POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Executive Summary of the Thesis

Customer Churn prediction in a slow fashion e-commerce context: an analysis of the effect of static data in customer churn prediction

Laurea Magistrale in Computer Science and Engineering - Ingegneria Informatica

Author: Luca Colasanti

Advisor: Prof. Marcello Restelli

Academic year: 2021-2022

1. Introduction

As defined by [13], a customer is labelled as a churner if they have no events (purchases in this instance) in a set period of time. Thus, the "customer churn" event is the point in time after which a customer has no purchases. In the specifics of a fashion application, churn rate becomes central in ensuring the profitability and survivability of a business. Due to the increasing costs of customer acquisition [11] and the tendency of customer to churn in the presence of non-personalized marketing [10], it becomes fundamental to invest on customer retention processes. This is even more relevant in the context of emerging slow fashion, which emphasizes slowing down both the production and the consumption processes [1][7].

The problem of churn retention represents an instance of survival analysis, a subfield of statistics where the goal is to analyse and model the data where the outcome is the time until the occurrence of an event of interest [14]. Developing from medical applications survival analysis also captures an element of time to an event rather than simply addressing frequency. It also incorporates censorship, in which data about the event of interest are unknown because of withdrawal of the patient from the study [12].

Most of the early proposed solutions addressing survival analysis have drawn on non sequential ML models being fed with static features, that remain immutable during the analysis period, to come to conclusions. More recent studies, especially after the introduction of sequential ML and DL models, have seen a shift towards the use of dynamic features that instead evolve and change during the analysis. The rapid shift of models and data to more data hungry and powerful sequential models that leverage on dynamic features has left little time to experiment on the possible combinations of non sequential models with aggregates of dynamic or even sequential models enriching the data representation with static features. As shown by [3] the combination of static and dynamic features has in fact proven to be an interesting investigation in the context of RNN,that warrants further investigation.

This study thus investigates a comparisons of error performances of different sequential and non sequential architectures when trained and tested with two combinations of datasets: datasets composed of either dynamic only (RFM) features or a combination of static and dynamic features.

We organise this summary as follows: in Section

2 we provide a more detailed overview of the survival analysis literature and of the theoretical concepts applied in the study, in Section 3 we present an overview of the applied methodology, the data and how it is structured for the investigation, in Section 4 we provide the empirical results garnered from the analysis and their assessment and in Section 5 we proceed to discuss about the accomplishments as well as possible avenues of further research.

2. Related Work

Many of the early applications of survival analysis have found solutions that relied on static data being fed to non sequential models. Such works were often organised as comparison of various ML models or introduction of new frameworks of analysis with a common denominator being the use of static features that remained constant during the analysis period. Out of the various proposed solutions, one went on to become a staple in many survival analysis problems: the RFM framework. It introduced a different scope to the study of survival analysis in the guise of churn prediction [6][2] and characterized a customer along 3 aspects:

- R (Recency): the period since the last purchase.
- F (Frequency): number of purchases made within a certain period.
- M (Monetary): the money spent during a certain period.

The prototype of a very loyal customer thus presents low recency, high frequency and high monetary value. Like more recent work, this framework tried to appreciate the evolution of customers' behaviour over time and not limit itself from drawing conclusions just from immutable, static features.

Most recent works and those that can be regarded as State of the Art in the survival analysis field [4][8][5][15] have finally started approaching it through sequential DL architectures, such as RNN, able to make predictions based on series of observations instead of static datasets. Such solutions often use attributes that can be regarded as *dynamic*, which means that their change over the analysis window allows the model to better understand the evolution of the phenomenon. Examples of such attributes can be the different observations of temperature gathered on various components of jet engines like in [8], or different diagnoses and analyses performed over time in a medical application, users recurrent musical interests like in [5] and finally features related to location checkins like in [15]. Static features are basically ignored in such studies, even though error scores are not affected.

Some very promising findings in the matter of defining how to model churn, are showcased by [8], where the author, rather than focusing on "the absence of an event", decides to shift the attention towards the concept of Time To Event, that is literally the amount of time until the next event (e.g. a purchase). This TTE could potentially tend to infinity, indicating churn, or at least give an estimate of the Remaining Useful Life of a customer. In his approach the author uses a specific model introduced in [9], the so called WTTE-RNN, a recurrent neural network leveraging on the Weibull distribution to perform predictions on the Time To Event. The event is thought of under the influence of survival analysis and can thus be interpreted as the moment a machine becomes unusable as well as the next purchase in time, like in a fashion application case.

3. Methodology

3.1. Data

To the ends of this work, we employed transactional data belonging to customer orders placed on a clothing ecommerce website between July 2015 and up until June 2022. The features are of varying nature, mainly relating to orders, shipment and website interactions amounting to a total of 35. In order to ensure the quality of data, customers with at least 3 purchases during the analysis period have been selected for the analysis, thus reducing the overall amount of available customers to 24k between July 2015 and June 2022.

3.2. Data Mining

In order to achieve a less complex configuration of the available data. different dimensionality reduction techniques have been applied. Most importantly FAMD has been employed to try and obtain a reduced representation of the data and together with an analysis of the correlation matrix the available features have been reduced in number while maintaining most of their expressive power.

3.3. Survival analysis approach

The analysis of the performances of models mainly relies on an inspection of the MAE, MSE and R2-Score based on two different data configurations. The first configuration includes only dynamic features from the datasets along with a cluster features. The dynamic features are essentially drawn from the RFM framework, which has proven to be very apt at providing such representation. The cluster feature instead has been obtained by running a clusterization task on the dataset in order to obtain a distinction of the various datapoints that would make the interpretation of results and assessment easier to conduct by identifying 5 different clusters. The second configuration of data instead is comprised of the same features of the first one to which we add the other static features present in the dataset.

According to the nature of the model further tweaks have been made to the dataset. For sequential models, the dataset will be divided in monthly timeframes. In the case of non sequential models instead the dataset will not be organized in different timeframes,but with one datapoint per customer where the dynamic features have been aggregated.

As commonly adopted for ML applications, models' performances and errors have been analysed through the use of cross-validation techniques. In particular a training-test split of 70/30 has been applied in order to comfortably allow for a good portion of customers to be unseen to the model and thus provide as good an evaluation of the error (and eventual over or underfitting) as possible. Regarding the validation set instead a further 20% of the dataset has been at each run dedicated to it, with a k=3 employed in the k-fold cross validation technique.

3.4. Models

As referenced in Table 1 the investigation is conducted on both sequential and non sequential models. All models have been chosen based on their occurrence in previous studies, as well as to provide a certain degree of variety to the ends of the investigation.

Table 1: List of models employed in the analysis.

Non sequential ML models	Sequential models
SVM	LSTM
RF	
xGboost	
MLP	

Among non sequential models there are both tree-like structures and a DL model, while for sequential models the choice has fallen on the LSTM as the most commonly used representative of this category in previous studies.

4. Results

Table 2: List of models errors and metrics with a pure dynamic dataset.

TTE prediction	MAE	MSE	R2-Score
SVM	0.029	0.131	0.115
RF	0.029	0.139	0.086
xGboost	0.034	0.147	**-0.014**
MLP	0.035	0.138	-7.397
LSTM	**0.025**	**0.12**	-1.55

In Table 2 we can observe the error measurements of the various models when trained on the dataset composed only of dynamic features. Through the use of just RFM variables it can be clearly observed that the LSTM model was able to provide a better all around predictive performance when compared to other models that are unable to handle data in a sequential way and had to rely on aggregations of the dynamic data. The ability of the LSTM model to collect an evolving and time-bound representation of the data has definitely helped it in bulding a better abstraction of it and thus a better performance when handling predictions on unseen data. In particular we can observe a **14%** reduction in MSE, **8,4%** reduction in MAE and **20,3%** reduction in average negative offset when comparing the LSTM to the best performing non-DL model (all the traditional ML models except the MLP) for each metric. It is also worthy of observation that the non-DL models were instead able to provide a less chaotic predictive performance according to the R2 score, probably because of the unnecessary complexity of the MLP and LSTM models compared to the task at hand. It is also because of this that such "less powerful" models were able to keep up in performances given the variety in the data could still be represented without unnecessarily complex abstractions.

Table 3: List of models errors and metrics with dataset consisting of both dynamic and static features.

TTE prediction	MAE	MSE	R2-Score
SVM	0.028	0.133	0.127
RF	**0.026**	0.13	0.203
xGboost	0.029	0.133	**0.117**
MLP	0.04	0.149	-8.54
LSTM	0.028	**0.121**	-2.08

In table 3 the error and performance measurements can be observed when evaluating the models with a combination of both dynamic and static features.
The error and metrics put an edge of traditional ML model over the LSTM, in particular: MSE sees a reduction of error of **7%** of the RF compared to LSTM and a reduction of **7%** in MAE. Differently from before we can observe a different trend in performances: in terms of errors the traditional ML models seem to have benefitted most from the addition of static features in their performances. While the number of samples has not changed, the addition of static features has created a more complex and rich data representation that because of its nature has not benefitted in the same way the LSTM model, that does not deal well with unchanging data across different time steps.
The increased complexity of the data representation is evident when observing a shift in R2 score behaviour, although in this case the difference between values in the two configurations does not necessarily mean the model improved or not, but rather the absolute value the score assumes, which is generally higher in the second configuration, tells the prediction tends to be more chaotic.

In table 4 it is possible to observe the difference between the errors and scores of the model in the configuration with static and dynamic data against the one with only dynamic data.

Table 4: Percentage changes between model performances when using only dynamic features and dynamic + static ones.

Variation	MAE	MSE
SVM	-3,5%	+1,5%
RF	-10,3%	-6,4%
xGboost	-14,7%	-9,5%
MLP	+14,3%	+6,5%
LSTM	+12%	+ >1%

As already briefly mentioned, the inclusion of static attributes in the analysis seem to have provided different benefits to the various models and we can summarise such changes in three points:

- Non-DL models: For the SVM, RF and xGboost models the introduction of static features has aided the models in reducing their prediction errors, in particular in the cases of tree structures. This shows how a configuration including only dynamic attributes actually affects the predictive performances of such models negatively, since they are unable to process this kind of information effectively after the required aggregation and instead prefer working on features and attributes that are natively built to represent invariant, static information regarding the whole analysis window.
- DL, non-RNN models: in the case of MLP, while the model is not meant to handle sequential data, the introduction of static features did not improve the error performances, but rather made the model struggle with predictions even more. This is probably due to the increase of complexity in the prediction that may require better optimization on the number of samples compared to amount of features. DL models are in fact traditionally data hungry.
- RNN models: the LSTM models, similarly to the MLP, does not benefit from the introduction of static features, mostly because of the increased data complexity that it causes and the lack of an increase of available samples at the same time. In general though, such reduction is less felt and it

still manages to retain most of its predictive power and its best performance in terms of MSE, thanks to the reliance on the sequential configuration of data. Still the inclusion of static attributes can be regarded as the wrong step in an attempt to increase data complexity.

The specific implications of such results are very diverse and strictly dependent in each single category of models on the way such models themselves handle different kinds of data.

5. Conclusions

This work focuses on the task of applying survival analysis to a specific application, churn prediction, by analysing the contribution to reducing performance errors when using ML models.

By observing how the addition of static features to a dataset consisting of dynamic ones, derived from the RFM framework, contributes to the error performances of models has shed light on how to optimise such a task and makes a case on whether or not such additions make sense in order to continue improving on the current State of the Art.

In the specific application of Time To Event prediction problems the contribution provided by static features can be seen as a positive one, being that the enriched contextual representation that it provides to the data tends to provide equal if not smaller error measurements when other operational factors hindering the performances are not present. What can be gathered from the experience of this thesis is that expanding on its findings in an application where the lack of data does not hinder DL model performances, it can be further shown that static features have a positive effect on reducing the error.

This research in itself is twofold: on the one hand improving on RNN architectures in order to allow them to handle more features without the risk of overfitting can push their performances to the best achievable. On the other hand instead such models are very complex, power and data hungry and maybe not always suited to applications in smaller studies where data may be a problem. In this case non-DL architectures are to be preferred and possibly expanded upon in order to make them able to perform equally if not better than such more complex ones.

References

[1] Hazel Clark. Slow fashion an oxymoron or a promise for the future? *Fashion theory*, 12(4):427–446, 2008.

[2] Arno De Caigny, Kristof Coussement, and Koen W. De Bock. A new hybrid classification algorithm for customer churn prediction based on logistic regression and decision trees. *European Journal of Operational Research*, 269(2):760–772, 2018.

[3] Cristobal Esteban, Oliver Staeck, Stephan Baier, Yinchong Yang, and Volker Tresp. Predicting clinical events by combining static and dynamic information using recurrent neural networks. pages 93–101, 10 2016.

[4] Peter Fader, Bruce Hardie, Yuzhou Liu, Joseph Davin, and Thomas Steenburgh. 'how to project customer retention' revisited: The role of duration dependence. *SSRN Electronic Journal*, 43, 01 2018.

[5] How Jing and Alexander J. Smola. Neural survival recommender. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, WSDM '17, page 515–524, New York, NY, USA, 2017. Association for Computing Machinery.

[6] Mahboubeh Khajvand, Kiyana Zolfaghar, Sarah Ashoori, and Somayeh Alizadeh. Estimating customer lifetime value based on rfm analysis of customer purchase behavior: Case study. volume 3, 01 2010.

[7] Amanda Langdown. Slow fashion as an alternative to mass production: A fashion practitioner's journey. *Social Business*, 4(1):33–43, 2014.

[8] Egil Martinsson. Wtte-rnn - less hacky churn prediction · focus on the objective, 12 2016.

[9] Egil Martinsson. Wtte-rnn: Weibull time to event recurrent neural network a model for sequential prediction of time-to-event in the case of discrete or continuous censored data, recurrent events or time-varying covariates. 2017.

[10] Teemu Mutanen. Customer churn analysis– a case study. *Journal of Product and Brand Management*, 14(1):4–13, 2006.

[11] Desai Neel. how is cac changing over time?, 2021.

[12] Louis L. Nguyen and Rebecca E. Scully. *Chapter 1 - Epidemiology and Research Methodology*, volume 1, pages 1–12. Elsevier inc., ninth edition edition, 2019.

[13] Ana Perišić, Dubravka Šišak Jung, and Marko Pahor. Churn in the mobile gaming field: Establishing churn definitions and measuring classification similarities. *Expert Systems with Applications*, 191:116277, 2022.

[14] Ping Wang, Yan Li, and Chandan K. Reddy. Machine learning for survival analysis: A survey. 51(6), 02 2019.

[15] Guolei Yang, Ying Cai, and Chandan K Reddy. Spatio-temporal check-in time prediction with recurrent neural network based survival analysis. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 2976–2983. International Joint Conferences on Artificial Intelligence Organization, 7 2018.