Deep Neural Pipeline for Churn Prediction and Mitigation

Andrei Ionut DAMIAN
Chief Data Scientist
University POLITEHNICA of Bucharest
Bucharest, Romania
damian@cloudifier.net

Andrei SIMION-CONSTANTINESCU
Computer Science & Engineering Student
University POLITEHNICA of Bucharest
Bucharest, Romania
andrei.simion.c@gmail.com

Nicolae TAPUS
Computer Science & Engineering Department
University POLITEHNICA of Bucharest
Bucharest, Romania
ntapus@cs.pub.ro

Bogdan DUMITRESCU
General Manager
Hight-Tech System&Software
Bucharest, Romania
bogdan.dumitrescu@htss.ro

Abstract— **Customer Churn is a metric used to quantify the number of customers who left the company. All retail and business to consumer companies carefully analyse customer behaviour to prevent them to cease his or her relationship with the company, in other words to make churn. With the latest development of Artificial Intelligence and the ones particularly related to Deep Learning we have a new set of powerful tools ready to employ within multiple horizontal and vertical domain – such as the horizontal of predictive business analytics domain. One of the main goals of predictive analytics is the research and development of the *almost-perfect* churn detection system. This paper objective is to propose a beyond state-of-the-art churn prediction and mitigation model together based on deep neural models and employing Big Data processing on massive parallel computing using GPU cells.**

Keywords— **machine learning, predictive analytics, massive parallel computing, GPU, deep learning, customer churn, big data**

I. INTRODUCTION

Customer Churn is a crucial metric in evaluating customer satisfaction over periods of time. Predicting customers behaviour has always been an important research area for all retail and business to consumers companies with high impact in the structure of the marketing campaigns. Identifying early who is at risk to leave is the top priority, because it is more expensive and time-consuming to acquire a new customer than retaining old ones.

This paper presents some churn prediction models applied on Pharma Industry taking advantage of big data provided from the largest pharmaceutical group in our country. Churn prediction techniques attempt to understand customer behaviours and attributes which signal the risk and timing of customer churn.

Pharma industry has been used as a target industry for our applied industrial research and we do have actual successful research experiment in our work.

The first step in creating our model was the exploratory analysis, in which you took the raw data for all client, we analyse and process keeping only the attributes that were proved to be relevant to our model. After pre-processing the raw data, we split the customers into 4 clusters, from the “weakest” clients to the best ones using RFM analysis [1]. Then, we split every macro-segment into other micro clusters, to have clients grouped by their behaviour into every large segment.

Another important stage of our project was to determine the definition of churn in our concrete example: A client is flagged as making churn if in the previous year it has a minimum number of one transaction and in the next one has none. We will also analyse the impact of “relaxing” the churn definition in our models.

The next step is choosing the model architecture. In this paper, we will analyse all range of experimented models beside the actual final research results, starting from a simple linear model and all the way down to a complex Directed Acyclic Convolutional-Recurrent Graph (DACOREG). As we will within the paper, our experimentation has been conducted using a pharma industry database of over 150 million observations thus resulting in a real base for benchmarking in conjunction with the plethora of proposed concurrently models.

The proposed and concurrently used models are:

- Logistic Regression
- Simple Decision Trees
- Union of Trees (Forests)
- XGBoost

Our model performance will be quantified using standard metrics, from which we are especially interested in two performance evaluating measures: precision and recall. Recall rate tell us who may churn clients our model identifies from the real ones and precision give us the false positive rate of our model. We tried to balance between a high recall rate and a maximal precision one, trying to get a balance in which we accept only a tiny decrease in recall for a better increase in precision.

Our first models were capable to see year-to-year evolution. We want a model that can see quarterly changes to have a more accurate prediction. Customers behaviour regarding pharma industry is directly influenced by seasons, that is why we need a model able to see couple of months to couple of months. Also, regarding RFM analyses, from our raw data we can collect quarterly information, meaning that we have a total year RFM and a RiFiMi for every quarter. This make the introduction of seasoning in our model more naturally.

II. RELATED WORK

A. *Logistic regression*

In machine learning a pair $(x^{(i)}, y^{(i)})$ represents a point made of the $n-th$ dimension predictors value (input value) with his associated output value.

Logistic regression [2] is a linear model used in classification problems where the dependent variable (the output y) is categorical. The goal is to find a function $h: D_x \rightarrow D_y$ which approximate the correspondent y-value for an entry x.

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n = \theta^T x$$
$$\textit{where n is the number of predictors}$$

In churn problem, the output is Boolean, meaning that we are in the case of a binary dependent variable. To retrieve the probability of the two classes (*churn* and *not churn*), the sigmoid $g(z) = \frac{1}{1+e^{-z}}$ function has been introduced. The hypothesis function $h_\theta(x) = g(\theta^T x)$ approximates the probability $P\left(y_q \middle| S_p, x_q\right)$ that if the underlying system is S_p, corresponding to a certain input x_p, the system output is y_p. Then a threshold is defined, to decide the output in term of churn/*not churn*.

Logistic regression performs well for simple patterns, since it is defined by a linear hypothesis function which values are transposed into [0,1] interval in order to define a probability. Logistic regression also works well for simple pattern with noise, but fails for complex ones (Figure 1). Since churn prediction is not a linear problem, the models based on this method does not provide very good results.

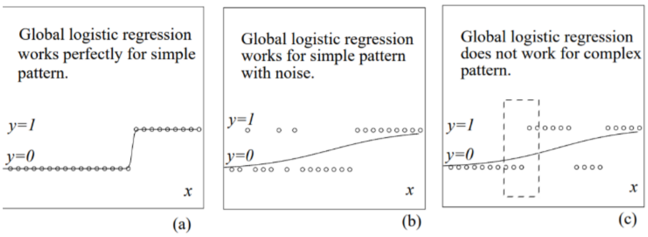
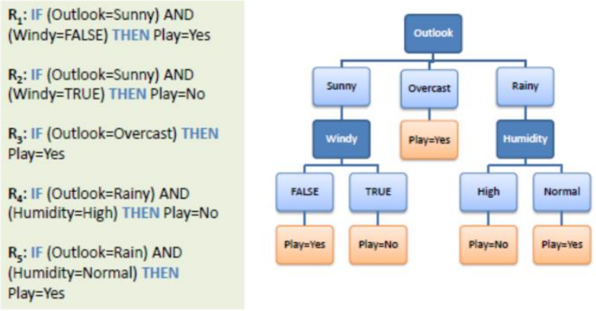


Figure 1

B. *Decision trees*

The idea of a decision tree [3] is to partition the input space into small segments, and label them with one of the various output categories. With the growth of the tree, the input space can be partitioned into tiny segments to be able to recognize more subtle patterns. However, overgrown trees can lead to overfitting.

A decision tree can easily be transformed to a set of rules by mapping from the root node to the leaf nodes one by one. (Figure 2).


R$_1$: IF (Outlook=Sunny) AND (Windy=FALSE) THEN Play=Yes

R$_2$: IF (Outlook=Sunny) AND (Windy=TRUE) THEN Play=No

R$_3$: IF (Outlook=Overcast) THEN Play=Yes

R$_4$: IF (Outlook=Rainy) AND (Humidity=High) THEN Play=No

R$_5$: IF (Outlook=Rain) AND (Humidity=Normal) THEN Play=Yes

Figure 2

An optimization for simple decision tree is the algorithm for boosting trees. The general idea is to compute a sequence of simple trees, where each successive tree is built for the prediction residuals of the preceding tree. Even if a boosted trees model is behaving better then a simple logistic regression, given the complex nature of churn analysis it is impossible to provide excellent results using them.

C. *Deep convolutional neural network*

Alex et al. [4] proposed a different approach in the image classification field using a large deep convolutional neural network (CNNs). For classifying high-resolution images from the ImageNet dataset into the 1000 different classes, his trained model achieved considerably better results than the previously state-of-the-art. Since then, CNN become specialized for image classification tasks. As you can see in Figure 3, a CNN consists of multiple convolutional layers, some pooling layers and in the end for classification proposes one or more fully connected layers.

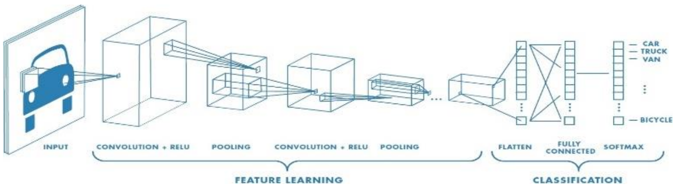


Figure 3

The power of image classification of CNNs were used to analyse churn problem in the telecommunication field [5]. Customer temporal behaviour data was represented as images based on their usage behaviour (columns as predictors) over a 30-day period (rows) as seen in Figure 4.

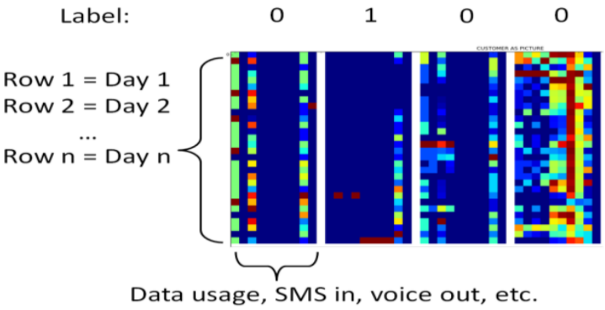


Figure 4

III. OUR WORK

A. Data format

Formalizing data used for training, each customer is defined by the values of the predictors variables $X = [X_i \, for \, i = 1 \, to \, n]$ where n is the number of predictors. The output the needed to be predicted is the binary variable $y \in \{0, 1\}$ where 0 is *not churn* and 1 is *churn*. Since our model trains on millions of data, it is mandatory to choose the relevant predictors for our problem.

Given the fact that our data is skewed, to evaluate our model we can not use accuracy metric. For a normal distribution, which is characterised by no skew, using accuracy make sense since the mean value is exactly at the peak. For our data, acquiring a *naive model* which predicts all observation into the dominate category can get a false high accuracy. The metrics needed in this case are *recall* and *precision*.

For classification tasks, we use terms like true positives (eqv. with hit: tp), true negatives (eqv. with correct rejection: tn), false positives (eqv. with false alarm: fp) and false negatives (eqv. with miss: fn).

$$Recall = \frac{tp}{tp + tn} \qquad\qquad Precision = \frac{tp}{tp + tn}$$

Recall and precision have antagonistic behaviour, meaning that an increase in recall will determine a decrease in precision.

REFERENCES

[1] Segmentation approaches in data-mining: A comparison of RFM, CHAID, and logistic regression, John A. McCarty and Manoj Hastak
[2] http://www.cs.cmu.edu/afs/cs/usr/kdeng/www/thesis/logistic.pdf
[3] Induction of Decision Trees, J.R.Quinlan
[4] ImageNet Classification with Deep Convolutional Neural Networks, Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton.
[5] Churn analysis using deep convolutional neural networks and autoencoders Artit Wangperawong, Cyrille Brun, Olav Laudy, Rujikorn Pavasuthipaisit