

Sequential User Retention Modelling

HELDER MARTINS

Master in Machine Learning

Date: March 7, 2017

Supervisor: Hedvig Kjellström (KTH) and Sahar Asadi (Spotify)

Examiner: Patric Jensfelt

Swedish title: Detta är den svenska översättningen av titeln

School of Computer Science and Communication

Abstract

English abstract goes here.

Sammanfattning

Träutensilierna i ett tryckeri äro ingalunda en faktor där trevnadens ordningens och ekonomiens upprätthållande, och dock är det icke sällan som sorgliga erfarenheter göras ordningens och ekon och miens därmed upprätthållande. Träutensilierna i ett tryckeri äro ingalunda en oviktig faktor, för trevnadens ordningens och och dock är det icke sällan.

Contents

| | |
|---|------------|
| Contents | iii |
| 1 Introduction | 1 |
| 2 Background | 3 |
| 2.0.1 Definitions | 3 |
| 2.0.2 Feature Selection | 3 |
| 2.0.3 Predictor Models | 3 |
| 2.0.4 Evaluation Metrics | 3 |
| 3 Related Work | 6 |
| 3.1 Churn Prediction | 6 |
| 3.2 Sequential Modelling for User Behaviour Data | 9 |
| 3.3 Evaluating and Training Churn Prediction Models | 10 |
| 4 Methods | 13 |
| 5 Results | 14 |
| 6 Conclusions and Future Work | 16 |
| Bibliography | 17 |
| A Unnecessary Appended Material | 20 |

Chapter 1

Introduction

The last decade has witnessed an overwhelming increase in popularity in all kinds of streaming services around the globe. Applications like Netflix and Spotify have changed the industry by offering a legal and affordable way of accessing content by through a subscription-like contract instead of the classical pay-by-content commonly used by traditional media providers. This change in business model has proven to be successful by the sheer amount of customers that the most popular service providers have nowadays. Netflix for instance has reported to its shareholders a user base of 93 million worldwide at the end of 2016 [1], while Spotify recently reached the amazing feat of 50 million paying subscribers, 20 million of that only in one year [2]. Even though bureaucratic challenges like the dispute between the service providers and the labels regarding the value of licenses are still on debate, the adoption of this new model by consumers indicates no interest of going back to the old ways of getting access to digital content.

Music streaming follows this trend of growth on the latest economical reports. Over the five year period to 2015, the observed revenue from the industry grew four times to US \$2.89 billion as can be seen in Figure (TODO: add figure IFPI p17) from the IFPI organisation [3]. The increase in the number of paying subscribers is also of notice, seeing an increase from 8 to 68 million people over the same period. This new model also helped bring some countries where licensed music market was losing ground to piracy like China and Mexico, and it corresponds to around 20% of the industry's revenue in the top five markets in the world.

Spotify is for some time the top music streaming service globally, both in terms of revenue and number of subscribers. However the promise of growth in the area has attracted well known brands to the dispute like Apple and Google, and maintaining your current users within your service grew in importance since the process of acquiring new users is costly and difficult to execute properly. One of the important steps towards that goal is to predict accurately what is the chance of a user leaving the service provider at any given time with the intent of acting before it happens, a problem which is called *churn prediction*. Several classical techniques for predicting churn like Decision Trees and Logistic Regression have been used on recent work, however most of them make use of user behaviour data at a single point in time, normally the most recent one when the dataset was created.

Intuitively one can think that latent factors hidden in the temporal axis of the user behaviour data could yield a better prediction accuracy when comparing to a model which leverage a single and static point in time. For instance, a user that is gradually

reducing its consumption of the service over time can be easily thought of a prospect churner. While this intuition is trivial to come up with, we are also interested in learning strong correlations between temporal aspects of the data and the churn rate which are not as clear to the eye. Our hypothesis is that making use of the properties hidden on user behaviour data over time, a churn rate predictor could improve its accuracy greatly when compared to methods which are static on time.

A churn rate predictor which can identify possible churners accurately is just a trigger on the user retention process that can span several different stages. One of these stages is to identify what actions can be taken by the service provider as to avoid the user abandoning the service. An important problem to solve beforehand is to identify which features of the data have a strong statistical correlation with the churn rate. With that information, the providers could for example set up automated actions as to influence the value on these features. Learning which are the most important features is also of relevance for the task of choosing what data to use for training the predictor models, since commonly service providers have a vast amount of data about the user but only a fraction of that is of importance for predicting churn: training models with a full dataset will commonly introduce error on the system and take a large amount of computing resources to train.

The goal of this project will be to research different models for churn prediction that can make use of the temporal aspect of user behaviour data at its fullest. Our hypothesis is that latent factors hidden on usage patterns may increase the accuracy of state-of-art predictors which considers only the state of the data at a single point in time. We shall experiment on different models that are known to perform well on time series data and compare their accuracy using different evaluation methods. Detecting possible churners, while important, does not improve user retention without an associated action performed by the service provider. To derive insights on what can be done to improve this metric, a deep data exploration and feature analysis is also a goal of this project. Features shall be correlated to their influence on the churn rate, and feature selection techniques shall be researched and implemented in an attempt to reduce bias on the churn classifiers.

Chapter 2

Background

Definitions

Definition 1 *A session is defined as the time period the user is engaged into the application, be it to stream music or to navigate through the available content. A session starts when a user interaction trigger is captured (eg. a button click), and the session ends when no such trigger is captured and no continuous activity is being performed (eg. listening to a song) for at least 15 minutes.*

The session is the basic measurement of activity of the user on the service. Users are considered *active* if they are currently on a session, and *inactive* otherwise. (TODO Check Sahar)

Definition 2 *A user is defined to have churned if no session of his was registered in the last 30 days.*

Note that this definition is not concerned whether the user is a paying customer or not, but only if he is actually making use of the service. For example, a user may be subscribed to a Premium paid account and still be considered to have churned if no activity was registered in the last 30 days. However, it should be noted that empirical observations by the service provider suggest that there is a significant behavioural difference between the two classes of users that will be taken into consideration when the experiments are performed. (TODO Check Sahar)

Feature Selection

TODO: add feature selection techniques (Information gain, L1-norm, Recursive Feature Elimination, Variance Threshold...)

Predictor Models

TODO: add models (Decision Trees, Logistic Regression, Random Forests, LSTMs, CNNs...)

Evaluation Metrics

Confusion Matrix

A *confusion matrix* (also called contingency table) is a table layout representing the performance of a classifier's output, judging by its predictions against the actual true values.

In a binary classification problem, the confusion matrix is a 2×2 table where commonly the rows represent the true labels while the columns the predicted classes. Each cell contains a count of how many samples were classified on that category, and the values in the diagonal represent the correctly classified samples (if the features are ordered). Table (TODO: Add table) depicts a confusion matrix for a binary classifier.

The confusion matrix is an excellent visualization tool to estimate the performance of a model. The values that should be maximized, the *true positives* (TP) and *true negatives* (TN), represent the samples that were correctly labelled as possessing the feature of interest or not, respectively. On the other hand, the *false positives* (FP) and *false negatives* (FN) are the misclassified samples where the algorithm predicted that the feature was present while in truth it was not and vice-versa. In this work, the positive class will always represent a churning user, and it follows that the negative class represents the non-churners.

Classification Accuracy

Several different metrics can be derived from the confusion matrix table. The *classification accuracy* (CA) of a model is a common metric that corresponds to the fraction of the correctly classified samples on the test set, and can be calculated as follows:

$$CA = \frac{TP + TN}{TP + TN + FP + FN} \quad (2.1)$$

While trivial to understand, this metric may lead to erroneous conclusions when class imbalance is present in the test set, which is a common occurrence on the churn prediction domain. For example, if 9 out of 10 users of a dataset are non-churners, any classifier that simply outputs a negative class for all samples will result in an accuracy of 90%, however its ability of detecting churners is non-existent. For a service provider, detecting churn cases is always more important than detecting the loyal users, and this metric by itself cannot represent this goal. (TODO Add ref? Hassouna 2015)

Precision, Recall, Fall-out and False Alarm

To address the class imbalance problem of the classification accuracy, others metrics are also commonly used. The *positive predictive value* (PPV, also known as precision) is the proportion of the samples labelled as positive which are true positives, and describes the performance of the algorithm. The *true positive rate* (TPR, also called sensitivity and recall) of a model corresponds to the number of correctly predicted positive samples divided by all positive samples. *True negative rate* (TNR, also called specificity and fall-out) is the number of correctly predicted negatives divided by all true negatives. The *false positive rate* (FPR, also called false alarm ratio) is the probability of receiving a false positive as output of an experiment, and is calculated by dividing the number of false positives by the total number of positive samples.

$$PPV = \frac{TP}{TP + FP} \quad (2.2)$$

$$TPR = \frac{TP}{TP + FN} \quad (2.3)$$

$$TNR = \frac{TN}{TN + FP} \quad (2.4)$$

$$FPR = \frac{FP}{TN + FP} = 1 - TNR \quad (2.5)$$

Depending on the distribution of classes of the dataset, it is often difficult (although desirable) to maximize both metrics at the same time. A compromise must be reached that achieves the best trade-off between the two, which is commonly a business decision. For a music streaming service like Spotify, maximizing sensitivity is preferred due to the costs associated with a churning user, however a reasonable specificity is also a metric that should be strived for. (TODO Check with Sahar)

Receiver Operating Characteristic

The *receiver operating characteristic* (ROC) is a visualization tool that plots the relationship between the true positive rate (commonly the y-axis) and the false positive rate (the x-axis) of a binary classifier system. The curve is drawn by selecting different parameters of a model or levels of threshold for the decision boundary between the positive and negative classes. For example, when the output of a classifier is a probability value (like in logistic regression), different thresholds can be chosen to decide whether a user is a churning user or not, depending if the goal is to minimize FPR or maximize TPR. An example of a ROC curve can be seen in Figure (TODO Add figure)

A good classifier would score values close to the upper-left corner of the plot, where the point (0,1) represents a perfect classifier with 100% TPR and 0% FPR. On the other hand, an algorithm that outputs a curve alongside the diagonal where TPR and FPR are almost the same at different threshold levels can be considered close to a random guess, like the flip of a coin. A classifier would underperform if its scores are closer to the bottom-right corner of the plot, however this result can always be mirrored by updating the model to simply invert the positive and negative labels of the classified samples.

ROC curves can be used to compare the performance of different models by measuring the *area under the curve* (AUC) of its plotted scores, which ranges from 0.0 to 1.0. The greater this area, the better the algorithm is to find a specific feature. Moreover, models with an area close to 0.5 can be assumed to perform not much better than random guess, since this is the total area under the diagonal line.

TODO: Add eval methods (F-score, Lift chart, Top Decile Lift (TDL)...))

Chapter 3

Related Work

Churn Prediction

Identifying which users are most likely to churn is an important step in any customer relationship management (CRM) framework interested in improving user retention. It has been shown that several economical values are directly correlated with the customer lifetime metric that can be crucial in a competitive market, like how the costs of acquiring new customers surpass those of retaining existing ones and how loyal users usually spend more and can bring in new customers [4]. Thus, churn rate prediction is an established topic in the literature and has seen studies being performed on a wide range of domains like the telecommunication industry [5][6][7], social multiplayer games [8][9] and community-based question answering services (CQA) [10][11].

The very first aspect of the study that any researcher has to take into consideration is the formal definition of what churn is, which depends on the domain that the data set belongs to. [12] divides the definition into three types: *active* where the contract is officially terminated, *hidden* when the customer is inactive (not using the service) for a significant amount of time, or *partial* when the customer is not using the service at its fullest, opting for a competitor instead. Subscription-based services like mobile companies [5][7] and massive multiplayer games [8] commonly utilizes the active churning definition caused by the explicit termination of the agreement. Social games [9][13] and CQA services [10][11] however usually model the user interest by measuring its level of activity and engagement since no contract is formally established, opting for the hidden churn definition instead.

Even though Spotify has a contract-based Premium service, this project is interested in the clients who are inactive for some time, whether they are paying customers or not. Delving into the hidden churn definition, one could ask what is considered a "significant amount of time" as to judge a user to have churned or not, and different domains have distinct approaches depending on the way the interaction between user and service takes place. For example, for a mobile network like the one studied in [6] this definition can span months of inactivity since customers on that market are commonly more loyal, even though this paper focus on pre-paid accounts with "soft" contracts. On the other hand, rapid-changing markets like free-to-use services as social games and CQA sites prefer to define the churn period as small as possible (commonly days or weeks), since the absence of an agreement makes it easy for customers to switch for the competition, having the service the obligation to act faster if any sign of loss of users is identified [6][13]. On

[11] the focus is set on new users so the the churn period is explicitly set to be their first week of activity. [10] however focus on both new and experienced users by considering the first days and posts (questions and answers) as parameters on the models trained, evaluating each independently.

Current literature for prediction of churn rate uses a wide range of machine learning technologies, such as decision trees [10][7] [14] [6], logistic regression [4], neural networks [9] and its convolutional variant [15], random forests [11] and support-vector machines [16]. The number of techniques is vast, however it can be mentioned that current research suffers from a difficulty to reach insights that can generalize well for other areas, since the applications that the data sets belong to are domain-specific, greatly influencing how the users interact with the service. Interpretability is a key component in any application since the end goal is always not only to detect the users most likely to defect but also the reasoning behind their departure, as to plan reasonable actions to increase retention. With that in mind, even though some models like weighted random forests [17] and neural networks [9] offer a higher accuracy for some domains, the difficulty in getting key drivers behind the consumer's behaviour in some "black-box" type of models have led to researchers opting for a more interpretable technique like logistic regression which is commonly a close second place in terms of performance [9] [11]. For that reason, decision trees and logistic regression are as of today the most popular techniques for churn prediction, yet a consensus of the best model cannot be reached by researchers as stated by [18].

The telecom industry is by far the most well researched area for predicting churn, and a proper review would be incomplete without mentioning some examples. In [7], two popular models for predicting user churn were empirically compared: decision trees and logistic regression. By making use of the data set provided by a mobile operator, two models were created by independently training and selecting their best performing versions. Evaluation was executed by comparing the AUC of their ROC curves, as also their Lift score and overall accuracy. The conclusion is that decision trees consistently perform better when compared to logistic regression models, and thus should be a preferred choice. In [5], churn classification was performed on users in a rather unique way. The Gentle AdaBoost boosting algorithm was used alongside a logistic regression base learner as to train a model to do the separation between the churning and non-churning classes. However, one further step was performed where the users were split into two clusters based on the weights assigned by the boosting algorithm. One of the clusters was identified to have a considerably larger churn rate than the other, and thus another logistic model was trained using the clusters as labels, and its performance evaluated using ROC curves. The idea behind this approach is to use a model to mark users who have a high risk of leaving the service (and be the focus of user retention actions), instead of labelling them as churners and non-churners directly. On [6], billions of call detail records from a mobile phone operator was the focus of a churn prediction study and feature analysis. The initial data set was filtered to the calls made by 100.000 subscribers during a 6 month period. A brute-force approach to feature engineering was then performed to create 12.914 out of the initial 10 features by combining every feature from each of the manually split 8 different groups. Feature selection is then performed in two distinct ways: first, an individual Student's t-test score is computed for each individual feature to evaluate how well it can differentiate between the churners and non-churners sets. Second, a tree-based method was used as to estimate the the accuracy of a

joint classifier by adding features one at a time. Features are then ordered by their statistical correlation with churn, and the top 100 features were selected to train several different classifiers. Evaluation was performed mainly by comparing the AUC of each model, where AdaBoost using a decision tree classifier performed the best, followed closely by a logistic regression model.

The rapid increase in Internet popularity has spawned a plethora of new services in the last decade, and thus a wider range of domains like CQA sites and gaming applications are being researched for churn prediction. In [11], an explorative study was made on the Yahoo! answers website as to discover an efficient churn predictor model and also the features that correlates with the user leaving the service, however differently from other works this paper focused on new users with less than a week of activity on the service. Features were grouped into Question, Answer and Gratification categories, and were used for training several different classifiers. For this dataset, random forests performed the best with logistic regression as close second. Features were also ordered by the amount of information gain that they provide, and the number of questions and answers are the top features on that regard (inversely correlating with churn), followed by the period of time the user is active and gratification features for answers given and questions made. In [10], the younger CQA service Stack Overflow was the focus of a study on user behaviour characterization and churn rate prediction. An extensive data exploration was made as to correlate features to the chance of a user leaving a service, and with those insights classical modelling techniques were used, where the best performing one was a decision tree. The approach used for extracting and categorizing features (temporal, frequency, gratitude, etc) and the insights that follow the study (like the importance of temporal features for predicting churn) is of high value and can be mapped to concepts on a different domain like a music streaming service with minor modifications. In [9], players responsible for generating the most profit on two casual social games were the focus of a study on churn analysis and prediction. Formal definitions for active players and churning were made, and the problem was defined as a binary classification task where users were labelled as leaving the service or not in the following week of when the models were trained. Four different models were then trained, and a neural network classifier obtained the best area under the curve (AUC) score, with logistic regression as a close second. The formal definitions of "churn" and "activity" can be applied to a music streaming service in a similar way, as the evaluation of performance using a series of receiver operating characteristic (ROC) curves. In [8], the activity log of a subscription-based multiplayer online game was the source of an experiment focused on comparing two different methods for predicting the users with a higher tendency to leave the service, a theory-driven and data-driven approaches. For the modelling, an ensemble method was used with several different classifiers, where the choice of model was made based on whether the cluster of the data set (found with K-means) that will be used to train the classifier has a significant proportion of churners or not. Even though the performance of the data-driven model was superior, the difference was negligible when taking into consideration the complexity of the models and also its interpretability. Moreover, if marketing resources are constrained the theory-driven model was considerably superior for the first 40% of users on the lift chart.

Sequential Modelling for User Behaviour Data

Even though there is currently a wealth of research on churn prediction techniques, most of them treat the user behavioural features as static points in time by aggregating or summarizing their values over a time window [19]. It is a simplification that eases the hassle of training models with a data set which is commonly high-dimensional and difficult to work with, however information that could be used to improve the predictors' performance is lost in this process [4]. For example, there is no information as to when over the time window being analysed a user decides to churn, it could be that all defectors are grouped into a period where an external event (like the end of an offer) has taken place. Environmental variables, that is those that varies through time but are common to all users, are also omitted on static models, leaving it to the experts the task of manually integrating these features on their analyses. Newer models like [10] attempt to integrate some temporal data by manually engineering features related to time (like the duration of an event, for example), however since each user is represented as a sample in a single point in time the sequence of data transformations is inherently lost in this process.

It is natural to think that the human behaviour evolves through time. Our opinions and desires changes in accordance to the stimuli that we receive from the world around us, and summarizing our persona by looking at a single point in our life is an oversimplification that does not make justice to the path that lead us to where we are. Dealing with temporal data is by no means trivial however, and to leverage fully its properties some careful considerations about this type of data must be done. On [20], an overview of the challenges on creating models that make use of the time component on data sets are presented to the reader. Also, since hand-crafted features are generally domain-specific and difficult to create, a review of the current research on unsupervised feature learning applied for time-series data is the focus of this paper. From the challenges, it is worth of mention the uncertainty that there is enough information to understand the underlying process, its non-stationary characteristics like mean, variance and frequency, and its common high dimensionality and noise. The right representation is deemed then crucial for any model to be successful in its goal. Several different techniques currently being used for unsupervised feature learning are then presented in details to the reader, where it is worth of quick mention the conditional Restricted Boltzmann Machine (cRBM), Recurrent Neural Networks (RNN, with is Long Short-Term Memory extension), the Gated RBM, the Time-Delay Neural Networks, and the Space-Time Deep Belief Network. Finally, several classical time-series problems are reviewed alongside the best models currently being used for each domain.

The hypothesis that sequential patterns hidden on temporal data can improve the accuracy of predictor models is not an uncharted territory however, and has being recently explored by researchers on similar domains as churn prediction that could be mapped to with minor modifications. Sparked by positive results of recurrent neural networks on domains like speech recognition [21] and video classification [22], some papers attempt to recreate similar deep architectures on their own prediction problems. On [23], a technique was presented as to predict the next event and its timestamp on a running case of a business process (a help desk process, for example). Three different LSTM architectures were experimented on, one for each of the problems being tackled: estimating the next activity and its timestamp, all the remaining activities in a use case and the remain-

ing time of a process. This technique could be used for churn prediction by interpreting the user interaction with the service as actions, and "churn" could be an event that may or may not exist in the process chain. The LSTM architecture was also present on [19] alongside a quantile regression (QR) model as to predict which of the past buyers of a store chain will return after acquiring a product on sale. The hypothesis is that a performance gain can be achieved by combining models that are known to perform well on both temporal and aggregate features, respectively. It was empirically shown that a mixture of experts approach between these two techniques can reach a significant mean-square error improvement when comparing to any of the models when evaluated independently.

Deep architectures like LSTMs could theoretically be applied in a similar way for the task of predicting which users are more likely to abandon a service, however the lack of research using techniques like these suggests that this mapping may not be as easy as it seems, nevertheless some examples can still be found in the literature. Inspired by results obtained on image recognition [24], a convolutional neural network (CNN) model was used as a churn predictor on the work by [15]. On it, users were represented as 2-dimensional images where columns are the tracked features and rows are days organized in sequences. Two different CNN models were used, and both outperformed a baseline decision tree model. On [4] the researchers propose a new data set generation framework that can better leverage the temporal aspect of user behaviour data for churn prediction models. The hypothesis being tested is that by boosting the data with features on different time periods instead of only the last one, a significant performance gain can be achieved. The main framework of this study is called Multiple Period Training Data (MPTD), which basically consists of aggregating to each user their features at each predefined time step along the time range of the data set, alongside their churn labels at each point in time and also environment variables which varies through time but are common for all users. This framework was tested against a classical static data set, demonstrating a significant improvement (by *p-value* comparison) on AUC and TDL for both logistic regression and decision tree models for predicting if the user is going to churn on the next test period or not. Another experiment was focused on predicting churn several periods on the future, and for that a common survival analysis method called Cox regression was used as a benchmark for comparison against logistic regression models trained on MPTD. It was empirically concluded that using several independently trained binary classifiers, each trained on a different parameter to the "churn within δ periods" ($W\delta C$) feature, a significant performance gain can be achieved when comparing their respective AUC and TDL.

Evaluating and Training Churn Prediction Models

Data sets from service industries often have a common characteristic that may be problematic while training models, which is the uneven distribution between the churning and non-churning classes. It is a common scenario for the number of non-churners to heavily outweigh the number of churning samples on real data sets. Class imbalance is thus a recurrent concern in almost all domains, however research in this area lacks the proper attention. [17] aims to solve this mystery by focusing entirely on how the uneven class distribution affects the performance of several different classifiers. The performance boost estimated to be received by using a cost-sensitive learning algorithm (where false negatives are assigned a greater cost than false positives) is also evaluated, and it has

taken the form of a weighted random forest model which was compared to other classical baseline models like logistic regression and random forests. Under-sampling, where fewer samples from the majority class are incorporated in the training data as to artificially change the distribution between the labels, is proved to significantly improve the performance of the underlying models, however the exact ratio between churners and non-churners is confirmed to be case-dependant, not necessarily being the even distribution the perfect choice.

To deal with the problems brought by the unbalanced distributions mentioned above, proper evaluation metrics must be used as to avoid the predictors to be biased towards the majority class. It is a well know fact that the accuracy of a classifier is not an appropriate method when comparing the performance of different models [25], and so no paper reviewed used this metric by itself. The overwhelming majority of the literature abides by the use of AUC as a proper metric for predicting churn rate, [14] [4] [6] [5] to name a few. [17] delves into this topic and defends the use of AUC as a proper metric, while adding the cumulative gains chart as to graphically represent the percentage of customers has to be targeted to reach a percentage of churners. The F-score is also sometimes added as a metric of interest, normally in conjunction with AUC [11] [6]. In respect to graphical representations, ROC curves and lift charts dominates the literature as appropriate methods, and it is present in works like [5] and [17].

Another question that can be made is how far in the user history models must be trained on as to achieve the best trade-off between accuracy and computational burden. It is intuitive to imagine that user actions far in the past exerts little influence on predicting whether the user is going to churn or not in the near future, however the exact threshold that should be used is not trivial to find. [14] attempts to answer this question by experimenting with three different models (logistic regression, decision trees with and without bagging) on a newspaper data set consisting of 16 years of user history. The models were evaluated for their performance by training with data ranging on this interval with a 1-year gap between each measurement. It has been concluded that after year 5 no gain in performance (measured by AUC) is statistically relevant enough to warrant the increase in computational power needed for the training. However, the researchers also question whether this result can be extended for other domains or not, leaving that for future work.

Current data sets from service providers have no lack of user historical information stored, however the probability that all available features have a significant correlation to churn rate is pretty slim. Using too many features may lead to undesirable effects like overfitting and falling into the "curse of dimensionality" problem, reducing the classifiers performance as cited by [26]. It is a common step to pre process the data in a way as to mitigate this risk, but still there is no single method in the literature that dominates the scene. Several papers do a manual subset selection of features based on theories, testing each using baseline classifiers and selecting the subset with the best performance [10][9]. Others works like [8] and [11] use information gain as a metric to classify the features on their expected reduction in entropy. One step further is taken by works like [5] and [6] where full decision trees are built specifically for feature selection. Another more exotic method was used by [15] with an autoencoder to discover which features influenced the churn rate the most.

Churn prediction has been explored in different kind of applications on recent literature, however music streaming services are as of today a quite under-researched area,

even though there is a prominent feature that makes it significantly different from other more researched and similar areas like video streaming, which would be the absence of the constant attention factor that video demands. Studying the user behaviour in each application can possibly help optimizing parameters that are difficult to tune without expert knowledge on how the interaction between costumer and application occurs. The work by [27] aims to fill that gap by analysing patterns of usage of Spotify's Premium subscribers on features like session and playback arrival patterns, user behaviour on single and multiple devices and favourite times of day for streaming. The main contributions of the study can be summarized as follows: First, daily patterns can be observed on features like session arrival, playback arrivals and session length. Second, there is a high probability of users to continue on the same device for consecutive sessions. Third, users have their unique times of day when they prefer to stream music on the platform. Fourth, a session length can be used as a good indicator of the next session length and also downtime.

Chapter 4

Methods

- * Baseline classifiers (Decision Trees, Logistic Regression, Random Forests?)
 - * Temporal classifiers (LSTM, CNN?)
 - * One classifier vs different values of class distribution (skew)
 - * Different subset of features

Chapter 5

Results

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam

pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetur.

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

Sed commodo posuere pede. Mauris ut est. Ut quis purus. Sed ac odio. Sed vehicula hendrerit sem. Duis non odio. Morbi ut dui. Sed accumsan risus eget odio. In hac habitasse platea dictumst. Pellentesque non elit. Fusce sed justo eu urna porta tincidunt. Mauris felis odio, sollicitudin sed, volutpat a, ornare ac, erat. Morbi quis dolor. Donec pellentesque, erat ac sagittis semper, nunc dui lobortis purus, quis congue purus metus ultricies tellus. Proin et quam. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent sapien turpis, fermentum vel, eleifend faucibus, vehicula eu, lacus.

Chapter 6

Conclusions and Future Work

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetur.

Bibliography

- [1] Netflix Quarterly Earnings. <https://ir.netflix.com/results.cfm>. Accessed: 2017-03-01.
- [2] Spotify Press. <https://press.spotify.com/us/about/>. Accessed: 2017-03-01.
- [3] IFPI Global Music Report 2016. <http://www.ifpi.org/downloads/GMR2016.pdf>. Accessed: 2017-03-01.
- [4] Özden Gür Ali and Umut Aritürk. Dynamic churn prediction framework with more effective use of rare event data: The case of private banking. *Expert Systems with Applications*, 41(17):7889–7903, 2014. ISSN 09574174. doi: 10.1016/j.eswa.2014.06.018.
- [5] Ning Lu, Hua Lin, Jie Lu, and Guangquan Zhang. A customer churn prediction model in telecom industry using boosting. *IEEE Transactions on Industrial Informatics*, 10(2):1659–1665, 2014. ISSN 15513203. doi: 10.1109/TII.2012.2224355.
- [6] Muhammad Raza Khan, Joshua Manoj, Anikate Singh, and Joshua Blumenstock. Behavioral Modeling for Churn Prediction: Early Indicators and Accurate Predictors of Custom Defection and Loyalty. *Proceedings - 2015 IEEE International Congress on Big Data, BigData Congress 2015*, pages 677–680, 2015. doi: 10.1109/BigData-Congress.2015.107.
- [7] Mohammed Hassouna, Ali Tarhini, Tariq Elyas, Mohammad Saeed Aboutrab, United Kingdom, United Kingdom, and Saudi Arabia. Customer Churn in Mobile Markets: A Comparison of Techniques. 8(6):224–237, 2015. doi: 10.5539/ibr.v8n6p224.
- [8] Zoheb Borbora, Jaideep Srivastava, Kuo Wei Hsu, and Dmitri Wil Iams. Churn prediction in MMORPGs using player motivation theories and an ensemble approach. *Proceedings - 2011 IEEE International Conference on Privacy, Security, Risk and Trust and IEEE International Conference on Social Computing, PASSAT/SocialCom 2011*, pages 157–164, 2011. doi: 10.1109/PASSAT/SocialCom.2011.122.
- [9] Julian Runge, Peng Gao, Florent Garcin, and Boi Faltings. Churn prediction for high-value players in casual social games. *IEEE Conference on Computational Intelligence and Games, CIG*, 2014. ISSN 23254289. doi: 10.1109/CIG.2014.6932875.
- [10] Jagat Pudipeddi, Leman Akoglu, and Hanghang Tong. User Churn in Focused Question Answering Sites: Characterizations and Prediction. pages 469–474, 2014.
- [11] Gideon Dror, Dan Pelleg, Oleg Rokhlenko, and Idan Szpektor. Churn prediction in new users of Yahoo! answers. *Proceedings of the 21st International Conference on World Wide Web*, pages 829–834, 2012.

- [12] Vladislav Lazarov and Marius Capota. Churn prediction. *Bus. Anal. Course. TUM Comput. Sci*, 2007.
- [13] Anders Drachen, Eric Thurston Lundquist, Yungjen Kung, Pranav Simha Rao, Diego Klabjan, Rafet Sifa, and Julian Runge. Rapid prediction of player retention in free-to-play mobile games. *CoRR*, abs/1607.03202, 2016.
- [14] Michel Ballings and Dirk Van Den Poel. Customer event history for churn prediction: How long is long enough? *Expert Systems with Applications*, 39(18):13517–13522, 2012. ISSN 09574174. doi: 10.1016/j.eswa.2012.07.006. URL <http://dx.doi.org/10.1016/j.eswa.2012.07.006>.
- [15] Artit Wangperawong, Cyrille Brun, Dr. Olav Laudy, and Rujikorn Pavasuthipaisit. Churn analysis using deep convolutional neural networks and autoencoders. pages 1–6, 2016.
- [16] Kristof Coussement and Dirk Van den Poel. Churn prediction in subscription services: An application of support vector machines while comparing two parameter-selection techniques. *Expert systems with applications*, 34(1):313–327, 2008.
- [17] J. Burez and D. Van den Poel. Handling class imbalance in customer churn prediction. *Expert Systems with Applications*, 36(3 PART 1):4626–4636, 2009. ISSN 09574174. doi: 10.1016/j.eswa.2008.05.027. URL <http://dx.doi.org/10.1016/j.eswa.2008.05.027>.
- [18] Vishal Mahajan, Richa Misra, and Renuka Mahajan. Review of data mining techniques for churn prediction in telecom. *Journal of Information and Organizational Sciences*, 39(2):183–197, 2015.
- [19] Gaurangi Anand Auon, Haidar Kazmi, Pankaj Malhotra, Lovekesh Vig, Puneet Agarwal, and Gautam Shroff. Deep Temporal Features to Predict Repeat Buyers. (February 2016), 2015.
- [20] Martin Långkvist, Lars Karlsson, and Amy Loutfi. A review of unsupervised feature learning and deep learning for time-series modeling. *Pattern Recognition Letters*, 42(1):11–24, 2014. ISSN 01678655. doi: 10.1016/j.patrec.2014.01.008.
- [21] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. In *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on*, pages 6645–6649. IEEE, 2013.
- [22] Joe Yue-Hei Ng, Matthew Hausknecht, Sudheendra Vijayanarasimhan, Oriol Vinyals, Rajat Monga, and George Toderici. Beyond short snippets: Deep networks for video classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4694–4702, 2015.
- [23] Niek Tax, Ilya Verenich, Marcello La Rosa, and Marlon Dumas. Predictive Business Process Monitoring with LSTM Neural Networks. 2016. URL <http://arxiv.org/abs/1612.02130>.
- [24] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

- [25] David Martin Powers. Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation. 2011.
- [26] Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *Journal of machine learning research*, 3(Mar):1157–1182, 2003.
- [27] Boxun Zhang, Gunnar Kreitz, Marcus Isaksson, Javier Ubbillos, Guido Urdaneta, Johan a. Pouwelse, and Dick Epema. Understanding user behavior in Spotify. *2013 Proceedings IEEE INFOCOM*, pages 220–224, 2013. ISSN 0743-166X. doi: 10.1109/INFCOM.2013.6566767. URL <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6566767>.

Appendix A

Unnecessary Appended Material