information

Article

An Approach to Churn Prediction for Cloud Services Recommendation and User Retention

José Saias [1,2]*, Luís Rato [1,2] and Teresa Gonçalves [1,2]

[1] Department of Computer Science, University of Évora, 7000-671 Évora, Portugal; lmr@uevora.pt (L.R.); tcg@uevora.pt (T.G.)
[2] Centro ALGORITMI, Vista Laboratory, University of Évora, 7000-671 Évora, Portugal;
* Correspondence: jsaias@uevora.pt

Abstract: The digital world is very dynamic. The ability to timely identify possible vendor migration trends or customer loss risks is very important in cloud-based services. This work describes a churn risk prediction system and how it can be applied to guide cloud service providers for recommending adjustments in the service subscription level, both to promote rational resource consumption and to avoid CSP customer loss. A training dataset was built from real data about the customer, the subscribed service and its usage history, and it was used in a supervised machine-learning approach for prediction. Classification models were built and evaluated based on multilayer neural networks, AdaBoost and random forest algorithms. From the experiments with our dataset, the best results for a churn prediction were obtained with a random forest-based model, with 64 estimators, having 0.988 accuracy and 0.997 AUC value.

Keywords: machine learning; churn; decision analysis; forecasting

Citation: Saias, J.; Rato, L.; Gonçalves, T. An Approach to Churn Prediction for Cloud Services Recommendation and User Retention. *Information* **2022**, *13*, 227. https://doi.org/10.3390/info13050227

Academic Editors: Marco Polignano, Giovanni Semeraro and Costas Vassilakis

Received: 7 March 2022
Accepted: 26 April 2022
Published: 28 April 2022

1. Introduction

Subscriptions to cloud-based services have been growing in recent years, with the existence of many products, each with different configurations or feature packs. Churn, or customer loss, has an impact on the service providers' business and sometimes even on customers when they migrate to a new product that forces them to adapt to a new context, regardless of the eventual financial advantage. According to [1], churn rates between 1% and 5% are common in the software-as-a-service (SaaS) field, and sometimes these rates can exceed 15%, which represents a significant impact on service provider revenue. To avoid losing customers, it is necessary to understand the reasons that lead a customer to abandon or give up the service. Modern data science techniques and machine learning can help find explanatory variables for the abandonment of a service and develop predictive models that help to guide a commercial strategy to retain the customer.

In this paper, we describe the `Cloud Service churn Prediction` system (CSCP), a solution for churn prediction developed as part of a productivity and business leverage project in cloud services. The goal of the CSCP is to support service providers by suggesting subscription levels more suitable for the service needs of the customer at the time, thereby retaining customer loyalty.

Based on the behavior history of cloud product usage, the system aims at estimating a quantitative churn risk indicator for a customer in relation to a subscribed service. The predictor output can later be considered in alarmistic modules and recommendations and also for *What-If* analysis tools. Service providers will thus have a valuable business support module to evaluate different usage scenarios and to design timely offers or retention campaigns tailored to each client's needs and to build customer loyalty.

The main contributions of this work are:

- The study of churn classification approaches and the enumeration of domain concepts, whose understanding is relevant for data selection and for modeling in this domain;

- The development of a machine-learning-based churn risk predictor designed to be used on a comprehensive subscription level service recommendation platform;
- Incorporation of remarks on data correlation, statistical significance and dataset adequacy into the distinct approaches that are tried and assessed.

In the next section we have a brief survey on work related to churn prediction in different business areas. Section 3 describes this work's grounding concepts and the approach taken to model churn and develop the proposed solution. The results of the evaluation performed are presented in Section 4. Finally, Section 5 includes some discussion of this work and its assessment, and Section 6 ends with general conclusions.

2. Related Work

In 2017, Aditya Kapoor published a report on customer retention and churn in the American telecommunications market [2]. A study was reported that indicated a 1.9% churn rate in the top four operators (AT&T, Verizon, T-Mobile, Sprint). These operators cover 100 million customers. On average, churn occurred at 19 months. It said that the normal lifespan of a customer is 52 months, so the lost revenue in each churn case was estimated to be more than USD 1100 (from USD 34 $\times$33 months). The author describes the modeling of a predictive system for churn based on a repository of around 100,000 records and 150 attributes. Data include information on phone calls (quantity, duration, for the last 3 and 6 months), contractual information, details about data consumption, and the client's socioeconomic profile. With a *Filter* approach, 20 features were selected. Data were divided between training and test sets, with sizes 70% and 30%, respectively. Through the Azure platform, classification models were created with logistic regression (LR), boosting, random forest [3], neural networks and support vector machine (SVM). Assessment results were not reported. The author mentions the optimization of models through the search for a threshold oriented towards profit maximization.

In April 2018, a `datascience.com` publication from Sowmya Vivek [4] described the use of the linear discriminant analysis (LDA) algorithm for predicting churn with customer data from a telecommunications company, with special focus on service-related factors. Using the LDA as a segmentation mechanism, the work seeks to divide customers between churn and nonchurn groups. A multivariate analysis of variance (MANOVA) was performed, and the relative importance of each independent variable was studied. The reported accuracy for the base prediction model is 85.67%, but with a low sensitivity of 12.7%. After some tests to balance the indicators, the threshold used was 0.16, which allowed stabilizing both accuracy and sensitivity measures at 74%.

In [5], the authors addressed the challenge of computational efficiency of data mining approaches over large scale data, using spark and caret tools for a churn prediction task on a telecommunication dataset. In both spark and caret, a random forest classifier model was trained with equal data partitioning and tuning parameters. The results showed that the classifier's operation in spark was more efficient when compared to caret (50.25 s vs. 847.20 s execution time), and in both cases the accuracy was approximately 80%.

In 2016, Dalvi et al. proposed a model for churn prediction for telecommunication companies [6]. A statistical survival analysis tool was proposed to predict churn, and it was based on a comparison between decision trees and logistic regression. R programming was chosen to build the prediction model. The evaluation process was planned, but the results were not presented.

A 2020 master's thesis from Christian Jensen [7] sought to answer which features best explained the reason behind telecom customer churn and which machine-learning algorithms best solve the task of data mining. The author sought to test some existing models on new data that contain both demographic and expenses variables. The best model only predicted 67.6% of the cases correctly.

In the financial area in 2018, Erdem Kaya et al. investigated spatiotemporal patterns and entropy in financial decisions, and the relationship they can have with churn [8]. Inspired by computationally based works in the area of social sciences, the authors designed

a predictive model sensitive not only to temporal and spatial aspects, but also to behavioral aspects of how consumers spend money. Data included demographic information, credit card transactions or bank transfers arranged in two sets, A and B. Set A had about 45 million transactions relating to 100,000 customers. Set B had about 22 million transactions relating to 60,000 customers. Set B was extracted from the same source, but had more specific criteria: customers with 10 or more credit card transactions where 60% or more of the transactions were associated with POS equipment from the bank that provided the data. Because there was an imbalance between classes, the evaluation of the models was made with the AUC measure, with an eight-fold cross-validation method. For the various models trained with random forest, reported AUC values were between 0.513 and 0.79.

Amuda and Adeyemo studied the use of a multilayer perceptron to predict customer churn for the financial domain [9]. In this study, the authors developed a model to reduce manual feature engineering. The dataset came from a Nigerian financial institution related to 50,000 customers and included 42 attributes. A total of 80 percent of the dataset was used for training, 10 percent was used for testing, and the remaining 10 percent was used for model validation. Two artificial neural network software solutions were used: Python (dropout and L2 regularization techniques for overfitting) and Neuro Solution Infinity software. Both obtained comparable performance, with accuracy rates of 97.53% and 97.4% with 0.89 and 0.85, respectively, for AUC.

In 2017, Robert Aman developed a study to identify relevant parameters to the strength of the relationship with customers in a software-as-a-service company in Stockholm in the digital marketing business for B2B [10]. The author describes the application of statistical processes and data mining techniques to software usage survey data. Monitoring tools collected detailed data on product usage/consumption by users for each client company. Data were treated with ANOVA (https://statistics.laerd.com/statistical-guides/one-way-anova-statistical-guide.php, accessed on 2 March 2022) analysis techniques, linear regression and logistic regression. The model for classification of customer status was evaluated in terms of accuracy, with the best variant obtaining 0.745 on this indicator. The results show different patterns of software usage between customers who remain active and cases where there is abandonment, with greater use of software generally associated with a greater propensity to remain as a customer. It is also mentioned that the product usage value perceived by the customer explains or is associated with the strength of the bond with the client. If this bond is weak, there is a greater chance of abandonment.

In [11], churn analysis was performed on data from a software-as-a-service company selling an advanced cloud-based business phone system. Oversampling, undersampling, and time series cross-validation methods were tried to diminish the impact of the imbalanced data. Logistic regression and random forest models were used to both predict and explain churn. The author reported that the resulting model was more useful to explain churn than to predict it.

Tsai and Lu [12] applied back propagation artificial neural networks (ANN) and also hybrid models by combining back propagation artificial neural networks and self-organizing map techniques for churn prediction. In the hybrid models, the technique started with a data reduction task by filtering out unrepresentative training data and then using the remaining representative data to create the prediction model based on the second technique. The assessment result indicated that the ANN + ANN hybrid models performed better than the baseline ANN models over the five testing subsets, having accuracy values between 90% and 94%.

The authors of [13] proposed a cloud-based *extract–transform–load* (ETL) framework for data fusion and aggregation with applications in churn prediction, service outage prediction, fraud detection and actionable information for a timely recommendation system. The authors described a use case for a customer churn prediction problem for a video-streaming service operating in the Canadian market. The dataset for this prediction use case include demographic and personal data (e.g., number of avatars, account age, number of profiles, number of children profiles, payment history, account type, etc.), customer

surveys data and dynamic data from user interactions logs. The distribution between classes was highly unbalanced: 200,000 subscribers, of which about 10% churned. Decision trees, random forest, extra trees, AdaBoost and XGBoost algorithms were used in the churn classification. The evaluation results showed 97% for accuracy and 98% AUC.

Using data about B2C e-commerce customers' shopping behaviors, [14] described a churn prediction approach based on the combination of k-means for customer segmentation and SVM for prediction. The method divided customers into three categories (core groups/clusters) according to their shopping behaviors, and then predictions were made for these customer types. Support vector machine and logistic regression were compared. The authors wanted to evaluate the effectiveness of customer segmentation and its effect on the predictive model. The imbalance between the number of nonchurn (580) and churn (7576) customers was remedied with oversampling with SMOTE, a minority oversampling technique. The results showed that the accuracy of the SVM predictor was higher than that of the logistic regression predictor and also that the prediction performance improved when customer segmentation was applied. After customer segmentation, accuracy, recall and precision results for the SVM model were 0.9156, 0.9721 and 0.861; those of the LR model were 0.9066, 0.9498 and 0.8533, respectively.

3. Materials and Methods

This section describes our churn prediction solution, starting with some grounding concepts. Then the approach to model churn, and the details of the system architecture and its software components are examined.

3.1. Overall Requirements

Before introducing the `CSCP` system components or their features, it is important to clarify some concepts:

- Client is a private individual, a company or an institution who can purchase a product or subscribe to a cloud service;
- Provider is a cloud service provider (CSP); Reseller is a service provider in an intermediary role whose products depend on other CSPs;
- Cloud Business Analytics and Recommendation (CBAR) is a productivity and business leverage solution for service providers with which our CSCP system will interact, which is owned by a partner company;
- Active subscription is a subscription with active status that has one or more licenses in use;
- Subscription Value is a numeric indicator proportional to the value that a subscription has for the service provider;
- Customer Value is a numerical indicator proportional to the value that a customer represents to the provider, considering the subscription's combined value.

The prototype system to be developed should include the following features:

1. Query or import data about customers, services and subscriptions as well as data on service consumption and billing data to support the predictive models;
2. Train a model for predicting churn risk by using machine learning with the data available on a given date;
3. Predict a customer's propensity to churn for asubscribed service;
4. Given a list of service subscriptions, prioritize such a list on the basis of multiple indicators, combining the risk of customer loss with the subscription or customer value;
5. Interoperate with other systems, including the CBAR analytic dashboards and recommender modules.

This system is not directly operated by customers or CSPs. A graphical interface is not required for this system. Operations will be made available via an API as a service and with communication standards independent of the technology in use by each interlocutor, thus facilitating the integration and interoperability. Privacy and data protection are important. The CSCP system does not access more data than strictly necessary for its operations. For model training and prediction, no real identifiers are needed. On the other hand, whenever the CSP's data protection policy prevents the use of its data by third parties, it is possible to restrict their use to the provider's exclusive models.

3.2. Data and Churn Modeling

In machine learning, classification aims to predict the category of an item from a model based on one or more numeric or categorical input variables that are referred to as predictive attributes or features. Given a subscription characterized by data about the subscriber, the aim is to classify between a case of imminent abandonment and its opposite, a low risk of abandonment. In a binary classification task, we have the churn and nonchurn classes. Additionally, machine-learning algorithms may even debit the estimated probability for the observed data to fall into each class. Thus, a high probability for the churn class means a high risk of losing the customer.

Prediction success depends largely on the quality and adequacy of the chosen data [15]. As with supervised learning, models will be trained with a set of instances, each described by a set of attributes or features from known cases. When fitting the model, one of these features corresponds to the class to be predicted (churn; nonchurn). In our work, each classifier training instance describes, for a given moment in time, a subscription to a service. The minimum set of instance features includes:

- Customer data: sociodemographic, customer type (business, private, academic), if the customer has other active subscriptions and whether the customer has abandoned any subscription in the last 30 days. Customer identity is never used. Data are accessed in a context of consent, which is managed by the CSP;
- Service data: service type and provider;
- Contractual and subscription usage data: date subscription start/activation date, loyalty commitment end date, billing cycle, promotion benefit, number of purchased licenses, number of licenses in use, subscription status (active, suspended, canceled), number of days in which there are subscription records, number of days the subscription was active, if it is within some loyalty program, number of days until the end of loyalty, number of days after the end of loyalty, number of days after subscription start, average number of activation days after loyalty for the subscribed service, existence of a reduction in the number of licenses used in the last 20 and in the last 5 days and billing data;
- Target variable: churn or nonchurn. This attribute is the key concept to model, where the notion of time is very important. This attribute will take the value churn if within 5 days following the time of observation (that is, the date to which the instance refers) the subscription is not active. Otherwise its value is nonchurn.

Our data source is managed by University of Évora's partner company in this work. We obtained access to a subset of the CBAR data after anonymization and only for specific use within this project's scope.

In preprocessing, categorical data variables are transformed with the one-hot encoding method, with a new column of data containing a binary numeric value for each label in the original domain. Some examples that receive this encoding are the columns billing type and cycle, service type, customer type, country and status.

The dataset built to support the predictive system contains 196,977 instances, corresponding to different points in time of 26,418 service subscriptions. We emphasize that an instance represents the status of a subscription on a certain day, and there may be several instances for the same subscription. The churn class has 62,193 (31%) instances, while the nonchurn has 134,784 (69%) instances.

Two tools were used in our churn modeling experiences: KNIME (https://www.knime.com/, accessed on 15 December 2021) [16] an open source analytics platform allowing visual workflows and Scikit-learn (https://scikit-learn.org/, accessed on 15 December 2021) [17], an open source machine-learning library. To manage, explore and visualize the data, we used a web-based interactive data analytics tool, Apache Zeppelin (https://zeppelin.apache.org/, accessed on 15 December 2021). This allowed for collaborative work, where different user profiles can query different data subsets and possibly through different permissions and distinct query languages such as SQL or Python. Figure 1 shows two boxplots with the distribution of values per class for two features: the number of subscription records and the number of days in those records time spans. In addition, from our exploratory data analysis, the matrix in Figure 2 exhibits the Pearson product–moment correlation coefficient for each pair of data variables. In each table cell, the color ranges from dark red to dark blue (strong negative correlation to strong positive correlation). On the main diagonal, we see a variable's correlation with itself, therefore having the blue color. The correlation between each feature and the target label is represented in the last row. We can see that churn has a positive correlation with the variable for warnings on quantity reduction; has a negative correlation with the variable is_recently_active_and_having_quant; and has no correlation with recent discontinuity of active state (warnings_c3active_discontinuity).

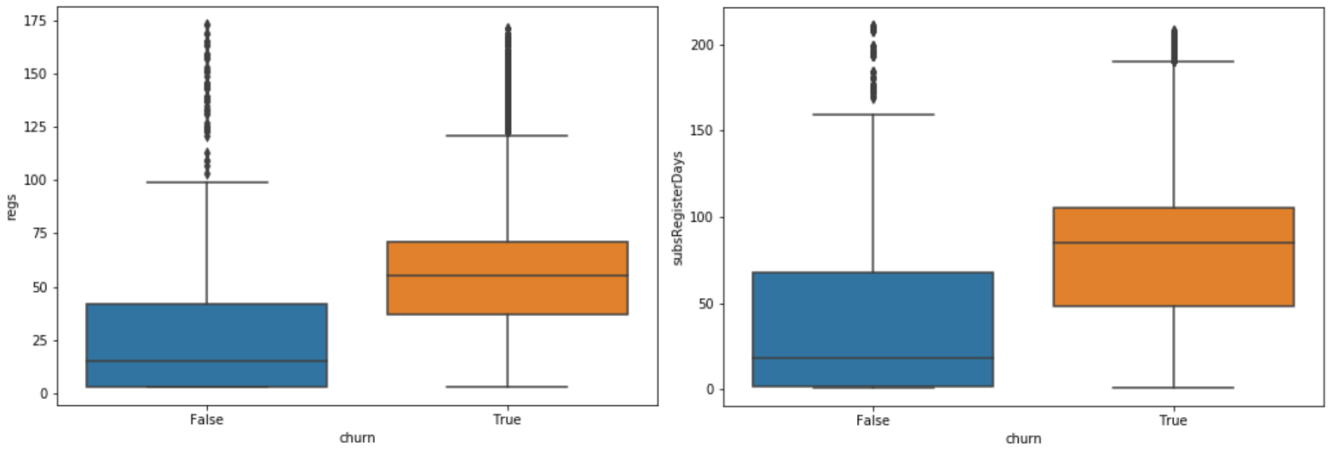

Figure 1. Distribution of values of two variables, for each class.

In addition to the correlation coefficient between variables, their respective statistical significance was also analyzed. Table 1 shows the *p*-value (probability describing how likely the data could have occurred by random chance) for some columns. Usually, a *p*-value higher than 0.05 means the case is not statistically significant. Thus, we find that the strong negative correlation between active state and churn (−0.929) is not significant. On the other hand, it was observed that the positive correlation that the target variable has with days after the end of loyalty period, canceled or suspended state or the warnings on quantity reduction is statistically significant.

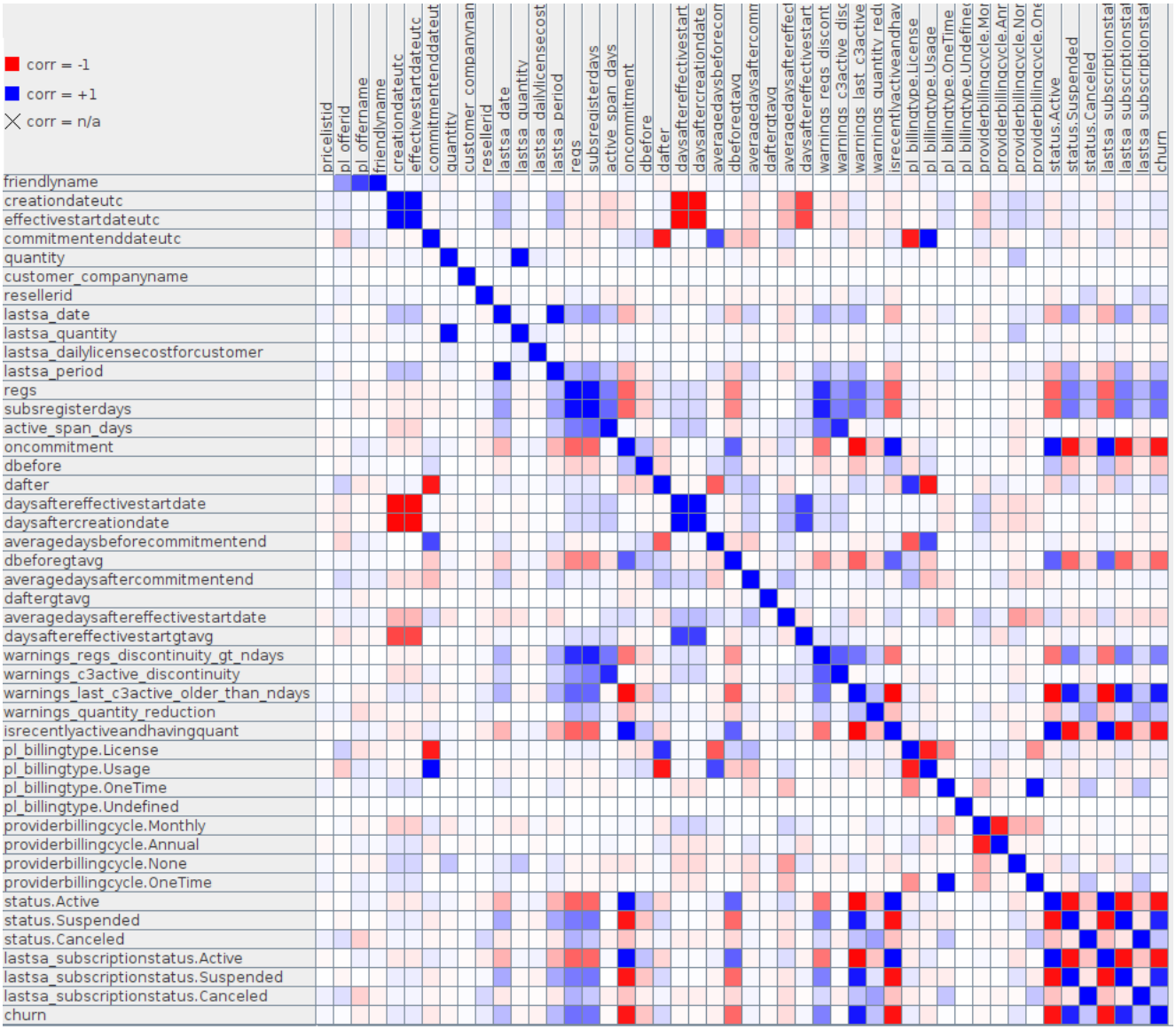

Figure 2. Correlation matrix: pair-wise correlation values of selected data columns.

Table 1. Churn correlation and statistical significance.

Data Column	Correlation with Churn	*p*-Value
days after the end of loyalty period	0.156	<0.01
days before the end of loyalty period	−0.229	>0.9
on loyalty commitment period	−0.917	>0.9
status: Active	−0.929	>0.9
status: Canceled	0.229	<0.01
status: Suspended	0.872	<0.01
warnings on quantity reduction	0.244	<0.01

3.3. Algorithms and Evaluation Metrics

Evaluating a predictive system can involve several performance metrics. Accuracy is the hit rate that corresponds to the number of correct classifications on the total number of cases to be classified. Despite being easy to understand, it is sometimes misleading about the actual success, especially if there is a large imbalance between classes. In binary classification tasks, for instance with positive and negative classes, it is common to use a confusion matrix, which is a table with performance by class, including: false or wrongly

predicted positives (*FP*), false negatives (*FN*), true or correctly predicted positives (*TP*) and true negatives (*TN*). With these elements, the following evaluation measures can be calculated: precision, sensitivity (*recall* or *true positive rate*) and specificity (*true negative rate*) according to the following formulas.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \qquad Precision = \frac{TP}{TP + FP} \tag{1}$$

$$Recall\ or\ TPR = \frac{TP}{TP + FN} \qquad Specificity = \frac{TN}{TN + FP} \tag{2}$$

The F-Measure is a performance measure also used for classification that combines precision and recall in a single quantitative indicator. The *receiver operating characteristic* curve, or ROC curve, is a graphical representation that relates sensitivity (in ordinates) with one-specificity (on the abscissa). The *area under the ROC curve*, or AUC, is a performance indicator often used for classification tasks, particularly when the distribution of instances across classes is not balanced.

Model fitting experiments were carried out with neural networks and ensemble classifiers AdaBoost [18] and random forest [3]. The random forest algorithm follows an approach of dividing the training set into randomly generated subsets to train different decision trees and then aggregates the results by choosing the majority class. For neural network-based models, we used the multi-layer perceptron (MLP) supervised learning algorithm, using the MLPClassifier class from Scikit-learn, and also MLP with the RPROP algorithm [19] in KNIME. Experiments with AdaBoost and random forest were performed in Scikit-learn. A first experimental evaluation was made using the full dataset with a cross-validation method for a direct model assessment, but later we changed the evaluation procedure to the three-way holdout method [20], to have a more adequate evaluation and comparison between models using different algorithms. With this approach, we keep an independent test dataset for a final evaluation with data that were not used during the training and validation stage. This test set includes 20% of the original dataset, keeping the proportion between classes. The remaining 80% of instances are used for model development in training and validation sets (with a split of 80% and 20% of this portion of the dataset). After this hyperparameter optimization phase, the best model for each type of algorithm was chosen, which was then retrained on the combined training plus validation sets. The process concludes with these models' performance evaluation when applied to the unseen data from the test set. In Section 4, we present the main prediction algorithm setups we tested and their observed performance.

For a more in-depth analysis of algorithms' setup variants, a more complete machine-learning hyperparameter optimization was initially planned with the sampling and pruning strategies in the Optuna framework [21], but after the prototype evaluation, we deferred this task to a second phase. After the data analysis and experimental modeling phase, a random forest algorithm was selected for the prediction task.

3.4. CSCP Architecture

Figure 3 shows an overview of the system components and how they interconnect. The dashed horizontal line on top represents the boundary between our system and CBAR. The CSCP solution involves five components: three computing-oriented applications (represented in rectangles) and two repository-type applications (at the bottom of the figure). This modular organization makes it easy to make changes or improvements to a module while maintaining the API without harming the system.

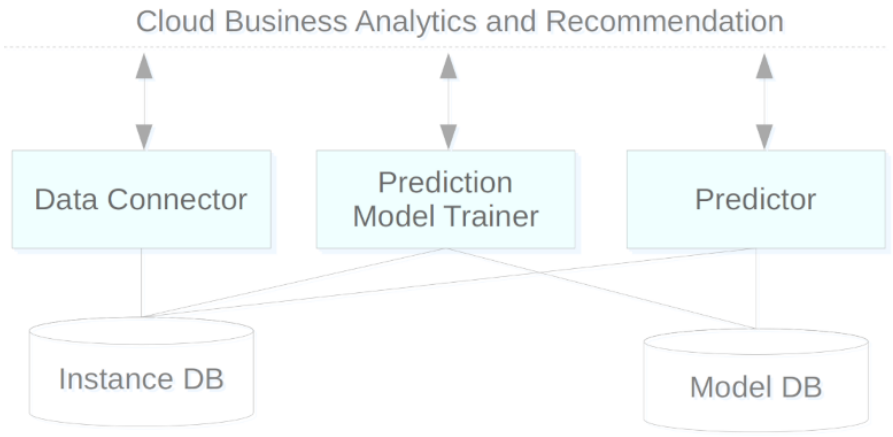

Figure 3. CSCP architecture.

The *data connector* module is, as the name suggests, a connector between CSCP and CBAR repositories and other data sources. Its function is the acquisition of key data about customers and services and processing this data to feed an instance repository. Data transformation can include cleaning and conversion procedures, data reduction, such as discretizing an attribute value and filling in new calculated fields. Data are loaded from CBAR analytics tables and directed to the local *Instance DB* module. Data retrieval can be triggered on-demand to update the features of an instance of the subscription for which a prediction is requested or as a preload anticipating the update of required instances for a model or prediction.

The *instance DB* module is a repository to support the training of predictive models with data on service subscriptions and their respective customers and details of consumption, access, costs and type of payment. Each record, considered an instance, characterizes the essentials about a subscription to feed machine-learning algorithms. This module was implemented on top of a common relational database in *Azure SQL*.

The *prediction model trainer* component is a predictive model generator, parameterized for a given type of objective and algorithm. The purpose of the model may be classification or regression, and the employed modeling process was described in Section 3.2. The CSCP system has a continuous operating cycle from model generation to its implementation and use. Then monitoring detects when it will be necessary to retrain the model for the prediction to keep up with new data and trends. After years of operation, if the data history grows so large that it makes training difficult,

subsampling will be used, truncating the oldest part of this data and giving priority to the most current behavior patterns.

These models that are gradually built are all kept in the *model DB*, where they will be loaded from by *predictor*. Storing the various versions of a predictive model allows future maintenance operations such as comparing models, or revoking a modem by automatically replacing it with a previous consistent version.

The *model DB* component is a repository of predictive models. Adjusting or training a model can be time-consuming. Storing these models allows for more flexible management, making the deployment of a previous model easier without the need to repeat the training. It also allows for the combined use of multiple models. This module also includes a repository with the history of predictions answered by each model, which can be used as a temporary cache.

Finally, the *predictor* module is, as the name implies, a predictive engine. It is currently used only for the churn probability prediction operation regarding a subscription, but in the near future, we may estimate propensity for UpSell in scenarios where a customer makes full use of a certain service plan. In each request, the module uses a previously trained model for this purpose and loads the data that characterizes the subscription and its context detail, and outputs the predicted churn risk.

Optionally, and for efficiency reasons, predictions for a broad set of subscriptions can be scheduled for early processing, with the results kept in cache for immediate response on later requests. This cache has a validity period of 2 days and contains the subscription and model identifiers, the output prediction and a timestamp.

A RESTful web service was implemented for the churn risk prediction, whose requests identify the subscription and may include optional parameters about the cache or the model version to employ, as listed in Figure 4a. Then, the system chooses the most appropriate predictive model (or models), feeds it with the features that describe the subscription on that date and returns the result in a JSON format, such as the example of Figure 5. Each field description can be found in Figure 4b.

Name	Description	Parameter Type	Value Type	Default
SubscriptionId	the subscription for which the prediction is intended	Path	string	
skip_cache	determines whether it is acceptable to use a cached value or to force rerun	Query, optional	boolean	false
model_version	the model to be used in the prediction	Query, optional	string	latest

(**a**)

Name	Description	Type
sid	SubscriptionId	string
churn_prediction	The target class prediction: churn or no-churn. When "churn", the prediction foresees the service abandonment to happen in the following days.	string
churn_prob	The estimated probability for churn	numeric (0.0 to 1.0)
no-churn_prob	The estimated probability for no-churn	numeric (0.0 to 1.0)
prediction_timestamp	When the prediction was executed. If the cache is not used, this timestamp is the current time.	string, iso format timestamp
sid_last_seen_by_model_on	Date of the most recent data for that subscription. It may be useful to validate the recentness of the data used for the prediction.	string, iso format date
model_version	The identifier of the model used in the given prediction.	string
status	Request execution status: 0 or 1 for normal cases; >10 for unexpected events or problems (described by the status_message field).	numeric (integer)
status_message	• empty string, status 0: the prediction was made and is presented • "the prediction is in queue for processing", status 1: the request is being processed, but the data connector may take some time to collect the necessary data for the classifier; the response time depends on the load in the CSP database; a few seconds later, the processing ends and the prediction is available in the local cache, being delivered immediately on the next request • "problems: data connector: sid N was not found", status 21: unexpected situation where the subscription was not found • "problems: data connector: some required data is not available for sid N", status 22: some required data for this subscription was not found	string

(**b**)

Figure 4. Service request parameters and answer details: (**a**) prediction request parameters; (**b**) prediction answer message fields.

```
{
 “sid”: “NN-EXAMPLE-SUBSCRIPTION”,
 “churn_prediction”: “no-churn”,
 “churn_prob”: 0.02,
 “no-churn_prob”: 0.98,
 “prediction_timestamp”: “2020-05-04 17:29:40”,
 “sid_last_seen_by_model_on”: “2019-12-24”,
 “model_version”: “2020-04-29”,
 “status”: 0,
 “status_message”: ""
}
```

Figure 5. CSCP prediction answer example.

4. Results

The recommendation engine, designed to suggest the ideal subscription level for the client over time and also for signaling the churn risk and thus allowing for reduced client loss, is at an embryonic stage. We expect to have results from recommending upgrade/downgrade on service subscription levels to CSP customers and also on the effect that churn prediction has on CSP's customer-targeted retention campaigns. The evaluation of the recommendation engine requires an operation dataset that has not yet been established, with only one primary evaluation carried out by the project partner company, which controls the CSCP deployment environment and the CBAR.

Being a core input to the recommender module, the churn risk prediction was developed from an incremental process using various approaches. Table 2 shows the assessment results for the development phase, with several machine learning setup variants tested within the scope of this work. The first two lines have the results for the multilayer perceptron classifier tested in Scikit-learn using the *lbfgs* solver. The first neural network has three hidden layers with 10 neurons each, and the second also has three hidden layers, but 12, 6 and 3 neurons each. The MLP KNIME methods refer to the multilayer perceptron classifier built in KNIME, using 50, 40 or 30 learning interations (li) and two to three hidden layers (hl) with 10 neurons per layer. The last four lines in the table present the evaluation for AdaBoost and random forest algorithms, where 32 and 64 are the number of estimators employed in the ensemble. Parameters not mentioned were given their default value for the framework.

Table 2. Development phase: assessment of models' setup variants.

Method	AUC	Accuracy	F-Measure per Class	
			Non-Churn	Churn
MLP SciKit 10, 10, 10	0.499	0.610	0.740	0.200
MLP SciKit 12, 6, 3	0.947	0.890	0.920	0.830
MLP KNIME 50 li, 2 hl	0.964	0.960	0.969	0.938
MLP KNIME 50 li, 3 hl	0.966	0.963	0.973	0.943
MLP KNIME 40 li, 3 hl	0.965	0.963	0.962	0.942
MLP KNIME 30 li, 3 hl	0.961	0.961	0.971	0.939
AdaBoost 32	0.994	0.981	0.984	0.970
AdaBoost 64	0.995	0.984	0.989	0.973
Random Forest 32	0.995	0.987	0.989	0.974
Random Forest 64	0.996	0.988	0.989	0.975

In Table 3, we can find the evaluation for the best performing models when assessed on the testing set. In each case, the chosen setup was the one that achieved the best result in the previous phase. The final evaluation to support the selection of the model to be used in the CSCP system is shown in Table 4. Here, each selected algorithm was trained with all developmental data by combining training and validation sets and then assessed with the new data from the testing set.

Table 3. Models' performance on the testing set (using training set data).

Method	AUC	Accuracy	F-Measure per Class	
			Non-Churn	Churn
MLP KNIME 50 li, 3 hl	0.965	0.963	0.973	0.942
AdaBoost 64	0.994	0.983	0.984	0.974
Random Forest 64	0.993	0.985	0.989	0.974

Table 4. Models' performance on the testing set (training with all development data, training and validation).

Method	AUC	Accuracy	F-Measure per Class	
			Non-Churn	Churn
MLP KNIME 50 li, 3 hl	0.968	0.965	0.975	0.946
AdaBoost 64	0.995	0.984	0.989	0.974
Random Forest 64	0.997	0.988	0.989	0.981

5. Discussion

The tests with neural networks in Scikit-learn always gave us lower results than MLP with RPROP in KNIME. Regardless, both ensemble experiments led to better results, whether in accuracy, AUC, or F-Measure metrics. The random forest classifier in its best variant (64 estimators) obtained the best result in the development phase as shown in Table 2, with an accuracy of 0.988 and a high F-Measure in both classes, 0.989 for the nonchurn class and 0.975 for instances of the minority class.

We noticed a close performance between the two emsemble methods in all evaluation metrics, as shown in the three tables. In both algorithms, during the model development stage, there was a slight improvement when we increased the number of estimators from 32 to 64.

When assessing the models on the testing set, once again, the random forest classifier had the best performance, as shown in Tables 3 and 4. In general, models trained with the training set only perform slightly below what they did in the development phase. However, considering Table 4, with extended training, they already recover their performance. This fact, and comparing the testing accuracy against the training accuracy, leads us to believe that with this dataset the model adjustment did not result in overfit.

The time needed to adjust the model was also analyzed, and the train was faster with random forest setup variants. The MLP models took longer to adjust.

In the data analysis, we found that part of the stronger correlations between input variables and the target label were not statistically significant. Given the high accuracy and F-Measure results, we also question whether there might be a bias in this dataset's instances generation process through several snapshots for each subscription context over time, and therefore, there might be some similar (or correlated) instances.

We are waiting for some CSCP usage history to find out if this current dataset and the models based on it allow us to generalize different out-of-sample data patterns. At this stage of work, considering both the prediction performance and the required time for model adjustment, random forest in Scikit-learn emerged as the most promising solution, and it was chosen for CSCP service.

It is difficult to make a fair direct comparison between predictor models based on different datasets. Nevertheless, establishing a comparison between this work and [13], we note that the type of data involved is similar, (including the use of calculated features, such as the number of days after the subscription start), as is the type of classifiers chosen for the predictive model (ensembles, including random forest). In both works' classifiers, evaluation obtained similar values, not differing more than 1% in accuracy or AUC; however, in [13] the F-Measure result is higher for the minority class (churn), while for us this metric has the best result for the nonchurn class. Other examples of high accuracy rates had already been reported on churn prediction, such as in [9] for the financial area, where neural networks were used in a Python implementation, among others. We also found that random forest is often used for churn risk classification, as reported in [2,5,8,10,11,13].

6. Conclusions

In this paper, we describe a churn prediction solution included in a broader recommendation platform for the level of subscription of cloud services, and to support CSPs' customer retention strategies. By signaling CSPs about their customers with the greatest propensity to churn, actions can be taken to decrease customer loss. In addtion to possible discounts or promotions, these actions may include changes to the service subscription level to better address the real customer needs at the time.

As a complement to the products or service subscription details and the trends in other users with a similar profile, the risk of customer churn is crucial for the cloud service providers' retention strategy. Faced with a high number of cases with equal churn risk, the recommendation system can prioritize cases of greater subscription value, or greater customer value that have the greatest impact on the CSP's business.

Based on the literature review [22], we found that there is no globally optimal algorithm or method for churn prediction with excellent performance in all types of data and business areas. After complex experiments with CBAR-related data provided by this project's partner company, we designed a churn prediction solution based on conventional machine learning, with random forest, which allows quick training of models and ensures a good prediction performance as shown in the previous section.

We conclude with two thoughts to consider for future work about the data and about the technique used for the churn predictor. The dataset we used allowed the adjustment of models with high performance, but we will follow the operation of the CBAR system to verify if the model is representative and suitable for more recent data. Currently, and considering the available data, using deep learning is not advantageous. As the solution matures, we may change the classification algorithm if other methods prove to be more efficient.

Author Contributions: Methodology, J.S., L.R. and T.G.; software, J.S.; validation, L.R. and T.G.; resources, J.S.; writing, J.S., L.R. and T.G. All authors have read and agreed to the published version of the manuscript.

Funding: Project APRA-CP.v2, with reference ALT20-03-0247-FEDER-038500, is supported under the "Programa Operacional Regional do Alentejo 2014/2020".

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AUC	Area Under the ROC Curve
CBAR	Cloud Business Analytics and Recommendation platform
CSCP	Cloud Service churn Prediction
CSP	cloud service provider
ETL	Extract, Transform and Load
FN	false negatives
FP	false positives
LDA	Linear Discriminant Analysis
MANOVA	Multivariate analysis of variance
MLP	Multi-layer Perceptron neural network
SaaS	Software-as-a-Service
SVM	Support Vector Machine
TN	true or correctly predicted negatives
TP	true or correctly predicted positives

References

1. PayPro Global Inc. Tackling SaaS Churn. 2014. Available online: http://docs.payproglobal.com/documents/white-papers/PayPro-WP-Tackling-SaaS-churn.pdf (accessed on 27 January 2022).
2. Kapoor, A. Churn in the Telecom Industry–Identifying Customers Likely to Churn and How to Retain Them. Technical Report. 2017. Available online: https://wp.nyu.edu/adityakapoor/ (accessed on 18 January 2022).
3. Liaw, A.; Wiener, M. Classification and Regression by randomForest. *R News* **2002**, *2*, 18–22.
4. Vivek, S. Using Linear Discriminant Analysis to Predict Customer Churn. 2018. Available online: https://www.datascience.com/blog/predicting-customer-churn-with-a-discriminant-analysis (accessed on 18 January 2022).
5. Olasehinde, O.; Johnson, O.V.; Fakoya, J.T. Computational Efficiency Analysis of Customer churn Prediction Using Spark and Caret Random Forest Classifier. *Inf. Knowl. Manag.* **2018**, *8*, 8–16.
6. Dalvi, P.K.; Khandge, S.K.; Deomore, A.; Bankar, A.; Kanade, V.A. Analysis of customer churn prediction in telecom industry using decision trees and Logistic Regression. In Proceedings of the Symposium on Colossal Data Analysis and Networking (CDAN) 2016, Indore, India, 18–19 March 2016; pp.1–4. [CrossRef]
7. Jensen, C. Customer Churn Prediction: A Study of Churn Prediction Using Different Algorithem from Machine Learning. Master's Thesis, Copenhagen Business School, Frederiksberg, Denmark, 2020.
8. Kaya; Dong, E.; Suhara, X.; Balcisoy, Y.; S; Bozkaya; B. Behavioral Attributes and Financial churn Prediction. *Epj Data Sci.* **2018**, *7*, 41. [CrossRef]
9. Amuda, K.; Adesesan, A. Customers churn Prediction in Financial Institution Using Artificial Neural Network. *arXiv* **2019**, arxiv:1912.11346.
10. Åman, R. Understanding When Customers Leave: Defining Customer Health and How It Correlates with Software Usage. Master's Thesis, Uppsala University, Uppsala, Sweden, 2017.
11. Sergue, M. *Customer Churn Analysis and Prediction Using Machine Learning for a B2B SaaS Company*; Degree Project in Engineering Physics; School of Engineering Sciences, KTH Royal Institute of Technology: Stockholm, Sweden, 2020.
12. Tsai, C.F.; Lu, Y.H. Customer churn prediction by hybrid neural networks. *Expert Syst. Appl.* **2009**, *36*, 12547–12553. [CrossRef]
13. Zdravevski, E.; Lameski, P.; Apanowicz, C.; Ślezak, D. From Big Data to business analytics: The case study of churn prediction. *Appl. Soft Comput.* **2020**, *90*, 2020, 106164. [CrossRef]
14. Xiahou, X.; Harada, Y. B2C E-Commerce Customer churn Prediction Based on K-Means and SVM. *J. Theor. Appl. Electron. Commer. Res.* **2022**, *17*, 458–475. [CrossRef]
15. García, D.L.; Nebot, À.; Vellido, A. Intelligent Data Analysis Approaches to churn as a Business Problem: A Survey. *Knowl. Inf. Syst.* **2017**, *51*, 719–774. [CrossRef]
16. Berthold, M.R.; Cebron, N.; Dill, F.; Di Fatta, G.; Gabriel, T.R.; Georg, F.; Meinl, T.; Ohl, P.; Sieb, C. KNIME—The Konstanz information miner. *Acm Sigkdd Explor. Newsl.* **2009**, *11*, 26. [CrossRef]
17. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
18. Zhu; Zou, H.; Rosset, S.; Hastie, T. Multi-class AdaBoost. *Stat. Interface* **2009**, *2*, 349–360.
19. Riedmiller, M.; Braun, H. A direct adaptive method for faster backpropagation learning: The RPROP algorithm. In Proceedings of the IEEE International Conference on Neural Networks (ICNN), San Francisco, CA, USA, 28 March–1 April 1993; IEEE: Piscataway, NJ, USA, 1993; Volume 16, pp. 586–591.
20. Raschka, S. Model Evaluation, Model Selection, and Algorithm Selection in Machine Learning. *arXiv* **2018**, arxiv:1811.12808.
21. Akiba, T.; Sano, S.; Yanase, T.; Ohta, T.; Koyama, M. Optuna: A Next-generation Hyperparameter Optimization Framework. arXiv **2019**, arXiv:1907.10902.
22. Saias, J.; Maia, M.; Rato, L.; Gonçalves, T. *Estudo Sobre Modelos Preditivos Para Churn*; Relatório Técnico do Projeto APRA-CP; Universidade de Évora: Évora, Portugal, 2018.