Information Systems 61 (2016) 40–52

Contents lists available at ScienceDirect

Information Systems

journal homepage: www.elsevier.com/locate/infosys

Benchmarking regression algorithms for income prediction modeling

Azamat Kibekbaev *, Ekrem Duman

Industrial Engineering Department, Özyeğin University, 34794 Istanbul, Turkey

ARTICLE INFO

Article history:
Received 21 January 2016
Received in revised form
30 April 2016
Accepted 2 May 2016
Recommended by: D. Shasha
Available online 12 May 2016

Keywords:
Regulation
Income prediction
Regression techniques

ABSTRACT

This paper aims to predict incomes of customers for banks. In this large-scale income prediction benchmarking paper, we study the performance of various state-of-the-art regression algorithms (e.g. ordinary least squares regression, beta regression, robust regression, ridge regression, MARS, ANN, LS-SVM and CART, as well as two-stage models which combine multiple techniques) applied to five real-life datasets. A total of 16 techniques are compared using 10 different performance measures such as R2, hit rate and preciseness etc. It is found that the traditional linear regression results perform comparable to more sophisticated non-linear and two-stage models.

1. Introduction

Credit is the borrowing capacity provided to an individual by the banking system, where a borrower receives some monetary value and agrees to repay the lender on some date in the future with interest. There are various types of credit for different segments of the society to cater their needs. These include: personal loans, credit cards, overdrafts, short-term commercial loans, long-term commercial loans, equipment leasing, and letters of credit. Our study focuses on credit cards, which is a very common and frequently used item in today's world.

The credit card market has exploded over the years with increasing interest in the computerization of society. It is one of the most widely accepted ways of payment around the world. Credit cards revolutionized consumers' spending habits and changed the face of business. In today's market, consumers are demanding more personal attention. They expect companies to understand their needs and to offer products and services that meet those needs almost instantaneously. As a result, today's lenders compete by offering complex and customized products to specific individuals. In doing so, they consider consumers' tastes for different cards, try to predict and monitor how valuable these consumers are as customers, and watch and respond to offers these individuals might receive from their competitors. For this reason, in today's economy, credit cards represent an important part of household, business, and global commercial activities.

Credit limit is the most important feature of a credit card. It is the amount of credit that a financial institution extends to a client or, simply, the total amount that can be spent with a credit card. The credit card limit of an individual will be decided by a combination of factors, which lenders are likely to look closely at the spending power and income, as these allows them to determine how much the individual can afford to borrow. Income information of an individual consumer enables the banks to validate the application data, target new prospects, and segment existing customers. Income can be defined as the sum of all earnings (wages, salaries, profits, interests payments, rents, etc.) in a given period of time (generally a month). Income is a crucial demographic element that is used at a wide variety of customer touch points. Therefore, it is very important to have income prediction for the existing and

* Corresponding author. Tel.: +90 5544640665.
E-mail addresses: kibekbaev.azamat@ozu.edu.tr (A. Kibekbaev), ekrem.duman@ozyegin.edu.tr (E. Duman).

http://dx.doi.org/10.1016/j.is.2016.05.001
0306-4379/

potential customers. However, collecting accurate indicators of income is difficult, yet it is essential for companies that need to prepare high-quality revenue budgets, especially in an uncertain economic environment with changing government policies. Increased competitions have made banks and financial institutions to search for new ways to minimize the denial of credit to creditworthy customers and to identify fraudulent ones as early as possible. Accordingly, banks and financial institutions need a rapid and accurate decision making process in order to maintain high lending power. The advantage of financial institutions is their individual segmentation of customer's data that facilitate analysis to categorize the optimum combination of outcome of risks and assets overtime. Under these conditions, the subjective judgment of decision makers is a crucial factor in making accurate forecasts to provide solutions that not only assess the creditworthiness, but also keep the per-unit processing cost low, while reducing turnaround time for the customers.

Our aim and motivation in this paper is to predict the incomes of customers for Turkish banks, in relation to some new banking regulations in Turkey. Turkey's banking regulator (BDDK) has announced new regulations that brought about a series of strict rules on the use of credit cards. A major rule was to launch a “single limit” for credit cards, where consumer's credit card limit could not exceed four times the amount of his/her monthly income to offset the uncontrolled use of loans and credit cards in the domestic market and to decrease the household debt levels. All these limits will be applicable to all the banks in Turkey, so that the sum of the limits from different banks cannot exceed this “single limit.” Income prediction will let us determine the credit card limit for each customer and will let us prevent credit card holders from exceeding their upper limits. Therefore, it is crucial to have models that estimate income as accurately as possible. In the near future, similar limitations will also be applied to personal loans.

Many empirical studies on modeling income are found in the literature, but this will be the first comprehensive benchmarking work in this area. It is very difficult to get the exact information about individuals' incomes, wealth, and their characteristics. Swan [39] showed that observed characteristics (such as age, occupation, sex, industry, and length of paid employment) explain a relatively small proportion of variability in income. In this paper, we concentrate on individual income models and predictions, yet the modeling is often limited because of lack of adequate data. For example, Carrier and Shand [10] argued that employers do not easily volunteer to provide salary data. Bone and Mitchell [5] showed that obtaining more appropriate data and good modeling for retirement income can lead to better estimation. Using U.S. sample data, Dominitz [15] presented income prediction by comparing datasets from 1993 and 1994. He found that income expectations are optimistic on average. In contrast, Das and van Soest [14] examined data from the Dutch Socio-Economic Panel between 1984 and 1989, where they had found that income expectations are too pessimistic on average. By examining 18 years of monthly data from the Michigan Survey of Consumer Attitudes and Behavior, Souleles [37] showed that most variables appear to have been biased and inefficient while predicting income.

On the basis of the literature survey, it is found that most income prediction studies were about the determination of future incomes for college students. They study the effect of a students college GPA, major, and standardized test scores in order to see what is most influential on future income. Thomas [40] and Smart [36] stated that college performance leads to higher earnings after graduation. Chia and Miller [12] used data from the University of Melbourne in Australia in order to study the effect of college performance on future income. They found that GPA plays a major role in determining the starting salaries.

Regression analysis is a statistical technique for investigating or estimating the relationship among variables. It is used when you want to predict a relationship between a dependent variable and one or more independent variables. Regression analysis can be of two types: nonlinear and linear. In this study, we will use regression techniques; hence, provided some articles from the literature related to income estimation by regression models.

Carlos et al. [9], by using regression models and types of neural network, made an accurate estimation of gross margin of farms. Results from artificial neural network (ANN) models have provided the most accurate gross margin predictions rather than regression models. Chen and Yang [11] proposed quasi-stepwise regression variable selection method based on the validation of least absolute deviation regression to forecast rural household net income. According to the study, the models constructed by the quasi-stepwise regression variable selection method and the least absolute deviation estimation method have lower errors and higher validation and can be applied to many other forecasting problems. Also, benchmarking works were done by Loterman et al. [27] to estimate loss given default (LGD) using various regression techniques for six real-life datasets. According to their observations, non-linear techniques such as least squares support vector machines (LSSVM), ANN, and MARS perform significantly better than the linear ones. Also, their work shows that the variance in LGD remains poorly explained, because the resulting techniques have limited explanatory power. Higgins and Sinning [21] proposed the importance of dynamic earnings modeling for the design of income contingent student loans by using regression models. According to the results, earnings modeling has considerable implications for the calculation of loan subsidies.

This paper is one of the first attempts to predict incomes to regulate credit limit of bank customers in Turkey to help banks in their income prediction problems. Since little is known about income prediction in credit cards, we want to fill the void of academic literature with our large-scale income prediction benchmarking study using 18 different regression techniques as shown in Table 1 and five real-life income datasets from Turkish banks to estimate and regulate the income for credit limits.

The rest of the paper is structured as follows: Section 1 provides introduction and literature review. Section 2 describes different regression methods tabulated in Table 1 and the performance measures used in this study. Section 3 briefly explains datasets and experimental set-

Table 1
Regression techniques.

Techniques	
Linear:	OLS
	Beta Regression
	Beta-OLS
	Box-Cox OLS
	Ridge Regression
	Robust Regression
Non-linear:	CART
	M5P
	MARS
	ANN
	LSSVM
	AdaBoost
	Random Forest
Linear + Non-linear:	OLS + CART
	OLS + M5P
	OLS + MARS
	OLS + ANN
	OLS + LSSVM

ups in regression techniques. Results of the estimation are discussed in Section 4. Finally, Section 5 concludes the paper and provides the directions for future research.

2. Regression techniques and performance measures

2.1. Applied techniques

When regression analysis is considered, the first regression that comes to mind is the linear regression. It is the first type of regression analysis to be studied rigorously and the most widely used of all statistical techniques in practical applications. This is because models that depend linearly on their unknown parameters are easier to fit and the statistical properties of the resulting estimators are easier to determine. A linear regression line has an equation of the form:

$$Y_i = \alpha + \beta_1 X_1 + \cdots + \beta_p X_p$$

where α is the intercept and the β_j are the slopes or coefficients.

The most common method for fitting a regression line is the method of ordinary least-squares (OLS). It is the simplest and thus the most common estimator. OLS calculates the best-fitting line for the observed data by minimizing the sum of the squares of the vertical deviations from each data point to the line.

The beta regression (BR) has become more popular in recent years, which was introduced by Ferrari and Cribari-Neto [16]. It is useful for modeling continuous variables "y" that assume values in the open standard unit interval (0, 1). BR is a parametric approach and performs the maximum likelihood estimation to produce a generalized linear model variant that allows for a beta-distributed dependent variable of density function:

$$f(y;\mu,\varphi) = \frac{\Gamma(\varphi)}{\Gamma(\mu\varphi)\Gamma((1-\mu)\varphi)} y^{\mu\varphi-1}(1-y)^{(1-\mu)\varphi-1}, \quad 0<y<1$$

where $\Gamma(.)$ denotes the gamma function; $0<\mu<1$ and $\phi>0$, with $a=\mu\phi$ and $b=(1-\mu)\phi$ are the usual parameterization of the beta density function and $\mu=E(Y)$. The parameter ϕ is related to the variance of the beta distribution, since $\mathrm{Var}(Y)=\mu(1-\mu)/(1+\phi)$. BR uses this aspect of the beta distribution to associate the explanatory variables with changes in mean and variance simultaneously.

The beta distribution is a flexible distribution that can produce a unimodal, uniform, or bimodal distribution of points that can be either symmetrical or skewed. Beta transformation with OLS (B-OLS) is an algorithm that fits a beta distribution to the dependent variable (income) before estimating an OLS model, based on which that variable is transformed to better meet the OLS normality assumption.

Box-Cox transformation [6] represents a family of power transformations to help researchers easily find the optimal normalizing transformation for each variable. It is designed for strictly positive responses and chooses the transformation to find the best fit to the data. It includes the logarithmic transformation as a limiting case:

$$BC(y;\lambda) = \begin{cases} \frac{y^\lambda - 1}{\lambda} & \text{if } \lambda \neq 0 \\ \log y & \text{if } \lambda = 0 \end{cases}$$

The parameter λ can be chosen by maximizing a log-likelihood function. To select an appropriate transformation, we need to try values of λ in a suitable range. Useful values of λ are often found to be in the range $(-2, 2)$. Unfortunately, the resulting models cannot be compared in terms of their residual sums of squares because these are in different units. We therefore use a likelihood criterion.

Ridge regression (Tikhonov regularization, method of regularization of ill-posed problems; [41]) involves the introduction of some bias into the regression equation in order to reduce the variance of the estimators of the parameters. When we have multiple regression and there is multicollinearity in the data, the OLS estimator performs "poorly." For this reason, we use ridge regression proposed by Hoerl and Kennard [22] which is a crude form of regularization. The regression coefficients can be estimated as

$$\beta = \left(X^T X + kI\right)^{-1} X^T y$$

where k is the ridge parameter and I is the identity matrix. The ridge parameter (k) plays a vital role to control the bias of the regression toward the mean of the response variable.

Robust regression limits the influence of outliers [20]. Barnett and Lewis [2] defined that outliers are observations that appear inconsistent with the rest of the data. In order to deal with this influential points, robust methods such as least trimmed squares (LTS), MM estimation, least absolute value method (LAV), S estimation, Huber M estimation and least median of squares (LMS) are used [24,31,32,44].

Classification and regression tree (CART) is a recursive partitioning method for predicting continuous dependent variables (regression) and categorical predictor variables (classification). It was first popularized by Breiman et al. [7] for fitting trees to data. CART is powerful because it deals with incomplete data, multiple types of features

(floats and unumerated sets) both in input features and predicted features, extremely resistant to outliers, and the trees it produces often contain rules which are readable by humans. Pseudocode for CART:

1. Start at the root node
2. Calculate the minimum of sum of squared errors between the observation and the mean in each node to split the tree (for regression). Gini index, entropy or twoing is used for classification.
3. Stop, if stopping criteria is reached and go to step 4. Otherwise go to step 2.
4. Prune to get right sized tree.

Multivariate adaptive regression splines (MARS) was introduced by Friedman in 1991. It is a nonparametric regression procedure that makes no assumption about the underlying functional relationship between the dependent and independent variables [18,33]. Instead, MARS constructs this relation from a set of coefficients and so-called basis functions that are entirely driven from the regression data. MARS divides the whole space of input variables into various sub-regions, each with its own regression equation. This equation relates input variables to output variables for each sub-region. This makes MARS particularly suitable for problems with higher input dimensions. MARS uses the following two-sided truncated power functions as spline basis functions [35]:

$$[-(x-t)]_+^q = \begin{cases} (t-x)^q & \text{if } x < t \\ 0 & \text{otherwise} \end{cases}$$

$$[+(x-t)]_+^q = \begin{cases} (x-t)^q & \text{if } x \geq t \\ 0 & \text{otherwise} \end{cases}$$

where the interface points between the pieces are called knots, t and q are the power to which splines are raised and determines the degree of smoothness of the resultant function estimate. The final MARS model has the following form:

$$\hat{y} = \hat{f}(x) = a_0 + \sum_{m=1}^{M} a_m B_m(x)$$

where y is the output variable, x is the input variable, a_0 is the coefficient of the constant term, M is the number of spline functions, and B_m and a_m are the mth spline function and its coefficient, respectively.

Final MARS model uses the following two stages: forward phase and backward phase algorithms. In the forward phase, basis functions introduced to the equation above until the maximum number of basis functions M_{max} was reached. The MARS model developed as such can have an overfitting problem because of large number of basis functions. For this reason, we use backward phase in order to prevent from overfitting. In this stage, redundant basis functions that made the least contributions are deleted. Generalized cross-validation (GCV) was used as the deletion criterion [13].

The M5 algorithm, the most commonly used classifier of model tree family, is an adaptation of a regression tree algorithm by Quinlan [29]. It can deal effectively with

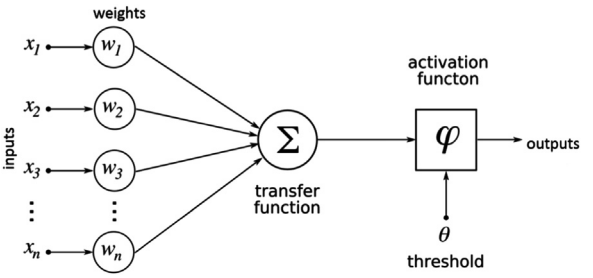

Fig. 1. Artificial neural network.

enumerated attributes and missing values. In order to build a model tree, using the M5 algorithm, we start with a set of training instances. The tree-based model is constructed by the divide-and-conquer method. It calculates a linear model (using linear regression) for each node of the generated tree. Then the splitted tree is pruned back from each leaf. M5' simplifies the tree by deleting the nodes of the linear models whose attributes do not increase the error. Finally, to avoid sharp discontinuities between the subtrees and to improve prediction accuracy of the tree-based model, a smoothing procedure is applied.

ANNs emerged after the introduction of simplified neurons by McCulloch and Pitts in 1943 [28]. It was inspired from the structure of biological neural networks and their way of encoding and solving problems. We usually refer to these artificial neurons as “perceptrons”. The first ANN called perceptron (Fig. 1) was invented in 1958 by psychologist Frank Rosenblatt [30].

There are two types of neural network: single-layer perceptron and multi-layer perceptron (MLP). MLP uses backpropagation for learning in training. The concept of back-propagation is that instead of using desired activities to train the hidden units, use error derivatives with respect to hidden activities. Each hidden activity can affect many output units and can therefore have many separate effects on the error. These effects must be combined and the error derivatives can be computed for all hidden units efficiently at the same time. Once we have the error derivatives for the hidden activities, it is easy to get the error derivatives for the weights going into a hidden unit.

LS-SVM models are an alternate formulation of SVM regression [38] proposed by Vapnik and Lerner [42]. The standard SVM classifier of Vapnik is modified for transforming the quadratic programming (QP) problem to a linear system, more precisely a Karush–Kuhn–Tucker (KKT) system, which leads to solving linear KKT in a least squares sense. Another advantage of the LS-SVM formulation is that it involves fewer tuning parameters and avoids the problem of local minima in SVM. It has been successfully applied for solving various problems in engineering [3,34,43].

These modifications are formulated as

$$\hat{y}(x) = \omega^T \varphi(x) + b$$

where $\phi(x)$ is a function that maps the input space into a higher dimensional feature space, x is the M-dimensional vector of inputs x_j, and ω and b are the parameters of the model.

LS-SVM for function estimation formulates the following optimization:

$$Minimize: \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w}+\gamma\frac{1}{2}\sum_{k=1}^{N}e_k^2$$

$$Subject\ to: y(x)=\boldsymbol{w}^T\varphi(x_k)+b+e_k, k=1,\ldots,N$$

where e_k is an error variable and γ is a regularization parameter. Solving this optimization problem in dual space leads to find the α_i and b coefficients in the following solution:

$$y(x)=\sum_{k=1}^{N}\alpha_k K(x,x_k)+b$$

where $K(x, x_k)$ is the kernel function defined as the dot product between the $\phi(x)^T$ and $\phi(x)$ mappings.

AdaBoost and Random Forest (RF) are ensemble learning methods for classification and regression. Both of them are divide-and-conquer approaches that are used to improve the performance, and the main principle of these algorithms is that a group of "weak learners" can come together to form a "strong learner." AdaBoost, formulated by Freund and Schapire [17], is sensitive to noisy data and outliers, where RF is proposed by Breiman [8], which adds an additional layer of randomness to bagging. AdaBoost works by choosing a base learner and assigns a weight coefficient to it. Initially, all the weights are set equally but on each iteration, the weights of incorrectly classified examples are increased, so that the weak learner is forced to focus on the hard examples in the training set. The smaller the error of the weak learner, the larger is its weight in the final model. As a result, final model is decided by a weighted sum of the base algorithms as training iteration proceeds. Original AdaBoost algorithm resolves classification problems, but to deal with regression, we changed the exponential loss function to least square loss function where it yields a new model of boosting,-LSBoost. RF starts with a standard machine learning technique called "decision tree," where trees are weak learners and the RF is a strong learner. In an RF, the split that is chosen is not necessarily the best split among all variables. Instead, if we have N variables, n variables are selected randomly, where $n \ll N$ is selected at each node and the best split on these n variables is used to split the node. This strategy enables RF to achieve accurate prediction by decreasing variance, while boosting achieves this by decreasing bias.

Table 1, provides the list of regression methods that are used in our study. So far, we have explained one-stage linear and non-linear techniques. In the two-stage models, non-linear one-stage models are combined with OLS. The purpose of these two-stage techniques is to combine the good comprehensibility of OLS with the predictive power of a non-linear model. In the first stage, a linear regression is built with OLS and in the second stage, the residuals of this linear regression are estimated using a non-linear regression technique. This estimate of the residuals is then added to the OLS estimate to obtain a more accurate prediction of income. Finally, all these algorithms were coded in MATLAB programming language.

2.2. Performance measures

The performance of prediction models can be assessed using various methods and metrics. Performance measure helps us decide whether one algorithm is better or worse than the other and measures how well a data mining algorithm performs on a given dataset. We have two different metric categories: calibration and discrimination, as seen in Table 2. Calibration is a measure of how well the predicted values agree with the actual observed values, while discrimination is the ability of the model to correctly

Table 2
Performance measures.

Metric	Descriptions	Type
RMSE	Root-mean-square error measures the differences between values predicted by a model and the values actually observed.	Calibration
MAE	Mean absolute error is a quantity used to measure how close forecasts or predictions are to the eventual outcome.	Calibration
AUC	The area under the curve (AUC) of a receiver operating characteristic (ROC) curve is a graphical plot which illustrates the performance of a binary classifier system as its discrimination threshold is varied. Which is used here to plot how good the model is at distinguishing values that are higher than average from those that are lower than average.	Discrimination
AOC	REC curves [4] in order to select a good threshold value, so that only residuals greater than that value are considered as errors. The REC curves facilitate visual comparison of regression functions and they are qualitatively invariant to choices of error metrics and scaling of the residual.	Calibration
RSquare	R-square is the square of the correlation between the response values and the predicted response values. It is also called the square of the multiple correlation coefficients and the coefficient of multiple determinations.	Calibration
Pearson's *r*	Pearson correlation coefficient is a measure of the strength of a linear association between two variables and is denoted by *r*.	Discrimination
Spearman's ρ	Spearman's rho is a statistical measure of the strength of a monotonic relationship between paired data.	Discrimination
Kendall's τ	Kendall's tau provides a distribution free test of independence and a measure of the strength of dependence between two variables.	Discrimination
Hit rate	Hit rate is a term used to describe the success rate of an effort or it is a ratio or a proportion. This rate specifically compares the number of times an initiative was successful against the number of times it was attempted. In our study, we use hit rate to measure how successful are our algorithms to make an approximate prediction of incomes. In other words, if the predicted income is within $\pm\alpha$ percent of actual income, we will consider it as a hit.	Calibration
Preciseness	Preciseness provides a quality or state of being very accurate. It measures success of algorithm; does our predicted income is within acceptable interval or it overestimates or underestimates the real income. In other words, level of success in shooting accurately at targets in a given range.	Calibration

separate the subjects into different groups or the ability to provide an ordinal ranking of the dependent variable. Ranges of RMSE and MAE are from 0 to infinity with 0 being the perfect score. AUC and AOC can range between 0 and 1. In AUC higher values show how good the model is, whereas in AOC lower values are better. In general, R^2 or coefficient of determination takes a number on the scale between 0 and 1, where 1 is perfect positive correlation and 0 no correlation. Discrimination metrics such as Pearson, Kendall and Spearman range from -1 to 1. Correlation is 1 if the agreement between the two rankings is perfect, -1 if the disagreement is perfect, and 0 if there is no correlation. Hit rate and preciseness are in calibration category in evaluating model performance. Both of them range from 0 to 1, higher values show how good the model makes predictions (ratio of the number of predicted hits to the number of observed).

3. Datasets and settings

3.1. Dataset and preprocessing

The number of dataset entries are given in Table 3, where we have five real-life datasets from Turkish banks. There are 10,000 observations in each dataset except for Dataset 3. The number of available input variables ranges from 10 to 15 and all of them are continuous. Each dataset is randomly shuffled and separated into 30% test set and 70% train set. The training data are used to build the model by pairing the inputs with the expected output. The test data are used to estimate how good models have been trained and to estimate the model properties or to assess the prediction performance of the models. Also, we applied log transformation to our dependent variable (income) in order to remove skewness.

After some data preprocessing, variable selection procedure was implemented in order to eliminate redundant or irrelevant features. The objectives of attribute selection are to provide more cost-effective predictors and to improve the performance of regression model. In our study, variables in all datasets were similar, and we used stepwise selection and in some cases expert opinion in order to decrease the number of attributes. Stepwise selection is a combination of backward elimination and forward selection. It is a method that allows moves in either direction, dropping or adding variables at various steps.

Pseudocode of Stepwise selection:

1. Start with no variables in the model
2. Add variable with the largest *F*-statistics
3. Refit with this variable added. Recompute all *F*-statistics for adding one of the remaining variables and add the variable with the largest *F*-statistic
4. Do backward elimination
5. After backward steps, do forward selection
6. Continue until no remaining variable is significant at cut-off level

3.2. Parameter settings

Regression techniques used in our study are mostly similar to that of Loterman et al. [27]. Yet, their research targets the LGD prediction, whereas our research aims to predict the income of each customer. As an additional measure to Loterman et al. [27], in our study, we implemented M5P model tree with two-stage OLS combination technique. Also, new performance measures as hit rate and preciseness were used to show the efficiency of regression algorithms.

OLS does not need any parameter tunings, and we used it as a traditional linear model. But in BR and B-OLS, dependent variable should be in the standard unit interval (0,1). For this reason, before estimation, dependent variable is divided by the maximum income to fit between (0,1) interval and multiplied back to predicted values (after estimation) to get exact income for the given instance. The BC-OLS is a member of a class of transformations called power transformations that raise dependent variable to an exponent (power). The values for power parameter that we used in this model varied over a range from -2 to 2, and optimal value was chosen on the basis of a maximum likelihood criterion.

One of the main obstacles in using RiR is choosing an appropriate value of k to the diagonal elements of the correlation matrix. By using least squares coefficients, we first obtained the value of k, and then implemented Hoerl and Kennard [22] iterative method for selecting the best k value. Also, ridge parameter was tuned by 10-fold cross validation on the training set and mean squared error (the variance plus the bias squared) is used as the selection criterion. RoR works by assigning a weight to each data point, as a weight function “bisquare” is chosen with its parameter set to $w=4.685$ (default constant value).

M5P is a tree-based piecewise linear model or a regression-based decision tree algorithm, based on the M5 algorithm by Quinlan [29]. In this research, we used M5PrimeLab matlab toolbox written by Jekabsons [25], where toolbox allows building regression trees and model trees using M5 method. M5PrimeLab accepts input variables to be continuous, binary, and categorical, as well as deals with the missing values. To predict appropriate output values for the dependent variable (income) by using this toolbox, we played with parameters of the M5 method. Minimum number of training data cases for one node was increased with respect to training set size (0.001*size of training data). Also, we performed smoothing with smoothing coefficient taken as 10*(0.001*size of training). In order to implement MARS in our study, we operated ARESLab written by Jekabsons [26] as in M5 model. ARESLab is a matlab toolbox for building piecewise-linear and piecewise-cubic regression models using Friedmans [18], 1993 MARS technique. In this method, we used default parameter values without any modification inside the ARESLab toolbox in order to predict the output values.

CART is one of the famous algorithms in the field of data mining and machine learning. It follows the divide and conquer strategy, and it can handle missing values. For CART model, the training set is further split into training

and validation subsets. Validation set is used to select the criterion for evaluating candidate splitting rules (pruning) based on the mean squared error and training set was used to build a regression tree. In ANN, we again split training dataset into a validation set and a training set. To select the target layer activation function (logistic, linear, exponential, reciprocal, square etc.) and determine the hidden neurons (from 1 to 20 is considered), the validation set with mean squared error was implemented.

Table 3
Dataset characteristics.

Dataset	Inputs	Total size	Training	Test
Dataset 1	10	10,000	7000	3000
Dataset 2	10	10,000	7000	3000
Dataset 3	15	13,018	9100	3918
Dataset 4	13	10,000	7000	3000
Dataset 5	12	10,000	7000	3000

In AdaBoost and RF, we used ready toolboxes in matlab; that is, fitensemble and TreeBagger, respectively. Fitensemble is used to create an ensemble model that predicts responses to data where it has some input arguments. We used default parameter values except method type, because we needed to change it to LSBoost in order to use it in our regression prediction. TreeBagger bags an ensemble of decision trees for either classification or regression. In this method, as in fitensemble, we need to build an ensemble of regression trees by setting the optional input argument "method" to "regression." Also, we set the number of trees to 50 for predicting the output as a function of inputs.

And finally, we used LS-SVMlab matlab toolbox written by Suykens et al. [38], which is a reformulation to standard SVM. LS-SVM model is a class of kernel-based learning method. This toolbox contains different LS-SVM techniques such as regression, classification, unsupervised learning, and time-series prediction. Our study focuses on regression, so that we can obtain regression LS-SVM algorithm with the radial basis function (RBF) kernel

Table 4
Results for Dataset 1.

	Low income			Middle income			High income			Total		
	ACC	OE	UE	ACC	OE	UE	ACC	OE	UE	ACC	OE	UE
OLS	0,00%	100,00%	0,00%	25,27%	65,06%	9,67%	21,05%	14,04%	64,91%	21,32%	55,42%	23,25%
BetaReg	0,00%	100,00%	0,00%	0,00%	100,00%	0,00%	31,58%	29,70%	38,72%	8,38%	81,34%	10,28%
Beta/OLS	0,00%	100,00%	0,00%	9,03%	90,97%	0,00%	28,95%	6,27%	64,79%	13,31%	69,49%	17,20%
BoxCox	0,00%	100,00%	0,00%	14,90%	83,49%	1,60%	25,31%	10,53%	64,16%	16,00%	65,97%	18,03%
Ridge	0,00%	100,00%	0,00%	25,28%	62,88%	11,85%	21,56%	14,75%	63,68%	21,41%	54,59%	24,01%
Robust	0,00%	100,00%	0,00%	23,96%	64,99%	11,05%	22,22%	15,95%	61,83%	20,50%	56,99%	22,52%
CART	0,00%	100,00%	0,00%	25,03%	61,50%	13,47%	17,65%	17,14%	65,21%	20,14%	54,68%	25,18%
M5P	0,00%	100,00%	0,00%	33,33%	51,44%	15,22%	20,18%	17,92%	61,90%	26,11%	47,97%	25,91%
MARS	0,00%	100,00%	0,00%	29,75%	56,30%	13,94%	22,68%	17,79%	59,52%	24,55%	50,96%	24,48%
LSSVM	0,00%	100,00%	0,00%	28,95%	59,29%	11,75%	22,56%	16,67%	60,78%	24,02%	52,53%	23,45%
ANN	0,00%	100,00%	0,00%	27,99%	60,52%	11,49%	17,54%	15,41%	67,04%	22,09%	52,96%	24,95%
OLS+M5P	0,30%	99,70%	0,00%	33,76%	50,00%	16,24%	21,43%	17,04%	61,53%	26,75%	46,81%	26,45%
OLS+MARS	0,89%	99,11%	0,00%	29,54%	57,05%	13,41%	23,18%	17,04%	59,77%	24,65%	51,13%	24,22%
OLS+CART	0,00%	100,00%	0,00%	28,10%	59,99%	11,91%	20,18%	16,42%	63,41%	22,85%	52,89%	24,25%
OLS+ANN	35,42%	61,01%	3,57%	32,16%	32,43%	35,42%	20,43%	18,17%	61,40%	29,41%	31,84%	38,76%
OLS+LSSVM	0,00%	100,00%	0,00%	33,17%	52,62%	14,21%	22,31%	18,30%	59,40%	26,58%	48,80%	24,62%
AdaBoost	11,48%	88,52%	0,00%	48,07%	36,92%	15,01%	23,35%	8,83%	67,81%	33,34%	43,69%	22,96%
Random Forest	17,28%	82,72%	0,00%	48,83%	37,05%	14,12%	25,60%	9,58%	64,82%	**35,71%**	42,46%	21,83%

	MAE	RMSE	AUC	AOC	R^2	Pearson	Kendall	Spearman	Hit 15%	Hit 30%	Hit 50%
OLS	898,564	1874,445	0,845	3365932,5	0,48	0,696	0,433	0,604	0,21	0,48	0,79
BetaReg	1695,777	2494,267	0,837	5016817,6	0,186	0,664	0,421	0,59	0,1	0,2	0,39
Beta/OLS	1190,394	2234,715	0,836	4784457,2	0,26	0,66	0,419	0,588	0,14	0,29	0,58
BoxCox	1024,011	1991,183	0,844	3791102	0,313	0,652	0,431	0,602	0,16	0,35	0,6
Ridge	871,843	1932,574	0,848	3496951,4	0,47	0,684	0,433	0,604	0,22	0,49	0,81
Robust	865,55	1855,757	0,856	3220477,7	0,478	0,694	0,438	0,608	0,21	0,47	0,8
CART	1035,446	2472,655	0,803	3808702,2	0,324	0,602	0,446	0,564	0,2	0,43	0,77
M5P	819,324	1787,96	0,873	3054610,5	0,527	0,726	0,47	0,649	0,27	0,55	0,81
MARS	831,167	1764,289	0,863	2986928,7	0,539	0,734	0,455	0,628	0,26	0,53	0,81
LSSVM	847,606	1833,852	0,863	3251637,6	0,502	0,708	0,463	0,639	0,25	0,52	0,81
ANN	871,648	1936,543	0,858	3598517,5	0,445	0,668	0,453	0,629	0,23	0,51	0,78
OLS+M5P	810,142	1772,471	0,877	3012730,5	0,535	0,732	0,473	0,653	0,28	0,55	0,81
OLS+MARS	827,139	1749,843	0,861	2943744,9	0,547	0,74	0,455	0,629	0,25	0,53	0,81
OLS+CART	885,764	1875,887	0,845	3370771,2	0,479	0,693	0,433	0,604	0,23	0,5	0,79
OLS+ANN	857,907	2145,871	0,868	4115697,7	0,318	0,667	0,464	0,641	0,29	0,53	0,8
OLS+LSSVM	817,296	1743,766	0,873	2901939,4	**0,55**	0,741	0,471	0,65	0,27	0,55	0,81
AdaBoost	826,146	1610,3	0,8339	2829570	0,316	0,664	0,394	0,558	0,30	0,59	0,82
Random Forest	810,222	1608,6	0,8389	2811670	0,318	0,677	0,416	0,590	**0,31**	**0,60**	**0,83**

because of its good overall performance [1]. Also, we tuned regularization parameter γ and squared kernel function parameter σ^2.

4. Results and discussion

The performance of the resulting techniques is measured on the test sets based on the 10 performance metrics (Table 3). Model performance results for 18 regression algorithms obtained from five real-life datasets are shown in Tables 4–8. The underlined and bold metric values indicate which technique performs well with respect to which measure. Note that differences in the type of dataset, number of observations and availability of independent variables are the probable causes of the observed variability in the actual performance levels between the five different datasets.

Considering Tables 4–8, linear models built by OLS, RiR and RoR showed similar performances in all the 10 metrics. In contrast, transformed linear models such as BR, B-OLS and BC-OLS performed much worse than original linear techniques, even though these approaches are specifically designed to cope with a violation of the OLS normality assumption.

According to our results, non-linear techniques such as MARS, RF, LS-SVM, and M5P outperformed the linear models in all the datasets except CART, while CART demonstrated worse performance in accordance with other non-linear models such as ANN, M5P, and MARS. Being better than linear techniques explains that income (dependent variable) and independent variables in the datasets are correlated non-linearly.

Two-stage models were observed by the combination of the linear and non-linear models. The prediction performance tends to increase to a level that is very close to the level of the corresponding one-stage non-linear technique in all the datasets. Note that this modeling method has also been applied successfully in LGD [27], where also two-stage algorithms increased the model performance.

In this research, in order to compare our model performances, we concentrated on R^2, *hit rate,* and *preciseness*

Table 5
Results for Dataset 2.

	Low income			Middle income			High income			Total		
	ACC	OE	UE	ACC	OE	UE	ACC	OE	UE	ACC	OE	UE
OLS	19,67%	79,51%	0,82%	32,80%	41,13%	26,07%	24,48%	24,48%	51,05%	28,95%	41,30%	29,75%
BetaReg	0,00%	100,00%	0,00%	0,05%	99,95%	0,00%	41,94%	17,34%	40,71%	11,38%	77,60%	11,01%
Beta/OLS	0,00%	100,00%	0,00%	19,72%	78,53%	1,75%	22,14%	4,55%	73,31%	17,97%	61,13%	20,90%
BoxCox	0,00%	100,00%	0,00%	35,38%	41,02%	23,60%	2,83%	0,37%	96,80%	22,26%	37,20%	40,53%
Ridge	21,82%	71,27%	6,91%	27,72%	40,94%	31,34%	26,18%	23,75%	50,06%	26,61%	40,11%	33,28%
Robust	20,54%	75,41%	4,05%	31,26%	39,73%	29,02%	23,21%	24,62%	52,18%	27,82%	40,20%	31,98%
CART	0,00%	100,00%	0,00%	31,13%	53,59%	15,29%	19,76%	19,11%	61,13%	24,34%	50,26%	25,40%
M5P	0,27%	99,73%	0,00%	35,60%	47,97%	16,43%	24,48%	19,56%	55,97%	28,29%	46,59%	25,12%
MARS	6,56%	93,44%	0,00%	35,98%	45,84%	18,18%	21,89%	21,16%	56,95%	28,59%	44,96%	26,46%
LSSVM	0,00%	100,00%	0,00%	32,48%	51,15%	16,37%	22,76%	21,40%	55,84%	25,89%	49,05%	25,06%
ANN	3,28%	96,72%	0,00%	34,50%	46,00%	19,50%	23,86%	18,33%	57,81%	27,82%	44,69%	27,49%
OLS+M5P	2,19%	97,81%	0,00%	34,94%	48,19%	16,87%	23,62%	20,30%	56,09%	27,89%	46,69%	25,42%
OLS+MARS	9,02%	90,98%	0,00%	35,76%	45,40%	18,84%	23,25%	21,28%	55,47%	29,12%	44,43%	26,46%
OLS+CART	0,86%	99,14%	0,00%	29,50%	36,86%	33,65%	29,42%	26,89%	45,69%	26,21%	40,20%	33,59%
OLS+ANN	3,28%	96,72%	0,00%	30,78%	56,08%	13,14%	22,88%	19,19%	57,93%	25,29%	51,05%	23,66%
OLS+LSSVM	5,19%	94,81%	0,00%	34,23%	49,40%	16,37%	24,35%	20,05%	55,60%	28,02%	46,99%	24,99%
AdaBoost	29,62%	63,45%	6,93%	31,13%	28,30%	40,57%	24,30%	22,14%	53,56%	28,88%	44,11%	27,00%
Random Forest	26,82%	69,19%	3,99%	34,43%	29,60%	35,97%	28,33%	21,05%	50,62%	**29,36%**	47,06%	23,58%

	MAE	RMSE	AUC	AOC	R^2	Pearson	Kendall	Spearman	Hit 15%	Hit 30%	Hit 50%
OLS	846,963	2122,255	0,877	4209273,6	0,521	0,73	0,472	0,652	0,29	0,55	0,78
BetaReg	1469,658	2824,984	0,878	5629127,5	0,152	0,717	0,476	0,656	0,12	0,25	0,47
Beta/OLS	1034,445	2644,592	0,878	4643973	0,257	0,715	0,475	0,656	0,18	0,38	0,72
BoxCox	988,608	2801,384	0,862	4703686,7	0,242	0,727	0,475	0,656	0,23	0,47	0,73
Ridge	830,314	2130,73	0,875	4115059,6	0,523	0,732	0,468	0,649	0,27	0,52	0,76
Robust	833,658	2108,917	0,871	4044522,2	0,53	0,728	0,464	0,643	0,28	0,53	0,77
CART	925,962	2279,768	0,851	4880525,7	0,353	0,638	0,457	0,597	0,25	0,5	0,77
M5P	814,055	2131,16	0,884	4265256,6	0,517	0,729	0,488	0,672	0,29	0,57	0,83
MARS	829,948	2100,448	0,877	4125658,6	0,531	0,737	0,475	0,655	0,29	0,56	0,81
LSSVM	864,177	2306,448	0,876	5031735,3	0,434	0,668	0,471	0,652	0,27	0,54	0,81
ANN	858,964	2191,112	0,862	4551941,7	0,49	0,705	0,455	0,633	0,28	0,55	0,81
OLS+M5P	815,989	2113,16	0,884	4185107,9	0,525	0,737	0,488	0,672	0,29	0,57	0,82
OLS+MARS	827,923	2101,164	0,878	4127626,4	**0,531**	0,738	0,476	0,656	0,3	0,56	0,81
OLS+CART	893,128	2109,845	0,855	4853864,6	0,372	0,658	0,462	0,593	0,28	0,55	0,82
OLS+ANN	846,459	2124,789	0,868	4214764,2	0,52	0,739	0,46	0,639	0,26	0,53	0,81
OLS+LSSVM	818,773	2110,477	0,883	4158552	0,527	0,74	0,482	0,665	0,29	0,55	0,82
AdaBoost	879,156	2085	0,862	4160900	0,442	0,674	0,484	0,662	0,29	0,55	0,80
Random Forest	884,806	2073,6	0,881	3954500	0,506	0,712	0,516	0,704	**0,30**	**0,59**	**0,84**

Table 6
Results for Dataset 3.

	Low income			Middle income			High income			Total		
	ACC	OE	UE	ACC	OE	UE	ACC	OE	UE	ACC	OE	UE
OLS	4,13%	95,87%	0,00%	54,17%	26,18%	19,65%	20,00%	7,86%	72,14%	47,24%	31,22%	21,54%
BetaReg	34,71%	62,53%	2,75%	17,56%	1,25%	81,19%	1,07%	0,00%	98,93%	17,96%	6,73%	75,31%
Beta/OLS	9,09%	90,91%	0,00%	51,43%	17,89%	30,68%	12,50%	1,79%	85,71%	44,86%	23,39%	31,75%
BoxCox	0,00%	100,00%	0,00%	46,90%	2,98%	50,12%	0,00%	0,00%	100,00%	39,35%	11,58%	49,06%
Ridge	3,43%	96,57%	0,00%	54,82%	25,35%	19,83%	21,21%	7,58%	71,21%	47,96%	30,52%	21,52%
Robust	4,78%	95,22%	0,00%	55,18%	25,25%	19,57%	23,02%	10,94%	66,04%	48,35%	30,70%	20,95%
CART	0,00%	100,00%	0,00%	52,77%	29,76%	17,47%	12,14%	3,21%	84,64%	45,13%	34,28%	20,59%
M5P	4,41%	95,59%	0,00%	55,07%	28,00%	16,94%	14,64%	5,36%	80,00%	47,64%	32,55%	19,81%
MARS	3,86%	96,14%	0,00%	54,89%	26,54%	18,57%	18,93%	7,86%	73,21%	47,74%	31,55%	20,72%
LSSVM	1,38%	98,62%	0,00%	56,68%	26,21%	17,11%	14,29%	5,00%	80,71%	**48,69%**	31,30%	20,02%
ANN	3,03%	96,97%	0,00%	55,10%	28,00%	16,91%	17,14%	5,36%	77,50%	47,71%	32,67%	19,61%
OLS+M5P	3,86%	96,14%	0,00%	55,78%	26,92%	17,29%	16,79%	5,00%	78,21%	48,34%	31,67%	19,99%
OLS+MARS	4,41%	95,59%	0,00%	54,47%	26,06%	19,47%	20,71%	7,50%	71,79%	47,56%	31,07%	21,37%
OLS+CART	0,00%	100,00%	0,00%	53,53%	27,77%	18,71%	12,18%	4,80%	83,03%	45,60%	32,94%	21,46%
OLS+ANN	5,51%	94,21%	0,28%	52,39%	26,98%	20,63%	15,36%	3,93%	80,71%	45,53%	31,47%	22,99%
OLS+LSSVM	4,68%	95,32%	0,00%	55,40%	27,19%	17,41%	19,64%	6,07%	74,29%	48,29%	31,90%	19,81%
AdaBoost	4.22%	95.78%	0.00%	54.01%	26.84%	19.15%	22.22%	6.61%	71.17%	47.11%	30.95%	21.94%
Random Forest	6.02%	93.98%	0.00%	55.79%	28.55%	15.66%	20.12%	4.80%	75.08%	48.56%	32.07%	19.37%

	MAE	RMSE	AUC	AOC	R^2	Pearson	Kendall	Spearman	Hit 15%	Hit 30%	Hit 50%
OLS	1191,496	2238,54	0,761	4745130,7	0,323	0,568	0,373	0,527	0,47	0,76	0,92
BetaReg	1750,304	2854,072	0,76	5829940,3	0,101	0,553	0,373	0,527	0,16	0,36	0,63
Beta/OLS	1293,31	2496,098	0,763	5785629,4	0,228	0,553	0,373	0,528	0,4	0,71	0,9
BoxCox	1452,764	2821,03	0,763	5931071,3	0,075	0,549	0,376	0,521	0,36	0,61	0,81
Ridge	1187,354	2331,313	0,774	4965719,9	0,294	0,543	0,389	0,548	0,47	0,76	0,93
Robust	1160,431	2306,595	0,775	4847947,1	0,291	0,54	0,392	0,553	0,48	0,76	0,92
CART	1250,399	2369,499	0,709	5290987,8	0,241	0,491	0,322	0,417	0,46	0,76	0,91
M5P	1153,814	2189,08	0,761	4519293,2	0,353	0,599	0,376	0,529	0,48	0,77	0,93
MARS	1193,219	2248,466	0,757	4781108,7	0,317	0,563	0,371	0,523	0,47	0,76	0,93
LSSVM	1162,148	2220,5	0,761	4640370,3	0,334	0,583	0,373	0,527	0,49	0,77	0,93
ANN	1164,726	2188,129	0,753	4511412,6	0,353	0,594	0,366	0,517	0,48	0,77	0,91
OLS+M5P	1148,568	2179,251	0,766	4481980,4	**0,358**	0,601	0,381	0,537	0,49	0,77	0,93
OLS+MARS	1188,062	2228,735	0,762	4701807,2	0,329	0,574	0,375	0,53	0,47	0,76	0,93
OLS+CART	1243,283	2469,738	0,709	5672656,3	0,24	0,493	0,335	0,433	0,45	0,73	0,88
OLS+ANN	1217,45	2236,619	0,746	4707303,6	0,324	0,571	0,349	0,496	0,46	0,75	0,92
OLS+LSSVM	1154,621	2217,089	0,766	4646490,8	0,336	0,582	0,384	0,541	0,49	0,77	0,93
AdaBoost	1169,1	2288,9	0,763	4917500	0,330	0,575	0,377	0,532	0,48	0,77	0,93
Random Forest	1119,3	2185,3	0,767	4478400	0,349	0,593	0,393	0,550	**0,49**	**0,79**	**0,93**

(Table 2). As can be noticed from Tables 4–8, other measures such as MAE, AUC, and Pearson, are related to R^2 and this is an indication of high correlation between mentioned metrics. Accordingly, the highest performance values for the given metrics are in non-linear and two-stage models, where two-stage models significantly outperformed most of the non-linear techniques in income prediction.

From Tables 4–8, we find that the results for hit rate and precision of R^2 do not always tend in the same direction in datasets. Hence, if a technique is bad in R^2, it can perform good in hit rate or preciseness. Therefore, R^2 values may not always be explanatory, and it depends on business objectives or needs. Good R^2 does not indicate whether a regression model is adequate for a given dataset. One can have a low R^2 value for a good model, or a high R^2 value for a model that does not fit the data. For this reason, using additional performance measures will be better for making decisions.

Another important point is the values of hit rate and preciseness, where we calculated each regression technique's predicted values by using preciseness with $\alpha=0.15$ and hit rates with $\alpha=0.15$, 0.3, 0.5. For our needs, the hit rate and preciseness are very important in terms of economy. For example, in credit card income prediction, we predicted customers' income and decided to set their credit limit according to the predicted values. After a while some of the customers who use their credit limit may not pay back on time because of their low income or other causes. In this situation, as a bank, we lose customers and liquid money or cash. For this reason, these metrics are important to determine the techniques, that most accurately predict the income of the customers. The higher the hit rate and preciseness, the better a bank is at making good decisions. As shown in Tables 4–8, the highest hit rates and precisenesses were detected on non-linear and two-stage models as in R^2. We also find that preciseness and hit rate for same α (tolerance) gives approximately similar results. Yet, we need to note that these two measures are different from each other. Difference between the hit rate and preciseness is that hit rate measures how successful our regression models are in making an approximate prediction of incomes, whereas preciseness measures how accurate the outcome of targets are in a

Table 7
Results for Dataset 4.

	Low income			Middle income			High income			Total		
	ACC	OE	UE	ACC	OE	UE	ACC	OE	UE	ACC	OE	UE
OLS	6,42%	93,21%	0,38%	55,45%	24,84%	19,71%	16,75%	7,61%	75,63%	48,61%	29,73%	21,67%
BetaReg	0,75%	99,25%	0,00%	54,94%	28,84%	16,22%	15,23%	0,51%	84,26%	47,27%	33,18%	19,55%
Beta/OLS	0,00%	100,00%	0,00%	54,19%	23,32%	22,49%	13,71%	8,63%	77,66%	47,75%	31,96%	20,29%
BoxCox	0,00%	100,00%	0,00%	53,19%	23,32%	23,49%	11,50%	8,00%	82,50%	46,81%	32,26%	20,94%
Ridge	5,68%	93,94%	0,38%	55,55%	24,92%	19,53%	23,00%	8,50%	68,50%	48,84%	30,01%	21,15%
Robust	2,95%	97,05%	0,00%	55,52%	26,20%	18,28%	15,79%	8,61%	75,60%	47,95%	31,41%	20,64%
CART	0,00%	100,00%	0,00%	54,51%	26,21%	19,28%	13,20%	6,60%	80,20%	47,01%	31,42%	21,57%
M5P	3,40%	96,60%	0,00%	57,60%	25,51%	16,89%	15,74%	6,60%	77,66%	**50,10%**	30,52%	19,38%
MARS	5,28%	94,72%	0,00%	56,03%	24,45%	19,51%	14,72%	8,63%	76,65%	48,87%	29,60%	21,53%
LSSVM	0,75%	99,25%	0,00%	57,76%	25,08%	17,16%	16,75%	5,08%	78,17%	50,07%	30,29%	19,64%
ANN	7,17%	92,08%	0,75%	54,19%	23,32%	22,49%	13,71%	8,12%	78,17%	47,41%	28,37%	24,22%
OLS+M5P	4,53%	95,47%	0,00%	56,74%	25,43%	17,83%	16,75%	5,58%	77,66%	49,54%	30,29%	20,17%
OLS+MARS	5,28%	94,72%	0,00%	56,19%	24,88%	18,93%	13,71%	8,63%	77,66%	48,94%	29,96%	21,10%
OLS+CART	3,79%	96,21%	0,00%	55,47%	27,11%	17,42%	10,50%	7,00%	82,50%	47,75%	31,96%	20,29%
OLS+ANN	7,92%	92,08%	0,00%	55,76%	25,67%	18,57%	13,71%	6,60%	79,70%	48,81%	30,26%	20,94%
OLS+LSSVM	6,79%	92,83%	0,38%	56,90%	25,47%	17,63%	17,26%	5,58%	77,16%	49,90%	30,09%	20,01%
AdaBoost	3,31%	96,69%	0,00%	53,33%	26,85%	19,82%	18,64%	9,47%	71,89%	45,78%	31,77%	22,45%
Random Forest	5,23%	94,77%	0,00%	55,66%	28,11%	16,23%	19,23%	4,73%	76,04%	47,93%	32,22%	19,85%

	MAE	RMSE	AUC	AOC	R^2	Pearson	Kendall	Spearman	Hit 15%	Hit 30%	Hit 50%
OLS	1132,884	2074,93	0,765	4058091	0,351	0,593	0,379	0,535	0,49	0,78	0,93
BetaReg	1548,152	2248,152	0,759	4301648,4	0,205	0,579	0,371	0,525	0,48	0,77	0,92
Beta/OLS	1335,499	2161,635	0,759	4300547,8	0,226	0,579	0,372	0,525	0,49	0,77	0,92
BoxCox	1563,169	2433,884	0,731	5530689,6	0,108	0,544	0,37	0,505	0,47	0,77	0,91
Ridge	1197,717	2166,951	0,761	3988007,2	0,347	0,629	0,381	0,536	0,49	0,77	0,93
Robust	1189,949	2101,444	0,763	4824412,4	0,316	0,567	0,381	0,538	0,48	0,78	0,93
CART	1203,599	2245,388	0,722	4800274,4	0,24	0,535	0,342	0,46	0,47	0,77	0,91
M5P	1102,301	2034,4	0,764	3907420,4	0,376	0,619	0,38	0,534	0,5	0,79	0,93
MARS	1136,453	2077,425	0,756	4075165,8	0,35	0,592	0,372	0,525	0,49	0,77	0,93
LSSVM	1121,177	2080,17	0,76	4099919,4	0,348	0,597	0,373	0,527	**0,51**	**0,78**	**0,93**
ANN	1170,616	2076,795	0,743	4064497,4	0,35	0,609	0,355	0,502	0,47	0,76	0,92
OLS+M5P	1102,344	2025,033	0,765	3875388,1	**0,382**	0,624	0,382	0,538	0,5	0,79	0,93
OLS+MARS	1131,455	2067,313	0,761	4031257,6	0,356	0,598	0,375	0,529	0,49	0,78	0,93
OLS+CART	1144,014	2282,591	0,71	4459847,1	0,287	0,577	0,325	0,443	0,48	0,77	0,91
OLS+ANN	1129,758	2032,852	0,757	3906104,1	0,377	0,617	0,373	0,527	0,49	0,79	0,93
OLS+LSSVM	1099,978	2060,151	0,768	3993901,3	0,361	0,605	0,388	0,546	0,5	0,79	0,93
AdaBoost	1159	2106,1	0,754	4197900	0,323	0,568	0,375	0,531	0,46	0,76	0,93
Random Forest	1115,6	2025,5	0,759	3891100	0,374	0,614	0,383	0,539	0,48	0,78	0,93

given range. Furthermore, hit rate is a term that is used to describe the success rate of an effort that compares the number of times an initiative was successful against the number of times it was attempted.

In each of Tables 4–8, the results of preciseness for our customers were divided into three groups according to their incomes as follows: low income (LI), middle income (MI), and high income (HI). Also, in order to understand and see how good our model is in predicting preciseness, we calculated acceptable (ACC), overestimated (OE), and underestimated (UE) incomes for every individual. In preciseness columns, total values give general information for the whole dataset about customers' income predictions whose predicted incomes are in an acceptable range and what is the average OE or UE values calculated from each technique. When customers predicted income taken from the model results is between the given interval ($\pm$ 15%) of the actual income, then it means that the technique's predicted customer income is acceptable. In some cases, the algorithm can overestimate or underestimate the incomes. As can be seen from Tables 4–8, in general, LI customers are overestimated and HI customers are underestimated. High acceptable ratio means that values predicted by the regression technique is good and it is suitable for use in income prediction for a given dataset at the bank. In contrast, both OE and UE are obviously not desired, although overestimation could be more harmful than underestimation, because of the fact that OE is more expensive and it assigns more loss than UE by increasing the risk of unpaid credit card limits. Underestimation, on the other hand, will cause loss of customers' future revenues that the bank could earn. Sometimes, assigning customers' income lower than actual may also adversely affect customer' satisfaction, which in turn may result in churn.

Eventually, we determined the relative ranks of each algorithm on all five datasets. Then, by taking the average of those ranks using Friedman [19], we calculated the overall rank as displayed in Table 9. It is a nonparametric statistical test performed to test the null hypothesis that all regression techniques perform alike, based on their rankings for a chosen performance metric. Once

Table 8
Results for Dataset 5.

	Low income			Middle income			High income			Total		
	ACC	OE	UE	ACC	OE	UE	ACC	OE	UE	ACC	OE	UE
OLS	28,42%	68,95%	2,63%	30,78%	38,33%	30,89%	21,35%	16,67%	61,98%	30,03%	38,88%	31,09%
BetaReg	0,00%	100,00%	0,00%	15,83%	71,59%	12,59%	0,00%	0,00%	100,00%	13,81%	68,81%	17,38%
Beta/OLS	0,00%	100,00%	0,00%	15,68%	77,61%	6,71%	9,38%	0,52%	90,10%	14,28%	74,10%	11,62%
BoxCox	0,00%	100,00%	0,00%	16,74%	69,18%	14,07%	0,00%	0,00%	100,00%	14,61%	66,71%	18,68%
Ridge	23,79%	71,36%	4,85%	29,11%	40,52%	30,37%	20,22%	15,85%	63,93%	28,18%	41,15%	30,67%
Robust	26,96%	72,06%	0,98%	32,75%	38,69%	28,55%	13,86%	13,86%	72,28%	31,07%	39,30%	29,63%
CART	0,00%	100,00%	0,00%	21,40%	61,48%	17,12%	8,33%	10,42%	81,25%	19,21%	60,65%	20,14%
M5P	0,00%	100,00%	0,00%	33,91%	45,50%	20,59%	13,54%	19,27%	67,19%	30,46%	47,27%	22,27%
MARS	0,00%	100,00%	0,00%	36,58%	40,43%	23,00%	16,15%	17,19%	66,67%	32,96%	42,71%	24,33%
LSSVM	0,00%	100,00%	0,00%	33,68%	46,91%	19,41%	18,23%	14,58%	67,19%	30,56%	48,20%	21,24%
ANN	0,00%	100,00%	0,00%	30,55%	47,94%	21,51%	13,54%	17,71%	68,75%	27,53%	49,30%	23,17%
OLS+M5P	0,00%	100,00%	0,00%	35,01%	44,74%	20,25%	14,06%	16,15%	69,79%	31,46%	46,40%	22,14%
OLS+MARS	1,58%	98,42%	0,00%	37,64%	40,05%	22,31%	16,67%	16,67%	66,67%	34,02%	42,24%	23,74%
OLS+CART	0,00%	100,00%	0,00%	21,95%	61,71%	16,34%	3,86%	5,31%	90,82%	19,24%	60,31%	20,45%
OLS+ANN	26,32%	72,11%	1,58%	30,93%	32,76%	36,31%	21,35%	14,06%	64,58%	30,03%	34,05%	35,92%
OLS+LSSVM	0,00%	100,00%	0,00%	34,02%	45,65%	20,33%	19,79%	15,63%	64,58%	30,96%	47,17%	21,87%
AdaBoost	26,43%	73,57%	0,00%	43,09%	27,04%	29,88%	25,00%	11,40%	63,60%	**35,43%**	42,34%	22,23%
Random Forest	15,33%	84,67%	0,00%	46,79%	27,96%	25,25%	25,74%	9,56%	64,71%	33,54%	46,67%	19,80%

	MAE	RMSE	AUC	AOC	R^2	Pearson	Kendall	Spearman	Hit 15%	Hit 30%	Hit50%
OLS	1380,735	2888,53	0,846	8037968,8	0,454	0,682	0,399	0,56	0,3	0,57	0,79
BetaReg	1793,34	3597,867	0,709	12532528	0,154	0,655	0,309	0,546	0,14	0,32	0,77
Beta/OLS	1776,939	3291,993	0,748	10452519	0,214	0,646	0,321	0,55	0,15	0,32	0,77
BoxCox	1719,824	3539,143	0,75	12125009	0,181	0,646	0,305	0,548	0,15	0,35	0,75
Ridge	1429,307	3125,167	0,845	9035550,6	0,434	0,662	0,383	0,541	0,28	0,53	0,77
Robust	1447,32	3258,443	0,847	11182530	0,44	0,664	0,395	0,555	0,31	0,57	0,79
CART	1582,671	3431,811	0,688	11292796	0,23	0,5	0,355	0,435	0,22	0,47	0,82
M5P	1294,391	2918,21	0,845	8180055,8	0,443	0,675	0,412	0,575	0,32	0,63	0,85
MARS	1324,733	2917,153	0,844	8187119,7	0,444	0,675	0,387	0,543	0,35	0,63	0,84
LSSVM	1330,562	3044,844	0,843	8925240,6	0,394	0,63	0,395	0,556	0,31	0,64	0,84
ANN	1407,283	3328,333	0,805	10339964	0,357	0,6	0,349	0,496	0,29	0,59	0,82
OLS+M5P	1274,121	2877,397	0,851	7976658,1	0,459	0,682	0,421	0,588	0,33	0,65	0,85
OLS+MARS	1316,163	2914,251	0,843	8158903,6	0,445	0,676	0,393	0,551	0,35	0,63	0,83
OLS+CART	1626,594	4020,791	0,7	15159630	0,2	0,448	0,364	0,445	0,21	0,47	0,83
OLS+ANN	1353,291	2874,196	0,838	7948147,7	0,43	0,684	0,371	0,526	0,3	0,56	0,77
OLS+LSSVM	1291,458	2865,749	0,847	7892468,2	**0,463**	0,683	0,403	0,566	0,32	0,64	0,84
AdaBoost	1283,3	2649,2	0,844	7772800	0,458	0,677	0,391	0,544	0,37	0,66	0,87
Random Forest	1251,4	2653,1	0,853	7783000	0,457	0,675	0,4	0,564	**0,37**	**0,68**	**0,88**

Table 9
Average ranking of algorithms.

Rank	R^2		Hit rate 15%		Acceptable		Total rank	
1	OLS+M5P	2.6	OLS+M5P	3	RF	3.8	OLS+M5P	3.533333
2	OLS+LSSVM	3.2	RF	3	OLS+M5P	5	OLS+LSSVM	4
3	OLS+MARS	4.8	OLS+LSSVM	3.4	OLS+MARS	5.4	RF	4.666667
4	M5P	5.6	M5P	4.2	OLS+LSSVM	5.4	OLS+MARS	5.133333
5	MARS	6.2	OLS+MARS	5.2	MARS	6	M5P	5.333333
6	OLS	7.2	MARS	5.4	M5P	6.2	MARS	5.866667
7	RF	7.2	AdaBoost	5.8	LSSVM	6.6	LSSVM	7.333333
8	OLS+ANN	8.6	LSSVM	6	AdaBoost	7.2	AdaBoost	7.6
9	ANN	9	OLS	8	Robust	8.6	OLS	8.2
10	Robust	9.4	OLS+ANN	9	OLS	9.4	OLS+ANN	9
11	LSSVM	9.4	Robust	9.2	OLS+ANN	9.4	Robust	9.066667
12	Ridge	9.8	Ridge	10.2	Ridge	9.6	Ridge	9.866667
13	AdaBoost	9.8	ANN	10.2	ANN	11	ANN	10.06667
14	OLS+CART	13.4	OLS+CART	12	OLS+CART	12.2	OLS+CART	12.53333
15	CART	14	Beta/OLS	14.2	CART	15.2	CART	14.53333
16	Beta/OLS	16	CART	14.4	Beta/OLS	15.8	Beta/OLS	15.33333
17	BoxCox	17.2	BoxCox	16	BoxCox	16.4	BoxCox	16.53333
18	BetaReg	17.6	BetaReg	16.6	BetaReg	17.4	BetaReg	17.2

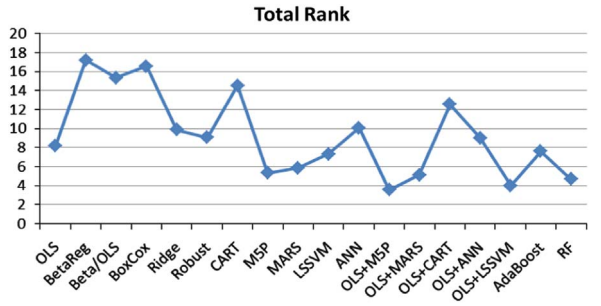

Fig. 2. Average ranking of algorithms.

Table 10
Comparison of three different algorithms in preciseness.

	Old model accpt. rate	Bought new model accpt. rate	Our model accpt. rate
Dataset 1	0.15	0.27	0.29
Dataset 2	0.12	0.32	0.34
Dataset 3	0.15	0.50	0.50

Table 11
Comparison of three different algorithms in R^2.

	Old bank model R^2	Bought new model R^2	Our model R^2
Dataset 1	0.24	0.57	0.59
Dataset 2	0.12	0.47	0.46
Dataset 3	0.09	0.37	0.38

Friedman's test rejected the null hypothesis, we proceeded with a Hommel's [23] post-hoc test method in order to find the concrete pairwise comparisons that produce rank differences with 95% confidence level. According to the demonstrated result of these tests, the difference between the models were small. Thus, Table 9 and Fig. 2 give a concrete suggestion to practitioners, so that, instead of implementing and testing all regression algorithms, one may select the top five best performing techniques (OLS+M5P, OLS+LS-SVM, OLS+MARS, M5P, and RF) for implementation and comparison.

Finally, we compared our best model results (OLS+M5P) with banks' old model and their new model that has been bought from another company. Tables 10 and 11 show the results of this three models according to preciseness and R^2. As we can see, our model outperforms the old model in all the three datasets and it also outperforms the new model except for Dataset 2 in R^2. Nevertheless, our presented algorithm is still better in income prediction, and banks in Turkey can use it in their income modeling problems.

5. Conclusion

In this large-scale income prediction benchmarking study, we evaluated 18 different regression techniques on five real-life datasets obtained from Turkish banks. The regression performance was assessed by 10 different performance metrics (MAE, RMSE, ROC, Spearman, Kendall etc.). It was found that non-linear and two-stage techniques except for CART yielded strong results in terms of all performance measures. However, it has to be noted that simple linear regression also had a very good performance, which clearly indicates that for most datasets, income can also be calculated by the traditional linear regression. The experiments also indicated that many regression techniques such as MARS, M5P, RF and LSSVM yield performances that are quite competitive. Yet, OLS+M5P model outperforms all other algorithms, and it was also better than the old model and the newly bought one that was used in the bank. We can comfortably say that banks can use this algorithm and other top five models in their income prediction problems.

Starting from the findings of this study, several interesting topics for future research can be identified. One of these promising avenues for future research would be to repeat the same experiment for customer groups of different characteristics (such as self employed ones, retired ones, etc.). Another interesting topic would be making classification work by considering dependent variable as nominal, for example, denoting incomes as high, medium, and low. Also, we intend to explore more methods and datasets from different countries to see the differences of country-based income prediction.

Acknowledgment

We would like to thank anonymous Turkish banks who made this study possible by providing us with in-house data on the income of their customers.

References

[1] B. Baesens, T. Van Gestel, S. Viaene, M. Stepanova, J. Suykens, J. Vanthienen, Benchmarking state-of-the-art classification algorithms for credit scoring, J. Oper. Res. Soc. 54 (6) (2003) 627–635.
[2] V. Barnett, T. Lewis, Outliers in Statistical Data, John Wiley and Sons, USA, 1998.
[3] A. Baylara, D. Hanbay, M. Batan, Application of least square support vector machines in the prediction of aeration performance of plunging overfall jets from weirs, Expert Syst. Appl. 36 (4) (2009) 8368–8374.
[4] J. Bi, K.P. Bennet, Regression error characteristic curves, in: Proceedings of the Twentieth International Conference on Machine Learning, 2003.
[5] C. Bone, O. Mitchell, Building netter retirement income models, N. Am. Actuar. J. 1 (1997) 1–12.
[6] G.E.P. Box, D.R. Cox, An analysis of transformations, J. R. Stat. Soc. 26 (1964) 211–252.
[7] L. Breiman, J.H. Friedman, R.A. Olshen, C.J. Stone, Classification and Regression Trees, 2nd Ed. Wadsworth, Pacific Grove, CA, 1984.
[8] L. Breiman, Random forests, Mach. Learn. 45 (1) (2001) 5–32.
[9] R.G. Carlos, T. Mercedes, H. César, Income prediction in the agrarian sector using product unit neural networks, Eur. J. Oper. Res. 204 (2010) 355–365.

[10] J. Carrier, K. Shand, New salary functions for pension valuations, N. Am. Actuar. J. 3 (1998) 18–26.
[11] Q. Chen, C. Yang, Quasi-stepwise regression variable selection and its application in rural household net income forecasting. Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100190, China, vol. 28(11), 2008, pp. 16–22.
[12] Grace Chia, Paul W. Miller, Tertiary performance, field of study, and graduate starting salaries, Aust. Econ. Rev. 41 (1) (2008) 15–31.
[13] P. Craven, G. Wahba, Smoothing noisy data with spline functions: estimating the correct degree of smoothing by the method of generalized cross-validation, Numer. Math. 31 (1979) 317–403.
[14] Marcel Das, Arthur van Soest, A panel data model for subjective information on household income growth, J. Econ. Behav. Org. 40 (1999) 409–426.
[15] Jeff Dominitz, Earnings expectations, revisions, and realizations, Rev. Econ. Stat. 80 (3) (1998) 374–388.
[16] S.L.P. Ferrari, F. Cribari-Neto, Beta regression for modelling rates and proportions, J. Appl. Stat. 31 (7) (2004) 799–815.
[17] Y. Freund, R.E. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, J. Comput. Syst. Sci. 55 (1) (1997) 119–139.
[18] J.H. Friedman, Multivariate adaptive regression splines, Ann. Stat. 19 (1991) 1–14.
[19] M. Friedman, A comparison of alternative tests of significance for the problem of m rankings, Ann. Math. Stat. 11 (1) (1940) 86–92.
[20] F.R. Hampel, E.M. Ronchetti, P.J. Rousseeuw, W.A. Stahel, Robust Statistics. The Approach Based on Influence Functions, John Wiley and Sons, New York, 1986.
[21] T. Higgins, M. Sinning, Modeling income dynamics for public policy design: an application to income contingent student loans, Econ. Edu. Rev. 37 (2013) 273–285.
[22] A.E. Hoerl, R.W. Kennard, Ridge regression: biased estimation for nonorthogonal problems, Technometrics 12 (1970) 55–67.
[23] G. Hommel, A stagewise rejective multiple test procedure based on a modified Bonferroni test, Biometrika 75 (1988) 383–386.
[24] P.H. Huber, Robust estimation of a location parameter, Ann. Math. Stat. 35 (1964) 7–101.
[25] G. Jekabsons, M5PrimeLab: M5' Regression Tree and Model Tree Toolbox for Matlab/Octave, 2010.
[26] G. Jekabsons, ARESLab: Adaptive Regression Splines toolbox for Matlab/Octave, 2011.
[27] G. Loterman, I. Brown, D. Martens, C. Mues, B. Baesens, Benchmarking regression algorithms for loss given default modeling, Int. J. Forecast. 28 (2012) 161–170.
[28] W.S. McCulloch, W.H. Pitts, A logical calculus of the ideas immanent in nervous activity, Bull. Math. Biophys. 5 (1943) 115–133.
[29] J.R. Quinlan, Learning with Continuous Classes, World Scientific, Singapore, 1992, 343–348.
[30] F. Rosenblatt, The perceptron: a probabilistic model for information storage and organization in the brain, Cornell Aeronautical Laboratory, Psychol. Rev. 65 (6) (1958) 386–408.
[31] P.J. Rousseeuw, V. Yohai, Robust regression by means of S-estimators, in: W.H.J. Franke, D. Martin (Eds.), Robust and Nonlinear Time Series Analysis, Springer-Verlag, New-York, 1984, pp. 256–272.
[32] P.J. Rousseeuw, A.M. Leroy, Robust Regression and Outlier Detection, Wiley-Interscience, New York, 1987.
[33] P. Samui, Multivariate adaptive regression spline (Mars) for prediction of elastic modulus of jointed rock mass, Geotech. Geol. Eng. (2012) 1–5.
[34] P. Samui, T.G. Sitharam, Least-square support vector machine applied to settlement of shallow foundations on cohesionless soils, Int. J. Numer. Anal. Method Geomech. 32 (17) (2008) 2033–2043.
[35] S. Sekulic, B.R. Kowalski, MARS: a tutorial, J. Chemom. 6 (1992) 199–216.
[36] John C. Smart, College influences on graduates' income levels, Res. High. Educ. 29 (1) (1988) 41–59.
[37] Nicholas S. Souleles, Consumer sentiment: its rationality and usefulness in forecasting expenditure – evidence from the Michigan micro data, J. Money, Credit, Bank. (2016). forthcoming (NBER working paper 8410), http://www.nber.org/papers/w8410.
[38] J.A.K. Suykens, T. Van Gestel, J. De Brabanter, B. DeMoor, J. Vandewalle, Least Squares Support Vector Machines, World Scientific Publishing Company, 2002.
[39] N. Swan, Problems in dynamic modeling of individual incomes, in: Proceedings of the Swedish Conference on Microsimulation, 1997.
[40] Scott L. Thomas, DefelTed costs and economic returns to college major, quality, and performance, Res. High. Educ. 41 (3) (2000) 281–313.
[41] A.N. Tikhonov, V.Y. Arsenin, Solution of Ill-posed Problems, Winston & Sons, Washington, 1977 ISBN 0-470-99124-0.
[42] V. Vapnik, A. Lerner, Pattern recognition using generalizd portait method, Autom. Remote Control 24 (1963) 774–780.
[43] X.D. Wang, M.Y. Ye, Nonlinear dynamic system identification using least squares support vector machine regression, in: Proceedings of the Third International Conference on Machine Learning and Cybernetics, Shanghai, 2004, pp. 26–29.
[44] V.J. Yohai, High breakdown-point and high effciency robust estimates for regression, Ann. Stat. 15 (1987) 642–656.