An Evaluation Model for Human Resource Management in Organizations using classification and association algorithms

Manoharan, Akshay Karthick, MS in Computer Engineering – 431030840
Soosai Michael, Joy Shalom, MS in Computer Science – 699737418
Parmar, Mansi, MS in Computer Engineering – 729052042
Basavani Shivashankara, Sachin, MS in Computer Engineering – 267871645
Vishwanathan, Vishnu Prasad, MS in Computer Engineering – 793782749

***Abstract*—** The development of an evaluation model for Human Resource Management is highly necessary because employees of an organization play a vital role in determining the organization's performance and picture the complete profit scale. Every strategy of an organization is directly or indirectly associated with the employee's talent. Hence it is important to retain the employees who serve to increase the organization's profit. The proposed model will help in evaluating how the attrition rate of the employees is increasing at a greater value by cutting the edge between the prevailing and departed employees based on the correlation identified between the given attributes in the Human Resource data. Data mining is widely used to extract the best knowledge out of available data. This paper is an attempt to provide an evaluation framework for Human Resource data comprising data mining techniques such as classification and association.

1. Introduction

Many people think that the main task of Human Resources is to source and employ new talent. However, the truth is that most HR activities revolve around retaining existing talent. Employee retention is a fine balancing act between company culture, remuneration and incentives. The HR department needs to provide each employee with the right combination of all three to satisfy the employee without compromising company interests in the process. Hence, the problem is to discover why the employees leave the organization and how to reduce the retention from the dataset. *Refer Dataset section in Appendix.* The chosen HR data set consists of the various attributes (Refer Table 2.a). Each attribute has its own significance in determining whether the employee will stay or leave the organization. Our work includes usage of classification and association algorithms in order to find which attributes influence the "Left" attribute more in the dataset. Classification models are built using Decision trees like 'C5.0, CART, Random forest' and 'Naïve Bayes classifier'. They serve as the predictive model in the future to determine when the employee might leave the organization. The accuracy obtained from the used decision trees are far best compared to the Naïve Bayes classifier. Association algorithm 'Apriori' is used to decide which employees leave the organization based on the high support, confidence and lift values. The most interesting rules are secluded to study in depth and evaluate based on the depending attributes.

2. Previous Work and Background Information

There are quite a lot of previous papers published on Human Analytics. Most of these papers focuses more on the organizational and intuitive features of retaining the employees and building a predictive model over it. The paper by Sujeet N. Mishra, Dev Raghvendra Lama, and Yogesh Pal is one of the best paper on Human Analytics that talks about the predictive analysis needed in an organization. Refer "http://www.ijstr.org/final-print/may2016/Human-Resource-Predictive-Analytics-hrpa-For-Hr-Management-In-Organizations.pdf". Paper published by Gary K. L. Tam, Vivek Kothari, and Min Chen is another very interesting source that clearly helps us to understand that classification algorithms perform better Human resource visualizations and yields a better result. Refer http://ieeexplore.ieee.org/document/7539314/ for more details on it. There must be a predefined intuitive analysis made on the dataset to manually understand how the human resource system works. To achieve this, the following excel workbook has been created which draws an idea of the attributes that influences employees who left the company. Refer the following workbook

HR_comma_sep_Intuitive Analysis.xlsx

for more details-

3.Data Pre-Processing and Feature Engineering

Data pre-processing is a crucial step to carefully screen out attributes which does not yield results and add attributes which will help us in making good judgement. The following are the data pre-processing and feature engineering steps done over the HR data to yield logical and well-reasoned results.

- *ImproperEvaluation* => "For all employees who have a good evaluation (0.87, as it's the 3rd quartile value), and the salary grade as low" => Assigned Boolean value as 'yes' for the new column.
- *Overrated* => "For all employees who have the satisfaction level, last evaluation and the number of projects less than the median value are technically low performing employees. Albeit if they are promoted, then they are considered as overrated employees. Hence this column holds a Boolean value as 'yes' for those who satisfy the criteria.
- *Average daily hours* => Column which helps us evaluate how much everyone has worked on average daily basis
- The column 'average_daily_hours' should be rounded off to two decimal places as the process of cleaning the data set.

The new variables can be used for data exploration and in other places whether categorical values are mandatory and needed for good analysis. *Refer code 4.a*

4. Data Exploration

Data exploration is an important task when working with complex data set. It is the process of summarizing the major characteristics of the attributes present. Data visualization acts as an important technique in laying out important attributes and helps in exploring the data in an effective manner. Following are few explorations which are to be considered and gives an initial insight about the HR data set.

- Only a small number of employees have a high salary. *Refer code 4.b and Fig.4.a.*
- Management Department, has the least attrition rate as it has a higher proportion of highly paid employees.
- There is almost no attrition in High Salary Paid Employees.
- Very few people have been promoted. *Refer code 4.c and Fig.4.b.*
- We can see that the employees with salary level low have not been promoted much in last 5 years.
- We cannot conclude people with low salary are under performing employees.
- Even employees who are under performing are over rated. *Refer code 4.d and Fig.4. c.*
- There are over rated employees who are under performing but promoted. *Refer code 4.e.*
- We can note that as the experience of the employee increases, they tend to leave the company. *Refer code 4.f and Fig.4.d.*
- Experienced employees are mostly not promoted. *Refer code 4.g and Fig.4.e.*
- Employees with number of projects as 2 have mostly left the company. Secondly employees with 6 and 7 projects have left the company.
- We can voice saying employees with few projects and the ones burdened with projects have left the company. *Refer code 4.h and Fig.4. f.*
- Employees with higher satisfaction level are evaluated incorrectly. *Refer code 4.i and Fig.4.g*
- The average satisfaction of the employees who left is lower than who haven't left. *Refer code 4.j and Fig.4.h.*
- Employees leaving the company have spent more years with salary levels low and medium. *Refer code 4.k, 4.l and Fig.4.i and Fig.4. j.*

- We can also identify that ‘Sales’ department has majority of the employees falling under salary level ‘low and medium’.
- Secondly, we can identify ‘support’ and ‘technical’ departments have salary level low and medium.
- Previously from ‘psales_left’ we can say that- sales, support and technical are departments where the salary level is not high and employees leaving the company is also more.

Observation from all the above interactions:

- Employees from low and medium salary grade have left the company.
- Employees with satisfaction level lower than the median value have left the company.
- Employees with high average daily hours have left the company.
- Employees with experience greater than 4, still no promotion have left the company.
- Employees who fall under the Improper evaluation category have left the company.
- Majority of the employees left fall under sales, support and technical departments.

5. *Data Mining*

5.1 Association Analysis

Association Analysis is a useful methodology for discovering interesting relationships hidden in large data sets. The uncovered relationships can be represented in the form of **Association Rules** or **Sets of Frequent Items**.

5.1.1 Apriori Algorithm for Association Analysis

The Apriori principle is: *If an item set is frequent, then all its subsets must also be frequent*. Based on this principle, a strategy of trimming the exponential search space based on the support measure is known as **Support-based pruning**. We use the **apriori()** function in R to mine **frequent item sets** or **association rules** from the data. The Apriori algorithm uses **level-wise search** for frequent item sets.

Usage:

apriori(data, parameter = NULL, appearance = NULL, control = NULL)

- data: Data in the form of transactions
- parameter: List, provides info as to the rules to mine with (viz. minimum support, minimum confidence, maximum length, minimum length etc.)
- appearance: List, which restricts items appearance
- control: List, controls the algorithmic performance

Apriori function generates rules of the form {X} => {Y}, where X and Y are attributes from the transactions of the data. {X} is called **LHS (Antecedent)** and {Y} is called **RHS (Consequent)**.

5.1.2 Implementation

After further processing the pre-processed data (*Refer table 2.a* for the pre-processing done for association analysis) we generate association rules on this data using **minsup = 0.1** and **minconf = 0.5**. *Refer code 5.1.a.* and *Fig.5.1.a*.

This generates several rules of varying support, confidence and lift on various attributes.
Table 5.1.a shows some of the rules generated with its corresponding support, confidence and lift values.

As, the goal of our project is to analyse on what factors the attrition rate is dependent on, we focus our analysis on the attribute “left” as the RHS.

From Exploratory Analysis, we can see that the attributes satisfaction_level, last_evaluation, average_daily_hours, number_project, and salary strongly influence the left attribute. That is, the employee’s satisfaction level, number of projects that he/she is involved in, salary, duration of working hours, his/her performance evaluation is a strong indicator of the attrition rate (attribute left).
Along with the above attributes we also include attributes “ImproperEvaluation” and “Over_Rated”. These attributes were included into the dataset during data exploration. Attribute “ImproperEvaluation” is YES if salary is low even when last evaluation is high and attribute “Over_Rated” is YES if satisfaction level is average, last evaluation is average, number of projects is average and yet if the employee was promoted in the last five years. The presence of these attributes helps us confirm that the observed rules are reliable.

Splitting our analysis into two parts:
Part A. Analysis of the characteristics of employees who are still working {left = 0}.
Part B. Analysis of the characteristics of employees who have left the company {left = 1}.

5.1.2.1 Part A

Analysis of the characteristics of employees who are still working {left = 0}.
Here, we maintain RHS as {left = 0} and we include all the categories of different attributes so that we can compare how each category is dependent on other attributes. *Refer code 5.1.b* and *Fig. 5.1.b* for the scatter plot of the rule.

Refer code 5.1.c, *Fig.5.1.c* and *Fig.5.1.d* for the distribution plot and the parallel co-ordinate plot for the rule.

For our analysis, we focus only on the rules with high confidence and support with high lift values. That is, we focus on points (rules) in the top portion of the scatter plot.

From the plots of the rules, we can summarize the following observations for {left = 0}:

- The attribute "last_evaluation=high" is strongly associated with the attributes: "satisfaction_level=average", "number_project=low".
- The attribute "number_project=average" is strongly associated with the attributes: "average_daily_hours=low", "satisfaction_level=high".
- The attribute "average_daily_hours=average" is strongly associated with the attributes: "last_evaluation=high", "satisfaction_level=average".
- The attribute "salary=medium" is strongly associated with the attributes: "satisfaction_level=high", "number_project=average", "last_evaluation=average"

From the above observations, we can say that the characteristics of an employee who will continue to work in the company are:

- An employee with satisfaction level of "average" or "high"
- An employee working on "average" number of projects
- An employee whose salary is in the "medium" range
- An employee whose average daily hours is either "low" or "average"
- An employee whose last evaluation was either "average" or "high"

Apart from visualizing the rules through the plots, we can also analyse the rules manually. Therefore, we convert the generated rules into data frames and save it as a .csv file and filter out the rules with low support, confidence and lift values. *Refer code 5.1.d* and *Table 5.1.b* (*Table 5.1.b* shows some of the rules with high support, confidence and lift values for RHS of {left=0}).

The observations made by manually analysing the rules are similar to the observations made through distribution plots. Also, the presence of the attributes "ImproperEvaluation=NO" and "Over_Rated=NO" in the rules further confirms that the observations made above are reliable and not spurious.

5.1.2.2 Part B

Analysis of the characteristics of employees who have left the company {left = 1}
Here, we maintain RHS as {left = 1} and we include all the categories of different attributes so that we can compare how each category is dependent on other attributes. *Refer code 5.1.e* and *Fig. 5.1.e* for the scatter plot of the rule.

Refer code 5.1.f, *Fig.5.1.f* and *Fig.5.1.g* for the distribution plot and the parallel co-ordinate plot for the rule.

From the plots, we can summarize that an employee will leave the company if his/her satisfaction level is "average", if the salary is "low" and has less number of projects.

Refer code 5.1.g and *Table 5.1.c* (*Table 5.1.c* shows some of the rules with high support, confidence and lift values for RHS of {left=1}).

By manually analysing the rules we can observe that, the attributes "last_evaluation=low", "number_project=low", "time_spend_company=low", "salary=low", "average_daily_hours=low" and "satisfaction_level=average" are strong characteristics of an employee who has left the company. Also, the presence of the attributes "ImproperEvaluation=NO" and "Over_Rated=NO" in the rules further confirms that the observations made are reliable.

So, we can summarize that the characteristics of an employee who has left the company are:

- An employee with satisfaction level of "low" or "average"
- An employee whose last evaluation was "low"
- An employee working on "less" number of projects
- An employee with "low" salary
- An employee whose average daily hours is "low"

Conclusion from Association rule mining

From the observations, we can see that there exist two categories of employees who are still working in the company or who might continue working.

- Employees who a. Work on average number of projects. b. Work a fairly average number of hours and are not stressed out. c. Are compensated well for the work they have put in. d. Are evaluated well for their efforts. e. Are well satisfied with their work and for the above reasons.
- Employees who a. Work on less number of projects. b. Work fewer number of hours. c. But still are compensated well. d. Are evaluated well. e. Are well satisfied with the work conditions

And for the employees who have left the company or who might leave.

- Employees who are kind of disengaged with the company for the following reasons. a. Works on less number of projects or less challenging projects or he/she might be less competitive. b. Therefore, works less number of hours daily. c. Consequently, the employee has a low evaluation. d. Is compensated less. e. For the above reasons, the employee has an average satisfaction level or more on the lower side.

5.2 Classification

The Data Classification process includes two steps:

- Building the Classifier or Model
- Using Classifier for Classification

Building the Classifier or Model

This step is the learning step or the learning phase. The classifier is built from the training set made up of database tuples and their associated class labels.

Using Classifier for Classification

In this step, the classifier is used for classification. Here the test data is used to estimate the accuracy of classification rules.

5.2.1 Decision Tree

The basic learning approach of decision tree is greedy algorithm, which use the recursive top-down approach of decision tree structure. A decision tree is a structure that includes a root node, branches, and leaf nodes. Each internal node denotes a test on an attribute, each branch denotes the outcome of a test, and each leaf node holds a class label.

5.2.1.1 C5.0 Decision Tree

The model can take the form of a full decision tree or a collection of rules (or boosted versions of either). When using the formula method, factors and other classes are preserved (i.e. dummy variables are not automatically created). This model handles non-numeric data of some types (such as character, factor and ordered data). *Refer Table 5.2.a* to understand the default syntax and the values passed into each argument.

5.2.1.1.1 C5.0 Implementation

The first step in the process of implementing decision tree is that we should divide the data set into testing and training data.

Refer code 5.2.a, to follow the R code for splitting the data set. We have taken 5000 records at random into testing data set and the remaining has been assigned to the train data set. The accuracy of the model w.r.t to training data is 97.88%. *Refer code 5.2.b*, which presents the code for the model built and its accuracy rate. The trained model must be validated using the testing data. The confusion matrix statistics over the test data against the attribute 'left' gives an accuracy of 97.64% which is surprisingly nearly perfect accuracy rate. *Refer code 5.2.c* for details on confusion matrix.

To understand the accuracy in a better way, plotting of AUROC curve is necessary. *Refer Fig.5.2.b* where the AUROC curve gives an accuracy of 95.5%. This type of graph is called a Receiver Operating Characteristic curve (or ROC curve.) It is a plot of the true positive rate against the false positive rate for the different possible cut points of a diagnostic test. The closer the curve follows the left-hand border and then the top border of the ROC space, the more accurate the test.

The closer the curve comes to the 45-degree diagonal of the ROC space, the less accurate the test.

Observation:

Though C5.0 decision tree is used to build an efficient model in classifying whether the employee has left the company or not, to specifically plot out the attributes due to which the employees are leaving, we must remove attributes one by one and build the model again, so that there is only a slight change in the AUC and other parameters. This is a tedious modelling task. Hence proceeding on to CART decision tree which is the most widely used decision tree.

5.2.1.2 CART Decision Tree

Recursive partitioning is a fundamental tool in data mining. It helps us explore the structure of a set of data, while developing easy to visualize decision rules for predicting a categorical (classification tree).

CART Modelling via rpart:

- Grow the tree

rpart (*formula*, **data=**, **method=, control=)** where the values passed on to each argument is explained in *Table.5.2. a*

5.2.1.2.1 CART Implementation:

Model 1 in CART:

Attempting to classify without using any specific attribute list. *Refer code 5.2.d* which presents the code for the model built. The cross-validation estimates are shown in *Refer code 5.2.e t*o understand the model much better.

Information Gain playing an important role:

The information gain is based on the decrease in entropy after a dataset is split on an attribute. Constructing a decision tree is all about finding attribute that returns the highest information gain (i.e., the most homogeneous branches). From *code 5.2.e*, we can conclude that the 4 important parameters chosen for splitting are "***average monthly hours, last evaluation, number of project, satisfaction level and time spent in the company***". During the construction of decision tree without specific attributes given, it picks up the attributes with increasing information gain. Hence, we can conclude that these attributes have the highest information gain.

The number of splits taken place are 9 and at each level we can find that the error rates are reduced. By plotting the obtained decision tree, we can identify that Satisfaction level has the highest information gain. Hence it has been taken as the root node. *Refer Fig.5.2.d* for model 1 decision tree

Observation:

To conclude that these attributes play a key role for an employee to leave the company, we need to obtain the accuracy rate. *Code 5.2.f* scenes the accuracy rate for both testing and training data with 96.77% and 97.02% respectively.

- Satisfaction level appears to be the most important piece. If you're above 0.46 you're much more likely to stay (which is what we observed from *Fig.5.2.d*).
- If you have low satisfaction, the number of projects becomes import. If you're on more projects you're more likely to remain. If you're on fewer projects – perhaps employees leave.
- If you're happy, have been at the company for more than 4.5 years, and score over 81% on your last evaluation, you're very likely to leave. And, it appears as if the "decider" is monthly hours over 216.

Model 2 in CART:

Repeat the same steps as per Model 1 using the following attributes:

"satisfaction_level+number_project+time_spend_company+promotion_last_5years"

The cross-validation estimates are shown in *code 5.2.g*. Among the given attributes, the chosen attributes are satisfaction_level, number_project, time_spend_company. The accuracy rate remains good and closer to perfection. Refer *code 5.2.h* which scenes the accuracy rate for both testing and training data with 96.90% and 96.88% respectively. *Refer Fig.5.2.e* for model 2 decision tree which show cases a similar scene as CART model 1.

Observation:
We identify the same scenario – if you're good and overworked you leave; if you're unhappy you tend to leave, especially if you're not getting enough work.

Model 3 in CART:
Repeat the same steps as per Model 1 using the following attributes:
"last evaluation, average daily hours, sales and salary"

The cross-validation estimates are shown in *code 5.2.j* Among the given attributes, the chosen attributes are last evaluation and average_daily_hours. The accuracy rate drops down to a greater extent. Refer *code 5.2.i* which scenes the accuracy rate for both testing and training data with 79.54% and 80.01% respectively. *Refer Fig.5.2.f* for model 3 decision tree which show cases a totally different scene from data exploration.

From Data exploration, we identified that sales and salary play a key role in an employee staying and leaving the company. Unfortunately, in decision tree, sales and salary are not considered for splitting. The reason can be figured out based on complexity parameter. The **complexity parameter (cp)** is used to control the size of the decision tree and to select the optimal tree size. If the cost of adding another variable to the decision tree from the current node is above the value of cp, then tree building does not continue. We could also say that tree construction does not continue unless it would decrease the overall lack of fit by a factor of cp.

Observation: We can clearly connect the dots saying employees leaving have last evaluation less than 0.57, average daily hours less than 7.3. Similarly, employees with last evaluation value greater than 0.76 as well leave the company. Since the accuracy of the model is not good, we cannot trust this model completely.

5.2.1.3 Random Forest

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of overfitting to their training set.

Model Random Forest
The pre-processed data is taken and the train set and test set are created. The random forest tree is used now on the train set as shown in *code 5.2.k.* The test data is predicted using the decision tree developed and the confusion matrix is created as shown in *code 5.2.l.* The accuracy is 0.9913. Now the given dataset is modified based on the understanding from the Data Pre-processing as per *code 5.2.n* where **salh <- HR$salary == "high"**. The modified data is again used to create another model and prediction in the same way as above. The results are in *code 5.2.m*

Observation
The second model with the modified data has an accuracy of 0.9926. We can see that the accuracy has increased after changing the data using the pre-processing techniques. The following conclusions are made about the factors why employees leave the company:1. Employees from low and medium salary grade have left the company 2. Employees with satisfaction level lower than the median value have left the company 3. Employees with experience greater than 4, still no promotion 4. Majority of the employees left fall under sales, support and technical departments
The ROC curve plots for both the given data and the modified data can be found in *Fig.5.2.g*

5.2.2 Naïve Bayes Classifier

In machine learning, naive Bayes classifiers are a family of simple probabilistic classifiers based on applying Bayes' theorem with strong (naive) independence assumptions between the features. Naive Bayes is a simple technique for constructing classifiers: models that assign class labels to problem instances, represented as vectors of feature values, where the class labels are drawn from some finite set. It is not a single algorithm for training such classifiers, but a family of algorithms based on a common principle: all naive Bayes classifiers assume that the value of a particular feature is independent of the value of any other feature, given the class variable. An advantage of naive Bayes is that it only requires a small number of training data to estimate the parameters necessary for classification.

Working Of Naïve Bayes

Naïve Bayes classifier mainly works on probabilities and prediction works based on probabilities. Here we train the data. The dataset which is pre-processed earlier is taken here and read into hra. Then the first column which is not needed. Then we split the dataset into training and test dataset by the ratio 80:20. This is the standard specification of splitting the dataset. Then we train the dataset with the attribute of number of people who left the company is taken as the class label. Now every attribute is weighed against this attribute and this is displayed in table. For example we analyze the number of people who left the company according to attributes such as salary, sales, Number of projects handled, Promotion for last 5 years and satisfaction level. These things are also plotted using ggplot function. We give out conclusions based on these findings. Then we predict the training dataset and print the confusion matrix which comes under the caret library. We now predict the results of the test dataset using the data that are trained with the naïve bayes classifier. Now both the accuracies of test dataset and training dataset is plotted under the ROC curve. The codes performing the above operations are given in the appendix . *Refer Code 5.3.a*

Observation:

Here first we make certain observations based on the dataset and probabilities we get from Naïve Bayes Classification: (*Refer Fig 5.3.a to 5.3.e* in Appendix)

a.)The Number of People receiving low and medium salary have a higher probability of leaving the company. *Refer Fig 5.3.a*

b.) People working in HR and sales have more probability of leaving the company. Refer *Refer Fig5.3.b*

c.)People with satisfaction level between 0.35 to 0.5 have a high probability of leaving the company. *Refer Fig5.3.c*

d.)People working on more number of projects or very less number of projects have a high probability of leaving the company. *Refer Fig 5.3.d*

e.)People getting less promotion over the last 5 years have a high probability of leaving the company. *Refer Fig 5.3.e*.

Also along with the above observations. We assess the reliability of the Naïve-Bayes Classifier.

From *Fig 5.3.f, Fig 5.3.g, Fig 5.3.h, Fig 5.3.i*, which gives the confusion matrix and ROC graph of training and test dataset, we find that training dataset has **80.41%** accuracy and test dataset has **80.17%** which are almost equal to each other. Thus proving Naïve-Bayes classifier is an efficient algorithm for classification.

6. *Discussion and Analysis*

Based on all the above work, and by reviewing the results, we can certainly say that Decision tree works as a better predictive model than Naïve-Bayes classifier. Since cross validation is not explicitly performed, the accuracy for the defined model is executed multiple times to firmly say that the model works best. Among all , the sole star performer is the decision tree built using random forest with the best accuracy of 99%, while the Naïve-Bayes classifier has only 80% accuracy. This is verified by comparing the ROC graphs of all the classifiers. The reason behind this is Naïve-Bayes is based on probabilities while Random Forest filters the attributes and gives out the prediction. Using random forest, we can efficiently predict that variables satisfaction level, last evaluation, number of projects play a key role.

7. Future work

Human resource analytics is a problem faced in every single organization. Hence an application involving all possible employee and organization view point attributes should be included as a part of input data to predict when an employee might leave the company. Predictive model of pre-judging the retention rate will be our future work and it can be used as a major application. Future work is to build a model which will first predict the employees who might leave the company. Additionally, the model will provide the needed value change in each attribute which might make employees stay back. Hence an organization can use this application to predict and make corrective measures before an employee leaves the organization.

8. Conclusion

As a concluding factor, based on Employee's Aspect: Employees with low satisfaction level leave the company. Employee working on more or very less number of projects, with no promotion even if he is experienced and low salary are the factors which influence the satisfaction rate to be low. Based on Organization's Aspect: Employees with Low evaluation rate have left the company. Employees working on more or very less number of projects, with less or high average working hours are important factors which influence the evaluation rate to be low. We provided an in-depth study of various algorithms to come to the above conclusion. In our in-depth analysis, we first collected various empirical evidence that helps to visualize a human resource analytics system. We quantitatively estimated the theory using R language to show that the estimated accuracy of the models built are reasonably good and make interesting predictions.

9. References

[1] Gary K. L. Tam, Vivek Kothari, and Min Chen, Member, IEEE. An Analysis of Machine- and Human-Analytics in Classification.

[2] Sujeet N. Mishra, Dev Raghvendra Lama, Yogesh Pal, Human Resource Predictive Analytics (HRPA) For HR Management In Organizations.

[3] https://www.kaggle.com/ludobenistant/hr-analytics.

[4] Introduction to Data Mining. Pang-Ning Tan, Michael Steinbach, Vipin Kumar.

[5] Association Analysis from Zhao_R_and_data_mining.