Comparison of the CatBoost Classifier with other Machine Learning Methods

Abdullahi A. Ibrahim[1], Raheem L. Ridwan[2], Muhammed M. Muhammed[3], Rabiat O. Abdulaziz[4], Ganiyu A. Saheed[5]
Department of Mathematical Sciences, Baze University, Abuja, Nigeria[1,3]
African Institute for Mathematical Sciences, Accra, Ghana[2]
Department of Energy Engineering, PAUWES, University of Tlemcen, Algeria[4]
Institute of Mathematics, University of Silesia, Katowice, Poland[5]

***Abstract*—Machine learning and data-driven techniques have become very famous and significant in several areas in recent times. In this paper, we discuss the performances of some machine learning methods with the case of the catBoost classifier algorithm on both loan approval and staff promotion. We compared the algorithm's performance with other classifiers. After some feature engineering on both data, the CatBoost algorithm outperforms other classifiers implemented in this paper. In analysis one, features such as loan amount, loan type, applicant income, and loan purpose are major factors to predict mortgage loan approvals. In the second analysis, features such as division, foreign schooled, geopolitical zones, qualification, and working years had a high impact on staff promotion. Hence, based on the performance of the CatBoost in both analyses, we recommend this algorithm for better prediction of loan approvals and staff promotion.**

Keywords*—*Machine learning algorithms; data science; CatBoost; loan approvals; staff promotion

I. Introduction

Machine learning and data-driven techniques have become very significant and famous in several areas. Some of the machine learning algorithms used in practice include; support vector machine, logistic regression, CatBoost, random forest, decision tree, AdaBoost, extreme gradient boosting, gradient boosting, naive Bayes, K-nearest neighbor, and many more. In supervised machine learning, classifiers have been widely used in areas such as fraud detection, spam email, loan prediction, and so on. In this work, we shall look into the applications of some machine learning methods in areas of loan prediction and staff promotion.

The issuance of loans is one of the many profit sources of financial institutions. However, the problems of default by applicants have been of major concern to credit providing institutions [1]. Studies conducted in the past were mostly empirical and as such the problems of default have not been definitively dealt with. The furtherance of time to the $21st$ century was accompanied by bulks of archived data collected from years of loan applications. Statistical techniques have been developed to study past data to develop models that can predict the possibility of defaults by loan applicants; thus, providing a score of creditworthiness. The availability of voluminous data called Big data necessitated the introduction of machine learning tools that can be used to discriminate loan applicants based on creditworthiness. This study considered some of these machine learning techniques to classify loan applicants based on available data to assess the probability of default and also recommend the technique that yields the best performance.

Since the advent of machine learning, several pieces of research has been conducted to discriminate against a loan applicants. In Goyal and Kaur [2], the authors developed an ensemble model by aggregating together Support Vector Machine (SVM), Random Forest (RF), and Tree Model for Genetic Algorithm (TMGA). The ensembled model was compared with each of these models individually and eight other machine learning techniques namely Linear Model (LM), Neural Network (NN), Decision Trees (DT), Bagged CART, Model Trees, Extreme Learning Machine (ELM), Multivariate Adaptive Regression Spline (MARS) and Bayesian Generalized Linear Model (BGLM) and was concluded from the analysis that the ensembled algorithm provided an optimum result. Alomari and Fingerman [3] tried to discriminate against loan applicants by comparing six machine learning techniques. The study compared DT, RF, K-Nearest Neighbour (KNN), OneR (1R), Naïve Bayes (NB), and Artificial Neural Networks (ANN) in which Random Forest gave the best performance with an accuracy of 71.75%. In Ibrahim and Rabiat [1], four classifiers were used to prediction in titanic analysis and XGBoost achieved the highest accuracy. Also, Ulaga *et al.* [4] conducted exploratory research where the suitability of RF was tested in classifying loan applicants and accuracy of 81.1% was achieved. In related research by Li [5], RF, BLR, and SVM were used to predict loan approvals and RF outperformed the other techniques with an accuracy of 88.63%. Xia *et al.* [6] predicted approvals for a peer-to-peer lending system by comparing Logistic Regression (LR), Random Tree (RT), Bayesian Neural Network (BNN), RF, Gradient Boosted Decision Trees (GBDT), XGBoost, and CatBoost and the results indicated that CatBoost gave the best performance over the other classifiers. The review of past literature showed tremendous developments in the applications of machine learning classifiers and how ensembled classifiers outperform single classifiers. However, only a few pieces of research considered CatBoost classifier in loan prediction approvals; hence, this research seeks to compare eight machine learning methods namely Binary Logistic Regression, Random Forest, Ada Boost, Decision Trees, Neural Network, Gradient Boost, Extreme Gradient Boosting, and CatBoost algorithms in the prediction of loan approvals.

The application of machine learning in employee promotion is another area we shall look into. Employees/staff play a significant role in the development of an enterprise. Employee promotion in an enterprise is a major concern to both the employer and employee. In human resource management, staff promotion is very vital for organizations to attract, employ, retain, and effectively utilize their employee's talents [7]. Promotion of staff in an organization is based on some factors among which are age [8], gender [9], education [10], previous experience [11] and communication strategy or pattern [12]. In Long *et. al* [7], the authors applied some machine learning algorithms on Chinese data to predict employee promotion. It was discovered that, among all the available features in their dataset, the number of the different positions occupied, the highest departmental level attained and the number of working years affect staff promotion. In Sarkar *et. al* [13], joint data clustering, and decision trees were used to evaluate staff promotion. Saranya *et. al* [14] researched why the best and performing employees quit prematurely and predicted performing and valuable employees likely to quit prematurely. The proposed algorithm was recommended to the human resource department to determine valuable employees likely to quit prematurely. Previous works showed tremendous developments in the applications of machine learning but only few researchers have considered the CatBoost classifier in staff promotion. This research seeks to compare four machine learning methods namely Random Forest, Gradient Boost, Extreme Gradient Boosting, and CatBoost algorithms in the prediction of staff promotion.

Some of these literature only discussed the applications without emphases on the mathematics behind this algorithm. This paper will differ from others by highlighting the mathematics of the algorithm, the process of data cleaning, applying the supervised learning algorithms and evaluating these algorithms. This paper aim to develop a predictive machine learning model from supervised machine learning in areas of loan prediction and staff promotion. To achieve this aim, we shall set some objectives which will also be our contribution:

- Perform data science process such as exploratory analysis, perform data cleaning, balancing, and transformation
- Develop a predictive model from machine learning methods
- Apply some model evaluation metrics to determine the performance of the implemented models.

The rest of the paper is structured as follows: some machine learning algorithms are given in Section II while designs and nomenclatures are presented in Section III. Section IV presents the analytical results and Section V concludes the paper.

II. Materials and Algorithms

The following algorithms; Binary logistic regression, Random forest, Adaptive Boosting, Decision trees, Neural networks, gradient boost, XGBoost and Catboost, shall be discussed in this section.

A. Binary Logistic Regression

Consider a dataset with response variable (Y) classified into two categories, Y = 'Loan approved','not approved' or $Y = \{'promoted',' notpromoted'\}$. Logistic regression models the probability of Y belongs to a specific category. With approach (1) below to predict this probability:

$$p(X) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_n X_n \tag{1}$$

The conditions $p(X) < 0$ and $p(X) > 0$ can be predicted for values of X, except for range of X is limited. To keep away from this, $p(X)$ must be modelled with the help of a logistic function that generates between 0 and 1 values as output. The function is defined as in (2)

$$p(X) = \frac{e^{\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_n X_n}}{1 + e^{\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_n X_n}} \tag{2}$$

The *'maximum likelihood'* method is used to fit (2). The unknown coefficients $\beta_0, \beta_1, \beta_2, \ldots, \beta_n$ in (2) should be approximated based on the data available for training the model. The intuition of likelihood function can be expressed mathematically as in (3):

$$\ell(\beta_0, \ldots, \beta_n) = \prod_{i:y_i=1} p(x_i) \prod_{i':y'_i=0} (1 - p(x'_i)) \tag{3}$$

The estimates $\beta_0, \ldots, \beta_n$ are selected to maximize this function [1]. More explanation can be obtained in [15].

Basic Assumptions of Binary Logistic Regression

(i) The response variable must be binary.

(ii) The relationship between the response feature and the independent features does not assume a linear relationship.

(iii) Large sample size is usually required.

(iv) There must be little or no multicollinearity.

(v) The categories must be mutually exclusive and exhaustive.

B. Random Forest

Random forest (RF) algorithm is a well-known tree-based ensemble learning method and the bagging-type ensemble [16]. RF differs from other standard trees, each node is split using the best among a subset of predictors randomly chosen at that node [17]. This additional layer of randomness is what makes RF more robust against over-fitting [18]. To improve the bagged trees in RF, a small tweak that de-correlates the trees are made. As in bagging, we build several decision trees on bootstrapped training sets. But when building these decision trees, each time a split in a tree is considered, a random sample of m predictors is chosen as split candidates from the full set of p-predictors [19]. The RF approach for both classification and regression is presented in Algorithm 1.

Algorithm 1 Random Forest Algorithm

```
(i) Draw $m_{tree}$ bootstrap samples from the initial data.

(ii) Initialize an unpruned tree, for every bootstrap sample, with the modification given as follow: instead of choosing the best-split among all predictors at each node, sample randomly $n_{try}$ of the predictors and select the best-split from among those features. Bagging can be seen as a special case of random forests which can be obtained when $n_{try} = k$, number of predictors.

(iii) new data is predicted by aggregating the predictions of the $m_{tree}$ trees.
```

C. Adaptive Boosting

Adaptive boosting (AdaBoost) algorithm is another machine learning method used to improve the accuracy of other algorithms. It is a boosted algorithm generated by training weaker rules to develop a boosted algorithm. In Adaboost, training sets $(x_1, y_1), \ldots, (x_m, y_m)$ is the input, where each x_i belongs to some *instance space X*, and each *feature* y_i is in some label set Y (in this case assuming that $Y = \{-1, +1\}$. This method calls repeatedly a given weak or base learning algorithm in a given series of rounds $t = 1, \ldots, T$. One of the significant and vital ideas of the algorithm is to keep a distribution or set of weights over the training set. The weight of this distribution on training samples *i* on round *t* is represented by $D_t(i)$.

At initial, all weights are set equally, but on each round, the weights of misclassified samples are increased so that the weak learner is forced to focus on the hard samples in the training set. The weak learner's job is to find a weak hypothesis $h_t : X \to \{-1, +1\}$ appropriate for the distribution D_t [20]. A metric used to measure the goodness of a weak hypothesis is its error. The algorithm procedure is presented in algorithm 2.

Algorithm 2 Adaboost Algorithm

```
Given: $(x_1, y_1), \ldots, (x_m, y_m)$ where $x_i \epsilon X$, $y_i \epsilon Y = \{-1, +1\}$

Initialize: $D_1(i) = \frac{1}{m}$ for $i = 1, \ldots, m$. For $t = 1, \ldots, T$ :
    • Train weak learner using distribution $D_t$.
    • Get weak hypothesis $h_t : X \to \{-1, +1\}$ with error
    • $\epsilon_t = Pr_{i \sim D_t}[h_t(x_i) \neq y_i]$
    • Choose $\alpha_t = \frac{1}{2}\ln(\frac{1 - \epsilon_t}{\epsilon_t})$
    • Update: $D_{t+1}(i) = \frac{D_t(i)}{Z_t} \times \begin{cases} exp(-\alpha_t) & \text{if} \quad h_t(x_i) = y_i \\ exp(\alpha_t) & \text{if} \quad h_t(x_i) \neq y_i \end{cases} = \frac{D_t(i)\exp(-\alpha_t y_i h_t(x_i))}{Z_t}$
      where $Z_t$ is a normalization factor (chosen so that $D_{t+1}$ will be a distribution).
      Output the final hypothesis:
      $H(x) = sign(\sum_{t=1}^{T} \alpha_t h_t(x))$
```

D. Decision Trees

Decision trees are one of the supervised learning algorithms that can be applied to both classification and regression problems [21]. We shall briefly consider regression and classification tree problems. There are two steps (as explained in [21]) for building a regression tree:

(i) Divide the set of feasible values $X_1, \ldots, X_n$ for into $I-$distinct and non-overlapping regions, $R_1, R_2, \ldots, R_i$.

(ii) For each sample that falls into R_i, the same prediction is made, which is the average of the dependent feature for the training sets in R_i.

In order to construct regions $R_1, \ldots, R_i$, we elaborate on step (i) above. In theory, $R_1, R_2, \ldots, R_i$ could take any shape or dimension.

However, for simplicity, we may split the predictor space into high-dimensional boxes and for easy interpretation of the predictive model. The aim is to obtain boxes $R_1, \ldots, R_i$ that minimizes the Residual Sum of Squares (RSS) as given in the mathematical expression in (4)

$$\sum_{i=1}^{I} \sum_{j \in R_i} (y_j - \hat{y}_{R_i})^2 \tag{4}$$

Where $(\hat{y}_{R_i})$ is the mean response of the training sets in the ith box.

The classification tree on the other hand predicts a qualitative response variable. In a classification tree, we predict that every observation belongs to the 'most frequently occurring' class of training sets in the region to which it belongs since we intend to allocate sample in a given region to the 'most frequently occurring' class of training sets in that region, the classification error rate is the part of the training sets in that region that do not belong to the most frequent class, as given in (5).

$$E = 1 - \max_{l}(\hat{p}_{ml}) \tag{5}$$

where $\hat{p}_{ml}$ denotes the ratio of training samples in the mth region from lth class. However, it turns out that classification error is not sensitive enough for tree-growing. The *Gini index* which is defined mathematically in (6)

$$G = \sum_{l=1}^{L} \hat{p}_{ml}(1 - \hat{p}_{ml}) \tag{6}$$

A measure of total variance over the L classes. Further details can be found in [21].

E. Neural Network

An Artificial Neural Network (ANN) is an imitation of the interconnections made up in the human brain. The inputs in ANN represent the dendrites in the human brain which receives electrochemical signals from other neurons into the cell body. Every input carries a signal which is obtained by the product of its weight and the input to a hidden layer in the neuron powered by an activation function usually a sigmoid function, other activation functions like tangent hyperbolic function, linear function, step function, ramp function, and Gaussian function can also be used [22]. The last layer is the output layer which represents the axon extending to the synapse that connects two different neurons. A typical ANN architecture has inputs, output, and a bias. The ANN architecture differs majorly by layers. The most common and simple architecture is a Perceptron which has two inputs, a hidden layer, and a single output. The neural networks are mostly backpropagated to be used for classification and prediction. The back and forth movement in a neural network between the input and output layers is referred to as an epoch. A neural network undergoes several epochs until a tolerable error is achieved and thus the training of an artificial neural network is achieved. ANN architecture is shown in Fig. 1.

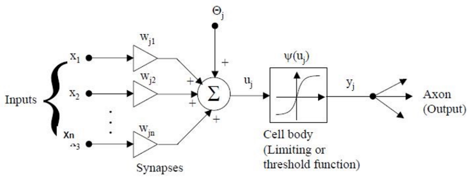

Fig. 1. Architecture of an Artificial Neural Network [23]

where $\Theta =$ external threshold, offset or bias $w_{ji} =$ synaptic weights $x_i =$ inputs $y_i =$ output as in (7)

$$y_i = \psi(\sum_{i=1}^{n} w_{ji}x_i + \Theta_i) \tag{7}$$

F. Gradient Boost

Gradient boost is a boosted algorithm used for regression and classification. It is derived from the combination of Gradient Descent and Boosting. It involves fitting an ensemble model in a forward stage-wise manner. The first attempt to generalize an adaptive boosting algorithm to gradient boosting that can handle a variety of loss functions was done by [24], [25]. The steps for gradient boosting algorithm is outlined in algorithm 3.

Algorithm 3 Gradient Boost Algorithm

```
Inputs:
    • Input data $(x, y)_{i=1}^{N}$
    • number of iterations M
    • choice of the loss-function $\psi(y, f)$
    • choice of the base-learner model $h(x, \theta)$
Algorithm:
    • intialize $\hat{f}_0$ with a constant
    • compute the negative gradient $g_t(x)$
    • fit a new base-learner function $h(x, \theta_t)$
    • find the best gradient descent step-size $\rho_t$ :
      $\rho_t = \arg\min_{\rho} \sum_{i=1}^{N} \psi[y_i, \hat{f}_{t-1}(x_i) + \rho h(x_i, \theta_t)]$
    • update the function estimate:
      $\hat{f} \leftarrow \hat{f}_{t-1} + \rho_t h(x, \theta_t)$
    • end for
```

G. Extreme Gradient Boosting

Extreme Gradient Boosting (XGBoost) is one of the boosted tree algorithms [16], which follows the principle of gradient boosting [24]. When compared with other gradient boosting algorithms, XGBoost makes use of a more regularized model formalization in other to control over-fitting of data, which gives it better performance [16]. In other to achieve this, we need to learn functions h_i, with each containing structure of tree and leaf scores [26]. As explained in [27], Given a data with m-samples and n-features, $\mathcal{D} = \{(X_j, y_j)\}(|D| = m, X_j \in \mathbb{R}^n, y_j \in \mathbb{R})$ a tree ensemble model makes use of L additive functions to predict the output as presented in (8).

$$\hat{y}_j = \phi(X_j) = \sum_{l=1}^{L} h_l(X_J), \qquad h_l \in \mathcal{H} \tag{8}$$

where $\mathcal{H} = \{h(X) = w_q(X)\}(q : \mathbb{R}^n \to U, w \in \mathbb{R}^U)$ is the space of regression trees. q denotes the structure of each tree that maps a sample to its corresponding leaf index. U denotes number of leaves in the tree. Each h_l corresponds to independent structure of tree q and leaf weights w.

To learn the set of functions used in the model, the regularized objective is minimized (9) as follows:

$$\mathcal{L}(\phi) = \sum_{j} l(\hat{y}_j, y_j) + \sum_{l} \Omega(h_l),\ \Omega(h) = \gamma U + \frac{1}{2}\lambda ||w||^2 \tag{9}$$

where l is differentiable convex loss function which measures difference between the target y_j and predicted $\hat{y}_j$. Ω

penalizes the complexity of the model to avoid over-fitting. The model is trained in an additive way. A score to measure the quality of a given tree structure q is derived as given in (10):

$$\hat{\mathcal{L}}^{(u)}(q) = -\frac{1}{2}\sum_{j=1}^{U}\frac{(\sum_{i=I_j} f_i)^2}{\sum_{i=I_j} g_i+\lambda}+\gamma U \tag{10}$$

where $f_i = \partial_{\hat{y}^{(u-1)}} l(y_i, \hat{y}^{(u-1)})$ and $g_i = \partial^2_{\hat{y}^{(u-1)}} l(y_i, \hat{y}^{(u-1)})$ are the gradient and second order gradient statistics, respectively. Further explanation can be obtained in [27].

H. CatBoost

Another machine learning algorithm that is efficient in predicting categorical feature is the CatBoost classifier. CatBoost is an implementation of gradient boosting, which makes use of binary decision trees as base predictors [28]. Suppose we observe a data with samples $D = \{(X_j, y_j)\}_{j=1,\ldots,m}$, where $X_j = (x_j^1, x_j^2, \ldots, x_j^n)$ is a vector of n features and response feature $y_j \in \mathbb{R}$, which can be binary (i.e yes or no) or encoded as numerical feature (0 or 1). Samples (X_j, y_j) are independently and identically distributed according to some unknown distribution $p(\cdot,\cdot)$. The goal of the learning task is to train a function $H : \mathbb{R}^n \rightarrow \mathbb{R}$ which minimizes the expected loss given in (11)

$$\mathcal{L}(H) := \mathbb{E}L(y, H(X)) \tag{11}$$

where $L(\cdot,\cdot)$ is a smooth loss function and (X, y) is a testing data sampled from the training data D.

The procedure for gradient boosting [24] constructs iteratively a sequence of approximations $H^t : \mathbb{R}^m \rightarrow \mathbb{R}, t = 0, 1, \ldots$ in a greedy fashion. From the previous approximation H^{t-1}, H^t is obtained in an additive process, such that $H^t = H^{t-1} + \alpha g^t$, with a step size α and function $g^t : \mathbb{R}^n \rightarrow \mathbb{R}$, which is a base predictor, is selected from a set of functions G in order to reduce or minimize the expected loss defined in (12):

$$\begin{aligned} g^t &= \text{arg min}_{g\in G}\mathcal{L}(H^{t-1} + g) \\ &= \text{arg min}_{g\in G}\mathbb{E}L(y, H^{t-1}(X) + g(X)) \end{aligned} \tag{12}$$

Often, the minimization problem is approached by the Newton method using a second-order approximation of $\mathcal{L}(H^{t-1} + g^t)$ at H^{t-1} or by taking a (negative) gradient step. Either of these functions is gradient descent [29], [30]. Further explanation of CatBoost algorithm can be obtained in [28].

III. Design and Nomenclatures

Some evaluation metrics such as confusion matrix, the area under the curve (AUC), accuracy, error rate, true positive rate, true negative rate, false-positive rate, and false-negative rate shall be discussed.

A. Confusion Matrix

A confusion matrix contains information about actual and predicted classifications from a classifier. The performance of such a classifier is commonly evaluated using the data in the matrix. Table I shows the confusion matrix for classifier [1], [31].

TABLE I. Confusion Matrix

		Predicted	
		Negative	Positive
Actual	Negative	True Negative (TN)	False Negative (FN)
	Positive	False Positive (FP)	True Positive (TP)

True Positive: The classifier predicted a true event and the event is actually true.

True Negative: The classifier predicted that an event is not true and the event is actually not true.

False Positive: The classifier predicted that an event is true but the event is actually not true.

False Negative: The classifier predicted that an event is not true but the event is actually true.

The confusion matrix can be interpreted as: the TN and TP are the correctly classified classes while FN and FP are the misclassified classes.

B. Model Evaluation Metrics

The Model training time, model accuracy, and memory utilized are some good metrics for comparing the performance of the classifiers. Also, the area under the Receiver Operating Characteristics Curve (ROC-AUC) is a performance metric for classification accuracy. The AUC is another metric which checks the performance of multiple-class classification accuracy [26]. Model accuracy is the proportion of the correct predictions (True positive and True negative) from the total predictions defined in (13).

$$\begin{aligned} \text{Accuracy} &= \frac{TN + TP}{TP + TN + FP + FN} \times 100\% \\ \text{Error Rate} &= \frac{FP + FN}{FP + FN + TP + TN} \times 100 \end{aligned} \tag{13}$$

The error rate is the proportion of all incorrect predictions divided by the total number of samples, given in (13).

True Positive Rate (TPR), also called the sensitivity or recall, is the proportion of correct positive predicted class from total positive class. The best sensitivity is 1.0 and the worst is 0.0. True Negative Rate (TNR), also called the specificity, is the proportion of correct negative predictions from the total number of negative classes. The best specificity is 1.0 and the worst is 0.0. The TPR and TNR are given in (14).

$$\begin{aligned} \text{True Positive Rate} &= \frac{TP}{FN + TP} \times 100 \\ \text{True Negative Rate} &= \frac{TN}{FP + TN} \times 100 \end{aligned} \tag{14}$$

Precision is the number of correctly predicted positive value out of the total number of positive class, as given in (15). False Positive Rate (FPR) is the number of incorrect positive prediction out of the total number of negatives as in (15).

$$\begin{aligned} \text{False Positive Rate} &= \frac{TP}{FNP+TP} \times 100 \\ \text{False Negative Rate} &= \frac{FP}{FP+TN} \times 100 \end{aligned} \tag{15}$$

C. Calibration Plots

Calibrated methods (classifiers) are probabilistic classifiers for which the outcome of the predicted probabilities of a particular classifier can be interpreted as a confidence interval. The metric is used to determine whether the predicted probability can be interpreted as a confidence interval.

D. System Specification

All classifiers were run on Jupyter notebook in python $3.7.4$ on Linux 19.10 version. The codes were run on 8GB HP elite book, core $i5$.

IV. Results and Discussion

In this section, we shall perform two analyses to determine the performance of all the machine learning algorithms discussed previously. We begin by exploring the data to obtain the numerical statistics, identify missing values, outliers, and if the independent feature is balanced or not. After initial exploration we were able to identify missing values and outliers, the independent feature is balanced.

A. Analysis 1: Predicting Mortgage Approvals from Government Data

The analysis is based on US Government data concerning predicting mortgage approvals [32]. This is a binary classification problem. Our analysis was based on the $500,000$ observations with 23 features from the training data-set of mortgage approvals government data, each containing specific characteristics for a mortgage application which will either get approval ("1"); or not ("0"). We tested our model on a data-set with 150,000 samples.

1) Exploratory Analysis: Before developing a predictive model, we need to understand the data-set by exploratory analysis. In the exploratory analysis, we intend to find answers to some questions such as (i) which features have missing values, (ii) features with outliers, (iii) is the response feature balanced? (iv) the distribution of the data points and so on. We present some visualizations in Fig. 2 and 3 to answer these questions.

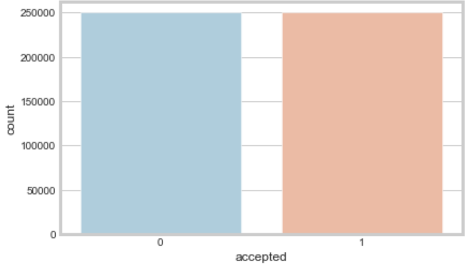

Fig. 2. Response Features

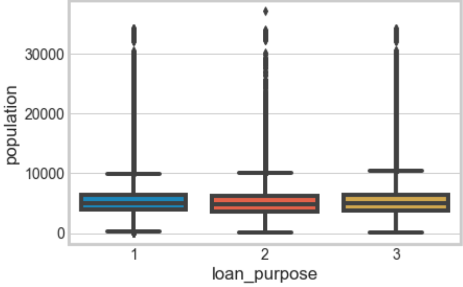

Fig. 3. Loan Purpose

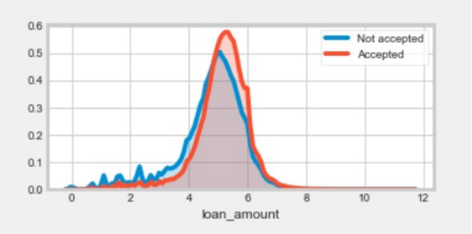

Fig. 4. Loan Amount

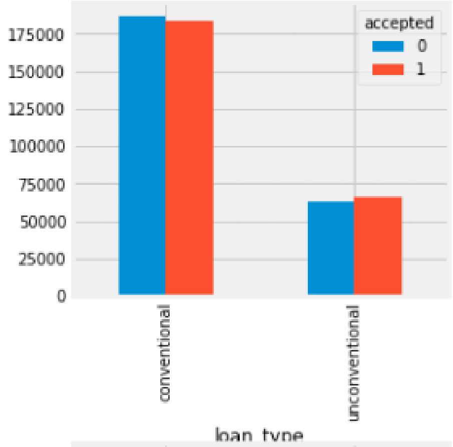

Fig. 5. Loan Type

Fig. 2 shows the both classes give almost the same frequency, with $250,114$ for the *accepted* data points and $249,886$ for *not accepted* data points. The data distribution seem balanced and other analysis can be performed. Fig. 3 shows the distributions of the three classes of the loan purpose that we have. The Loan amount follows a normal distribution for both the *accepted* and *not accepted* in Fig. 4. Fig. 5 shows the two classes of loan type. The accepted class of the conventional loan type has highest frequency with around $180,000$. In Fig. 6 state code 6 has the highest number ($\approx$ 890) of district lenders, with state code 50 having the least. Fig. 7 shows the features “msa-md, applicant-income, number-of-owner-occupied-units, number-of-1-to-4-family-units, tract-to-msa-md-income-pct” having numbers of $76982, 39948, 22565, 22530, 22514$, respectively. Generally, from our visualizations we can see that the major features that contribute to the prediction of mortgage loan are loan amount, loan type, applicant income and loan purpose.

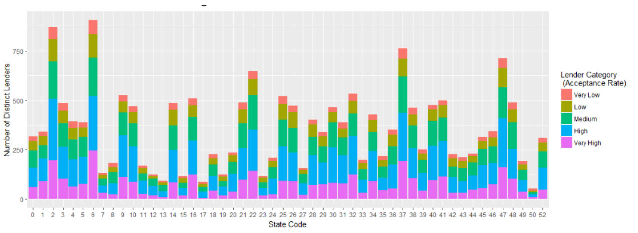

Fig. 6. District Lenders by State Code

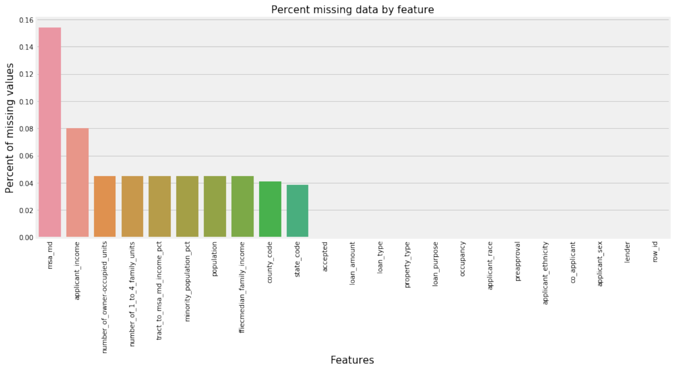

Fig. 7. Features with Missing Information

2) Data Pre-processing: To replace the missing values (NA’s) for both numerical and categorical features. Starting with the categorical features, the NA’s encountered were replaced with the mode of that feature. Also, categorical features were one-hot encoded, which means each of the distinct categories in a particular feature was converted to numerical fields. For numerical features, the NA’s were replaced on a case by case basis. Features like “applicant-income, number of owner-occupied units” were replaced with the median as it handles the presence of outliers, unlike mean imputation. The test set used has 150, 000 samples for each of the models.

TABLE II. PERFORMANCE COMPARISON OF THE ALGORITHMS

Algorithm	Score	Avg. time (fit)	Avg. time (score)(s)	F1 score	AUC	Precision
Logistic Regression	0.62	73.806	0.032	0.62	0.67	0.61
Random Forests	0.69	19.045	1.271	0.64	0.71	0.68
Adaboost	0.67	28.632	1.664	0.63	0.72	0.66
XGBoost	0.69	28.322	1.342	0.65	0.75	0.68
Neural Networks	0.68	27.234	1.123	0.73	0.66	0.66
Gradient Boosting	0.69	30.233	3.432	0.66	0.75	0.68
CatBoost	**0.732**	46.657	6.725	0.75	0.78	0.83
Decision Trees	0.68	4.272	1.012	0.054	0.62	0.66

3) Results and Discussion: False positive rate is a method of committing a type I error in null hypothesis testing when conducting multiple comparisons. For the problem used in this paper, the false positive rate is an important metric as it would be a disaster if the system predicts a client would be given a loan but in reality, he was not. From Table II, the CatBoost algorithm achieved the highest accuracy. This means that the confusion metrics for CatBoost, the value of correctly classified (TP + FN) is higher than the other six algorithms implemented. And with the least number of miss-classified. Other metrics such as f1 score, AUC, and precision are also shown in Table II.

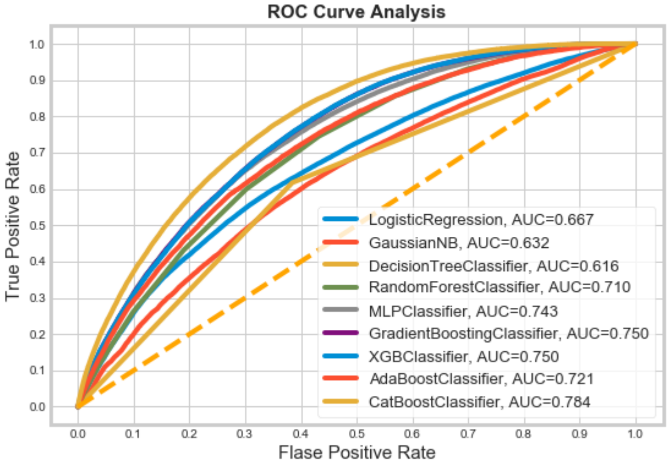

Fig. 8. ROC Curves for the Algorithms

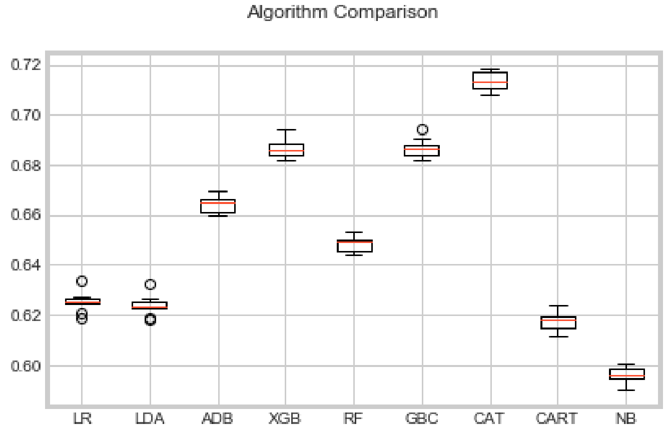

Fig. 9. Boxplots for the Algorithms

After training the models on the training set and predicted the probabilities on the test set, we then obtained the true positive rate, false positive rate, and AUC scores. From Fig. 8, CatBoost achieved highest AUC value of 0.78 which is closer to 1 than other classifiers. Also, Fig. 9 shows the comparison with other implemented algorithms. The names of the algorithms are written in the short form where LR denotes Logistic regression, ADB for AdaBoost, RF denotes random forest, GBC for gradient boosting, CART for decision trees, NB for naive Bayes, XGB denotes XGBoost, and CAT for CatBoost algorithms. Box 9 shows the spread of the accuracy scores across each cross-validation fold for each algorithm. Box 9 is generated based on the mean accuracy and standard deviation accuracy of the algorithms. In Fig. 10, the calibration plots for all the implemented algorithms are plotted, CatBoost method produced well-calibrated predictions as it optimizes log-loss. Fig. 11 shows the model performance graph for the CatBoost classifier.

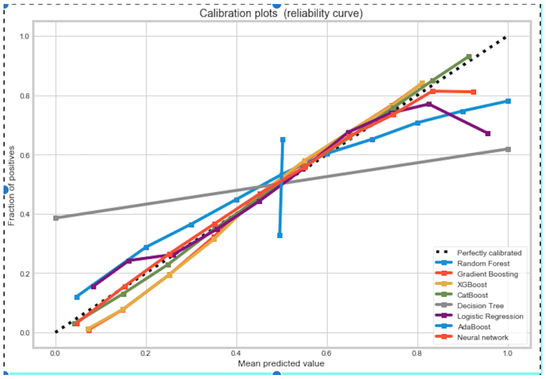

Fig. 10. Calibration Plots

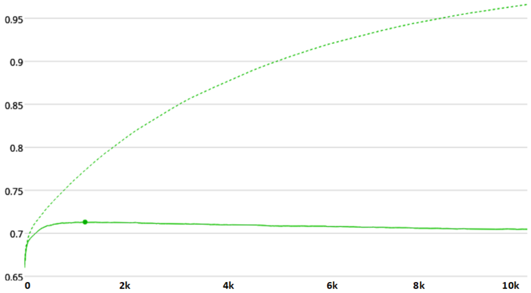

Fig. 11. Graph of the Model Training

From the plot 9, it would suggest that CatBoost is perhaps worthy of further study on this problem due to its performance. The result has been presented in Table II which contains the model accuracies, AUC, the average time to fit, and score. In summary, features such as *loan amount*, *loan type*, *applicant income* and *loan purpose* played a key role in predicting mortgage loan approvals. This mean, for an individual to get a mortgage loan, the amount of loan, what type of load, income of the loan applicant and purpose for wanting to secure loan are the key questions that needs to be addressed before a loan will be approved.

B. Analysis 2: Staff Promotion Algorithm

HR analytics using machine learning will revolutionize the way human resources departments now operate. This will lead to higher efficiency and better results overall. This analysis uses predictive analytics in identifying the employees most likely to get promoted or not using historical staff promotion datasets [32]. We trained the model on a dataset with $38,312$ samples and 19 features and tested it on a data-set with $16,496$ samples.

1) Exploratory Analysis: Before developing a predictive model, we need to understand the data points and have a pictorial view of what the data-set contains. In the exploratory analysis, we intend to find answers to some questions such as (i) which features have missing values, (ii) features with outliers, (iii) is the response feature balanced? (iv) the distribution of the data points and so on. We present some visualizations in Fig. 12, 13, 14, 15, 16, 17, and 18 to answer these questions.

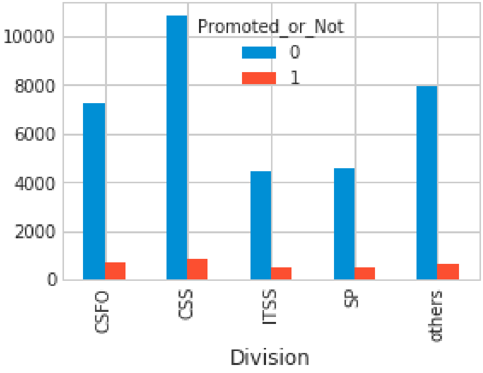

Fig. 12. Division

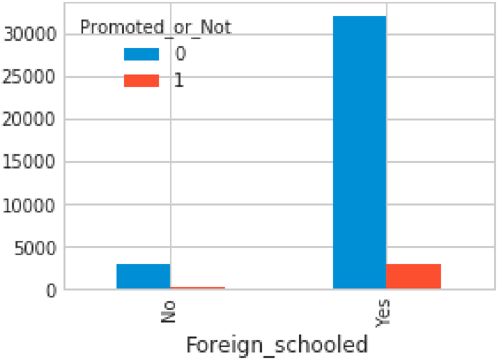

Fig. 13. Foreign School

Starting with Fig. 12, the CSS class have the highest number (over 1000) of staff that was promoted in the Division feature. In Fig. 13, more than 30000 staff who got promoted were foreign-schooled and $\approx$ 2500 studied locally. Fig. 14 shows the geographical zones of staffs, the South-West zone recorded the highest number of promoted staff while the North-East zone has the least number of promoted staff. Fig. 15 shows the frequency for the two classes in the response variable. It was observed that most of the staffs fall in the “not promoted” class, with a ratio of “promoted” to “not promoted” as $8 : 92\%$.

Furthermore, in Fig. 16, employees from Oyo state (in South-West) appears to have the highest number of working years (38 years) while employees from Zamfara state had 24 years of working experiences. This could further support Fig. 14 with staff from the South-West zone having the highest

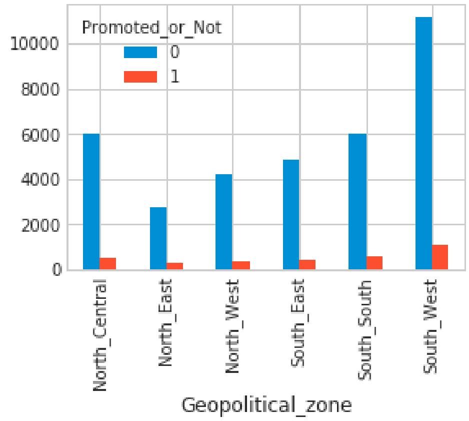

Fig. 14. Geographical Zones

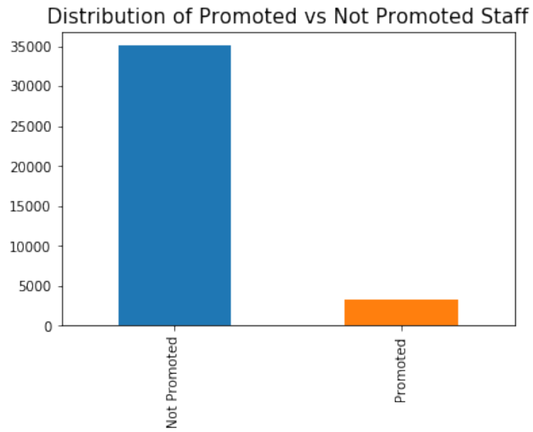

Fig. 15. Response Feature

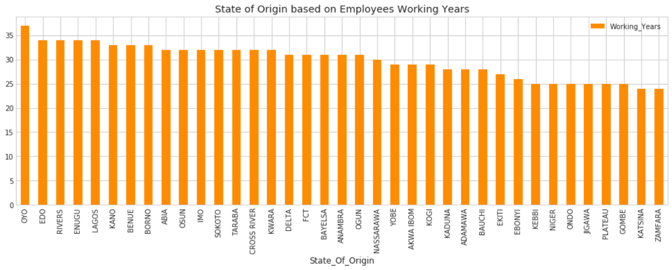

Fig. 16. State of Origin

number of promotion because of their working years. In Fig. 17, the graduate employees appear older than the Non-graduate employee. This could be due to the number of years spent studying before joining the workforce. The distribution of staff channel of recruitment is shown in Fig. 18. In summary, features such as division, foreign schooled, geopolitical zones, qualifications, and working years had a high impact on staff promotion.

2) Data-preprocessing: We replaced the missing values (NA's) for both numerical and categorical features. For the categorical features, the NA's encountered were replaced with the mode of that feature. For Numerical features, the NA's were replaced on a case by case basis. For the imbalanced response feature, it was balanced with a Resampling technique, to improve our prediction. After this step, we split the data and proceed to the prediction phase. The test set used has $16,496$ samples for each of the models.

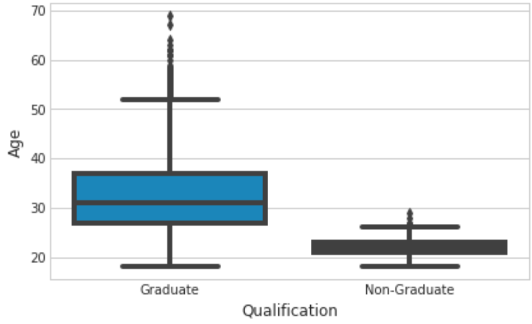

Fig. 17. Educational Status

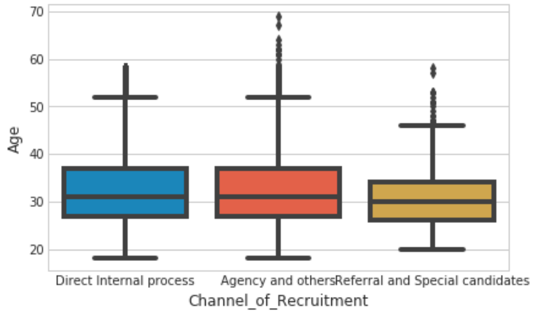

Fig. 18. Channel of Recruitment

3) Results and Discussion: Table III shows a summary of the evaluation metrics for implemented algorithms. CatBoost and XGBoost achieved the highest score with 94%. And when uploaded into Kaggle [33] online, we had a difference of 0.01. Other metrics are also shown in this table.

TABLE III. PERFORMANCE COMPARISON OF THE ALGORITHMS

Algorithm	Score (PC)	Score (Kaggle)	AUC	Precision	F1-Score
Random forest	0.93	0.88	0.71	0.70	0.94
XGBoost	0.94	0.93	0.82	0.93	0.92
Gradient Boost	0.90	0.84	0.82	0.93	0.95
CatBoost	**0.94**	0.93	0.82	0.91	0.95

In Fig. 19 the AUC value of the applied algorithms are plotted, the random forest classifier had the least value of 0.71 while other algorithms achieved 0.82. The distribution of the algorithms is shown in Fig. 20. Again, the random forest classifier achieved the least value while other algorithms achieved high values. The model performance graph for the CatBoost algorithm showing how the model was trained, the number of iterations, and the accuracy is shown in Fig. 21.

The proportion of the training set and the test error rate is plotted in Fig. 22. CatBoost and XGBoost had little error compared to other implemented algorithms. Fig. 23 shows that the CatBoost and XGBoost methods produced well-calibrated predictions.

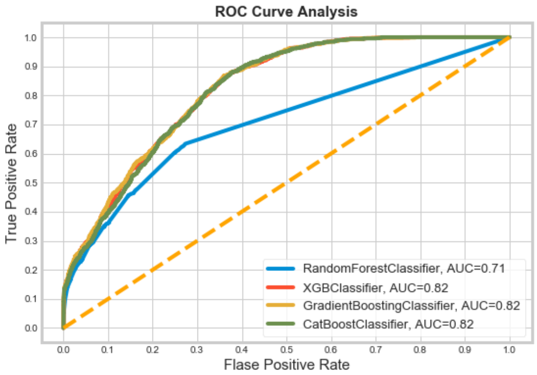

Fig. 19. ROC Curves for the Algorithms

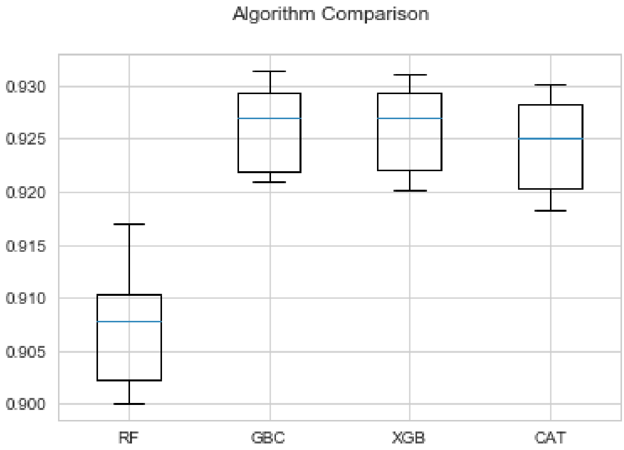

Fig. 20. Boxplots for the Algorithms

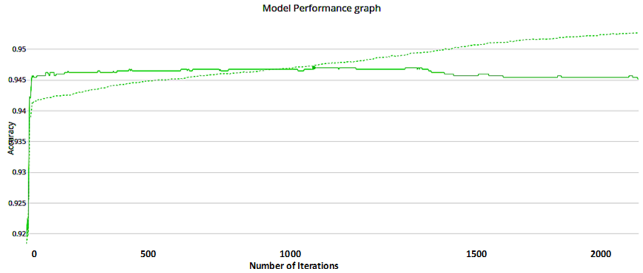

Fig. 21. Graph showing the Model Performance of the CatBoost Algorithm

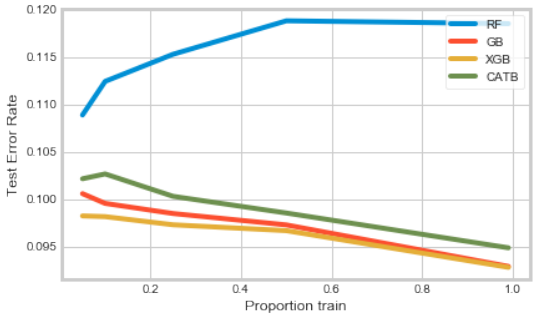

Fig. 22. Reliability Curves for the Test Error Rate

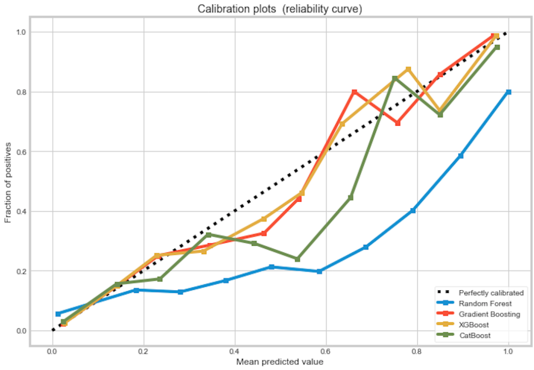

Fig. 23. Reliability Boxplots for the Algorithms

V. Conclusion

Having applied all the mentioned algorithms in our methodology, this paper aimed to compare some predictive machine learning algorithms from supervised learning with applications in areas of loan prediction and staff promotion. We performed two analyses: loan prediction and staff promotion. Each analysis started with exploratory analysis where we find insights from the data, then the data was cleaned, balanced, and transformed for prediction. The machine learning algorithms discussed in this paper were then implemented and some metrics were used to evaluate the implemented models' performance. CatBoost classifier did pretty well achieving the highest score (accuracy) in both applications (or analysis). Other evaluation metrics also support the performance of this algorithm. We thereby recommend the CatBoost classifier for the predictive model.

For the mortgage loan analysis, features such as *loan amount*, *loan type*, *applicant income* and *loan purpose* played a significant role in predicting mortgage loan approvals. And for the staff promotion analysis, features such as *division*, *foreign schooled*, *geopolitical zones*, *qualifications*, and *working years* had a significant impact towards staff promotion.

Future work might consider cross-validation. Cross-validation could also be used to compute the model's accuracy based on different combinations of training and test samples. Besides, some other classifiers with larger datasets may be applied.

Acknowledgment

The authors would like to appreciate Sowole Samuel for his suggestions.

References

[1] Abdullahi Adinoyi Ibrahim and Rabiat Ohunene Abdulaziz, *Analysis of Titanic Disaster using Machine Learning Algorithms*, Engineering Letters, vol. 28, no. 4, pp1161-1167, 2020.

[2] Anchal Goyal and Ranpreet Kaur, *Loan Prediction using Ensemble Technique.* International Journal of Advanced Research in Computer and Communication Engineering. Vol. 5(3), 2016.

[3] Zakaria Alomari and Dmitriy Fingerman, *Loan Default Prediction and Identification of Interesting Relations between Attributes of Peer-to-Peer Loan Applications.* New Zealand Journal of Computer-Human Interaction ZJCHI. Vol. 2(2), 2017.

[4] K. Ulaga Priya, S. Pushpa, K. Kalaivani and A. Sartiha, *Exploratory Analysis on Prediction of Loan Privilege for Customers using Random forest.* International Journal of Engineering and Technology. Vol. 7(2.21) Pp. 339-341, 2018. https://doi.org/10.14419/ijet.v7i2.21.12399

[5] Li Ying, *Research on Bank Credit Default Prediction Based on Data Mining Algorithm.* The International Journal of Social Sciences and Humanities Invention Vol. 5(6): 4820-4823, 2018.

[6] Y. Xia, L. He, Y. Li, N. Liu and Y. Ding, *Predicting Loan Default in Peer-to-Peer Lending using Narrative Data.* Journal of Forecasting. Pp. 1–21, 2019. https://doi.org/10.1002/for.2625

[7] Y. Long, J. Liu, M. Fang, T. Wang, & W. Jiang, *Prediction of Employee Promotion Based on Personal Basic Features and Post Features.* In Proceedings of the International Conference on Data Processing and Applications (pp. 5-10), 2018.

[8] C. S. Machado and M. Portela, *Age and Opportunities for Promotion.* IZA Discussion Paper No.7784, 2013.

[9] F. D. Blau and J. DeVaro, *New Evidence on Gender Differences in Promotion Rates: An Empirical Analysis of a Sample of New Hires.* Industrial Relations: A Journal of Economy and Society. 46, 3 (July. 2007), 511-550. DOI= https://doi.org/10.1111/j.1468-232x.2007.00479.x.

[10] S. Spilerman and T. Lunde, *Features of Educational Attainment and Job Promotion Prospects.* American Journal of Sociology. 97, 3 (Nov. 1991), 689-720, 1991. DOI= https://doi.org/10.1086/229817

[11] I. E. De Pater, A. E. Van Vianen, M. N. Bechtoldt and U. C. KLEHE, *Employees' Challenging Job Experiences and Supervisors' Evaluations of Promotability.* Personnel Psychology. 62, 2 (Summer 2009), 297-325. DOI= https://doi.org/10.1111/j.1744-6570.2009.01139.x

[12] A. W. Woolley, C. F. Chabris, A. Pentland, N. Hashmi and T. W. Malone, *Evidence for a Collective Intelligence Factor in the Performance of Human Groups.* Science. 330, 6004 (Oct. 2010), 686-688. DOI= https://doi.org/10.1126/science.1193147.

[13] A. Sarker, S. M. Shamim, M. S. Zama, & M. M. Rahman, *Employee's Performance Analysis and Prediction using K-Means Clustering & Decision Tree Algorithm.* Global Journal of Computer Science and Technology, 2018.

[14] S. Saranya and J. S. Devi, *Predicting Employee Attrition using Machine Learning Algorithms and Analyzing Reasons for Attrition.* 2018

[15] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani, *An Introduction to Statistical Learning: with Applications in R.*. Springer Texts in Statistics 103, DOI 10.1007/978-1-4614-7138-7 2017.

[16] R. Punnoose and P. Ajit, *Prediction of Employee Turnover in Organizations using Machine Learning Algorithms.* International Journal of Advanced Research in Artificial Intelligence (IJARAI), Vol. 5, No. 9, 1-5, 2016.

[17] A. Liaw and M. Wiener, *Classification and Regression by Random forest,* R news, 2(3), 18-22, 2002.

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

[19] Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani, *An Introduction to Statistical Learning with Applications in R. Springer textbook*, 2013.

[20] Yoav Freud and E. Robert Schapire, *A Short Introduction to Boosting.* Journal of Japanese Society for Artificial Intelligence. Vol 14(5):771-780, 1999.

[21] G. James et al., *An Introduction to Statistical Learning: with Applications in R,* Springer Texts in Statistics 103, DOI 10.1007/978-1-4614-7138-7 8

[22] Shiruru, Kuldeep, *An Introduction to Artificial Neural Network.* International Journal of Advance Research and Innovative Ideas in Education. 1. 27-30, 2016.

[23] Kumamoto University http://www.cs.kumamoto-u.ac.jp/epslab/ICinPS/Lecture-2.pdf

[24] J. H. Friedman, *Greedy Function Approximation: A Gradient Boosting Machine*, Annals of statistics, 1189-1232, 2001.

[25] N. Alexy and K. Alois, *Gradient Boosting Machines: A Tutorial. Frontiers in Neurorobotics.* Vol 7(21) pp 3, 2013.

[26] 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.

[27] T. Chen and C. Guestrin, *XGBoost: Reliable Large-scale Tree Boosting System*, 2015.

[28] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush & A. Gulin, *CatBoost: Unbiased Boosting with Categorical Features.* In Advances in Neural Information Processing Systems (pp. 6638-6648), 2018.

[29] J. Friedman, T. Hastie and R. Tibshirani, *Additive Logistic Regression: A Statistical View of Boosting.* The Annals of Statistics, 28(2):337–407, 2000.

[30] L. Mason, J. Baxter, P. L. Bartlett and M. R. Frean, *Boosting Algorithms as Gradient Descent.* In Advances in Neural Information Processing Systems, pages 512–518, 2000.

[31] R. Kohavi and F. Provost, *Glossary of Terms. Machine Learning—Special Issue on Applications of Machine Learning and the Knowledge Discovery Process.* Machine Learning, 30, 271-274, 1998.

[32] Microsoft Capstone Project https://www.datasciencecapstone.org/ (Assessed in April 2019)

[33] Kaggle https://www.kaggle.com/c/intercampusai2019 (Assessed in August 2019)