Prediction of Employee Turnover in Organizations using Machine Learning Algorithms

A case for Extreme Gradient Boosting

Rohit Punnoose, PhD candidate
XLRI – Xavier School of Management
Jamshedpur, India

Pankaj Ajit
BITS Pilani
Goa, India

***Abstract*—Employee turnover has been identified as a key issue for organizations because of its adverse impact on work place productivity and long term growth strategies. To solve this problem, organizations use machine learning techniques to predict employee turnover. Accurate predictions enable organizations to take action for retention or succession planning of employees. However, the data for this modeling problem comes from HR Information Systems (HRIS); these are typically under-funded compared to the Information Systems of other domains in the organization which are directly related to its priorities. This leads to the prevalence of noise in the data that renders predictive models prone to over-fitting and hence inaccurate. This is the key challenge that is the focus of this paper, and one that has not been addressed historically. The novel contribution of this paper is to explore the application of Extreme Gradient Boosting (XGBoost) technique which is more robust because of its regularization formulation. Data from the HRIS of a global retailer is used to compare XGBoost against six historically used supervised classifiers and demonstrate its significantly higher accuracy for predicting employee turnover.**

Keywords—turnover prediction; machine learning; extreme gradient boosting; supervised classification; regularization

I. Introduction

The problem of employee turnover has shot to prominence in organizations because of its negative impacts on issues ranging from work place morale and productivity, to disruptions in project continuity and to long term growth strategies. One way organizations deal with this problem is by predicting the risk of attrition of employees using machine learning techniques thus giving organizations leaders and Human Resources (HR) the foresight to take pro-active action for retention or plan for succession. However, the machine learning techniques historically used to solve this problem fail to account for the noise in the data in most HR Information Systems (HRIS). Most organizations have not prioritized investments in efficient HRIS solutions that would capture an employee's data during his/her tenure. One of the major factors is the limited understanding of benefits and cost. It is still difficult to measure the return of investment in HRIS [1]. This leads to noise in the data, which in turn attenuates the generalization capability of these algorithms.

In this paper, the problem of employee turnover and the key machine learning algorithms that have been used to solve it are discussed. The novel contribution of this paper is to explore the application of extreme gradient boosting (XGBoost) as an improvement on these traditional algorithms, specifically in its ability to generalize on noise-ridden data which is prevalent in this domain. This is done by using data from the HRIS of a global retailer and treating the attrition problem as a classification task and modeling it using supervised techniques. The conclusion is reached by contrasting the superior accuracy of the XGBoost classifier against other techniques and explaining the reason for its superior performance.

This paper is structured as follows. Section II gives a brief overview of the employee turnover problem, the importance of solving it, and the historical work done in terms of application of machine learning techniques to solve this problem. Section III explores the 7 different supervised techniques, including XGBoost, that this paper compares. Section IV outlines the experimental design in terms of the characteristics of the dataset, pre-processing, cross-validation, and the choice of metrics for accuracy comparison. Section V showcases the results of the study and its subsequent discussion. Section VI concludes the paper by recommending the XGBoost classifier for predicting turnover.

II. Literature Review on Employee Turnover

Employee turnover can be interpreted as a leak or departure of intellectual capital from the employing organization [2]. Most of the literature around turnover categorizes turnover as either voluntary or involuntary.

This analysis is centered on voluntary turnover. In a meta-analytic review of voluntary turnover studies [3], it was found that the strongest predictors for voluntary turnover were age, tenure, pay, overall job satisfaction, and employee's perceptions of fairness. Other similar research findings suggested that personal or demographic variables, specifically age, gender, ethnicity, education, and marital status, were important factors in the prediction of voluntary employee turnover [4], [5], [6], [7], [8]. Other characteristics that studies focused on are salary, working conditions, job satisfaction, supervision, advancement, recognition, growth potential, burnout etc. [9], [10], [11], [12].

High turnover has several detrimental effects on an organization. It is difficult to replace employees who have niche skill sets or are business domain experts. It affects ongoing work and productivity of existing employees. Acquiring new employees as replacement has its own costs like hiring costs, training costs etc. Also, new employees will have their learning curves towards arriving at similar levels of

technical or business expertise as a seasoned internal employee.

Organizations tackle this problem by applying machine learning techniques to predict turnover thus giving them the vision to take necessary action. Table 1 below briefly documents the literature review findings. Subsequent sections of the paper will highlight the inadequacy of the classifiers recommended here in handling noise of the scale in HRIS.

TABLE I. RELATED WORK ON TURNOVER PREDICTION

Research Authors	**Problem studied**	**Data Mining Techniques studied**	**Recommend**
Jantan, Hamdan and Othman [13]	Data Mining techniques for performance prediction of employees	C4.5 decision tree, Random Forest, Multilayer Perceptron(MLP) and Radial Basic Function Network	C4.5 decision tree
Nagadevara, Srinivasan and Valk [14]	Relationship of withdrawal behaviors like lateness and absenteeism, job content, tenure and demographics on employee turnover	Artificial neural networks, logistic regression, classification and regression trees (CART), classification trees (C5.0), and discriminant analysis)	Classification and regression trees (CART)
Hong, Wei and Chen [15]	Feasibility of applying the *Logit* and *Probit* models to employee voluntary Turnover predictions.	Logistic regression model (logit), probability regression model (probit)	Logistic regression model (logit)
Marjorie Laura Kane-Sellers [16]	To explore various personal, as well as work variables impacting employee voluntary turnover	Binomial logit regression	Binomial logit regression
Alao and Adeyemo [17]	Analyzing employee attrition using multiple decision tree algorithms	C4.5, C5, REPTree, CART	C5 decision tree
Saradhi and Palshikar [18]	To compare data mining techniques for predicting employee churn	Naïve Bayes, Support Vector Machines, Logistic Regression, Decision Trees and Random Forests	Support Vector Machines

III. METHODS

In machine learning, classification has two distinct meanings. We may be given a set of observations with the aim of establishing the existence of classes or clusters in the data. Or we may know for certain that there are a certain number of classes, and the aim is to establish a rule(s) whereby we can classify a new observation into one of the existing classes. The former type is known as Unsupervised Learning, the latter as Supervised Learning [19]. This paper deals with classification as supervised learning, because the data contains 2 classes – active and terminated. This section details the theory behind various classification algorithms compared.

A. Logistic Regression

Logistic regression/ maximum entropy classifier is one of the basic linear models for classification. Logistic regression is a specific category of regression best used to predict for binary or categorical dependent variables. It's often used with regularization in the form of penalties based on L1-norm or L2-norm to avoid over-fitting. An L2-regularized logistic regression for this paper. This technique obtains the posterior probabilities by assuming a model for the same and estimates the parameters involved in the assumed model. The form of the model is given below in (1):

$$p(\text{churn}|w) = \frac{1}{1+e^{-\left[w0+\sum_{i=1}^{N} wiXi\right]}} \quad (1)$$

The parameters w, are estimated using maximum likelihood estimation technique [20]

B. Naïve Bayesian

Naïve Bayes is a popular classification technique that has attracted attention for its simplicity and performance [21]. Naïve Bayes performs classification based on probabilities arrived, with a base assumption that all variables are conditionally independent of each other. To estimate the parameters (means and variances of the variables) necessary for classification, the classifier requires only a small amount of training data. It also handles real and discrete data [22].

The underlying logic to using the Bayes' rule for machine learning is as follows: To train a target function fn: $X \rightarrow Y$, which is the same as, $P(Y|X)$, we use the training data to learn estimates of $P(X|Y)$ and $P(Y)$. Using these estimated probability distributions and Bayes' rule new X samples could then be classified [21].

C. Random Forest

Random Forest algorithm is a popular tree based ensemble learning technique. The type of 'ensembling' used here is bagging. In bagging, successive trees do not depend on earlier trees — each is independently constructed using a different bootstrap sample of the data set. In the end, a simple majority vote is taken for prediction. Random forests are different from standard trees in that for the latter each node is split using the best split among all variables. In a random forest, each node is split using the best among a subset of predictors randomly chosen at that node [23]. This additional layer of randomness makes it robust against over-fitting [24].

D. K-Nearest Neighbor (KNN)

The intuition behind Nearest Neighbor Classification is to classify data points based on the class of their nearest neighbors. It is often useful to take more than one neighbor into account so the technique is more commonly referred to as k-Nearest Neighbor (k-NN) Classification [25].

The 2 stages for classification using KNN involve determining neighboring data points and then deciding the class based on the classes of these neighbors. The neighbors can be determined using distance measures like Euclidean

distance (used in this paper), Manhattan distance etc. The class can be decided on majority vote of neighbors or weighting inversely proportional to the distance. The data was scaled to [0, 1] range before building the KNN based model.

E. Linear Discriminant Analysis (LDA)

Discriminant analysis involves creating one or more discriminant functions so as to maximize the variance between the categories relative to the variance with the categories [14]. Linear Discriminant Analysis is explained as deriving a variate or z-score, which is a linear combination of two or more independent variables that will discriminate best between two (or more) different categories or groups.

The z-scores calculated using the discriminant functions is then used to estimate the probabilities that a particular member or observation belongs to a class. An important point to note with LDA is that the features used should be continuous or metric in nature.

F. Support Vector Machine (SVM)

An SVM is a supervised learning algorithm that implements the principles of statistical learning theory [26] and can solve linear as well as nonlinear binary classification problems. A support vector machine constructs a hyper-plane or set of hyper-planes in higher dimensional space for achieving class separation. The intuition here is that a good separation is achieved by the hyper-plane that has the largest distance to the nearest training data points of any class- the larger the margin the lower the generalization error of the classifier. For this reason, it is also referred to as maximum margin classifier. The data was scaled to [0, 1] range before building this model.

G. Extreme Gradient Boosting (XGBoost)

Boosting refers to the general problem of producing a very accurate prediction rule by combining rough and moderately inaccurate rules-of-thumb [27]. This involves fitting a sequence of weak learners on modified data. The predictions from all of them are then combined through a weighted majority vote (or sum) to produce the final prediction. The data modification at each step consists of assigning higher weights to the training examples that were misclassified in the previous iteration. As iterations proceed, examples that are difficult to predict receive ever-increasing influence. This forces the weak learner to concentrate on the examples that are missed by its predecessor.

XGBoost is a boosted tree algorithm. It follows the principle of gradient boosting [28]. Compared to other gradient boosted machines, it uses a more regularized-model formalization to control over-fitting, which gives it better performance. What we need to learn are the functions f_i, with each containing the structure of the tree and the leaf scores [29]. This can be formalized as seen in (2):

$$f_t(x)=w_{q(x)},\ w\in R^T,\ q:R^d\rightarrow\{1,2,\cdots,T\} \tag{2}$$

Where 'w' is the vector of scores on leaves, 'q' is a function assigning each data point to the corresponding leaf and 'T' is the number of leaves. The model complexity is formulated as:

$$\Omega(f)=\gamma T+\frac{1}{2}\lambda\sum_{j=1}^{T} w_j^2 \tag{3}$$

The objective function at the t^{th} iteration is as seen in (4):

$$Obj^{(t)}=\sum_{j=1}^{T}[G_j w_j+\frac{1}{2}(H_j+\lambda)\,w_j^2]+\gamma T \tag{4}$$

Solving this quadratic (4), the best w_j for a given structure q(x) and the best objective reduction we can get is:

$$w_j^*=-\frac{G_j}{H_j+\lambda} \tag{5}$$

$$Obj^*=-\frac{1}{2}\sum_{j=1}^{T}\frac{sqr(G_j)}{H_j+\lambda}+\gamma T \tag{6}$$

The score gained by splitting a leaf into 2 leaves is as seen in (7):

$$Gain=\frac{1}{2}\left[\frac{sqr(G_L)}{H_L+\lambda}+\frac{sqr(G_R)}{H_R+\lambda}-\frac{sqr(G_L+G_R)}{H_L+H_R+\lambda}\right]-\gamma \tag{7}$$

Where: $G_j=\Sigma_{i\in Ij}\ g_i$ and $H_j=\Sigma_{i\in Ij}\ h_i$; the definitions of which are as per [29].

IV. Exprerimental Design

The population under study was a particular level of stores leadership team of a global retailer over an 18 months period. The population chosen is distributed across various locations in the US. The data was pulled at a Quarterly level. There are 2 Class labels - Active and Terminated labeled 0 and 1 respectively. Each employee would have a record for every quarter of being active in the organization, until the quarter of turnover (if it occurs), at which time the data point changes class label from active to terminated. The dataset had 73,115 data points with each labeled active or terminated.

The features for the dataset were chosen based on the studies referenced in section II. The data was gathered from 2 sources: the HRIS database of the organization, as well as the BLS (Bureau of Labor Statistics). The HRIS database of the organization provided some key features like demographics features e.g. age etc.; compensation related features like pay etc.; team related features like peer attrition etc. The BLS data provided key features like unemployment rate, median household income etc.

Overall there were 33 features of which 27 were numeric while 6 were categorical in nature.

A. Data pre-processing

For categorical variables the missing values were imputed using the mode of that field. For numerical variables, missing values were imputed on a case-to-case basis. Zero-imputation was done on fields like number of promotions to prevent inflating data around employee promotions. Domain knowledge directed the imputation of certain numeric fields. For instance time since last promotion was imputed using tenure-in-position, as was known to be a good approximation. Certain other numeric variables were median-imputed as it handles the presence of outliers unlike mean imputation. As part of the data preparation, the categorical features were One-Hot Encoded, by which each of the distinct values in the categorical fields was converted to binary fields.

B. Model validation technique

The dataset was split 80:20 into training and hold out sets. A grid-search was performed over tuning parameters, including regularization or penalty hyper-parameters, for each algorithm. The optimal configuration of hyper-parameters for each algorithm was chosen based on a 10-fold cross validation on the training set. The models were trained using their optimal-configuration on the training dataset. The trained model from each algorithm was then used to predict and test on the 20% holdout sample.

C. Evaluation criteria for model(s)

The Area under the receiver operating characteristic curve (ROC-AUC) is the measure chosen here to compare classification accuracies. The AUC is a general measure of 'predictiveness' and decouples classifier assessment from operating conditions i.e., class distributions and misclassification costs [30]. Furthermore, AUC is preferable over alternative indicators like, e.g., error-rate because it measures the probability that a classifier ranks a randomly chosen positive instance higher than a randomly chosen negative one, which is equivalent to the Wilcoxon test of ranks [31].

Additionally, model run time and memory utilization are also used to compare the performance of the classifiers. These 2 measures are important to report on, as they build a case from a practitioner's perspective on determining what algorithm is good to implement for real-life business problems, solving for scalability and performance.

D. System specification

All classifiers except XGBoost are used from the scikit-learn package in Python 2.7. XGBoost classifier was used from the XGBoost package. The codes were run on a 16 GB MacBook OS 10.10.5 version.

V. Results

TABLE II. Model Results

Algorithm	AUC (Training)	AUC (Holdout)	Run-time (Training)	Maximum Memory Utilization (Of 16 GB)
XGBoost	**0.88**	**0.86**	16 min 12 sec	**12%**
Logistic Regression	0.66	0.50	**52 sec**	20%
Naïve Bayesian	0.64	0.59	59 sec	20%
Random Forest (Depth controlled)	0.79	0.51	23 min 10 sec	29%
SVM (RBF kernel)	0.68	0.52	105 min 30 sec	21%
LDA	0.74	0.52	6 min 51 sec	35%
KNN (Euclidean distance)	0.52	0.5	180 min 12 sec[a]	35%

[a] *Since KNN is a lazy learner, we are measuring the run time till final output for this model*

A. Lift Charts

The output obtained as the prediction is the probability of attrition, which is then converted to a risk ranking of employees. The model was further validated by checking the performance of each risk decile by means of a lift chart as depicted in Figure 1. A Lift Chart visualizes the improvement that a particular model provides when compared against a random guess.

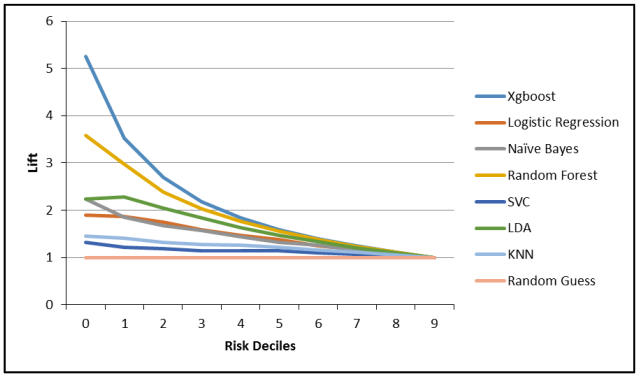

Fig. 1. Lift Chart for the Classifiers

It can be gauged from figure 1 that the XGBoost model has better decile performance than other models till the 7th decile (inclusive). It is also consistently and considerably better than a random guess.

B. Discussion

The population in this dataset is representative of a workforce that is distributed across the United States, comprising of people at different stages of their careers, different levels of performance and pay, and from different backgrounds. Hence, it's intuitive to assume that a rule based approach or a tree-based model will most likely perform best, considering the various themes and groups naturally occurring in the data. This intuition is validated by the observations in Table 2. It is seen that the two tree-based classifiers in Random Forest and XGBoost performs better than the other classifiers during training and that XGBoost is significantly better than Random Forest during testing. The XGBoost classifier outperforms the other classifiers in terms of accuracy and memory utilization.

Algorithmically, Random Forests trusts its stages of randomization to help it achieve better generalization but as is seen from the table it's still insufficient to prevent over-fitting in this case. On the other hand the XGBoost tries to add new trees that compliments the already built ones. Boosting serves to improve training for the difficult to classify data points. Another important point is the over-fitting suffered by classifiers other than XGBoost despite regularization or introduction of randomness, as the case maybe. XGBoost overcomes this problem due to its excellent inherent regularization (as shown mathematically in Section III, G) and hence works perfectly for the noisy data from the HRIS.

The XGBoost classifier is also optimized for fast, parallel tree construction, and designed to be fault tolerant under the distributed setting [29]. XGBoost classifier takes data in the form of DMatrix. DMatrix is an internal data structure used by

XGBoost which is optimized for both memory efficiency and training speed. Here, DMatrixes were constructed from numpy arrays of the features and the classes.

VI. Conclusions and Future Work

The importance of predicting employee turnover in organizations and the application of machine learning in building turnover models was presented in this paper. The key challenge of noise in the data from HRIS that compromises the accuracy of these predictive models was also highlighted. Data from the HRIS of a global retailer was used to compare the XGBoost classifier against six other supervised classifiers that had been historically used to build turnover models. The results of this research demonstrate that the XGBoost classifier is a superior algorithm in terms of significantly higher accuracy, relatively low runtimes and efficient memory utilization for predicting turnover. The formulation of its regularization makes it a robust technique capable of handling the noise in the data from HRIS, as compared to the other classifiers, thus overcoming the key challenge in this domain. Because of these reasons it is recommended to use XGBoost for accurately predicting employee turnover, thus enabling organizations to take actions for retention or succession of employees.

For future studies, the authors recommend the capture of data around interventions done by the organization for at-risk at employees and its outcome. This will transform the model into a prescriptive one, addressing not just the question "Who is at risk?" but also "What can we do?". It is also recommended to study the application of deep learning models for predicting turnover. A well-designed network with sufficient hidden layers might improve the accuracy, however the scalability and practical implementation aspect has to be studied as well.

References

[1] S. Jahan, "Human Resources Information System (HRIS): A Theoretical Perspective", Journal of Human Resource and Sustainability Studies, Vol.2 No.2, Article ID:46129, 2014.

[2] M. Stoval and N. Bontis, "Voluntary turnover: Knowledge management – Friend or foe?", Journal of Intellectual Capital, 3(3), 303-322, 2002.

[3] J. L. Cotton and J. M. Tuttle, "Employee turnover: A meta-analysis and review with implications for research", Academy of management Review, 11(1), 55-70, 1986.

[4] L. M. Finkelstein, K. M. Ryanand E.B. King, "What do the young (old) people think of me? Content and accuracy of age-based metastereotypes", European Journal of Work and Organizational Psychology, 22(6), 633-657, 2013.

[5] B. Holtom, T. Mitchell, T. Lee, and M. Eberly, "Turnover and retention research: A glance at the past, a closer review of the present, and a venture into the future", Academy of Management Annals, 2: 231-274, 2008

[6] C. von Hippel, E. K. Kalokerinos and J. D. Henry, "Stereotype threat among older employees: Relationship with job attitudes and turnover intentions", Psychology and aging, 28(1), 17, 2013.

[7] S. L. Peterson, "Toward a theoretical model of employee turnover: A human resource development perspective", Human Resource Development Review, 3(3), 209-227, 2004.

[8] J. M. Sacco and N. Schmitt, "A dynamic multilevel model of demographic diversity and misfit effects", Journal of Applied Psychology, 90(2), 203-231, 2005.

[9] D. G. Allen and R. W. Griffeth, "Test of a mediated performance – Turnover relationship highlighting the moderating roles of visibility and reward contingency", Journal of Applied Psychology, 86(5), 1014-1021, 2001.

[10] D. Liu, T. R. Mitchell, T. W. Lee, B. C. Holtom, and T. R. Hinkin, "When employees are out of step with coworkers: How job satisfaction trajectory and dispersion influence individual-and unit-level voluntary turnover", Academy of Management Journal, 55(6), 1360-1380, 2012.

[11] B. W. Swider, and R. D. Zimmerman, "Born to burnout: A meta-analytic path model of personality, job burnout, and work outcomes", Journal of Vocational Behavior, 76(3), 487-506, 2010.

[12] T. M. Heckert and A. M. Farabee, "Turnover intentions of the faculty at a teaching-focused university", Psychological reports, 99(1), 39-45, 2006.

[13] H. Jantan, A. R. Hamdan, and Z. A. Othman, "Towards Applying Data Mining Techniques for Talent Managements", 2009 International Conference on Computer Engineering and Applications, IPCSIT vol.2, Singapore, IACSIT Press, 2011.

[14] V. Nagadevara, V. Srinivasan, and R. Valk, "Establishing a link between employee turnover and withdrawal behaviours: Application of data mining techniques", Research and Practice in Human Resource Management, 16(2), 81-97, 2008.

[15] W. C. Hong, S. Y. Wei, and Y. F. Chen, "A comparative test of two employee turnover prediction models", International Journal of Management, 24(4), 808, 2007.

[16] L. K. Marjorie, "Predictive Models of Employee Voluntary Turnover in a North American Professional Sales Force using Data-Mining Analysis", Texas, A&M University College of Education, 2007.

[17] D. Alao and A. B. Adeyemo, "Analyzing employee attrition using decision tree algorithms", Computing, Information Systems, Development Informatics and Allied Research Journal, 4, 2013.

[18] V. V. Saradhi and G. K. Palshikar, "Employee churn prediction", Expert Systems with Applications, 38(3), 1999-2006, 2011.

[19] D. Michie, D. J. Spiegelhalter, and C. C. Taylor, Machine Learning, Neural and Statistical Classification. Ellis Horwood Limited, 1994.

[20] G. King and L. Zeng, "Logistic regression in rare events data", Political Analysis, 9(2), 137–163, 2001.

[21] T. Mitchell, Machine learning. 2nd ed. USA: McGraw Hill, 1997.

[22] H. A. Elsalamony (2014), "Bank direct marketing analysis of data mining techniques", International Journal of Computer Applications, 85(7).

[23] A. Liaw and M. Wiener, "Classification and regression by randomForest", R news, 2(3), 18-22, 2002.

[24] L. Breiman, Random forests. Machine Learning, 45(1), 5–32, 2001.

[25] P. Cunningham and S. J. Delany, "k-Nearest neighbour classifiers", Multiple Classifier Systems, 1-17, 2007.

[26] C. Cortes and V. Vapnik, Support-vector networks. Machine learning, 20(3), 273-297, 1995.

[27] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting", Journal of computer and system sciences, 55(1), 119-139, 1997.

[28] J. H. Friedman, "Greedy function approximation: a gradient boosting machine", Annals of statistics, 1189-1232, 2001.

[29] T. Chen and C. Guestrin, "XGBoost: Reliable Large-scale Tree Boosting System, 2015", Retrieved from http://learningsys.org/papers/LearningSys_2015_paper_32.pdf. Accessed 12 December 2015.

[30] S. Lessmann and S. Voß, "A reference model for customer-centric data mining with support vector machines", European Journal of Operational Research 199, 520–530, 2009.

[31] T. Fawcett, "An introduction to ROC analysis", Pattern Recognition Letters 27 (8), 861–874, 2006.