

On the Operational Efficiency of Different Feature Types for Telco Churn Prediction

Sandra Mitrović^{a,*}, Bart Baesens^{a,b}, Wilfried Lemahieu^a, Jochen De Weerd^a

^a*Department of Decision Sciences and Information Management, KU Leuven, Leuven, Belgium*

^b*School of Management, University of Southampton, Southampton, United Kingdom*

Abstract

Churn prediction in telco remains a very active research topic. Due to the uptake of social network analytics and the results of previous benchmarking studies showing a rather flat maximum performance effect of predictive modeling techniques, the focus has mainly shifted to expanding and exploring the relevant feature space. While previous studies generally agree that adding features typically increases predictive performance, they rarely discuss the accompanying issues such as data availability and computational cost. In this work, we bridge the gap between predictive performance and operational efficiency by devising a new feature type classification and a novel reusable method to determine optimal feature type combinations based on Pareto multi-criteria optimization. Our results provide several insights that can serve as a guideline for industry practitioners.

Keywords: Feature Engineering, Feature Type Classification, Optimal Feature Type Combinations, Operational Efficiency, Churn Prediction, Pareto Multi-Objective Optimization

*Corresponding author

Email address: sandra.mitrovic@kuleuven.be (Sandra Mitrović)

1. Introduction

Churn prediction (CP), i.e. predicting which customers will stop using a company's services, is probably the most frequently tackled predictive task in the telecommunication industry, since retaining a customer is several times more beneficial than acquiring a new one (Kim et al., 2014). Consequently, many different approaches have been suggested in the literature, exploiting different modeling techniques and a variety of explanatory features. Lately, the latter focus is becoming more prevalent due to the fact that several benchmarking studies demonstrated that simple classification techniques perform well (Verbeke et al., 2010) and the fact that the availability of social network information enables expansion of the feature space.

Many previous works have analyzed the impact of including an additional variety of data on predictive performance (PP), e.g. network data (Backiel et al., 2014, 2016; Dasgupta et al., 2008; Richter et al., 2010; Zhang et al., 2012), operational circuit/packet switch data (Huang et al., 2015), price sensitivity data (Zhang et al., 2012), and volume (historical) (Huang et al., 2015; Zhang et al., 2012). However, the current CP literature suffers from at least three problems. First, apart from the local vs. network feature type classification, there is no standardized and more fine-grained feature type classification available. Second, existing studies do not unanimously agree on the predictive power of local and network feature types (e.g. Kim et al. (2014); Backiel et al. (2014) claim network features are better while Kusuma et al. (2013) show the opposite). Hence, a more thorough analysis, considering a more comprehensive feature type hierarchy, is required. Finally, the most important driver for this study is that none of the existing studies discusses the resource efforts related to data availability, data collection, feature engineering (e.g. time and effort needed for extracting features from huge networks, retrieving historical data), and model evaluation. Despite the fact that PP is often prioritized by researchers,

feature set design should also be driven by factors influencing the operational efficiency (OE) in real business settings as practitioners need to be capable of making informed decisions given their own circumstances and priorities. More specifically, they would benefit from proper guidelines indicating whether or not additional investments in e.g. computational time or data collection can generate a relevant increase in PP. Although it can be speculated that in today's big data era, distributed computing has drastically reduced the importance of computational time, it is still valuable to know whether a set of features provides a return of investment. Additionally, given the low switching costs between telco providers, online real-time CP is becoming more and more important (Diaz-Aviles et al., 2015). Hence, not just accurately, but also timely detection of dissatisfied customers is crucial (Flores-Méndez et al., 2016), which makes that considering appropriate features becomes fundamental.

As such, we shift the focus from a single point of view where only PP is considered, to one in which the OE/PP trade-off becomes imperative. To scrutinize this trade-off, we propose a detailed classification of feature types based on data recency (recent vs. historical), data locality (individual customer-related or local vs. inter customer-related or network), and data rendering (absolute vs. trend). Next, we resort to Pareto multi-criteria optimization of AUC (Area Under the ROC-Curve) and CT (computational time) so as to determine the optimal order in which different feature types are conjoined and to identify the final set of Pareto optimal solutions (transforming OE/PP trade-off into CT/AUC trade-off). As for the given problem this is the first trial to consider, and also to avoid model-related dependencies, CT is chosen to measure the feature engineering efforts. Our results are based on one prepaid and one postpaid dataset of a European telecommunication provider.

The key contributions are threefold: 1) a new feature type classification; 2) a novel reusable methodology for determining optimal feature type combinations

based on Pareto multi-objective optimization, and 3) an experimental investigation and discussion of the trade-off. Our experimental results show that considering the trade-off between conflicting objectives provides better insights into the limitations of expanding the feature space. As such, our methodology and results can serve as a guideline for CP modeling in practice. For example, in contrast to our expectations, using more complex features (e.g. Page Rank Page et al. (1999), trends), does not improve PP while decreasing OE.

The paper is organized as follows: in Section 2, we provide an overview of related work. Section 3 introduces the methodology with Section 4 detailing the experimental setup. In Section 5, we present results, which are discussed in Section 6, before the paper is concluded in Section 7.

2. Related Work

In this section we provide a concise overview of related work.

2.1. Feature Engineering for Churn Prediction

Feature classification oftentimes entails a distinction between for instance sociodemographic, subscription-, payment-, and complaint-related features. However, with the uptake of social network analytics, the most adopted feature type classification became the one distinguishing between local (also known as traditional, individual) and network features (Dasgupta et al., 2008). Remarkably, no other feature type classifications have been proposed in the literature (see the first column in Table 1). As such, our proposal is considered a first step towards establishing a more fine-granular feature type classification for CP.

Regardless the absence of a ‘standardized’ classification, the existence of a wide variety of features types for CP is evidenced by the literature. This variety can be categorized along several dimensions, as shown in Table 1 (columns 3-6). However,

notice that the distinction between these dimensions is sometimes blurred. For example, usage-related information in (Kusuma et al., 2013) as well as call drop rate and page response success in (Huang et al., 2015) are considered as local features, although the underlying information essentially comes from Call Detail Records (CDRs), typically used to derive network features. Furthermore, these examples illustrate that features referred to as local might require a comparable or sometimes higher computational effort than network ones.

While several studies compare different feature types, these approaches significantly differ from the one presented in this paper, for several reasons. First, there exist differences in feature classification and churn definition between our and previous studies (and among themselves as well). Second, the comparison scope is typically very limited to either a strict comparison between different feature types (Dasgupta et al., 2008; Kim et al., 2014), or in the best case their combination as well (e.g. local vs. network vs. local+network in (Zhang et al., 2012; Kusuma et al., 2013) denoted as L:N:L+N in Table 1). However, none of the previous studies devises a customized way of combining many different feature types in the way our study does (see Table 1). Third, even though some works compare different modeling techniques and feature categories together (Dasgupta et al., 2008; Zhang et al., 2012; Backiel et al., 2014, 2016), comparisons of different modeling techniques are out of the scope of this study. Fourth and most importantly, other studies only consider a PP perspective, except partially for (Kim et al., 2014), while our study takes into account the OE/PP trade-off.

Finally, even though different studies undoubtedly agree that adding more features improves PP, they disagree on different feature types' importance: Kusuma et al. (2013) claim that network features alone do not express enough predictive power, while Kim et al. (2014); Backiel et al. (2014) claim the opposite. This provides an excellent motivation for analyzing the impact of feature types on PP at a

more fine-grained level.

Table 1: Overview of related work. The meaning of columns 3-6 is as follows: column 3 is an indicator whether recent (R) and/or historical (H) features were used. The number between parenthesis denotes the number of months used to calculate historical features, with symbol ‘:’ denoting that a comparison between feature types is performed. Column 4 reflects the use of local (L) vs. network (N) features, while Column 5 denotes the use of absolute (A) vs. trend (T) features, with symbol ‘*’ meaning that aggregation functions have been applied. Finally, column 6 indicates the use of the egonet (E) vs. higher-order neighborhood (H) for deriving network features. The last 4 columns position this work in terms of whether a customized OE/PP trade-off is made, whether MOO is applied, whether flexible prioritization of the multiple objectives is possible, and finally which general types of evaluation metrics were used.

Authors	New FT classification	Recent vs. Historical	Local vs. network	Absolute vs. trend	Egonet vs. Higher-ord. neigh.	Custom. FT comb.	OE/PP trade-off	MOO	Flexible priorit.	Evaluation measure
Buckinx & Van den Poel (2005)	/	R,H(4)	L	A*	/	/	/	/	/	PP
Dasgupta et al. (2008)	Yes	R,H(2)	N	A*	E,H	/	(OE)	/	/	PP
Coussement & Van den Poel (2009)	/	R,H(30)	L	A*	/	/	/	/	/	PP
Richter et al. (2010)	/	R	N	A*	E	/	OE	/	/	PP,OE
Wang et al. (2010)	/	R,H(11)	L	A*	/	/	Yes	Yes	No	PP,OE
Lee et al. (2011)	/	R,H(1)	L,N	A	/	/	OE	/	/	PP,OE
Benoit & Van den Poel (2012)	/	R,H	L:N	A*	E,H	/	/	/	/	PP
Huang et al. (2015)	/	R,H(3)	L:N	A,T	E	/	(OE)	/	/	PP
Zhang et al. (2012)	/	R,H(3)	L:N:L+N	A*	E,H	/	/	/	/	PP
Coussement & De Bock (2013)	/	R,H(16)	L	A*	/	/	(OE)	/	/	PP
Kusuma et al. (2013)	/	R,H(2)	L:N:L+N	A*	E,H	/	(OE)	/	/	PP
Modani et al. (2013)	/	R,H(1)	L,N	A	E,H	/	/	/	/	PP
Podgorelec & Karakatic (2013)	/	R	L,N	A	E	/	/	Yes	/	PP
Kim et al. (2014)	/	R,H(1)	L:N	A	H	/	Partly	/	/	PP
Backiel et al. (2014)	/	R,H(5)	L:N	A	E	/	/	/	/	PP
Huang et al. (2012)	/	R,H(6)	L,N	A	E,H	/	Partly	/	/	PP
Backiel et al. (2016)	/	R,H(3)	L:N	A,T	E	/	/	/	/	PP
This study	Yes	R:H(1)	L:N	A:T	E:H	Yes	Yes	Yes	Yes	PP,OE

2.2. Operational Efficiency and its Trade-Off with Predictive Performance

The necessity of measuring OE of classification models is recognized in several works (Dasgupta et al., 2008; Kusuma et al., 2013; Coussement & De Bock, 2013). Moreover, Richter et al. (2010) report the computational time required for experiments. However, a majority of CP-related studies, including these ones, do not consider the OE/PP trade-off (see column OE/PP trade-off in Table 1). Among those which at least mention the OE/PP trade-off are (Kusuma et al., 2013), observing that the small gain in PP of a hybrid (local+network) approach is not worth

the computational effort, and (Huang et al., 2012), discussing the computational complexity (and cost) of the applied models. Additionally, Lee et al. (2011) compare predictive accuracy, computational time and interpretability of artificial neural networks (ANN) and decision trees (DT) finding that ANN take more time while performing better than DT but without considering different feature types.

Kim et al. (2014) perform a more detailed discussion on the OE/PP trade-off, however, similar to the previous study, only considering variations of the modeling technique. Furthermore, Huang et al. (2015) is one of the rare studies to discuss the OE/PP trade-off taking into account resources needed for feature engineering (even though they only explicitly measure PP and not OE). Wang et al. (2010) conduct the most thorough study on the OE/PP trade-off is, looking into several PP measures such as accuracy, AUC, precision, recall, mean absolute error as well as OE measures such as train and test time. However, in contrast to our study, they do not perform any evaluation with respect to features. As such, our standpoint is that OE should be perceived from a broader perspective, primarily focusing on the data availability and feature engineering aspect, since the companies need insights into the return on investment of different feature types when starting their CP projects.

2.3. Multi-Objective Optimization (MOO)

Real-life problems in a variety of domains typically depend on optimization of several conflicting objectives. In these circumstances, a single optimal solution does not exist and thus a trade-off must be made. A Pareto multi-criteria optimization approach (Deb et al., 2016), based on concepts of Pareto dominance and Pareto optimality, has been widely used to solve such tasks. To efficiently retrieve Pareto optimal solutions in case of a high number of objectives, studies mainly resort to exploiting stochastic search algorithms like genetic algorithms (GA), either in their classical settings or proposing different customizations (e.g. Multi-Objective GA,

Non-dominated Sorting GA (NSGA), Multi-Objective Particle Swarm Optimization etc. (Coello et al., 2007)). A number of alternatives to Pareto optimization have been proposed as well. These include the so-called *lexicographic* method, where each of the objective functions is optimized separately (one at a time), followed by converting each obtained optimal solution to constraints and continuing the procedure for other objectives (Miettinen, 2012), and the *scalarization* method, where the task is reduced to single-objective optimization. A relevant example for our work, is the efficiency method $E_S(f) = \frac{\sum_i w_i m_i^+(f)}{\sum_j w_j m_j^-(f)}$, which enables evaluation of learning algorithms with respect to different measures of either positive (m^+) or negative (m^-) influence (w_i and w_j are the respective weights) Japkowicz & Shah (2011). However, we deem the aforementioned strategies inconvenient for handling the CT/AUC trade-off since they not only require an a priori assumption of objective preference and a convex search space, but also provide a single solution which cannot properly balance different objectives. Similarly, Wang et al. (2010) propose two different methods, TOPSIS+ and PROMETHEE II, used for MOO but in both cases a priori prioritization of objectives is required. Furthermore, even though their optimization objectives include training and testing time, they only consider MOO with respect to different classification techniques (variants of decision trees, logistic regression, sequential minimal optimization clustering-based classifiers, decision table, NaïveBayes and rule-based classifier) and not features. Podgorelec & Karakatic (2013) use the MPGA method to induce an optimal decision tree. In contrast, our methodology is based only on Pareto optimality and therefore, does not depend on the choice of the predictive model. Additionally, they consider only PP measures (true positives and true negatives) as objectives. Finally, observe that in (Huang et al., 2010), MOO using NSGA-II has been applied for feature selection in CP, but again focusing solely on performance metrics.

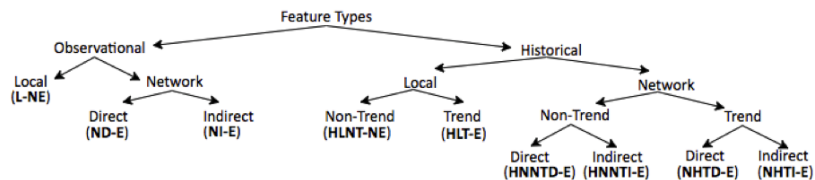


Figure 1: Hierarchy of feature types.

3. Methodology

This section introduces the proposed fine-grained feature type classification as well as our Pareto-based method for finding optimal feature type combinations.

3.1. A Fine-Grained Feature Type Classification for Churn Prediction

One of the data aspects that we focus on in this work is to expand the predictive potential of our data, which we achieve by carefully engineering additional features. Based on the considerable amount of literature discussing experiments with different types of features for CP, we devise a fine-grained classification consisting of nine feature types, hierarchically represented in Figure 1 and described in more detail in Table 2. A first distinction is made between features using the most recent data (referred to as observational) and features using older data (referred to as historical, in our case, the previous month). Next, locality of the data is exploited, distinguishing between local and network features. However, unlike current CP-related literature, we further categorize network features into direct (simple) and indirect (complex), an idea from the fraud detection domain (Baesens et al., 2015). As for direct network features we consider those which can be derived from the 1st-order neighborhood (also known as egonet), while those whose computation requires exploring higher orders of the neighborhood (and sometimes even the whole network, hence the name complex) are considered as indirect features. Since both node degree information and RFM (Recency-Frequency-Monetary) features (Huges, 1994)

are typically used in the literature to capture customer behaviour, we consider these as our direct network features. To adjust for our particular domain, the node degree information is represented by the number of different MSISDNs¹, that is, the number of unique mobile phone numbers that a customer interacts with. With RFM, we account for recency of calls, number of calls and monetary value of calls per customer within a certain period of time. In order to capture more detailed information, in the final list of features each of the RFM (and degree-related) features is assessed in several different flavors, along following dimensions for the observational version: outgoing calls towards home operator, outgoing calls towards other operators and incoming calls, and additionally along incoming w.r.t. churners and outgoing w.r.t. churners for historical version. E.g. for call recency, we calculate five features: recency of incoming calls, recency of outgoing calls towards home operator/other operators, recency of incoming calls from/outgoing calls towards churners (see Table 2).

For indirect network features, we propose taking into account 2^{nd} -order degree of a node, the number of triangles and quadrangles that a node belongs to and Page Rank score of a node.

Finally, as the same features are calculated for two consecutive months, a non-trend vs. trend classification arises as a logical consequence². Next to each feature type, we add a ‘non-engineered’ (‘NE’)/‘engineered’ (‘E’) label, to ease the distinction between features not-requiring/requiring any further preprocessing (model-dependent pre-processing is not considered)³.

¹acronym for Mobile Station International Subscriber Directory Number

²Relative trend is calculated $\left(x = \frac{x_M - x_{M-1}}{x_{M-1}}\right)$, not the absolute one $(x = x_M - x_{M-1})$.

³Initially, we considered also dividing local features further in NE/E, but eventually decided to abandon this idea due to the fact that in case of local features, this characteristic heavily depends on the dataset at hand. For example, the same feature, age, if provided in the dataset should be considered as ‘NE’, while, if calculated from date of birth, should be considered as ‘E’. In case local

Table 2: The explanation of devised feature types. The definition column explains the nature and origin of different feature types and as such can be applied to any type of dataset (even beyond the telco domain).

Abbrev.	Feature Type Name	Definition	Example
L	Local	Features that characterize individual customer (also known as traditional)	gender; number of reloads; handset characteristics (e.g. POLYPHONIC);
ND-E	Network Direct Engineered	Features calculated from the customer ego-network (1 st -level neighbourhood) across different dimensions/granularities	number of incoming/outgoing toward home operator calls in month M
NI-E	Network Indirect Engineered	Network features which cannot be calculated from customer ego-network only	Page Rank score; 2 nd degree of a node in month M
HLNT-NE	Historical Local Non-Trend Non-Engineered	The same type of features as local (L), except that they refer to the one month before the observed month (month $M-1$)	handset characteristics; number of reloads in month $M-1$
HLT-E	Historical Local Trend Engineered	The same type of features as local (L), except that they refer to the one month before the observed month (month $M-1$)	trend in number of reloads; recharge trend
HNNTD-E	Historical Network Non-trend Direct Engineered	The same type of features as ND-E, except that they refer to the one month before the observed month (month $M-1$)	number of incoming/outgoing toward home operator calls in month $M-1$
HNNTI-E	Historical Network Indirect Engineered	Historical (one month ahead i.e. month $M-1$) versions of the NI-E variables	Page Rank score; 2 nd degree of a node in month $M-1$
NHTD-E	Historical Network Trend Direct Engineered	Trend features calculated based on direct network features corresponding to month M (ND-E), and direct network features corresponding to month $M-1$ (HNNTD-E)	trend in number of incoming calls; trend in number of outgoing calls
NHTI-E	Historical Network Trend Indirect Engineered	Trend features calculated based on indirect network current (NI-E) and indirect network historical (HNNTI-E) features	trend in 2 nd degree of a node; trend in Page Rank of a node

3.2. Finding Optimal Feature Type Combinations

The main goal of this work is to develop a method that can identify feature types (or combinations) which provide the best CT/AUC trade-off. Hereto, a novel forward-backward approach based on Pareto multi-objective optimization is devised, whose schematic illustration can be seen in Figure 2.

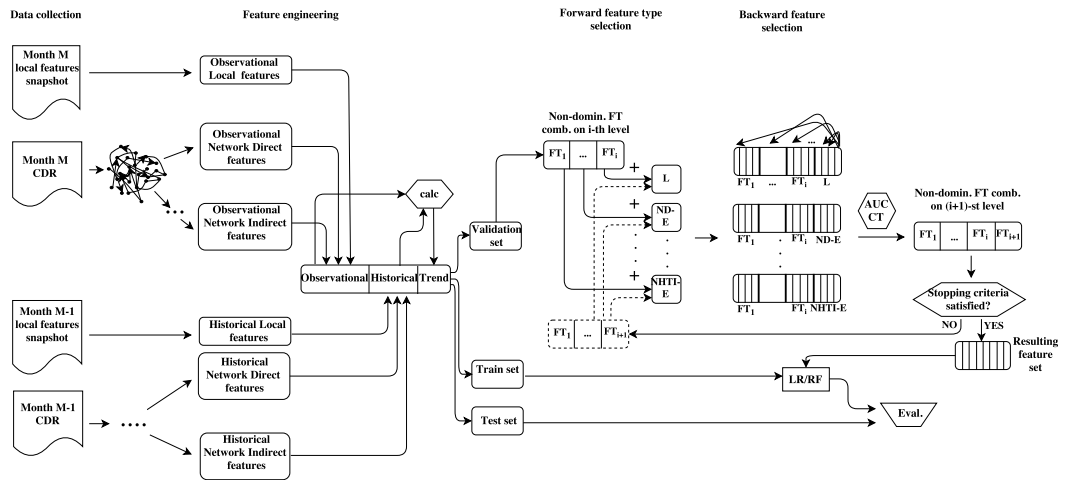


Figure 2: A schematic illustration of our approach.

3.2.1. Pareto Multi-Criteria Optimization

Pareto multi-objective optimization takes into account the trade-off between conflicting objectives and proposes several optimal results, known as Pareto optimal

features require significant computational effort, e.g. Huang et al. (2015) as explained in Subsection 2.1, this division can be reconsidered.

solutions (Deb et al., 2016). More formally, given objective functions $f_i, i \in I; |I| \geq 2$, the task $\min_i(f_i(x)), x \in X$ is called the multi-objective optimization task.⁴ Let $Z = \{z^t\}$ be the set of all feasible solutions. We say that a solution z^1 is a dominating solution over z^2 if $\forall j \in I : f_j(z^1) \leq f_j(z^2)$ and $\exists k \in I$ such that: $f_k(z^1) < f_k(z^2)$. In other words, one solution is dominating another if it is better than it in at least one objective and not worse in all other objectives. Solution z^* is optimal if $\nexists z^{**} \in Z$ s.t. z^{**} dominates z^* (no other solution dominates it). The set of all Pareto optimal solutions is called the Pareto set, while the set of corresponding objective values is called the Pareto frontier.

3.2.2. Customized Forward-Backward Feature Selection

Our approach is an iterative two-step approach whereby, in the first step, the process of combining feature types is performed in a forward manner (in increments of one feature type) based on Pareto optimal solutions, while in the second step, the feature selection for each combination of feature types follows a backwards approach. The reason for proposing such a design is threefold. First, by incrementally adding feature types, our methodology allows for maximal flexibility in terms of exploiting different feature type combinations. Second, backward feature selection ensures that only informative features within a particular feature type combinations are retained, which is relevant as not all features of the same feature type contribute equally in PP. Third, by guiding the whole process using Pareto multi-criteria optimization, we ensure that the most optimal path in selecting features is performed, allowing for reaching optimal solutions from the trade-off perspective.

The forward part represents the skeleton of our approach, driven by Pareto multi-criteria optimization, which has as an outcome a Pareto optimal set of fea-

⁴This scenario was adapted for simplicity and ease of explanation, but similarly, the task $\max_i(f_i(x))$ or any combination of maximizing/minimizing goals could be defined as well.

ture types, as shown in Algorithm 1.

Algorithm 1 Finding Pareto optimal sets among different feature types

Input: $\mathcal{F} = \{FTS\}_k$, a collection of different feature type sets FTS, $k \in \{1, \dots, 9\}$

- 1: Calculate an initial set ND_1 of non-dominated solutions in \mathcal{F} (w.r.t. AUC & CT)
- 2: $prev := 1$;
- 3: **repeat**
- 4: $ND_{prev+1} := \{\}$;
- 5: **for** $\forall NDS \in ND_{prev}$ **do**:
- 6: $Candidate_NDS := \{\}$; /* Collection of Candidate_NDS */
- 7: **for** $\forall FTS \in \mathcal{F}$ **do**:
- 8: **if** $FTS \notin NDS$ **then**:
- 9: $Candidate_NDS = NDS \cup FTS$;
- 10: $Calc_AUC_CT(Candidate_NDS)$;
- 11: $Candidate_NDS += Candidate_NDS$;
- 12: **end if**;
- 13: **end for**;
- 14: Add all non-dominated solutions from $Candidate_NDS$ to ND_{prev+1} ;
- 15: **end for**;
- 16: $prev += 1$;
- 17: **until** stopping criteria; /* all FTS exhausted or a certain performance reached*/
- 18: $ND = \bigcup_i ND_i$;

Output: the non-dominated solutions ND ;

At first, the set of all feasible solutions is the collection of all the possible subsets (except the empty set) of a collection of different feature type sets \mathcal{F} . To determine the optimal ones, we perform several iterations, each of which (except the first), starts from the non-dominated solutions that are found in previous iterations. Within the new iteration, each of these non-dominated solutions is extended with every single remaining feature type (without repetition of already explored feature types) and, based on the achieved AUC and CT, the new set of non-dominated solutions for that iteration is identified. Therefore, the cardinality C_i of each combination of feature types per iteration i is exactly one feature type higher than the

cardinality C_{i-1} of the previous iteration (forward step). The process terminates when either no additional feature types can be added or when the obtained performances of at least one non-dominated solution dominates the solution obtained using all feature types (stopping criteria, line 17). As already mentioned, every feature type combination is evaluated in terms of CT and AUC. In Algorithm 1 this part is embedded into the *Calc_AUC_CT* function, but the details are provided in Algorithm 2. Logistic regression (LR) without regularization and Random Forests (RF) are used for model construction. Both LR and RF are well established methods and have been successfully applied in previous studies on CP, e.g., LR was used in (Buckinx & Van den Poel, 2005; Coussement & Van den Poel, 2009; Wang et al., 2010; Zhang et al., 2012; Kusuma et al., 2013; Lee et al., 2011), while RF was employed in (Benoit & Van den Poel, 2012; Buckinx & Van den Poel, 2005; Coussement & Van den Poel, 2009; Coussement & De Bock, 2013).

For the backward part of our approach, a slightly modified version of the classical backward feature selection approach has been devised. Similarly, we start with all features and eliminate exactly one feature in every step, namely the one without which we are obtaining the highest AUC performance as compared to all other features. However, differently from the classical approach, we do not define a stopping criterion beforehand. Instead, we first complete the whole feature elimination process (until no further features are left). Then, the optimal feature set is selected as the iteration having retained the lowest number of features among all those iterations where the AUC score is still higher than the maximum AUC score minus one standard deviation (see line 18 in Algorithm 2). There are several reasons for this approach. First, given that AUC is one of our objectives, we opt for an AUC-driven approach, in contrast to other approaches based on AIC/BIC criteria and/or p-values. Second, it was not feasible to apply the classical stepwise approach since, due to very subtle and, even more importantly, non-monotonic AUC fluctu-

Algorithm 2 Calculating AUC & CT for each feature type set combination

Input: $FC := \bigcup_{i,j} \{f_i^j\}$, $f_i^j \in FTS_i$, $FTS_i \in$ a particular feature type set combination

```
1:  $Res := \{\}$ ;          /* result set */
2:  $s := 1$ ;             /* step id */
3:  $Remaining\_F := FC$ ;    /* set of remaining features */
4: while  $card(Remaining\_F) > 0$  do:
5:    $Res_s := \{\}$ ;      /* set of obtained results at step s */
6:   for  $\forall$  feature  $f \in Remaining\_F$  do:
7:     Create LR/RF model  $M_f$  using features  $Remaining\_F \setminus \{f\}$ ;
8:     Calculate  $AUC_f, CT_f$  for  $M_f$ ;
9:     Add  $(s, M_f, AUC_f, CT_f)$  to the result set  $Res_s$ ;
10:  end for;
11:   $f_s = argmax_f (AUC_f), AUC_f \in Res_s$ ;
12:   $Remaining\_F = Remaining\_F \setminus \{f_s\}$ ;
13:  Add  $(s, M_{f_s}, AUC_{f_s}, CT_{f_s})$  to the result set  $Res$ ;
14:   $s := s + 1$ ;
15: end while;
16:  $m = max_f (AUC_f), AUC_f \in Res$ ;
17:  $sd = std_f (AUC_f), AUC_f \in Res$ ;
18:  $ind\_step = max_i (s_i)$  s.t.  $AUC_{s_i} > m - sd$ ,  $s_i \in Res$ ;
Output:  $AUC_{ind\_step}, CT_{ind\_step}, M_{ind\_step}$ 
```

ations, determining appropriate thresholds for feature elimination/addition was unexpectedly hard. Despite a lot of different stepwise alternatives known in literature (Wagner & Shimshak, 2007), there are no concrete guidelines for determining these thresholds and our attempts to empirically evaluate them fell short, leading to either retention or elimination of almost all the features, thus conflicting the whole purpose of CT/AUC trade-off analysis. Moreover, for certain feature sets, additional constraints had to be imposed to avoid the repetition of the same feature combinations over time during the stepwise selection process. Third, opting for the stopping point with the minimal number of features, while still keeping AUC in the range of one standard deviation from the maximum AUC, complies with our targeted compromise between AUC and CT. Hence, the introduced modification allowed us to

make the feature selection process in line with both of the considered objectives, less biased by predefined choices and more data driven instead.

4. Experimental Setup

4.1. Data and Tools

Experiments were performed using two data sets (one for prepaid, one for post-paid). Both consist of CDRs and monthly snapshots of local features. CDRs contain information about customer calls (no SMS or other usage types), in the form of: caller, callee, date/time, call duration. Local features are mostly related to customer handset characteristics, the date of the first usage and tariff plan information. Additionally, for prepaid customers we have information about the last recharge and the amount spent on voice and SMS during the observed month, while for postpaid, we have information about whether the customer has a fixed-time contract and if so, the number of days left till its expiration. Regarding demographic data, we are only provided with the zip code plus the date of birth for postpaid customers. In addition, for postpaid, the monthly snapshot for ported-out customers (with ported-out dates) is provided as well. A customer has ported out if (s)he has officially requested to switch to services of another provider while keeping his/her current telephone number.

4.2. Network Construction and Direct/Indirect Network Feature Generation

Initially, we construct a single unweighted graph from CDRs in a standard way. In some of the previous works, during the process of generating call graphs, calls are neglected if their duration is less than 10 seconds (Nanavati et al., 2006), or if the appropriate weights are not among the top 10% (Richter et al., 2010). Unlike these, we retain every single call⁵ (note that the number of zero-length calls is less than

⁵Short calls can reveal a customer "dependency" on the other customer, which becomes even more relevant in case the latter churns. Additionally, short calls do not have to be less profitable. E.g. if

0.05% on a monthly level). Furthermore, we take into account the direction of the call, hence the obtained network is directed (and asymmetric) with approximately 5 million edges and 2.8 million of nodes.

The obtained network is used to calculate direct network features. Due to lack of the exact amount charged, monetary features (in RFM) are determined based on call duration.

For generating indirect network features, we first transform the initially obtained network into (two different) weighted networks, to quantify the strength of interaction between customers, as that supposedly increases CP performance (Dasgupta et al., 2008). Edge weights are assigned based on the total number of calls between two nodes, for the first, and total duration (that is, total number of seconds) of the calls between two nodes, for the second network (with both we account for direction). We are using these weighted networks to calculate two versions of node Page Rank scores and for this, we perform certain alterations. Namely, in the contrast to the approaches where each caller/callee is represented by a node or where all the callers/callees of other providers are represented with a single node in a network (Backiel et al., 2014) or even where each provider is also represented by a node (Zhuang et al., 2015), we completely leave out other providers' customers and interactions with them. This is motivated by the fact that we do not have the outgoing traffic of these customers and this, in both of the mentioned representations, could potentially lead to having multiple (in the first case) or single (in the second case) sink nodes when applying the Page Rank algorithm. When calculating the number of triangles and quadrangles that a node belongs to, we impose an additional simplification of weighted networks and consider them undirected, to reduce the

they both have a tariff plan charging every 60 seconds, customer A making a call of 5 seconds and customer B making a call of 55 seconds will be charged equally.

computational effort. We also calculated (two versions of) personalized Page Rank scores based on exponential time decays (to favor more recent churn dates from the less recent ones).

4.3. Definitions and Setting

One of the challenges of CP is to determine when the customer has actually churned. In our data sets, we are constrained by having neither ported-out information of prepaid customers nor the churn date for postpaid customers (except ported-out dates for only a dozen). Given these restrictions and also the limited number of explanatory features (see Section 6), we defined churners (for both prepaid and postpaid) as those customers having traffic in one month, but not having traffic in the consecutive month.

As mentioned earlier, not only accurately, but also timely predicting churn is crucial. Therefore, we introduce a one-month gap between observation and prediction period: for the customer base of month M , we are trying to predict who will churn in month $M+2$ (see Figure 3). Under these premises, the obtained churn

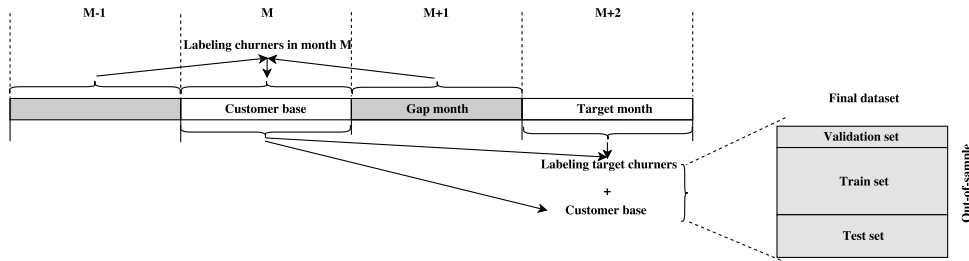


Figure 3: An illustration of our one-month gap approach.

rates are 7.15% and 1.54% for the prepaid and postpaid dataset respectively. The influence that one-month gap has on PP will be discussed later in Section 6.

The exact number of features per feature type per dataset can be seen in Table 3. Please note that the set of HLT-E features is omitted for the postpaid dataset as

none of the postpaid local features is continuous.

To avoid multi-collinearity, Chi-square and Spearman correlation tests with a confidence level of 95% were applied to all pairs of categorical and continuous features respectively. As a result, several features (mostly related to handset characteristics, but also personalized Page Rank scores) were eliminated.

Table 3: Number of features per feature type and the total number of features per dataset.

Dataset	L	ND-E	NI-E	HLNT-NE	HLT-E	HNNTD-E	HNNTI-E	NHTD-E	NHTI-E	Total
Prepaid	21	12	5	21	4	20	5	12	3	103
Postpaid	13	12	5	13	N/A	20	5	12	3	83

5. Results

Algorithm 1 can be driven by various stopping criteria. In our case, we decided to stop after having explored combinations of three different feature types, since for both prepaid and postpaid, the obtained AUC is comparable to the one obtained with all features. In the two following subsections, we will first focus on the trade-off before briefly analyzing the retained features.

5.1. The Trade-Off between Operational Efficiency and Predictive Performance

The results of the Pareto search for the prepaid dataset is depicted in Figure 4 for LR/RF (top/bottom), with CT represented relatively to the CT obtained using all features. All ‘shortlisted’ candidate solutions are shown, that is, all non-dominated solutions from the previous iterations (corresponding to line 18 of the Algorithm 1). Among these, the final set of non-dominated solutions is identified (filled circles/squares in red for LR/RF, respectively). Given our stopping criterion, no other solutions exist which could dominate any of the found non-dominated solutions. Consequently, these non-dominated solutions are Pareto optimal solutions forming the Pareto optimal frontier (dotted/dashed red line for LR/RF). Dominated solutions are marked with blue empty circles/squares for LR/RF. Taking a closer look into

the set of optimal solutions (seven for LR and four for RF), it can be seen that they consist of only five different feature types: L, ND-E, HLNT-NE, HNNTD-E and HLT-E in case of LR, and only four (the same as for LR, except for HLT-E) in case of RF. Not surprisingly, RF and LR provide different results in terms of AUC per combination. However, optimal solutions for LR and RF share the following three different combinations, very relevant from the perspective of the CT/AUC trade-off: 1) L&HLNT-NE (denoted by 1), due to the minimal CT required; 2) L&ND-E&HLNT-NE (denoted by 4), due to its maximal AUC performance (according to RF); 3) ND-E&HLNT-NE (denoted by 2), which can be perceived as a compromise between the two previous solutions. It is important noticing that all of these contain the historical local feature type. However, there are no linear dependencies between performances of different solutions (see Appendix A for exact figures). Additionally, the non-parametric pairwise DeLong, DeLong, Clarke-Pearson statistical test (DeLong et al., 1988) shows significant difference in AUC scores for these combinations at the 95% confidence level (see Table C.8 and Table C.9 in Appendix C). Therefore, additional expert information is needed to support the decision on which feature types to include.

The ‘shortlisted’ solution candidate set (line 18 of the Algorithm 1) for the post-paid dataset is depicted at the top (for LR) and bottom (for RF) of Figure 5. Here, out of an initial set of 10 non-dominated solutions (both in case of LR and RF), only four (for LR) and five (for RF) are Pareto-optimal (marked as before). As expected, RF and LR provide different AUC scores per combination, but again, optimal solutions obtained by both LR and RF have three different combinations in common (denoted by 1, 3 and 4). It is worth noticing that all of these contain local (observational) features. In this case, the pairwise DeLong, DeLong, Clarke-Pearson test does not show any significant difference between AUC scores of solutions 3 and 4. However, AUC scores for best scoring solutions (9 and 7) are significantly different

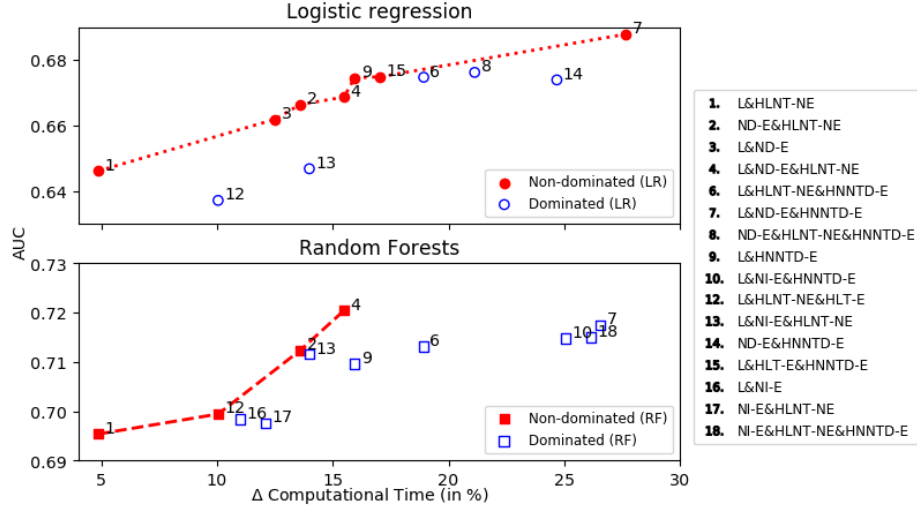


Figure 4: The shortlisted solutions for the prepaid dataset and their performance in CT (relatively to the CT obtained using all features; x -axis) and AUC (y -axis) obtained applying LR (top) and RF (bottom). The optimal solutions are marked with red filled circles/squares while the dominated solutions are blue empty circles/squares for LR/RF, respectively.

from the AUC scores of other optimal solutions at the 95% confidence level (see Table C.10 and Table C.11 in Appendix C). Similar as with prepaid, a mixture of local and observational direct network features (denoted by 4) scores better in terms of AUC, but it also requires more computational time (around 8%). Different combinations of observational local and historical network features (denoted by 6,7 and 9), although computationally more demanding (but still remain within a 10-20% margin), score by far the best in terms of AUC (both 9 and 7 outperform the “full” model for LR and RF, respectively). As in the case of prepaid, expert information would be needed to finally decide on which feature types to use. However, it is crucial to notice that local features are present in every optimal set.

Quite surprisingly, using trend and indirect network features does not improve AUC performance in spite of increased computational complexity. In general, the

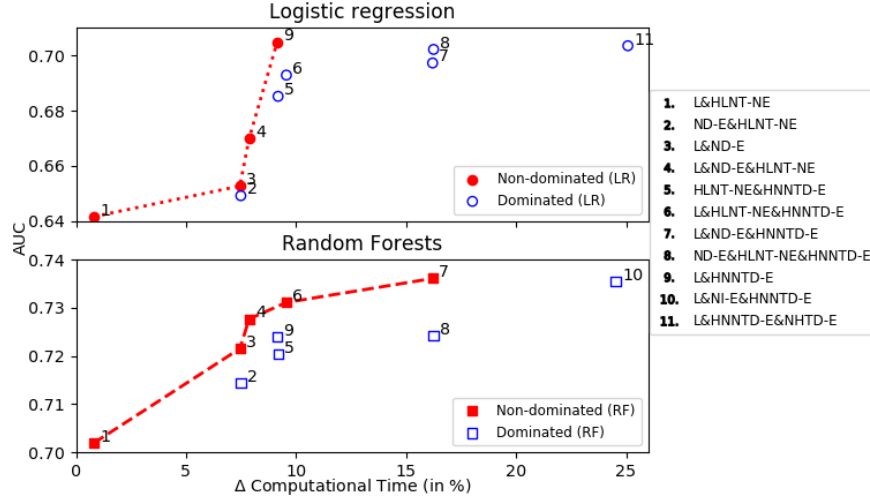


Figure 5: The shortlisted solutions for the postpaid dataset and their performance in CT (relatively to the CT obtained using all features; x -axis) and AUC (y -axis) obtained applying LR (top) and RF (bottom). The optimal solutions are marked with red filled circles/squares while the dominated solutions are blue empty circles/squares for LR/RF, respectively.

highest AUC scores for both prepaid and postpaid datasets are obtained using the combination of local and historical direct network features. While we ground our approach on AUC due to its robustness, we do include all evaluations in terms of AUC, top-decile lift and EMP (Expected Maximum Profit by Verbraken et al. (2013)) for all shortlisted solutions for both datasets and models (Logistic Regression and Random Forests) in the Appendix A.

5.2. Feature engineering

Next, we examine feature types and features retained in the model for each of the optimal solutions. The results are presented in Figure 6 for prepaid RF and Figure 7 for postpaid RF (due to better AUC scores we explain RF results in more detail, but equivalent figures are provided in Appendix B for LR). Each feature is uniquely represented by a symbol, determined by its color and shape. The color

corresponds to a particular feature type (e.g. blue for ND-E and red for HNNTD-E). The presence of the same shape in a different color indicates that the same feature is retained both in its observational and historical version, e.g. for prepaid, the amount spent on calls (denoted by ♠) is retained both in the observational and historical version.

It can be observed that for both prepaid and postpaid (and both for RF and LR), only features of four feature types remain in the finally retained feature sets: L, HLNT-NE, ND-E, HNNTD-E. However, local and direct network features seem to be of different importance for prepaid and postpaid datasets.

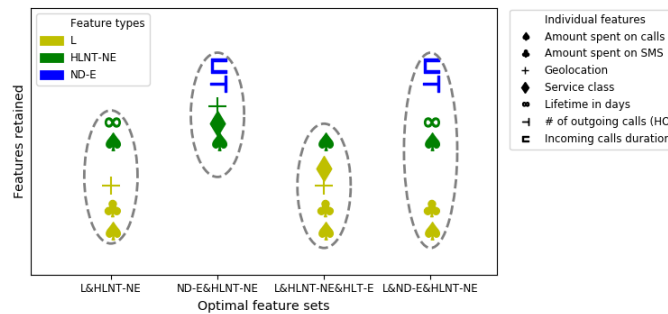


Figure 6: Retained features (in the RF model) for Pareto optimal feature type combinations for the prepaid dataset. The feature type combinations are sorted by increasing order of AUC performance (from left to right). HO stands for ‘home operator’, while OO stands for ‘other operator’

For prepaid RF, several different repetitive arrangements of retained features can be observed. First, only ten different features are retained for all four Pareto optimal sets, with four of them being local observational, four local historical and two observational network direct (hence, no trend features are retained, although HLT-E appears in one of Pareto optimal sets). Second, the amount spent seems to be of crucial importance since both observational and historical version of the amount spent on calls (♠) and observational version of amount spent on SMS (♣) remain retained in all feature sets. Third, for Pareto optimal sets containing the

(observational) network direct feature set, the combination of number of outgoing calls towards home operator (−) and incoming calls duration (⊞) is retained. Lastly, lifetime in days, service class that the customer belongs to and customer geolocation also reoccur in the retained feature sets, either in their observational or historical versions.

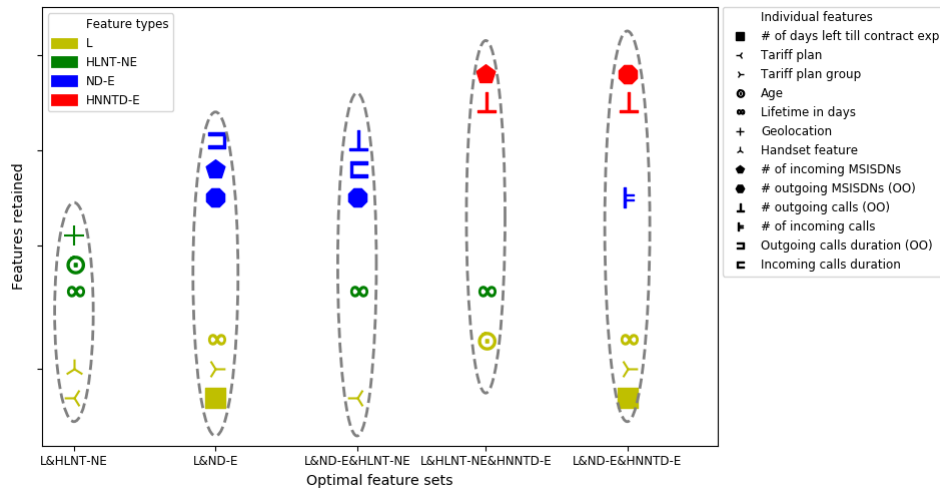


Figure 7: Retained features (in the RF model) for Pareto optimal feature type combinations for the postpaid dataset. The feature type combinations are sorted by increasing order of AUC performance (from left to right). HO stands for ‘home operator’, while OO stands for ‘other operator’.

For postpaid RF, in total 18 different features are retained, of which: six local (observational), three local historical, six direct network (observational) and three historical direct network. Similar to prepaid, several patterns can be observed. First, loyalty seems to play a vital role as the feature lifetime in days (∞) appears in all five Pareto optimal feature type combinations. Second, contract information is another important aspect since in four Pareto optimal solutions, either tariff plan (\leftarrow) or number of days left till contract expiration (\blacksquare) followed by tariff plan group (\rightarrow) feature are retained. Third, regarding network features, not surprisingly, the num-

ber of different outgoing MSISDNs towards other operators (◆) and the number of outgoing calls towards other operators (⊥) remain in three out of five final feature sets, either in observational or historical flavor. Although less frequently, the number of incoming MSISDNs (◆) also reoccurs in the final feature set. Likewise, the age (⊙) is retained twice in the final feature set.

6. Discussion

Our experiments showcase several important findings. First, in contrast to previous studies, the choice of modelling technique influences results, although the sets of optimal solutions overlap. Second, the results obtained for Pareto optimal solutions are discontinuous, hence there is no ‘middle’ solution which could be chosen as a real CT/AUC trade-off. This allows for more flexible decisions although subjective preference still remains a necessary element for decision making (e.g. if CT is more critical factor, local features alone could be used). Third, as it was shown, best results with respect to CT are obtained using local features (both observational and historical), while the highest AUC is obtained by a mixture of local and direct network features. This confirms the result of (Kusuma et al., 2013) (and contrasts the findings in (Kim et al., 2014; Backiel et al., 2014)) that local features should definitely not be neglected. Moreover, the demonstrated importance of local and direct network features gives an incentive to companies to invest into their quality.

We are aware of several limitations which led to lower AUC scores than usually reported in the CP studies. A first limitation refers to data availability given that many previous studies could benefit from additional data as indicated in Section 1 and Section 2.1. As for network features, we based our approach on RFM (already successfully applied in the CP domain (Benoit & Van den Poel, 2012; Coussement & De Bock, 2013; Modani et al., 2013)), Page Rank scores (also used in (Huang et al., 2015)) and other measures inspired by Baesens et al. (2015). However, we are

aware that exploiting other structural characteristics of the network (e.g. betweenness or eigenvector centrality like in (Kim et al., 2014)) might lead to better performances, as claimed by Dasgupta et al. (2008). Another reason for achieving lower AUC scores is our one-month gap approach. We confirmed the latter by applying our method to the scenario without the one month gap (predicting immediately for the next month), and we indeed obtained higher AUC scores. However, due to the importance of timely detecting churners, we decided to stick to the current setting.

Lastly, we opt for experimenting with combinations of up to three feature types which provided on average 0.78% better AUC score and only 16.85% of the required CT, as compared to the full model (measured for four cases: prepaid/postpaid LR/RF). However, the number of feature types can be easily adapted as required.

7. Conclusion

Experimenting with different features and feature types in order to improve PP has been the most recent and most popular research direction for telco CP. While performance is unquestionably an important factor, given the different levels of maturity regarding data availability, computational power and predictive analytics of different operators in different countries, with this work we raise a justified, yet still within current literature not addressed, question of examining the OE/PP trade-off.

The key contributions of this work are threefold. First, we devise a new feature type classification. Second, we propose a novel, reusable method for determining optimal solutions, based on Pareto multi-objective optimization. The method requires no a priori preference between conflicting objectives, while it still allows for making an informed decision based on the Pareto-optimal solutions. Third, we perform an experimental investigation of the trade-off. The obtained results demonstrate that the choice of modelling technique matters. Nevertheless, prediction-

wise, local and direct network features convey valuable information, which gives an incentive to telco operators to invest in their quality and availability. Furthermore, we observe that investing in certain more complex feature types like trends and indirect network features does not pay off in terms of PP. Therefore, even though in the end, the operational decision will probably have to account for some kind of subjective preference, we can claim that the right approach for choosing feature types for CP would be to start small, using (good quality) local features and the least complex network features (e.g. RFM based).

We performed our analysis on one prepaid and one postpaid dataset from the same mobile operator, which is an exceptional situation and raises justified concern of external validity. Therefore, to secure the validity of findings, for our future work we would like to experiment with a larger number of datasets and verify whether our findings are operator (and country) dependent.

References

- Backiel, A., Baesens, B., & Claeskens, G. (2014). Mining telecommunication networks to enhance customer lifetime predictions. In *International Conference on Artificial Intelligence and Soft Computing* (pp. 15–26). Springer.
- Backiel, A., Baesens, B., & Claeskens, G. (2016). Predicting time-to-churn of prepaid mobile telephone customers using social network analysis. *Journal of the Operational Research Society*, 67.
- Baesens, B., Vlasselaer, V. V., & Verbeke, W. (2015). Social network analysis for fraud detection. *Fraud Analytics Using Descriptive, Predictive, and Social Network Techniques: A Guide to Data Science for Fraud Detection*, (pp. 207–278).

- Benoit, D. F., & Van den Poel, D. (2012). Improving customer retention in financial services using kinship network information. *Expert Systems with Applications*, 39, 11435–11442.
- Buckinx, W., & Van den Poel, D. (2005). Customer base analysis: partial defection of behaviourally loyal clients in a non-contractual fmcg retail setting. *European Journal of Operational Research*, 164, 252–268.
- Coello, C. A. C., Lamont, G. B., Van Veldhuizen, D. A. et al. (2007). *Evolutionary algorithms for solving multi-objective problems* volume 5. Springer.
- Coussement, K., & De Bock, K. W. (2013). Customer churn prediction in the online gambling industry: The beneficial effect of ensemble learning. *Journal of Business Research*, 66, 1629–1636.
- Coussement, K., & Van den Poel, D. (2009). Improving customer attrition prediction by integrating emotions from client/company interaction emails and evaluating multiple classifiers. *Expert Systems with Applications*, 36, 6127–6134.
- Dasgupta, K., Singh, R., Viswanathan, B., Chakraborty, D., Mukherjea, S., Nana-vati, A. A., & Joshi, A. (2008). Social ties and their relevance to churn in mobile telecom networks. In *Proceedings of the 11th international conference on Extending database technology: Advances in database technology* (pp. 668–677). ACM.
- Deb, K., Sindhya, K., & Hakanen, J. (2016). Multi-objective optimization. In *Decision Sciences: Theory and Practice* (pp. 145–184). CRC Press.
- DeLong, E. R., DeLong, D. M., & Clarke-Pearson, D. L. (1988). Comparing the areas under two or more correlated receiver operating characteristic curves: a nonparametric approach. *Biometrics*, (pp. 837–845).

- Diaz-Aviles, E., Pinelli, F., Lynch, K., Nabi, Z., Gkoufas, Y., Bouillet, E., Calabrese, F., Coughlan, E., Holland, P., & Salzwedel, J. (2015). Towards real-time customer experience prediction for telecommunication operators. In *Big Data (Big Data), 2015 IEEE International Conference on* (pp. 1063–1072). IEEE.
- Flores-Méndez, M. R., Postigo-Boix, M., Melús-Moreno, J. L., & Stiller, B. (2016). A model for the mobile market based on customers profile to analyze the churning process. *Wireless Networks*, (pp. 1–14).
- Huang, B., Buckley, B., & Kechadi, T.-M. (2010). Multi-objective feature selection by using nsga-ii for customer churn prediction in telecommunications. *Expert Systems with Applications*, *37*, 3638–3646.
- Huang, B., Kechadi, M. T., & Buckley, B. (2012). Customer churn prediction in telecommunications. *Expert Systems with Applications*, *39*, 1414–1425.
- Huang, Y., Zhu, F., Yuan, M., Deng, K., Li, Y., Ni, B., Dai, W., Yang, Q., & Zeng, J. (2015). Telco churn prediction with big data. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data* (pp. 607–618). ACM.
- Huges, A. (1994). *Strategic Database Marketing: The Master Plan for Starting and Managing a Profitable, Customer-Based Marketing program*. Probus Publishing Co., Chicago, IL.
- Japkowicz, N., & Shah, M. (2011). *Evaluating learning algorithms: a classification perspective*. Cambridge University Press.
- Kim, K., Jun, C.-H., & Lee, J. (2014). Improved churn prediction in telecommunication industry by analyzing a large network. *Expert Systems with Applications*, *41*, 6575–6584.

- Kusuma, P. D., Radosavljevik, D., Takes, F. W., & van der Putten, P. (2013). Combining customer attribute and social network mining for prepaid mobile churn prediction. In *Proc. the 23rd Annual Belgian Dutch Conference on Machine Learning (BENELEARN)* (pp. 50–58).
- Lee, H., Lee, Y., Cho, H., Im, K., & Kim, Y. S. (2011). Mining churning behaviors and developing retention strategies based on a partial least squares (pls) model. *Decision Support Systems*, *52*, 207–216.
- Miettinen, K. (2012). *Nonlinear multiobjective optimization* volume 12. Springer Science & Business Media.
- Modani, N., Dey, K., Gupta, R., & Godbole, S. (2013). Cdr analysis based telco churn prediction and customer behavior insights: A case study. In *International Conference on Web Information Systems Engineering* (pp. 256–269). Springer.
- Nanavati, A. A., Gurumurthy, S., Das, G., Chakraborty, D., Dasgupta, K., Mukherjea, S., & Joshi, A. (2006). On the structural properties of massive telecom call graphs: findings and implications. In *Proceedings of the 15th ACM international conference on Information and knowledge management* (pp. 435–444). ACM.
- Page, L., Brin, S., Motwani, R., & Winograd, T. (1999). *The PageRank citation ranking: Bringing order to the web.*. Technical Report Stanford InfoLab.
- Podgorelec, V., & Karakatic, S. (2013). A multi-population genetic algorithm for inducing balanced decision trees on telecommunications churn data. *Elektronika ir Elektrotechnika*, *19*, 121–124.
- Richter, Y., Yom-Tov, E., & Slonim, N. (2010). Predicting customer churn in mobile networks through analysis of social groups. In *SDM* (pp. 732–741). SIAM volume 2010.

- Verbeke, W., Dejaeger, K., Martens, D., & Baesens, B. (2010). Customer churn prediction: does technique matter? In *Proceedings of the Joint Statistical Meeting, JSM2010, Vancouver, Canada*.
- Verbraken, T., Verbeke, W., & Baesens, B. (2013). A novel profit maximizing metric for measuring classification performance of customer churn prediction models. *IEEE Transactions on Knowledge and Data Engineering*, 25, 961–973.
- Wagner, J. M., & Shimshak, D. G. (2007). Stepwise selection of variables in data envelopment analysis: Procedures and managerial perspectives. *European Journal of Operational Research*, 180, 57–67.
- Wang, G., Liu, L., Peng, Y., Nie, G., Kou, G., & Shi, Y. (2010). Predicting credit card holder churn in banks of china using data mining and mcdm. In *Web Intelligence and Intelligent Agent Technology (WI-IAT), 2010 IEEE/WIC/ACM International Conference on* (pp. 215–218). IEEE volume 3.
- Zhang, X., Zhu, J., Xu, S., & Wan, Y. (2012). Predicting customer churn through interpersonal influence. *Knowledge-Based Systems*, 28, 97–104.
- Zhuang, Z.-Y., Gulzar, M. A., & Chang, S.-C. (2015). Call-distribution-based view: The theoretical ground for strategic routing management. *Journal of Communications*, 10, 221–230.

Appendix A.

Here we present detailed results obtained per each of the ‘shortlisted’ feature type combinations: final AUC score on validation set, top-decile lift (TDL), EMP (Expected Maximum Profit), number of initial (and retained) features, time (in seconds) needed for feature selection (FS), model training and validation (MTV), feature extraction (FE) and information if that combination is dominated. Best results per measure are marked in **bold**. Results are sorted in increasing order of AUC score.

Table A.4: Results for ‘shortlisted’ feature type combinations obtained with LR for prepaid dataset (Figure 3 top).

Feature type combination	AUC	TDL	EMP	Init.(Ret.)	num. feat.	FS time	MTV time	FE time	domin.
L&HLNT-NE&HLT-E	0.63719	1.51840	0.01431	46(7)		17875.36	44.38	1527.00	Y
L&HLNT-NE	0.64614	1.58483	0.01709	37(8)		32808.35	47.41	739.38	N
L&NI-E&HLNT-NE	0.64683	1.57397	0.02017	47(7)		16932.60	42.35	2125.99	Y
L&ND-E	0.66189	1.76284	0.04930	33(5)		11635.39	26.78	1900.11	N
ND-E&HLNT-NE	0.66641	1.78893	0.04868	33(5)		9287.22	24.13	2066.66	N
L&ND-E&HLNT-NE	0.66864	1.76086	0.04682	54(7)		22541.92	31.53	2353.08	N
ND-E&HNNTD-E	0.67388	1.40302	0.08684	32(5)		5150.83	16.86	3751.11	Y
L&HNNTD-E	0.67453	1.79429	0.04861	41(7)		19049.44	32.99	2423.83	N
L&HLNT-NE&HNNTD-E	0.67469	1.79133	0.05340	62(8)		19770.38	25.06	2876.80	Y
L&HNNTD-E&HLT-E	0.67617	1.77582	0.04691	45(8)		17244.09	37.72	3211.45	N
ND-E&HLNT-NE&HNNTD-E	0.68781	1.88019	0.10662	53(9)		19505.60	33.85	4204.08	Y
L&ND-E&HNNTD-E	0.68880	1.87399	0.10543	53(9)		17410.67	33.47	4037.53	N

Table A.5: Results for ‘shortlisted’ feature type combinations obtained with RF for prepaid dataset (Figure 3 bottom).

Feature type combination	AUC	TDL	EMP	Init.(Ret.)	num. feat.	FS time	MTV time	FE time	domin.
L&HLNT-NE	0.69537	1.93379	0.10112	42(5)		11793.65	148.85	739.38	N
NI-E&HLNT-NE	0.69759	1.94225	0.11724	26(4)		4021.77	447.22	1839.58	Y
L&NI-E	0.69836	1.96595	0.12330	26(5)		4389.02	129.22	1673.03	Y
L&HLNT-NE&HLT-E	0.69942	1.96398	0.11346	46(5)		9913.41	465.65	1527.00	N
L&HNNTD-E	0.70963	2.00855	0.13469	41(6)		10342.66	130.58	2423.83	Y
L&NI-E&HLNT-NE	0.71157	2.06257	0.15589	47(5)		10446.55	130.74	2125.99	Y
ND-E&HLNT-NE	0.71237	2.00164	0.14281	33(5)		5938.85	439.63	2066.66	N
L&HLNT-NE&HNNTD-E	0.71316	2.03295	0.15729	62(6)		19271.81	131.35	2876.80	Y
L&NI-E&HNNTD-E	0.71479	2.05693	0.16507	46(6)		9470.14	121.14	3810.45	Y
NI-E&HLNT-NE&HNNTD-E	0.71499	2.02872	0.15580	46(6)		9941.46	97.41	3976.99	Y
L&ND-E&HNNTD-E	0.71747	2.10192	0.18355	53(7)		13687.74	369.98	4037.53	Y
L&ND-E&HLNT-NE	0.72043	2.09797	0.17431	54(6)		12097.76	145.04	2353.08	N

The code for this work was implemented using the Python Scikit library, with the exception of the Page Rank algorithm which was implemented in C++. The

Table A.6: Results for ‘shortlisted’ feature type combinations obtained with LR for postpaid dataset (Figure 4 top).


Feature type combination	AUC	TDL	EMP	Init.(Ret.) num. feat.	FS time	MTV time	FE time	domin.
L&HLNT-NE	0.64149	1.71266	8.8E-06	26 (6)	3820.61	37.98	371.47	N
ND-E&HLNT-NE	0.64911	1.81520	0.00013	25 (6)	4177.09	43.77	3488.22	Y
L&ND-E	0.65263	1.82960	2.2E-06	25 (6)	4512.52	40.29	3468.20	N
L&ND-E&HLNT-NE	0.66985	1.91716	0.00013	38 (6)	9603.22	37.41	3663.94	Y
HLNT-NE&HNNTD-E	0.68514	2.03122	0.00002	33 (7)	7918.67	32.01	4277.68	Y
L&HLNT-NE&HNNTD-E	0.69284	2.05311	0.00003	46 (9)	17207.30	46.44	4453.41	Y
L&ND-E&HNNTD-E	0.69723	2.08307	7.4E-07	45 (7)	12527.38	41.30	7550.13	Y
ND-E&HLNT-NE&HNNTD-E	0.70216	2.11360	9.6E-07	45 (8)	12375.70	48.52	7570.16	Y
L&HNNTD-E&NHTD-E	0.70349	2.08825	0.00013	45 (8)	15601.11	38.83	11684.99	Y
L&HNNTD-E	0.70482	2.10496	0.00003	33 (7)	8119.42	40.79	4257.66	N

Table A.7: Results for ‘shortlisted’ feature type combinations obtained with RF for postpaid dataset (Figure 4 bottom).

Feature type combination	AUC	TDL	EMP	Init.(Ret.) num. feat.	FS time	MTV time	FE time	domin.
L&HLNT-NE	0.70193	2.08941	0.00246	26 (7)	3682.58	293.08	371.47	N
ND-E&HLNT-NE	0.71434	2.21787	0.00383	25 (6)	3255.33	355.14	3488.22	Y
HLNT-NE&HNNTD-E	0.72037	2.24264	0.00506	33 (6)	5403.37	435.01	4277.68	Y
L&ND-E	0.72152	2.24840	0.00416	25 (6)	3378.90	359.02	3468.20	N
L&HNNTD-E	0.72401	2.25935	0.00537	33 (5)	5342.20	325.68	4257.66	Y
ND-E&HLNT-NE&HNNTD-E	0.72422	2.28239	0.00386	45 (6)	7213.99	344.55	7570.16	Y
L&ND-E&HLNT-NE	0.72766	2.31004	0.00391	38 (6)	5052.95	324.87	3663.94	N
L&HLNT-NE&HNNTD-E	0.73109	2.29564	0.00392	46 (6)	7990.31	350.62	4453.41	N
L&NI-E&HNNTD-E	0.73548	2.35613	0.00482	38 (6)	4922.77	362.36	11430.28	Y
L&ND-E&HNNTD-E	0.73605	2.34691	0.00405	45 (6)	11678.60	360.26	7550.13	N

machine used for computation has two 64-bit Intel Xeon processors working at 2,3GHz with 36 cores and 256GB of RAM.

Appendix B.

In this section, we provide graphical representations of retained features using  LR model for both datasets.

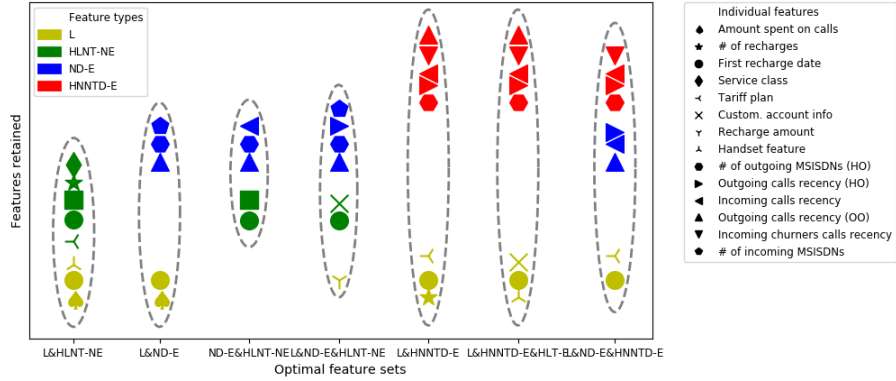


Figure B.8: Retained features (in the LR model) for Pareto optimal feature type combinations for the prepaid dataset. The feature type combinations are sorted by increasing order of AUC performance (from left to right). HO stands for ‘home operator’, while OO stands for ‘other operator’.

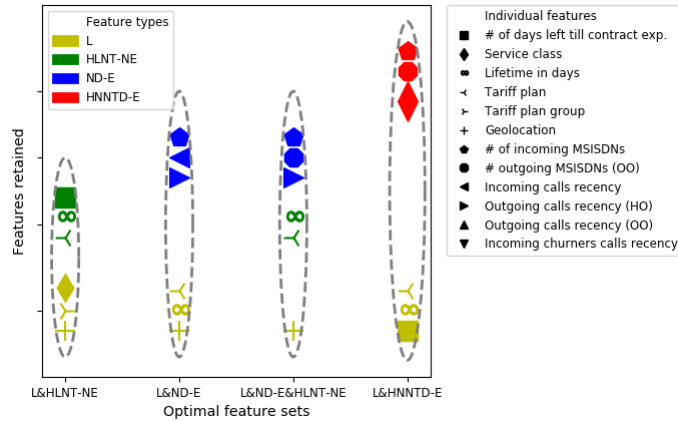


Figure B.9: Retained features (in the LR model) for Pareto optimal feature type combinations for the postpaid dataset. The feature type combinations are sorted by increasing order of AUC performance (from left to right). HO stands for ‘home operator’, while OO stands for ‘other operator’.

Appendix C.

In this section, we provide the results of non-parametric pairwise DeLong, DeLong, Clarke-Pearson statistical test to examine the potential statistical differences between AUC scores in four different cases: prepaid with LR, prepaid with RF, postpaid with LR and postpaid with RF.

