LightGBM: an Effective Decision Tree Gradient Boosting Method to Predict Customer Loyalty in the Finance Industry*

Marcos Roberto Machado[1], Salma Karray[1]

Abstract— **This study presents an implementation of a Machine Learning model, using LightGBM algorithm, in the financial sector to predict credit card customer loyalty, as a real company's problem. The offer of new algorithm options to be used in different industries have been increasing, and a result new models should be tested and compared with existent ones. This study compares the accuracy of LightGBM, which does not present applications in the financial industry, with XGBoosting, being both Gradient Boosting Decision Trees Models, these article presents the main method used by them. Also, an implementation of LightGB is performed over a Kaggle's competition, where a company aims to classify each customer of their base with a loyalty score for a given set of features. Results are presented, examined and discussed and customer loyalty prediction accuracy is measured through RMSE and compared between LightGBM and XGBoosting.**

I. INTRODUCTION

In order to maximize profits, companies have to invest their efforts in trying to retain customers, boosting customer satisfaction levels and as a consequence escalating their loyalty. The intensity of customers satisfaction is a key metric for a company's CRM (Customer Relationship Management) and as a consequence, loyalty can be measured and analyzed as well. Although customer satisfaction and loyalty are key mediators of profit, they cannot be taken as simple predictors of it. From a business standpoint, it is more important to identify and nurture relationships, specifically with profitable customers (Kumar and Reinartz, 2012 [1]). In this context artificial intelligence can be used through the application of Machine Learning (ML) and Deep Learning (DL) models given the extraordinary amount of customer information to businesses available in different sources.

ML and DL models can be applied within customer loyalty context from two different point of views. Firstly, it can be used to measure loyalty given customer information, and, secondly, it can be applied to improve the customer satisfaction by providing the right product or service in the right channel/place at the correct time. In both cases, ML/DL models would learn from previous customer interactions. It is so important to be focus on customer satisfaction and loyalty that, according to Gerry Brown from IDC, 2017 [2], 65% of the marketing executives surveyed pointed out that "real-time personalized advertising insertions" and "optimized message targeting" will convey a compelling value by 2020. Also, a research published by MIT and Google said that 50% of businesses intend to apply ML for customers insights, and 48% expect it to earn a competitive advantage.

Serkan, 2016 [3]; Ajay et al. 2019 [4] and Davies et al. 2018 [5] have applied ML in order to study customer loyalty in different settings. More general, other business metrics such as customer lifetime value (CLV), customer retention or customer churn were also explored using ML models and different applications were found in the literature (Jamalian and Foukerdi, 2018 [6], Jing and Xing-hua, 2008 [7], Amin et al. 2017 [8] and Amin et al. 2018 [8]). However, none of these applications studied customer loyalty for credit card customers using LightGBM model.

The aim of this study is to explore the application of LightGBM as a predictor for customer loyalty score. The problem was motivated based on an online competition (Kaggle, [9]), where a credit card company was looking for a ML model that make prediction of customer loyalty score for each single Card ID (for each single card issued) possible. Company was also requesting a RMSE (Root Mean Square Error) to be presented along to the loyalty score predicted. These type of problem is getting more and more common in the financial industry, where, with more customer data it is more important to anticipate customers needs in order to retain them, increasing their loyalty. LightGBM as applied as the main ML method, mainly based on its capacity to uses GPU and uses less memory, the possibility to get high speed and handle large size of data. Also, because it present better accuracy than other decision tree gradient boosting models such as XGBoosting (Ke et al., 2017 [10]) and by reason of its application was not found at the CRM/Loyalty literature.

II. LITERATURE REVIEW

A. Decision Tree Models

One of the most common classification technique is the Decision Tree (Seni and Elder, 2010 [11]). Figure 1, presents a flowchart, which exemplifies a decision tree. It is composed of decision blocks (rectangles) and terminating blocks (circles), which present the conclusion or final classification. Arrows in the flowchart represent branches that can lead to decision blocks or terminating blocks (Harrington, 2012 [12]). In this example, a decision between approval or denial of a credit card is build, where income and credit score are features or variables that will decide

*Submitted to Prof. Dra. Salma Karray as a report/article suggestion. This text was elaborated based on a ML grad course presentation performed at The Fields Institute in April 2019.

[1]University of Ontario Institute of Technology

final classification.

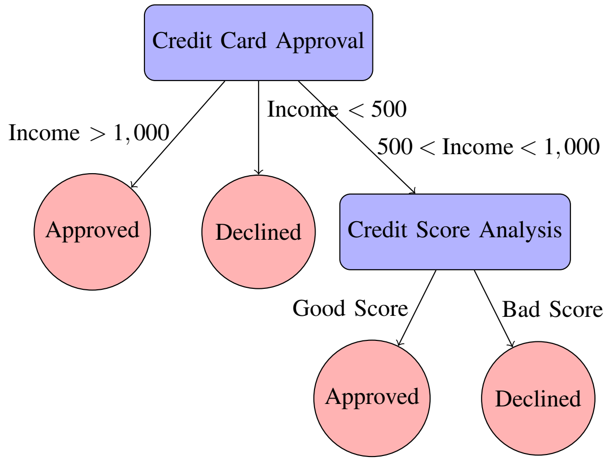

Fig. 1. Decision Tree example. A decision over issue customer with a credit card or not can be performed following this model, where customer income and credit score are features analyzed to make a decision.

Decision tree models are computationally cheap to use, easy for humans to understand learned results, handle missing value/information well, can deal with irrelevant features, works with numerical and nominal values, however, it tends to overfit (Bishop, 2006 [13] and Russell and Norvig, 2009 [14]). Another important concept with respect to these type of models is split. It is necessary to choose which feature will be choosen to split the data in order to guarantee the best results possible. The decision on how to split data should be performed the same way repeatedly until all data has been classified. Data split in decision tree models had been performed using different measures in different studies (Dietterich, 2000 [15], Kohavi et al., 2002 [16], Quinlan, 1986 [17] and Fayyad and Irani, 1992 [18]). Moreover, the most common metrics are the Gini Impurity, Information Gain (Entropy) and Variance Reduction (Harrington, 2012 [12] and Russell and Norvig, 2009 [14]). Variance reduction of Gain of Variance is the metric used in XGBoosting and LightGBM, which will be explored in detail in this study.

B. Gradient Boosting Models

Another machine learning technique used for regression and classification is gradient boosting. The output of the models built with the application of this technique does perform a prediction in the form of an ensemble of weak prediction models, decision trees (Bishop, 2006 [13] and Harrington, 2012 [12]). Leo Breiman ([19]) first presented the idea that boosting can be interpreted as an optimization algorithm for a given cost function. Later on, regressions gradient boosting algorithms were implemented by different authors (Friedman, 1999 [20], Friedman, 2000 [20] and Elith et al., 2008 [21]) using a different and practical viewpoint (Mason et al., 1999 [22] and Mason et al., 2000 [23]). As a consequence, the development of different boosting algorithms in different areas such as Statistics and AI (Artificial Intelligence) is observed (Grabner and Bischof, 2006 [24], Lee et al., 2010 [25] and Shabtai et al., 2009 [26]).

In order to perform predictions, gradient boosting uses a strong learner – a classifier arbitrarily well-correlated with the true classification – which is build from the combination of different weak learners – classifiers that are only slightly correlated with the true classification – this approach is applied in a iterative method (Li, 2017 [27]).

For the gradient boosting algorithm, as in any other machine learning supervised problem, there is a vector of features x that will be used to predict an output variable y through a probability distribution $P(x,y)$. Thus, a data set $\{(x_i,y_i)\}_{i=1}^n$ is needed and the aim of the algorithm is to find an approximation $\hat{F}(x)$ to a function $F(x)$ which will minimize the value of a certain loss function $L(y_i,F(x))$. Thus, assuming a value y, the gradient boosting algorithm will look for an approximation $\hat{F}(x)$ in the form of a weighted sum of functions, from different classes of (weak) learners.

Algorithm 1 presents the pseudo-code on how the gradient boosting algorithm works. For a given data set, $\{(x_i,y_i)\}_{i=1}^n$, and lost function, $L(y_i,F(x_i))$, the procedure initiate assuming a constant as the predictor that minimize the loss function. Then, in the step 2, the residuals (r_{im}) are calculated for each observation m in the data set (which, will logically provide the gradient boosting calculation), then a regression is fitted over the residuals calculation creating terminal regions (R_{jm}) for possible classifications, after that, the minimum value of each terminal (γ_{jm}), that were already defined is verified (which is the average in each region defined), and finally the constant (average) with which the iteration process started will be optimize in each integration time step in order to perform the final prediction $(F_m(x))$.

```
Input: Data $\{(x_i,y_i)\}_{i=1}^n$, and a differentiable loss
  function $L(y_i,F(x_i))$.
Step 1: Initialize model with a constant:
  $F_0(x) = \min_{\gamma} \sum_{i=1}^n L(y_i,\gamma)$.
Step 2: for $m = 1$ to $M$:
  • Compute $r_{im} = -\left[\frac{\partial L(y_i,F(x_i))}{\partial F(x_i)}\right]_{F(x)=F_{m-1}(x)}$ for
    $i = 1,2,...,n$;
  • Fit a regression tree to the $r_{im}$ values and create
    terminal regions $R_{jm}$ for $j = 1,2...,J_m$ ;
  • For $j = 1,2,...,J_m$ compute
    $\gamma_{jm} = \min_{\gamma} \sum_{x_i \in R_{ij}} L(y_i,F_{m-1}(x_i)+\gamma)$ ;
  • Update $F_m(x) = F_{m-1}(x) + \nu \sum_{j=1}^{J_m} \gamma_{jm} I(x \in R_{jm})$;
Step 3: Output $F_m(x)$.
```

Algorithm 1: Gradient Boosting Algorithm – Pseudo Code (Source: Scikit Learn, 2019 [28], adapted by the author).

C. GBDT Models

Gradient Boosting Decision Tree (GBDT) models combine the two techniques already presented in this study, usually through decision tree models, for instance XGBoosting, which is one of the most used GBDT algorithms (Torlay, 2017 [29], Zheng et al., 2017 [30] and Chen and Guestrin, 2016 [31]).

These models are characterized mainly by the way that the splits are performed when the trees are growing in the processes. Most of the GBDT models, like XGBoosting, does perform splits by calculating the Gain of Variance (Equation 1). For instance, letting O be the training set on a fixed node of the decision tree. The variance gain of splitting feature j at point d for this node is presented in Equation 1 (Ke at al., 2017 [10]).

$$V_{j|O}(d) = \frac{1}{n_O}\left(\frac{\left(\sum_{x_i \in O : x_{ij} \leqslant d} g_i\right)^2}{n^j_{l|O}(d)} + \frac{\left(\sum_{x_i \in O : x_{ij} > d} g_i\right)^2}{n^j_{r|O}(d)}\right) \tag{1}$$

where g_i is the negative gradient of the loss function with respect to the output of the model, $n_O = \sum I[x_i \in O]$ which is the total number of observations in the data set O, $n^j_{l|O}(d) = \sum I[x_i \in O : x_{ij} \leqslant d]$ is the total number of observations at the left of the data set and $n^j_{r|O}(d) = \sum I[x_i \in O : x_{ij} > d]$ is the total number of observations at the right of the data set. For a feature j, the decision tree algorithm selects $d^*_j = argmax_d V_j(d)$ and calculate the largest gain $V_j(d^*_j)$. Then, the data are split according feature j^* at the point d_{j^*} into the left and right child nodes.

Most decision tree learning algorithms, such as XGBoosting, grow trees by level (depth), which means that at each integration time step, all nodes in the deepest level grow (Figure 2).

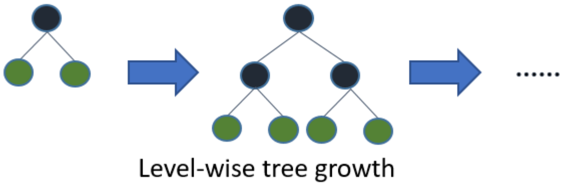

Fig. 2. GBDT Models - Growing tree level-wise within model training, all nodes grow in each iteration (Source: LightGBM Documentation, 2019 [32]).

1) LightGBM: LightGBM is an open source GBDT algorithm by Microsoft. It uses a histogram-based algorithm (Alsabti et al., 1999 [33]) to speed up the training process, reduce memory consumption and combines advanced network communication to optimize parallel learning, called parallel voting decision tree algorithm (Wang et al., 2017 [34] and Ke at al., 2017 [10]). Also, LightGBM uses the leaf-wise strategy to grow trees and to find a leaf with largest gain of variance to do the split (Figure 3).

LightGBM can be differentiated from other GBDT models by the way the gain of variation is calculated (Equation 2). Considering the same inputs presented for the calculation of the gain of variance in Equation 1, in lightGBM, the splits occur considering weak and strong learners (small and big gradients (g_i). In this case, the training instances are rank according to the absolute values of their gradients in the descending order; then, a top a percent of instances with the larger gradients are kept to form an instance subset A. Then, for the remaining set A^c formed by the $(1-a)$ percent of instances with smaller gradients, a subset B with size $b \times |A^c|$ is randomly formed; finally, the split of the instances according to a estimated variance gain $V^*_j(d)$ over the subset $A \cup B$ is performed (Equation 2, Ke at al., 2017 [10]).

$$V^*_j(d) = \frac{1}{n}\left(\frac{\left(\sum_{x_i \in A_l} g_i + \frac{1-a}{b}\sum_{x_i \in B_l} g_i\right)^2}{n^j_l(d)} + \frac{\left(\sum_{x_i \in A_r} g_i + \frac{1-a}{b}\sum_{x_i \in B_r} g_i\right)^2}{n^j_r(d)}\right) \tag{2}$$

where $A_l = \{x_i \in A : x_{ij} \leqslant d\}$, $A_r = \{x_i \in A : x_{ij} > d\}$, $B_l = \{x_i \in B : x_{ij} \leqslant d\}$, $B_r = \{x_i \in B : x_{ij} > d\}$, where d is the point in data where the split is calculated in order to find the optimal gain in variance, and the coefficient $\frac{1-a}{b}$ is used to normalize the sum of the gradients over B back to the size of A^c.

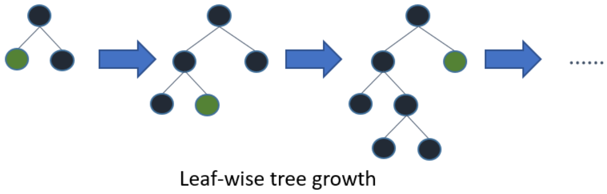

Fig. 3. LightGBM - Growing tree leaf-wise within model training, not all nodes grow in each integration time step, just the node that present optimal gain of variance is choose to grow and split (Source: LightGBM Documentation, 2019 [32]).

The gain of variance methodology, including the consideration of weak learners within the algorithms and the methodology to grow trees combined make LightGBM offer a better classification/prediction than other GBDT models (Ke at al., 2017 [10]). Applications of LightGBM algorithm can be found in the literature in different industries to deal with different problems (Wang et al., 2017 [34], Sun et al., 2018 [35] and Ma et al., 2018 [36]). However, because none of the applications of this algorithm were found in Business, Marketing of Financial setting, this study aims to compare its applicability, performance and accuracy when compared with another standard GBDT model (XGBoosting).

III. METHODOLOGY

A. Datasets

Elo is one of the largest payment brands in South America. Elo results from a partnership of three of the largest banks in Brazil: Banco do Brasil, Bradesco and CAIXA. Elo offers credit, debit, and prepaid cards. At the end of 2018, Elo had issued more than 50 million cards. Also, Elo credit card supports installment payments, which allows a customer to arrange payments over a period of time (Kaggle / Elo, 2018 [9]) .

Elo has built partnerships with merchants in order to offer promotions or discounts to cardholders. They have built machine learning models to understand the most important aspects and preferences in their customers lifecycle, from food to shopping. They have also built machine learning models to attribute a loyalty score for cluster of customers (per segment), however so far none of them is specifically tailored for an individual or profile (Kaggle / Elo, 2018 [9]). This is where the Kaggle competition and data made available comes in. The main objective is to develop algorithms to identify customer loyalty score for each card identified from the data sets.

Different data sets were made available (Figure 4). The historical data set has transaction information, features such as purchase date and amount, number of installments, city and state, merchant identification and a sub sector identification of the merchant where card was used. The merchant data set has additional information with respect to the places where cards were used. Features such as merchant identification group, average sales, average purchase, active month lags (amount of months in which card had not being used), city and state of the merchant location were available. New merchant data set has same type of information that historical data frame has, however this set of data contains material info of merchants visited for the first time. Training data set contains the first month of activation, the target (loyalty score) and other features. All five data sets also have anonymous information/features.

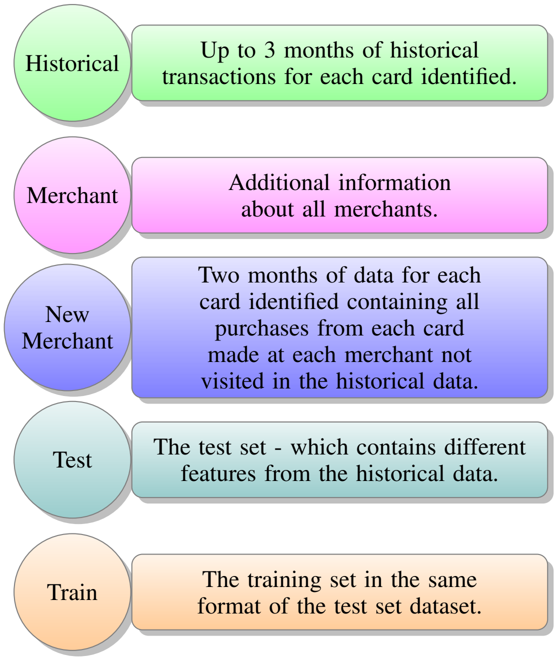
Historical
Up to 3 months of historical transactions for each card identified.
Merchant
Additional information about all merchants.
New Merchant
Two months of data for each card identified containing all purchases from each card made at each merchant not visited in the historical data.
Test
The test set - which contains different features from the historical data.
Train
The training set in the same format of the test set dataset.

Fig. 4. Five different datasets were made available from ELO, a credit card company. Dataset are briefly described, each dataframe has different set of infromations. Data was published at kaggle.com in 2018 [9]

B. Data Preparation

Almost all ML algorithms can not perform well with missing values. This was the very first point explored at the Data Preprocessing Treatment (DPT), which is part of what we are calling Data Cleaning. At this stage, for all data sets missing values were replaced based on the following rule: if the number of missing observations were less than 5% of the total of instances for that feature, those observations were drop, in the other case, the missing value was replaced by the mode of that feature. In addition, an analysis over possible outliers was performed. It is important to highlight that, mainly for the target, 1.1% (2,207) observations were possible outliers, and given the relevance of the feature in the data, those values were marked, considering the inclusion of a dummy feature in order to identify those.

In the second stage of DPT, categorical features were converted to numerical values in order to facilitate things later on in the pipeline process. Codes were attributed to different nominal observation and this standard was applied for all data sets. Then, feature engineering was performed. Firstly, all the data sets were unified/appended by the card identification, then more features were created based on the ones that the company made available. Feature transformers were applied in order to provide more data information for the models to learn better and predict information with higher accuracy. In the fourth stage, a correlation analysis over the features was implemented. When correlation between features and target was higher than 90%, features (new features created or previous ones) were drooped from the data sets. Finally, at the end of the workflow, a pipeline was implemented in order to select possible models and techniques to be used to train the data and make predictions. A workflow process is presented in Figure 5, where the bottom to top Data Preprocessing Treatment (DPT) performed in this study has been presented.

C. Model Training

In order to perform predictions on the test set, the main parameters used in LightGBM training process were:

- Number of leaves: 31;
- Minimum amount of data in a leaf: 30;
- Objective: Regression;
- Number of threads: 4;
- Boosting: GBDT

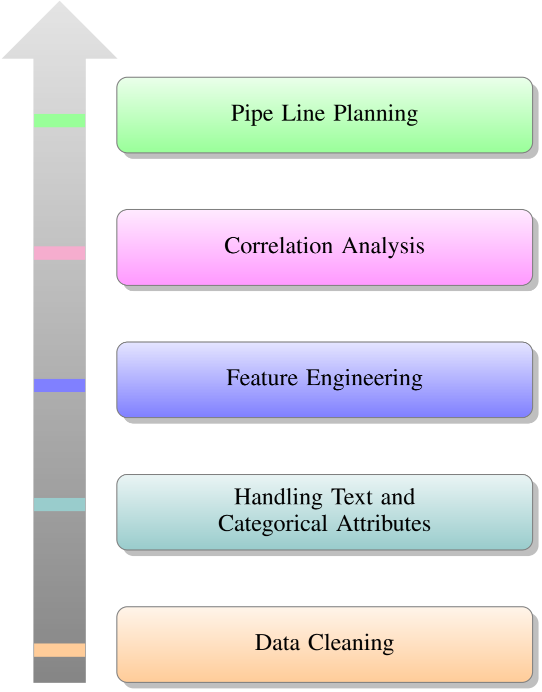

Fig. 5. Workflow from bottom to top of the Data Preprocessing Treatment (DPT) performed in the datasets in this study. All steps in this process was recommended by Geron, 2017 [37] and it is usually performed in an end-to-end ML modelling project.

Another important thing to mention with respect to the training process is that stratified random sampling method was used in this step. This is a method of sampling that was used to divide the data set that went through the DPT into smaller groups known as strata. In a stratified random sampling method, the strata are formed based on similar attributes or characteristics. Each new sample has the same size. In LightGBM, it is necessary to define the number of strata, which in this study is 5 for all training performed. By using stratified random sampling method, it is possible to have a higher number of subsets to train the model and by doing this we give the model a chance to obtain a better performance, because in this case, each new subset is used once as a test set while the others are used as training ones, until all folders be a test set at least once.

D. Evaluation Criteria

Among of all possible metrics, such as Absolute Error, Mean Absolute Error, Mean Squared Error, and Root Mean Squared Error (RMSE), RMSE was used to measure the differences between values predicted by LightGBM and XGboosting. These differences are called residuals and the RMSE is used to track the size of the errors in predictions and measure how well the model perform. Thus, RMSE is a measure of accuracy, to compare forecasting errors of different models for a particular dataset and not between datasets, as it is scale-dependent ((Harrington, 2012 [12] and Hyndman and Koehler, 2006 [38]).

According to Hyndman and Koehler, 2006 [38] RMSE is a quadratic scoring rule that measures the average magnitude of the error. Its the square root of the average of squared differences between prediction and actual observation:

$$RMSE = \sqrt{\frac{1}{n}\sum_{j=1}^{n}(y_j - \hat{y}_j)^2} \tag{3}$$

where n is the number of observations, y_j are the predicted values and $\hat{y}_j$ are the actual values.

As a consequence of Equation 3, RMSE is always non-negative, and a $RMSE = 0$ would indicate that the model fits the data set of the problem perfectly, which will never happen. Usually, researches aim to find the minimum RMSE, as it happens in this study.

IV. Results

This section presents how well LightGBM predicted customer loyalty in the financial sector, specifically for the case of Elo credit card company. RMSE has been analyzed altering some of the parameters within the ML training algorithm such as learning rate and number of iterations. Also, a comparison between LightGBM and XGboosting has been performed.

A. RMSE vs Learning Rate

The model was trained for different learning rates (0.001, 0.005, 0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.07, 0.08, 0.09, 0.1) and the RMSE for the predicted loyalty score was calculated. Figure 6 shows these results. It is possible to verify that when learning rate is approaching zero RMSE is not one of the best results. This happens because given a poor learning rate, when the algorithm is looking for the best solution, a local minimum is found and it is interpreted to be the global minimum. On the other hand, when learning rate is increasing, after a certain point (around learning rate equals to 0.04), the RMSE is no-optimal as well. This behavior is given by the reason of big learning rates lead makes the algorithm surpass the global minimum, finding at the end a local minimum. For this specific study, a learning rate of 0.01 is optimal, it offers a low RMSE as well as best loyalty score prediction.

B. RMSE vs Number of Iterations

For a fixed learning rate of 0.01, which provides the better RMSE as presented previously, different number of iterations was input during model training. Mainly because LightGBM has a Decision Tree model growing in its code, it is important to verify how the RMSE would behave for different number of iterations, mainly because in a decision tree model depending on the number of iterations the model will end it up with a different tree and as a consequence a different prediction. In this setting, different number of iterations were considered and RMSE measured, results are presented in Figure 7.

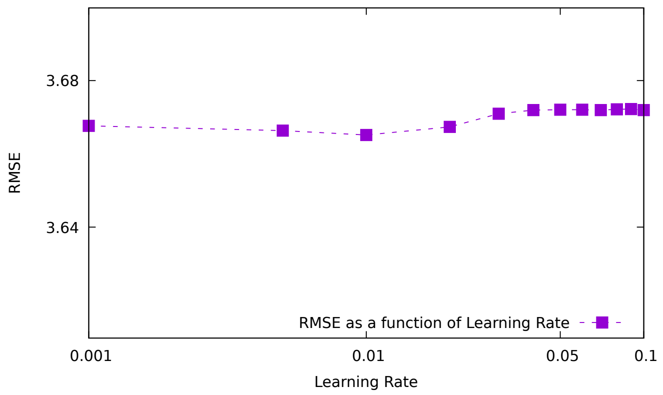

Fig. 6. LightGBM results: RMSE as a function of learning rate.

Figure 7 shows that for a small number of iterations, the model does not have time enough time to learn and find best solution and as a consequence a higher RMSE is observed. As the number of iterations increase, the RMSE decreases, up to a certain point where it does not matter how much more the number of iterations increases, the RMSE will not present a significant change, it will just increase the processing time. Therefore, it is plausible to choose between 900-1000 iterations in this study and guarantee the lower RMSE of the series.

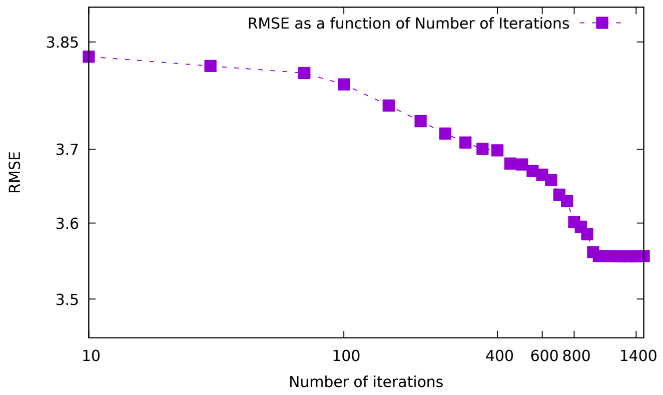

Fig. 7. LightGBM results: RMSE as a function of number of iterations.

C. Training Time

Another important thing to decide, from the point of view of a prediction model, is based on a trade-off: time versus accuracy. The number of iterations was fixed at 1000 and output measures such as time and RMSE were calculated (Figure 8). It is possible to observe that for small learning rate time is between the highest in the interval and the RMSE is not optimal, and as the learning rate increases the processing time decreases, however, the RMSE does not show a significant absolute difference in its value. Therefore, depending on the application the CRM will have to decide what is the primary focus - time or accuracy.

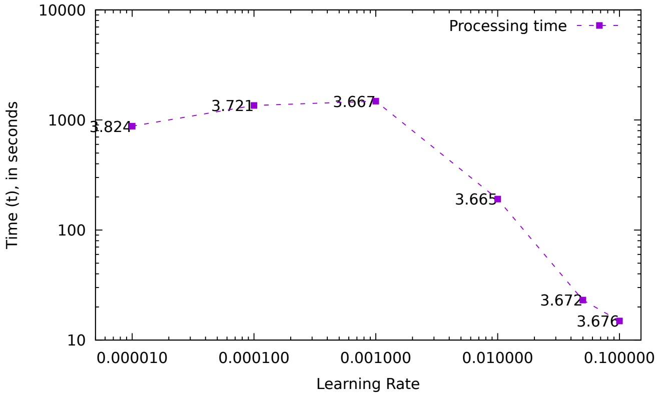

Fig. 8. LightGBM results: Time and RMSE as a function of learning rate. RMSE is presented as a label for each point on the dotted line.

D. Feature Importance

Before performing model training, we ended up with more than 230 features. Feature importance in predicting customer loyalty is given through the LightGBM model. Importance of each feature is calculated based on how many times a feature was used to slip information (was used as a node) during the tree growing in the modelling process. Figure 9 presents the top 8 most important features and the least important to predict customer loyalty in this study, feature importance measure and standard deviation are presented. From approximately 200 features, only as the ones that do not change when the number of iterations or learning rate were altered are presented. It is possible to observe that Month lag Mean, which is the average of months that a card identified was not used is the feature most used to split data in the decision tree process and the Month lag Minimum was the least used one. Standard deviation slightly decreases as the feature importance decreases.

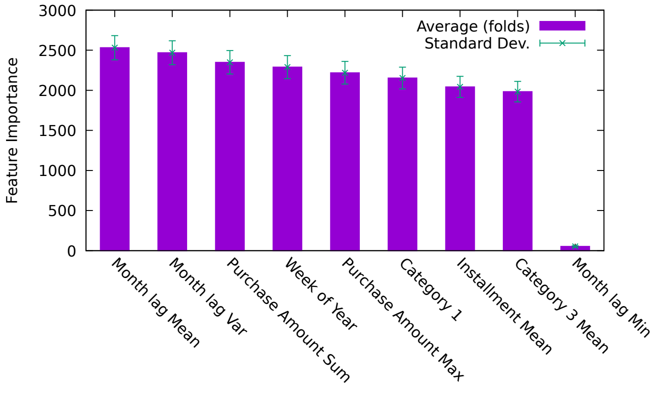

Fig. 9. LightGBM results: Feature Importance.

E. LightGBM vs XGBoosting

LightGBM and XGBoosting are based on Decision Tree and Gradient Boosting (DTGB) techniques. However, this section aims to verify Ke et al., 2017 [10] affirmation that LightGBM performs better than XGboosting as a DTGB combined model. Same set of parameters were set up for both scenarios, information such as learning rate and number of iterations were fixed and RMSE was measured for both cases and results are presented in Figure 10. It is possible to verify that, for this study, LightGBM perform better than XGBoosting for all interval of learning rate used. Therefore, it is plausible to affirm that LightGBM can also be used as a DTGB model in the financial sector to predict customer loyalty.

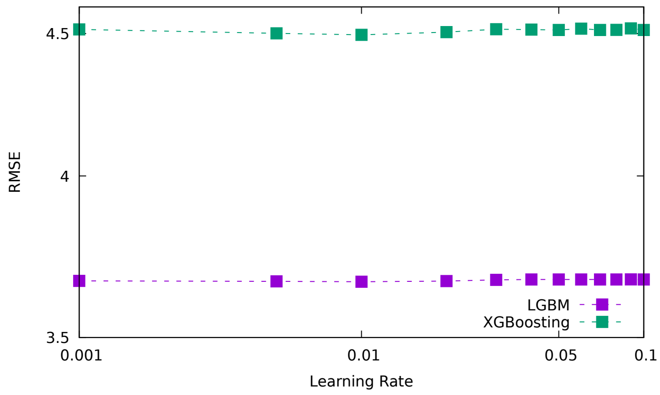

Fig. 10. LightGBM vs XGBoosting: RMSE for customer loyalty prediction.

V. Conclusion

In this study, LightGBM ML algorithm was applied in a financial setting in order to perform credit card customers loyalty score predictions. We proposed a GBDT model application in this setting, considering that in the literature this model does not appear to be applied exhaustively, specially in the Business/Financial industry. This paper aimed to verify if the accuracy of this model would be better than other types of GBDT algorithm, then a comparison between LightGBM and XGBoosting was also provided.

It is possible to verify from the results that, by utilizing a combination of good and weak learners (gradients) in order to look for a global minimum – which result as the best predictor for the problem, LightGBM does present good results, also when it is compared with XGBoosting.

The findings presented in this paper validate the usage of a relatively new GBDT model to be widely applied in the financial sector, once that it presents results with slightly better accuracy than usual regression and other GBDT models.

For future studies, comparisons with other type of ML models can be performed, also analyzing the feature importance and training the model with most important features (small number of variables) can boost management strategies in the business, results can be measured and loyalty programs can be developed or enforced based on the customer loyalty prediction performed over this study.

References

[1] V. Kumar and W. Reinartz, *Customer Relationship Management: Concept, Strategy, and Tools*, 2nd ed. Berlim: Springer Berlin Heidelberg, 2012.

[2] b. G. B. IDC International Data Corporation, “Can machines be creative? how technology is transforming marketing personalization and relevance,” 2019. [Online]. Available: https://www.criteo.com/digital-marketing-reports/can-machines-be-creative-how-technology-is-transforming-marketing-personalization-and-relevance/

[3] S. Varol, “Analyzing brand loyalty in automotive sector using the hidden markov model and support vector machine,” Ph.D. dissertation, 2016, copyright - Database copyright ProQuest LLC; ProQuest does not claim copyright in the individual underlying works; Last updated - 2017-09-13. [Online]. Available: http://search.proquest.com.uproxy.library.dc-uoit.ca/docview/1867750556?accountid=14694

[4] A. Aluri, B. S. Price, and N. H. McIntyre, “Using Machine Learning To Cocreate Value Through Dynamic Customer Engagement In A Brand Loyalty Program,” *Journal of Hospitality & Tourism Research*, vol. 43, no. 1, pp. 78–100, 2019. [Online]. Available: https://doi.org/10.1177/1096348017753521

[5] A. Davies, M. A. Green, and A. D. Singleton, “Using machine learning to investigate self-medication purchasing in england via high street retailer loyalty card data,” *PLoS ONE - Academic OneFile*, vol. 13, no. 11, p. p. e0207523, 2018. [Online]. Available: http://link.galegroup.com.uproxy

[6] E. Jamalian and R. Foukerdi, “A Hybrid Data Mining Method for Customer Churn Prediction,” *Engineering, Technology & Applied Science Research*, vol. 8, no. 3, pp. 2991–2997, 2018. [Online]. Available: www.etasr.com

[7] J. Zhao and X.-H. Dang, “Bank Customer Churn Prediction Based on Support Vector Machine: Taking a Commercial Bank’s VIP Customer Churn as the Example,” *2008 4th International Conference on Wireless Communications, Networking and Mobile Computing*, vol. 1, no. 3, pp. 1–4, 2008. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/

[8] A. Amin, F. Al-Obeidat, B. Shah, A. Adnan, J. Loo, and S. Anwar, “Customer churn prediction in telecommunication industry using data certainty,” *Journal of Business Research*, no. October 2017, pp. 1–12, 2018. [Online]. Available: https://doi.org/10.1016/j.jbusres.2018.03.003

[9] E. C. Cards, “Elo merchant category recommendation - help understand customer loyalty,” 2019. [Online]. Available: https://www.kaggle.com/c/elo-merchant-category-recommendation

[10] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, “Lightgbm: A highly efficient gradient boosting decision tree,” in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017, pp. 3146–3154. [Online]. Available: http://papers.nips.cc/paper/6907-lightgbm-a-highly-efficient-gradient-boosting-decision-tree.pdf

[11] G. Seni and J. F. Elder, “Ensemble methods in data mining: Improving accuracy through combining predictions,” *Synthesis Lectures on Data Mining and Knowledge Discovery*, vol. 2, no. 1, pp. 1–126, 2010. [Online]. Available: https://doi.org/10.2200/S00240ED1V01Y200912DMK002

[12] P. Harrington, *Machine Learning in Action*. Greenwich, CT, USA: Manning Publications Co., 2012.

[13] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg: Springer-Verlag, 2006.

[14] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 3rd ed. Upper Saddle River, NJ, USA: Prentice Hall Press, 2009.

[15] T. G. Dietterich, “Ensemble methods in machine learning,” in *Multiple Classifier Systems*. Berlin, Heidelberg”, pages=1–15, isbn=978-3-540-45014-6: Springer Berlin Heidelberg, 2000.

[16] R. Kohavi and J. R. Quinlan, “Handbook of data mining and knowledge discovery,” W. Klösgen and J. M. Zytkow, Eds. New York, NY, USA: Oxford University Press, Inc., 2002, ch. Data Mining Tasks and Methods: Classification: Decision-tree Discovery, pp. 267–276. [Online]. Available: http://dl.acm.org/citation.cfm?id=778212.778254

[17] J. R. Quinlan, “Induction of decision trees,” *Machine Learning*, vol. 1, no. 1, pp. 81–106, Mar 1986. [Online]. Available: https://doi.org/10.1007/BF00116251

[18] U. M. Fayyad and K. B. Irani, “On the handling of continuous-valued attributes in decision tree generation,” *Machine Learning*, vol. 8, no. 1, pp. 87–102, Jan 1992. [Online]. Available: https://doi.org/10.1007/BF00994007

[19] L. Breiman, “Arcing the edge,” Statistics Department, University of California, Berkeley., Tech. Rep., 1997.

[20] J. H. Friedman, “Greedy function approximation: A gradient boosting machine,” *Annals of Statistics*, vol. 29, pp. 1189–1232, 2000.

[21] J. Elith, J. R. Leathwick, and T. Hastie, “A working guide to boosted regression trees,” *Journal of Animal Ecology*, vol. 77, no. 4, pp. 802–813, 2008. [Online]. Available: https://besjournals.onlinelibrary.wiley.com/doi/abs/10.1111/j.1365-2656.2008.01390.x

[22] L. Mason, J. Baxter, P. Bartlett, and M. Frean, “Boosting algorithms as gradient descent in function space,” 1999.

[23] ——, “Boosting algorithms as gradient descent,” in *In Advances in Neural Information Processing Systems 12*. MIT Press, 2000, pp. 512–518.

[24] H. Grabner and H. Bischof, “On-line boosting and vision,” in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’06)*, vol. 1, June 2006, pp. 260–267.

[25] B. K. Lee, J. Lessler, and E. A. Stuart, “Improving propensity score weighting using machine learning,” *Statistics in Medicine*, vol. 29, no. 3, pp. 337–346, 2010. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/sim.3782

[26] A. Shabtai, R. Moskovitch, Y. Elovici, and C. Glezer, “Detection of malicious code by applying machine learning classifiers on static features: A state-of-the-art survey,” *Information Security Technical Report*, vol. 14, no. 1, pp. 16 – 29, 2009, malware. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1363412709000041

[27] C. Li, “A gentle introduction to gradient boosting,” 2017. [Online]. Available: http://www.ccs.neu.edu/home/vip/teach/MLcourse/

[28] S. Learn, “Ensemble methods - gradient boosting,” 2019. [Online]. Available: https://scikit-learn.org/stable/modules/ensemble.htmlgradient-boosting

[29] L. Torlay, M. Perrone-Bertolotti, E. Thomas, and M. Baciu, “Machine learning–xgboost analysis of language networks to classify patients with epilepsy,” *Brain Informatics*, vol. 4, no. 3, pp. 159–169, Sep 2017. [Online]. Available: https://doi.org/10.1007/s40708-017-0065-7

[30] H. Zheng, J. Yuan, and L. Chen, “Short-term load forecasting using emd-lstm neural networks with a xgboost algorithm for feature importance evaluation,” *Energies*, vol. 10, no. 8, 2017. [Online]. Available: https://www.mdpi.com/1996-1073/10/8/1168

[31] T. Chen and C. Guestrin, “Xgboost: A scalable tree boosting system,” in *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD ’16. New York, NY, USA: ACM, 2016, pp. 785–794. [Online]. Available: http://doi.acm.org/10.1145/2939672.2939785

[32] Microsoft, “Lightgbm - documentation,” 2019. [Online]. Available: https://lightgbm.readthedocs.io

[33] K. Alsabti, S. Ranka, and V. Singh, “Clouds: A decision tree classifier for large datasets,” in *Proceedings of the 4th Knowledge Discovery and Data Mining Conference*, 1998, pp. 2–8.

[34] D. Wang, Y. Zhang, and Y. Zhao, “Lightgbm: An effective mirna classification method in breast cancer patients,” in *Proceedings of the 2017 International Conference on Computational Biology and Bioinformatics*, ser. ICCBB 2017. New York, NY, USA: ACM, 2017, pp. 7–11. [Online]. Available: http://doi.acm.org/10.1145/3155077.3155079

[35] X. Sun, M. Liu, and Z. Sima, “A novel cryptocurrency price trend forecasting model based on lightgbm,” *Finance Research Letters*, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1544612318307918

[36] X. Ma, J. Sha, D. Wang, Y. Yu, Q. Yang, and X. Niu, “Study on a prediction of p2p network loan default based on the machine learning lightgbm and xgboost algorithms according to different high dimensional data cleaning,” *Electronic Commerce Research and Applications*, vol. 31, pp. 24 – 39, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S156742231830070X

[37] A. Géron, *Hands-On Machine Learning with Scikit-Learn*. O’Reilly, 2017.

[38] R. J. Hyndman and A. B. Koehler, “Another look at measures of forecast accuracy,” *International Journal of Forecasting*, pp. 679–688, 2006.