), University of Mumbai, India

[2]Associate Professor, Department of Information Technology, Vidyalankar Institute of Technology (VIT), Mumbai, India

Abstract: ***Industries, especially IT (Information Technology) today, are experiencing high employee attrition rate. The employee leaving voluntarily is not good for organization or to project in which they are working. Hence HR and senior managers and the policy makers of any industry are working together to reduce this voluntary exit. A good leader senses and understands employee needs and work with them and HR to fix the issues. However not all attrition causes are known to managers and when it actually happens it turns out as a surprise, then they are not able to do much. Some amount of attrition is certain and bound to happen like employee retiring or death of employee hence the scope of this work is only restricted to voluntary exit [1]. Organization and HR department has felt that if they would have known earlier, or they could have picked the sign of exit, they might have prevented good employees leaving. With vast amount of historical data available within the organization, and through analytics & machine learning it is possible to predict attrition. These tools not only predict but also show some clear pattern in attrition. Many organizations today uses cots attrition prediction tool or build their own in-house prediction tool. The scope of this work is implementation of my theory paper published "Attrition Prediction- Need of the Hour for Companies" [1]. A tool is developed to predict attrition and it also predicts reason for attrition, this tool is based on decision tree algorithm and developed in R language. The factor or reasons for attrition are then effectively used by managers and HR department to design a retention strategy for the employee or proactively find his replacement. At the same time management becomes aware of the situation and are in position to predict how much new backup recruitment can be done in future.***

Keywords: Attrition, COTS, C45, C50, ID3, exp, Model, Machine learning, Notice period, Retention, r_dt10

1. Introduction

Attrition is "the normal reduction of the employees caused by leaving the organization for reasons of retirement, death, or resignation" (National Performance Review, 1997) [2]. Certain attrition cannot be avoided like death or retirement; hence the scope of this work is only restricted to voluntary exit. Shortage of employee is another problem which becomes a concern when attrition is high. The work presented here is implementation of my theory paper published "Attrition Prediction- Need of the Hour for Companies" [1]. A tool is developed in R language and experiments are done and results are compared to draw conclusion. Results also explain how it is used by industry to predict attrition and how it could be better than some other tool available.

2. Proposed Solution

There is enormous amount data stored in every organization. Ability to collect and analyze data and then make decisions based on powerful insights has exploded in recent years. with the kind of tools available and domain expertise and data scientist organization today are able to make a correct predictive analysis.

Below are solution steps using decision tree and machine learning technique. This is depicted in block diagram shown in Figure 1

- First identify the features (variables / attributes) within data available which is likely to impact attrition.
- Select a model to predict employee attrition using the feature identified.
- Detect the most high contributor/ factor for employee attrition.
- HR/ Manager needs to work on the factor to reduce attrition or decide on backfill depending on the cost factor.
- HR / Manager domain knowledge is leveraged to provide feedback to model for improvement in future prediction. This is how the model learns to increase the accuracy.

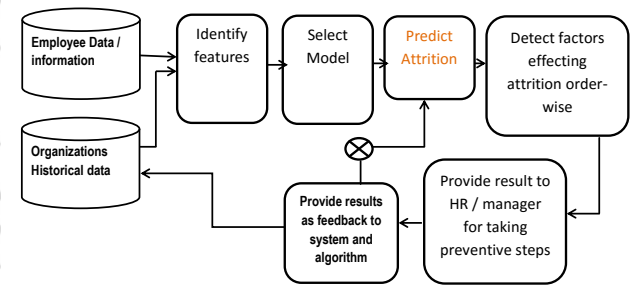


Figure 1: proposed solution block diagram

3. Implementation Snapshots

3.1 Model

It involves building a system based on classification algorithms (ID3 or C4.5/5.0) using the proposed block diagram shown in fig 1.

In this phase a model is created which is based on decision tree, ID3 [4] algorithm. The name of tool or model is given as r_dt10 which is r decision tree algorithm version 1.0 "r_dt10".

3.2 Training and Testing Data

Important part of this experiment is right input data which can be used for comparison. This data is closed to actual employee data, however due to confidentiality few attributes data is changed and hence this data is considered to be testing data. This data is employee sensitive data and cannot be known to any other department then to HR of the organization. Hence due to reason stated above the current data used here are considered to be testing data.

This testing data is around 600 records having 18 attributes of predictor variable and 1 as target or class attributes which is a Boolean value attributes. The target or class variable can have only either yes or no values means if yes then attrition occurred and employee left the organization if no then employee remained with the organization. For first experiment the testing and training data is only of 30 rows this data is kept in comma separate .csv file. For Experiment no 2 there is 600 record which is divided into training and testing set , the training set is of 500 data while the testing set is of 100 rows data. This is kept in 2 different comma separated file. Table 1 below shows the sample data which can best fit here. User who wants to use this tool can have their own sample data only care should be taken for class attribute as stated above.

Table 1: Sample attributes data

emp_id	age	gender	degree	work exp	occupation	join_date	sal	marital status	.....	attrition
CI0001	25	M	BE	4	IT	22-Dec-14	450000	M		yes
CI0002	35	F	G	12	SALES	14-Mar-10	1100000	Y		no
CI0003	32	M	BE	10	HR	29-Mar-10	900000	N		no
CI0004	45	M	PG	22	ADMIN	2-Sep-11	1800000	M		no
CI0005	27	M	G	4	IT	23-Apr-12	410000	M		no
CI0006	28	M	G	5	IT	19-May-13	500000	M		no
CI0007	30	M	BE	5	IT	20-Jan-14	500000	M		yes
⋮										
CI0500	21	M	G	0	IT	28-Nov-15	105000	S		no

3.3 Functional Module

This Phase involves building a system based on classification algorithms (ID3 & C4.5/5.0) using similarity and relevance computation.

Fig 2 shows functional diagram which is built on ID3 algorithm. It has mainly 4 function and 1 test script. The main four function are r_dt_create, r_dt_predict, r_dt_createTree, r_dt_drawTree respectively. Each are called using test script and test data provided.

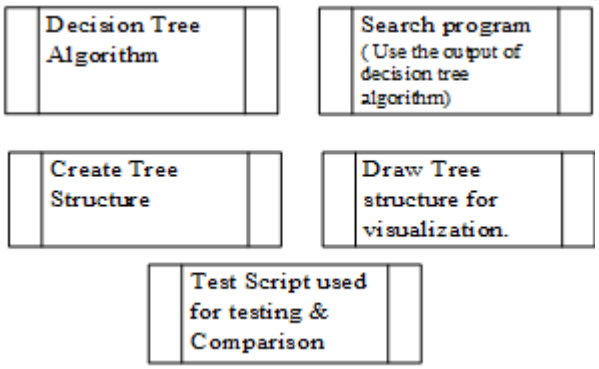


Figure 2: Functional module of r_dt10 attrition prediction

3.4 Software Requirements

1. Operating System: Windows 7 or LINUX
2. Programming Language: R [3]
3. IDE: R Graphic Viewer
4. Tools: Excel , File System , Text File

4. Results & Discussion

The aim of this section is to present the results of empirical analysis done for Attrition prediction. At the end of this section, results are compared which shows the effectiveness of the proposed solution approach and conclusion is presented.

4.1 Performance Results Exp. 1

The Data set consist of 30 records of multiple attributes. This data set is training set as well as testing set so that we can accurately predict if there is change in the result generated. The data set is stored in "emp_det_exp1.csv" file. Multiple iteration is performed to check improvements this is presented below.

4.1.1 Test results: Iteration 1

In this the actual testing scripts is executed in the R tool and results are recorded as below. The test script used for testing is in file "R_dt10_testscipt_exp1.R". the below are the steps & output.

Step 1: Load the dataset file, Run the **r_dt_create** function with valid input data file.

Step 2: create test set same as training set, Run the r_dt_predict function with valid input.
Step 3 : Compare the result with confusion matrix. Below Fig 3 is the output screen of confusion matrix with actual data and predicted data.

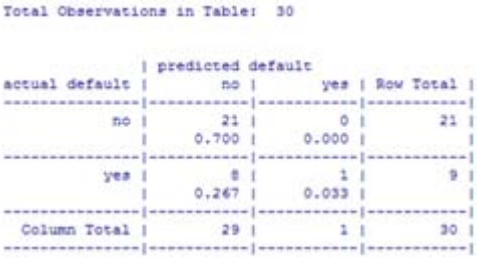


Figure 3: Confusion matrix for train & test data exp. 1

From the above matrix it is evident that the output True Positive (Yes) is not predicted rightly. Hence we will try to find the reason why it is so. Later perform corrective action in the next iteration.

Step 4: Using single record test set , from the test set, create tree structure using function r_dt_createTree by passing valid parameter.

Step 5: Output of Step 4 pass to the function r_dt_drawTree, the output of this function is shown in below screen Fig 4.

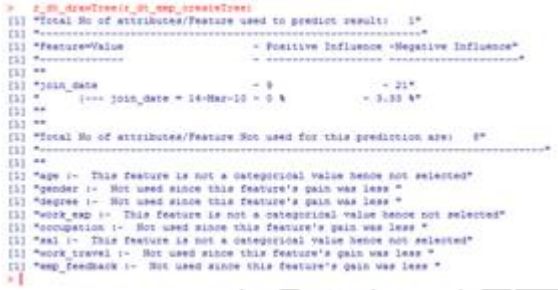


Figure 4: Tree structure for exp 1

From the above figure 4 it is evident that only join_date attribute is used to predict the output which is not even a factor type variable hence in next iteration we will remove this attribute and test.

4.1.2 Test results: Iteration 2

In this iteration all the steps are same except that we remove or suppress join date from the test data. Below are the steps & output for iteration 2.
Step 1: Load the dataset file, Run the r_dt_create function with valid input data file.
Step 2: Create test set same as training set , Run the r_dt_predict function with valid input.
Step 3: Compare the result with confusion matrix. Below Fig 5 is the output screen .

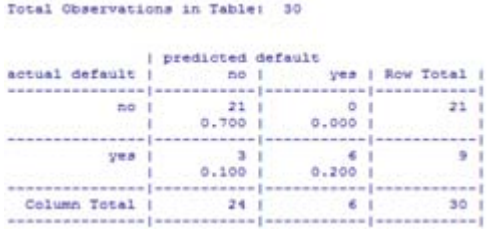


Figure 5: Confusion matrix for train & test data exp. 1 iteration 2

From the above confusion matrix it is proved that there is improvement in results for the True Positive (Yes) case. Now we will use single record steps to check what attributes which are used or not used.

Step 4: Using single record test set , from all the test set, create tree structure using function r_dt_createTree by passing valid parameter.

Step 5: Output of Step 4 pass to the function r_dt_drawTree, the output of this function is shown in below screen Fig 6.

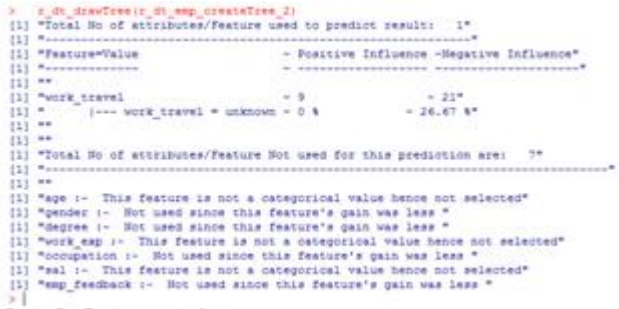


Figure 6: Tree structure for exp 1 iteration 2

From the above screen it is evident that although there is improvement in results only work_travel attribute is used to predict the output which could be the best feature for attrition but still we can try to add some more feature by changing the attribute in next iteration. Hence check if we can improve in prediction.

4.1.3 Test results: Iteration 3

In this iteration all the steps are same except that we make attribute age as categorical data. Below are the steps & output for iteration 2.

Step 1: Load the dataset file, Run the r_dt_create function with valid input data file.
Step 2: Create test set same as training set, Run the r_dt_predict function with valid input.
Step 3 : Compare the result with confusion matrix. Below Fig 7 is the output screen.

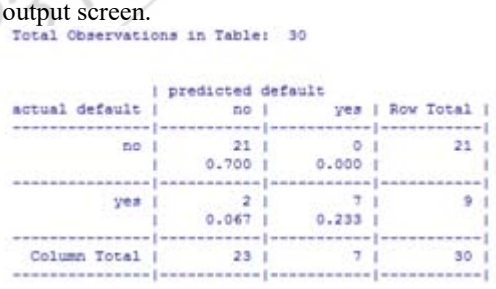


Figure 7: Confusion matrix for train & test data exp. 1 iteration 3

From the above confusion matrix it is proved that there is slight improvement in the True Positive (Yes) result. Now we will again do the single record steps to check what attributes are used or not used.

Step 4: Using single record test set, from all the test set, create tree structure using function r_dt_createTree by passing valid parameter.

Step 5: Output of Step 4 pass to the function r_dt_drawTree, the output of this function is shown in below screen Fig 8.

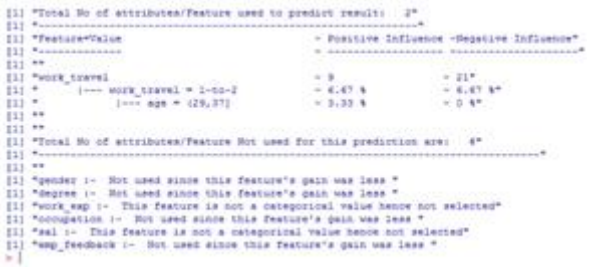


Figure 8: Tree structure for exp 1 iteration 3

From the above output screen it can be proved that we can improve the predictability by changing the attributes and increase the effectiveness of the algorithm. No further test is done now but the user can keep doing these iteration till the accuracy is maximum and no changes are observed in further iteration.

4.1.4 Comparison of all Iterations

From all 3 iterations test we have done in the above, let's put the output result in table format to compare the various values. Table 1 shows this comparison result. The same result is plotted in graph output shown in Fig 9.

Table 2: Comparison of results for exp 1

Iteration. No.	Precision	Recall - (TPR)	False Positive Rate (FPR)	True Negative Rate (TNR)	Accuracy
1	1.0000	0.1111	0.0000	1.0000	0.7333
2	1.0000	0.6667	0.0000	1.0000	0.9000
3	1.0000	0.7778	0.0000	1.0000	0.9333

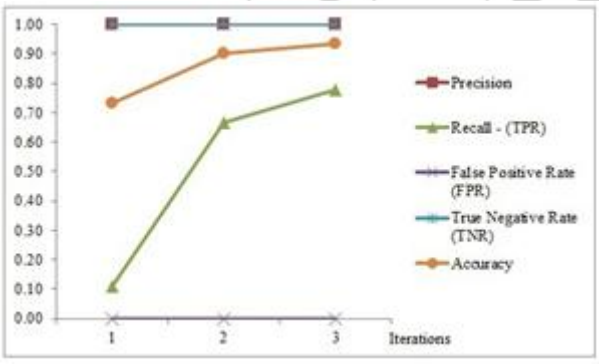


Figure 9: Comparison of results for exp 1

4.2 Performance Results Exp. 2

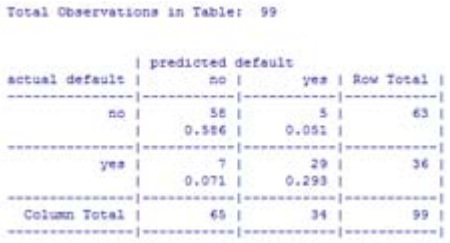
This experiment is to compare C50 algorithm with the r_dt10 algorithm created. The Data set consists of multiple attributes. This data set is of training set as well as testing set so that we can accurately predict if there is change in the result generated. The data is stored in file for training set "emp_det.csv" and for testing set "emp_det_test.csv".

4.2.1 Test results

The actual testing script created for this experiment is "R_dt10_testscipt_exp2.R". This script is executed in the R tool and results are recorded as below.

Step 1: Load the dataset from file both training and testing data set . training set is 500 records and testing set is 99 records.

Step 2 : Run C50 classifier with both training and testing data and check the confusion matrix. The output result of confusion matrix is shown below in Fig 10.

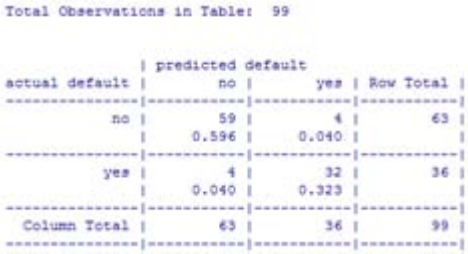


Total Observations in Table: 99

actual default	predicted default: no	yes	Row Total
no	58	5	63
	0.586	0.051	
yes	7	29	36
	0.071	0.293	
Column Total	65	34	99

Figure 10: Confusion matrix for test data set, exp. 2, C50 classifier

Step 3 : Run R_dt10 classifier with both training and testing data and check the confusion matrix. The output result of confusion matrix is shown below in Fig 11.



Total Observations in Table: 99

actual default	predicted default: no	yes	Row Total
no	59	4	63
	0.596	0.040	
yes	4	32	36
	0.040	0.323	
Column Total	63	36	99

Figure 11: Confusion matrix for test data set, exp. 2, r_dt10 classifier

4.2.2 Comparison of both classifier

Both the classifier output values let's put them in table format and check /compare the various results. Table 3 shows the results of both the classifier and Fig 12 shows graph plot of the same table. With the result it is evident that r_dt10 predicts slightly better that C50 in this condition.

Table 3: Comparison of results for exp 2

Classifier	Precision	Recall - (TPR)	False Positive Rate (FPR)	True Negative Rate (TNR)	Accuracy
C50	0.8529	0.8056	0.0794	0.9206	0.8788
r_dt10	0.8889	0.8889	0.0635	0.9365	0.9192

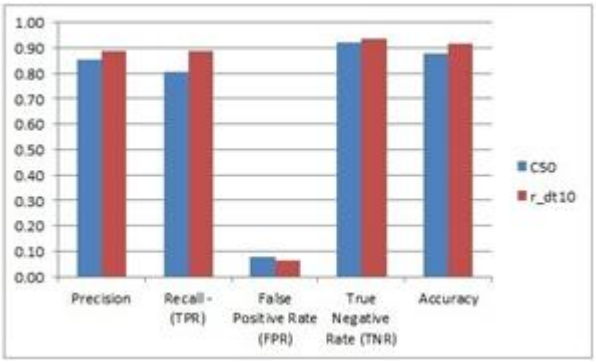


Figure 12: Comparison of results for exp 1

5. Conclusion

In conclusion, the solution/tool I created is novel solution. When combined with HR/ Manager domain expertise it improves. The tool not only predict attrition, it also gives the factor/reason for attrition. This helps the HR / Manager to take cost effective and faster decision for dissatisfied employee.

This research presented uses decision tree algorithm ID3 to create a prediction tool r_dt10, when compared with the existing tool C50 it shows accuracy of 91% as of 87% for C50. This clearly shows improvement. The other aspect is also compared where the tool is run in multiple iteration to test itself with each iteration changing the training data with the help of output generated by previous iteration. The results shows improvement in each iteration. This proves that model/algorithm r_dt10 improves with valid data set if performed in multiple iteration.

References

[1] Reshad Abdullah, Prof. Sachin Bojewar, "Attrition Prediction- Need of the Hour for Companies" | IJSER - International Journal of Scientific & Engineering Research, Volume 7, Issue 2, February-2016.
[2] Employee Attrition: A Study Of It Organizations | SetarehShokatSadry | GMP Review, 2015; V18(1).
[3] Machine Learning with R | Brett Lantz | First published , October 2013 | PACKT publishing
[4] Induction of Decision Trees. | Quinlan, J. R. 1986| First published , 1986 | Kluwer Academic Publishers, Boston