Predicting Multiple Domain Queue Waiting Time via Machine Learning

Carolina Loureiro[1], Pedro José Pereira[1,2], Paulo Cortez[2], Pedro Guimarães[1,2], Carlos Moreira[3], and André Pinho[3]

[1] EPMQ - IT Engineering Maturity and Quality Lab, CCG ZGDV Institute,
Guimarães, Portugal
{carolina.loureiro,pedro.pereira}@ccg.pt
[2] ALGORITMI Centre/LASI, Dep. Information Systems, University of Minho,
Guimarães, Portugal
pcortez@dsi.uminho.pt
[3] Qevo – Queue Evolution Lda., Lisboa, Portugal
{carlos.aj.moreira, andremsp95}@gmail.com

Abstract. This paper describes an implementation of the Cross-Industry Standard Process for Data Mining (CRISP-DM) methodology for a demonstrative case of human queue waiting time prediction. We collaborated with a multiple domain (e.g., bank, pharmacies) ticket management service software development company, aiming to study a Machine Learning (ML) approach to estimate queue waiting time. A large multiple domain database was analyzed, which included millions of records related with two time periods (one year, for the modeling experiments; and two year, for a deployment simulation). The data was first preprocessed (including data cleaning and feature engineering tasks) and then modeled by exploring five state-of-the-art ML regression algorithms and four input attribute selections (including newly engineered features). Furthermore, the ML approaches were compared with the estimation method currently adopted by the analyzed company. The computational experiments assumed two main validation procedures, a standard cross-validation and a Rolling Window scheme. Overall, competitive and quality results were obtained by an Automated ML (AutoML) algorithm fed with newly engineered features. Indeed, the proposed AutoML model produces a small error (from 5 to 7 minutes), while requiring a reasonable computational effort. Finally, an eXplainable Artificial Intelligence (XAI) approach was applied to a trained AutoML model, demonstrating the extraction of useful explanatory knowledge for this domain.

Keywords: CRISP-DM · Automated Machine Learning · Regression.

1 Introduction

Nowadays, human queues are still required in several service sectors (e.g., health, banks). Waiting in these queues is often stressful and exhausting, leading to unsatisfied and frustrated citizens. Therefore, providing a beforehand accurate estimation of citizens waiting time in queues would reduce such frustration, since it allows them to optimize their schedule, avoiding spending an excessive time waiting. Furthermore, this estimation enhances a better resource management by the responsible entities, allowing to avoid excessively long queues. However, an imprecise estimation could produce the opposite effect. If the queue waiting time is overestimated, citizens could loose their turn in the queue, while an underestimation would still force them to wait in the physical queue.

This paper addresses a multiple domain queue waiting time estimation task by adopting a Machine Learning (ML) approach. This research work was developed in collaboration with a Portuguese software development company that operates in the ticket management sector and has several customer companies from multiple domains (e.g., banking). Over the past years, the company collected and stored a large amount of data that holds valuable knowledge related with human queues. Hence, there is a potential in using Data Mining (DM) and ML [21] to extract valuable predictive knowledge that improves the queue waiting time estimation task. Currently, the analyzed company addresses this estimation by using a rather rigid formula that was based on their business expertise. In this work, adopted the popular Cross-Industry Standard Process for Data Mining (CRISP-DM) methodology [20], which provides a framework for developing successful DM projects. In effect, CRISP-DM has been widely used on multiple DM research studies (e.g., [4,16]).

The CRISP-DM methodology is composed by a total of six phases, namely business understanding, data understanding, data preparation, modeling, evaluation and deployment. In this paper, we describe the adopted CRISP-DM execution regarding all these phases. The company business goal is to accurately predict the queue waiting time of a specific ticket, which we addressed as a ML supervised regression task. The sample of data used in this study was collected from the company database server and it is corresponds to millions of tickets withdrawn from 58 stores related with five distinct domains (i.e., banking, insurance companies, pharmacies, public and private services). An initial one-year dataset (from January to December of 2022) was first analyzed and preprocessed, which included data cleaning (e.g., outlier removal), feature engineering and data scaling processes. Then, concerning the modeling stage, five state-of-the-art ML algorithms were adapted and compared: Decision Trees (DT), Random Forest (RF), Gradient Boosted Trees (GBT), deep Artificial Neural Networks (ANN) and an Automated Machine Learning (AutoML). Furthermore, in this phase we defined four input set scenarios (A, B, C and D), which include distinct input feature selections and new engineered attributes that feed the ML models. The ML algorithms and input set scenarios were evaluated under two modes of a robust cross-validation procedure, using both predictive performance and computational effort measures. The performance was evaluated in terms of four popular regression metrics: Mean Absolute Error (MAE), Normalized MAE (NMAE), Root Mean Squared Error (RMSE) and the Area under the Regression Error Characteristic (AREC) curve [4,6]. As for the computational effort, it was measured in terms of training and prediction times. After analyzing the cross-validation results, we selected the best ML model (AutoML method and scenario C, which includes newly engineered attributes). Then, an additional Rolling Window (RW) robust validation procedure [19] was executed, using a larger two-year time period dataset, collected from January 2021 and December 2022. The goal was to realistically simulate the ML model deployment phase performance, further comparing it with the company currently adopted queue time estimation method. Finally, we applied the SHapley Additive exPlanations (SHAP) method [13] to a trained RW model, aiming to demonstrate the extraction of eXplainable Artificial Intelligence (XAI) knowledge.

This paper is organized as follows. The related work is presented in Section 2. Next, Section 3 details the adopted DM approach in terms of the CRISP-DM methodology phases. Finally, Section 4 discusses the main conclusions and presents future steps.

2 Related Work

Accurately estimating beforehand human waiting time in a queue is crucial tool for ticket management systems, providing benefits for both citizens and organizations. Waiting time estimation is a challenging task, since it can be affected by a wide range of phenomena (e.g., sudden increase of customers, employee attendance slowness) that often are not directly measured by ticket management systems. Recently, several research studies have addressed this task, assuming traditional approaches, such as: Average Predictions (AP) [18]; Queuing Theory [17]; and DM/ML approaches [7]. In this work, we detail the ML based approaches, since they are more related with our CRISP-DM approach.

In 2018, a study was carried out in Portugal regarding the use ML algorithms to predict waiting times in queues, assuming a categorical format, thus a multi-class classification task (e.g., “very high”, “low”) [7]. To validate the results, the authors used a dataset from an emergency department of a Portuguese hospital, containing 4 years of data and around 673.000 records. Only one ML algorithm was used (RF). Interesting results were obtained for the most frequent time interval categories (e.g., “low”), although poor quality results were achieved for the infrequent classes.

In 2019, Sanit-in et al. [18] compared 3 different approaches to estimate queue waiting times: Queuing Theory [17], AP and ML. However, similarly to [7], instead of estimating the exact waiting time, the authors grouped the values into multi-class intervals (e.g., “very short”, “short”, “medium”, “long” and “very long”). In order to validate their results, two datasets related to the queuing sector in Thailand were used. The first (1,348 records) was related with a medical care service, while the second (3,480 rows) was related to a post office store. In terms of used input features, the authors selected (among others): the queue identification number; the day of the week, hour and corresponding period of the day when the ticket was withdrawn; and the number of tickets taken and served per minute. For both datasets, the ML best results were achieve by the RF algorithm, with an overall accuracy of 86% and 82%, respectively.

In a different context, in 2019, Kyritsis et al. [11] performed a study that aimed to present the benefits of using ML for predicting queue waiting times in banks, assuming a regression approach. The ML algorithm used was an ANN and it was tested using a four week dataset with around 52,000 records related to 3 banks in Nigeria. The ANN outperformed both AP and Queuing Theory estimation systems.

More recently, in 2020, Kuo et al. [10] studied a real-time prediction of queue waiting time in an hospital emergency department in Hong Kong. In terms of ML, 4 regression algorithms were compared: Linear Regression (LR), used as baseline; ANN; Support Vector Machines (SVM) and Gradient Boost Machines (GBM). The used dataset had nearly 13,000 records and two combinations of attributes were used: using attributes including patient triage category (non-urgent, semi-urgent and urgent), arrival time and number of doctors in the emergency room; and using the same attributes but complemented with information about the patients in the queue. Additionally, the authors applied outlier detection and removal techniques and feature selection, using a LR feature importance measure. In terms of results, the GBM achieved the better predictive metrics for both attribute combinations.

Finally, in 2023, Benevento et al. [3] analyzed the use of ML to predict, in real time, the waiting time in emergency department queues, aiming to improve the department resource management. The datasets used refer to two hospitals in Italy and each contained approximately 500,000 records. The ML regression algorithms used were: Lasso, RF, SVM, ANN and an Ensemble Method (EM). The attributes used include information regarding the patient age, mode of arrival at the emergency room, wristband color after triage, an average estimation of patient arrivals by wristband color, the number of patients in the queue, grouped by wristband color, among others. The results revealed that the ensemble (EM) provided the best predictive performance, achieving a MAE of approximately 30 minutes.

When compared with our study, the related works are focused in different and single queuing domains, mostly related with health institutions. In particular, emergency services were targeted in [7,18,10,3], while bank queues were considered in [11] and a post office store data was modeled in [18]. In contrast, our research targets a single global ML model for several stores from multiple domains (e.g., banks, pharmacies). This is a more complex task, since it can only use more general queuing attributes that are common to all analyzed domains. Thus, it is not feasible to employ very specific features, such as the wristband color from the emergency services. Aiming to improve the waiting time estimation performance, in this work we use both the company ticket management attributes and newly proposed engineered attributes, computed using the company queuing data. Additionally, in our study we explore much larger dataset, with more than 2 million records, when compared with the ones used by the related works (e.g., 673,000 in [7] and 500,000 in [3]). Furthermore, similarly to [11,10,3], we addressed the queue waiting time predictions as a pure regression task, instead of a classification of time intervals, which is less informative and that was performed in [7,18]. Moreover, we explore a recently proposed AutoML tool, which automatically selects the best predictive model among seven distinct ML algorithms. None of the related ML works have employed an AutoML. Finally, we employ two robust validation schemes to measure the predictive performance of the ML models, a standard cross-validation (under two modes) and a RW, comparing the ML results with the method currently adopted by the analyzed company. In particular, we note that the RW performs a realistic simulation of usage of the predictive models in a real environment, since it simulates several training and test iterations over time [19].

3 CRISP-DM Methodology

This section details the developed work in each of the CRISP-DM methodology phases for the multiple domain queue waiting time prediction task.

3.1 Business Understanding

Currently, the company under study uses their own solution to estimate multiple domain queue waiting time, which is quite complex and involves multiple steps. First, they remove outliers from the data and compute the average service time for each costumer, store and counter for the next day, using only data from homologous days. This step is performed daily, during the night, in order to avoid a computational system overload. Next, when there is a ticket withdrawing request, their solution queries the database to get the number of counters open to a specific service, the open counters status (e.g., servicing, paused) and the number of citizens in the queue and being serviced. Then, they simulate the allocation of all citizens to counters, order by their priority, resulting in multiple queues (one by counter). Lastly, it sums the average service times relative to the ticket being printed queue, returning it as their waiting time estimation.

The goal of this project was set in terms of predicting the queue waiting time by using supervised ML regression algorithms. Moreover, we considered the improvement of the current estimation method (termed here as “Company”) as a success criteria, measured in terms of the Mean Absolute Error (MAE) computed over a test set. This criteria was was validated by the company. Additionally, the ML model inference time (when producing a prediction) must be equal or less than 10 milliseconds, in order to ensure an acceptable ticket withdrawing time. Concerning the software, we adopted the Python programming language. In particular, due to the vast volume of data, we adopted the Spark computational environment for data preprocessing operations and MLlib, which is the Spark ML library, for the modeling phase, as well as H2O (for AutoML) and TensorFlow (for the deep ANN).

3.2 Data Understanding and Preparation

Two datasets were collected from the company database server using a Structured Query Language (SQL). At an initial CRISP-DM execution stage, the company provided us a sample that included 1,238,748 records and 52 attributes. The raw data was related with tickets withdrawn from a set of 58 stores, associated with five different ticket management sectors (banking, insurance companies, pharmacies, public and private services), from January 2022 to December 2022. We used this one-year dataset when executing the first five CRISP-DM stages, which includes the cross-validation ML comparison experiments that were held during the CRISP-DM modeling stage. Then, in a later research stage, we had access to a larger two-year company sample, with a total of 2,087,771 records from January 2021 to December 2022 and related with the same 58 stores. This second dataset was used only for the CRISP-DM deployment simulation experiments.

Using the one-year raw data, we first executed the CRISP-DM data understanding and preparation stages. The latter stage was performed in collaboration with the business experts. The preprocessing aimed to enhance the quality of the data used to feed the ML models and it included several operations: data cleaning, outlier removal, creation of new data attributes (feature engineering) and data transformations.

First, we discarded all null valued attributes (e.g., with no citizen information). We also ignored data variables that could only be computed after the ticket being printed (e.g., service duration, counter and user that served the citizen), thus unfeasible to be used in a real-time prediction. The remaining 23 data attributes are presented in Table 1 and were considered for the CRISP-DM modeling phase, under distinct input set combinations, as shown in Column **Scenarios** and detailed in Section 3.3.

In terms of outlier removal, several records presented queue waited times above 20 hours, which reflects errors in the costumers data gathering process. Together with the company experts, a maximum threshold value of 8 hours was set for the waited time, allowing to remove all records that did not fulfill this time limit. Additionally, the priority attribute (isPriority from Table 1), which is computed when the counter user calls the citizen, revealed several inconsistencies and led to the removal of several records. Then, we detected around 57% of null values in the company estimation of queue waited time (CompanyEstimation attribute). Since these null values do not affect the ML models, we decided to maintain them and adopt two evaluation modes. In the first mode (”All”), we compute the regression metrics using all the test records. In the second (”Sampled”), we compute the same performance metrics using only the records that have the company estimation, in order to ensure a fair comparison with the company solution (see Section 3.4). Finally, 1.5% of waited time values were null and we have calculated them by subtracting the printing hour to the calling hour attribute.

Aiming to further improve the ML results, the next step of the data preparation stage included the creation of 9 new attributes, as presented in Table 2. The first 8 new attributes concern with the average and standard deviation values of waiting times and service duration, in seconds, for both the previous and current days, for a given store, service and priority. Finally, the 9^{th} attribute is

Table 1: List of analyzed data attributes.

Context	Name	Description	Scenarios
Location	storeId	Identifier of the store where the ticket was withdrawn.	A,B,C,D
	entityId	Identifier of the store entity.	
	peopleInFront	Number of people in front for a given store and service.	
	serviceId	Identifier of the service.	
	storeProfileStoreId	Identifier of the in-store profile set up.	
	entityQueueId	Identifier of the entity queue.	
	partnerId	Identifier of the partner.	A, C
Printing device	inputChannelId	Identifier of where the ticket request was made.	A,B,C,D
	outputChannelId	Identifier of the channel where the ticket will be printed.	
	deviceId	Identifier of the device where the ticket was withdrawn.	A, C
Ticket info	printingHour	Time of ticket request.	A,B,C,D
	isPriority	If the ticker has priority.	
	isFastLane	If the ticket has a fast lane priority.	
	isForward	If the ticket is forwarded from other service/store.	A, C
	ticketLanguageId	Ticket language identifier.	
	ticketOutputId	Ticket format identifier.	
	ticketTypeId	Ticket type identifier.	
	ticketNumber	Ticket number.	
	originalServiceId	If forward, the initial service ID.	
	originalStoreId	If forward, the initial store ID.	
	subId	Number of times that a ticket was forward.	
Target	CompanyEstimation	Company waiting time estimation (in seconds).	–
	waitedTime	Queue waiting time (in seconds).	–

the waiting time of the last similar ticket withdrawn, with this similarity being defined as the same store, service and priority.

Table 2: List of newly computed attributes.

Name	Description
AvgPrev_waitedTime	Average waiting time for previous day.
AvgCurr_waitedTime	Average waiting time for current day.
AvgPrev_duration	Average service duration for previous day.
AvgCurr_duration	Average service duration for current day.
StdPrev_waitedTime	Standard deviation of waiting times for previous day.
StdDevCurr_waitedTime	Standard deviation of waiting times for current day.
StdPrev_duration	Standard deviation of service duration for previous day.
StdDevCurr_duration	Standard deviation of service duration for current day.
LastSimilarWaitedTime	Waited time of the last similar ticket.

All input data attributes are numeric and have different scales (e.g., week day ranges from 0 to 6; store ID ranges from 8 to 537), which often results in different ML algorithms impacts due only to scale differences [15]. Therefore, in the last data preparation step, we performed the scale normalization to all input attributes by applying a standard scaling (also known as z-scores) [8], which transforms each attribute to have a mean of zero and standard deviation equal to 1.

3.3 Modeling

The first task of CRISP-DM modeling stage concerns with the selection of modeling techniques. After analyzing the related studies, in terms of ML algorithm, the most popular choice were RF [7,18,3] and ANN [11,10,3] and therefore we tested them. Furthermore, we tested two other tree-based algorithms, Decision Trees (DT) and Gradient-Boosted Trees (GBT), and an Automated ML (AutoML) algorithm, as provide by the H2O tool [12].

All ML algorithms were implemented by using the Python programming language. In particular, we used the `pyspark` package for all tree-based methods, with all the default hyperparameters. In terms of defaults: RF uses a total of 20 trees, each one with a maximum depth of 5, and one third as feature subset strategy, i.e., each tree node considers one third of the total of features for split; DT uses maximum depth of 5; and GBT uses a maximum of 20 iterations, maximum depth of 5, all features for subset strategy and squared error as loss function.

As for the ANN implementation, since `pyspark` does not have an ANN implementation for regression tasks, we used the popular `TensorFlow` package [1]. The implemented ANN architecture, similarly to the ones used in [14,2], uses a triangular shape deep Multilayer Perceptron (MLP). Assuming the input layer size I, the H hidden layers with size L, and a single output neuron, each subsequent layer size is smaller in a way that $I > L_1 > L_2 > ... > L_H > 1$. After some preliminary experiments, assuming only the first iteration of the 10-fold cross-validation procedure (as detailed in the last paragraph of this section), we defined the following ANN setup. The ANN model includes a total of $H = 5$ hidden layers, with the following layer structure: (I, 25, 20, 15, 10, 5, 1). In each layer, the ReLu activation function was used. Furthermore, in order to avoid overfitting, we added: a dropout applied on the 2^{nd} and 4^{th} hidden layers, with a dropout ratio of 0.2 and 0.1, respectively, as in [14]; an inverse time decay to Adam optimizer, with an initial learning rate of 0.0001, a decay rate of 1 and a decay step of 30 epochs, similarly to [5]; and an early stopping monitoring of the Mean Squared Error (MSE) on the validation data, with a patience of 20 epochs, similar to [14]. Lastly, we trained our ANN with a batch size of 1000, for a maximum of 100 epochs, using the MSE as loss function.

Finally, concerning the AutoML algorithm, we selected the H2O tool based on recent AutoML benchmarking studies [6,15]. In terms of implementation, we used the `h2o` python package, assuming the default parameters in terms of the searched ML algorithms, which were: Generalized Linear Model (GLM), RF, Extremely Randomized Trees (XRT), Gradient Boosting Machine (GBM), XGBoost, a Deep Learning Neural Network (DLNN) and two Stacked Ensembles. Regarding the stopping metric, we selected MSE, which is also used to sort the learderboard on the validation data. Additionally, we set a maximum runtime limitation of 30 minutes for the model and hyperparameter selection process.

During the CRISP-DM modeling phase, we also designed multiple input selection scenarios, allowing us to test different hypotheses regarding the influence of attributes on the queue waiting time prediction. In particular, we compared 4 attribute combination scenarios: A) use of all 21 input attributes presented in Table 1 (from storeID to subId); B) use of domain knowledge selected attributes (as advised in [21]), which corresponds to the 11 input variables listed in in Table 1 and that were signaled as relevant by the domain experts; C) combination of scenario A) with the 9 new engineered attributes shown in Table 2 (e.g., mean and standard deviation values of waiting times), thus resulting in a set with 30 input features; and D) combination of scenario B) with the 9 created attributes, leading to 20 numeric inputs.

In order to evaluate the performance of the distinct input scenario and ML algorithm combinations, we executed the standard 10-fold cross-validation [8] using the whole one-year data (from 2022). The 10-fold procedure randomly divides the dataset into 10 equal sized data partitions. In the first iteration, the data included in 9 of the folds is used to train a ML model, which is then tested using the remaining data. This procedure is repeated up to 10 times, with each 10-fold iteration assuming a distinct fold as the external (unseen) test data. Regarding the two ML algorithms that require validation data (ANN and AutoML), the training data is further randomly split into fit (with 90%) and validation (with the remaining 10%) sets.

3.4 Evaluation

During this step, we performed the evaluation of all ML algorithms and input selection scenarios using two different modes. The first mode, termed here as "All", computes the performance metrics for each of the 10-fold test set partitions by using the entire test data. The second "Sampled" mode filters first the records with null values for the company queue waiting time estimation from the 10-fold test sets, keeping only the test examples for which there is a company method estimation value. Thus, the "Sampled" mode ensures a fair performance comparison between the ML algorithms and the estimation system currently used by the company.

In this work, ML algorithms are evaluated in terms of two relevant problem domain dimensions: the computational cost and predictive performance. For the former, we compute both the algorithm training time, in seconds, and the prediction time (i.e., the time to perform a single estimation), measured in microseconds. Regarding the latter, we selected four popular regression metrics[4,6]: Mean Absolute Error (MAE), Normalized NMAE (NMAE), Root Mean Squared Error (RMSE) and Area under the Regression Error Characteristic curve (AREC). Although multiple metrics are presented, the company agreed that a major focus should be given to the MAE measure and to the prediction time.

Table 3 presents the median 10-fold cross-validation predictive and computational measures obtained for all ML algorithms and input set scenarios, assuming the "All" evaluation mode. The predictive performance statistical significance is measured by adopting the nonparametric Wilcoxon test [9] over the 10-fold results. Regarding the predictive metrics, the results clearly show that the H2O is the best ML algorithm, regardless of the scenario, returning MAE values that are inferior to 11 minutes for all scenarios. In particular, the best MAE value (6.48 minutes) was achieved for H2O and scenario C. The second best ML performance is provided by GBT, with a median MAE of nearly 1 minute more, and then ANN, DT and RF, respectively. On the other hand, in terms of training time, H2O has the highest values in all scenarios, requiring the allowed 30 minute execution time for the model and hyperparameter selection. Although it is the slowest ML model it terms of training, H2O is the fastest one in terms of the predictive time, regardless the scenario, with the maximum inference time of 7.69 microseconds for scenario D. This time is much lower than the company 10 millisecond limit for a real-time ticket management time estimation. Regarding the computational cost of the remaining algorithms, DT is the fastest during the training process, followed by RF, GBT and ANN. In terms of time taken for each prediction, H2O is the best option, followed by DT, RF, GBT and ANN, which take almost 10 times more to perform predictions. As for the scenarios, all models achieve a better predictive performance when using the new 9 attributes calculated during the data preparation phase (scenarios C and D). In particular, H2O, GBT and ANN achieve a better predictive performance on scenario C, while DT and RF obtain their best predictive results on scenario D.

Table 4 displays a comparison of predictive metrics for all ML models across each scenario, in terms of median 10-fold cross validation measures, for the “Sampled” evaluation mode. Since the ML train and predictive time are the same as in mode “All” and we do not have access to the estimation time of the company solution, these values were not considered on this evaluation mode. In terms the predictive performance, the obtained results are similar to the ones obtained for the “All” mode. In effect, H2O also achieves the best predictive in all scenarios, with the best MAE value of 5.20 minutes for scenario C. In terms of MAE, the best performing ML algorithm is H2O, followed by GBT, DT, ANN and RF, respectively. Concerning the scenarios, H2O, ANN and GBT achieve a better predictive performance when using the features from scenario C, while the remaining ML models improved their performance when using the attributes from scenario D. In this evaluation mode, all the predictive results improved, when compared with the “All” mode, with the highest MAE value (10.52 minutes) being obtained by RF when using the attributes from scenario B.

A summary of the best ML algorithm predictive results (H2O), obtained for all scenario and evaluation modes, is presented in Table 5. In particular, we highlight the scenario C results, for which H2O obtained MAE values below 7 minutes for the “All” and “Sampled” modes. In case of the latter mode, we compare all explored scenarios with the company current estimation system. Clearly, the best results are provided by H20 regardless of the input set scenario. In effect, for scenario C, the company system achieves a median MAE value of 9.86 minutes, while the H2O method only required 5.20 minutes. Thus, an impressive 53% MAE improvement was obtained by the H2O algorithm. Following this results, we selected for the next CRISP-DM stage the H2O algorithm and the

Table 3: Comparative results for evaluation mode “All” (median cross-validation values; best values in **bold**).

Scenario	**ML Model**	**MAE (min.)**	**RMSE (min.)**	**AREC (%)**	**Train Time (s)**	**Prediction Time (μs)**
A	DT	12.20	23.54	67.58	9.09	10.04
	RF	12.07	23.29	67.47	12.46	11.23
	GBT	10.94	21.82	70.50	30.32	10.18
	H2O	**10.22***	20.48*	72.24*	1798.15	5.56
	ANN	11.01	22.80	70.64	624.58	106.86
B	DT	12.02	23.75	68.08	6.61	9.23
	RF	12.05	23.43	67.58	9.48	9.01
	GBT	11.16	22.46	69.96	26.57	9.50
	H2O	**10.76***	21.41*	70.94*	1796.78	6.33
	ANN	11.26	23.38	70.01	570.77	89.00
C	DT	8.90	18.40	75.08	11.06	10.37
	RF	9.12	17.67	73.91	14.67	10.53
	GBT	7.71	16.87	78.34	33.53	11.47
	H2O	**6.48***	14.20*	81.06*	1799.54	7.37
	ANN	7.74	16.58	77.97	586.48	110.11
D	DT	8.86	18.41	75.44	8.41	9.22
	RF	8.93	17.67	74.57	11.28	9.46
	GBT	7.84	17.25	78.14	29.58	9.79
	H2O	**6.84***	14.96*	80.20*	1798.87	7.69
	ANN	8.56	17.84	76.31	645.73	90.81

$\star$ – Statistically significant under a paired comparison with all other methods.

input set scenario C as the best predictive ML approach to be further compared with the company based method.

3.5 Deployment

In terms of deployment, we did not implement the DM approach on the company environment yet. Nevertheless, we performed a realistic simulation of its implementation potential performance by employing a RW validation scheme [19]. During this stage execution, we had access to a larger sample of two-year data, relative to the same 58 stores. The two-year dataset includes around 2 millions of records collected from January 2021 and December 2022. Using this larger sample, we first executed the same data preprocessing that was previously applied to the one-year data (described in Section 3.2), selecting then the input variables associated with scenario C, which led to best predictive results shown in Section 3.4.

Next, the RW simulation was executed over the two-year preprocessed data. The RW approach mimics what would occur in a real-world environment, since

Table 4: Comparative results for evaluation mode "Sampled" (median cross-validation values; best values in **bold**).

Scenario	**ML Model**	**MAE (min.)**	**RMSE (min.)**	**AREC (%)**
A	DT	9.23	15.98	72.67
	RF	9.94	15.73	69.64
	GBT	9.03	14.93	72.59
	H2O	**7.97***	14.38*	75.90*
	NN	9.18	15.53	73.19
B	DT	9.83	15.99	70.55
	RF	10.52	15.86	68.23
	GBT	9.27	15.23	71.86
	H2O	**8.45***	14.51*	73.90*
	NN	10.11	16.27	69.99
C	DT	7.56	12.41	76.60
	RF	7.83	11.55	75.34
	GBT	6.15	10.55	80.82
	H2O	**5.20***	9.30*	83.66*
	NN	6.21	11.16	80.80
D	DT	7.36	11.74	77.08
	RF	7.78	11.63	75.38
	GBT	6.39	10.79	79.93
	H2O	**5.52***	9.48*	82.62*
	NN	7.26	12.63	77.79

$\star$ – Statistically significant under a paired comparison with all other methods.

it assumes that data is time-ordered, thus the ML model is always trained using historical data and produces predictions for more recent unseen data. Moreover, it performs several training and testing iterations over time.

The RW training time window was set to one year and the testing and sliding windows were set to two weeks. In the first RW iteration, one year of the oldest records were used to train the ML algorithm, except for the last week data that was used as a validation subset for H2O model selection purposes. Then, the subsequent two weeks of data were used as the external (unseen) data, for predictive testing purposes. In the second RW iteration, we update the training data by advancing the testing period 2 weeks in time, thus discarding the oldest two weeks of data. The next two subsequent weeks of data are now used for test purposes. And so on. In total, this procedure produces in 26 iterations, advancing 2 weeks of data in each iteration, resulting in a total of 1 year of predictions. In order to reduce the computational effort, the H2O algorithm selection is only executed during the first RW iteration, assuming the last week of the available training data as the validation subset, allowing to select the best ML algorithm and its hyperparameters. Once this model is selected, in the

Table 5: Overall H2O and company method predictive results (median cross-validation values; best values in **bold**).

Mode	**Scenario**	**MAE (min.)**	**NMAE (%)**	**RMSE (min.)**	**AREC (%)**
"All"	A	10.22	2.15	20.48	72.24
	B	10.76	2.26	21.41	70.94
	C	**6.48**	**1.37**	**14.20**	**81.06**
	D	6.84	1.44	14.96	80.202
"Sampled"	A	7.97	3.14	14.38	75.90
	B	8.46	3.32	14.51	73.90
	C	**5.20**⋆	**2.00**⋆	**9.30**⋆	**83.66**⋆
	D	5.53	2.11	9.48	82.62
	Company	9.86	3.96	19.88	72.34

⋆ – Statistically significant under a paired comparison with the Company method.

remaining RW iterations (from 2 to 26), we just retrain the selected ML using the newer training data. In terms of the H2O setup, we used the same as in the previous experimentation (Section 3.3).

Table 6 presents the obtained RW results. In terms of mode "All", H2O obtained only a slight increase on the median MAE value (0.40 minutes) when compared with the previous experiments, which demonstrates a consistency of the H2O Scenario C model performance. Moreover, the H2O algorithm outperformed the current company estimation system by 4.7 minutes (improvement of 53%) in the "Sampled" evaluation mode, which suggests a strong potential predictive value for the company ticket management system.

Table 6: Overall results for the simulation system (median RW values; best values in **bold**).

Mode	**Estimator**	**MAE (min.)**	**NMAE (%)**	**RMSE (min.)**	**AREC (%)**
"All"	H2O	6.88	1.49	14.81	80.06
"Sampled"	H2O	**5.37**⋆	**2.48**⋆	**9.76**⋆	**83.15**⋆
	Company	10.07	4.60	19.11	72.53

⋆ – Statistically significant under a paired comparison with the Company method.

A further predictive analysis is provided in Fig. 1, which shows the median REC curves for all RW iterations (colored curve), associated with the respective Wilcoxon 95% confidence intervals (colored area), for both the systems tested. Particularly, the REC curve shows the model accuracy ($y - axis$), measured in terms of correct predictions for a given absolute error tolerance ($x - axis$). For instance, for a 5 minute tolerance, H2O has an accuracy of nearly 70%. In this graph, we limited the absolute deviation to 30 minutes, after which we consider that the predictions have low value for the company. When comparing both algorithms results, the company estimation system has a better accuracy (40%) for a very small error tolerance (less than 2 minutes), which is quite low. As for the ML approach, it achieves a better accuracy for the remaining absolute error values of the curve. Moreover, in the H2O (blue) curve, the Wilcoxon confidence intervals are practically unnoticed in Fig. 1 (cyan shadowed area), denoting a small variation of model accuracy for all absolute errors over the 26 RW iterations, which increases the confidence and reliability of the model. On the other hand, the company system presents a greater variation of results, denoted by the gray shadowed area, particularly for absolute deviations between 5 and 20 minutes. These results reflect a higher level of uncertainty associated to the model in the mentioned tolerance interval.

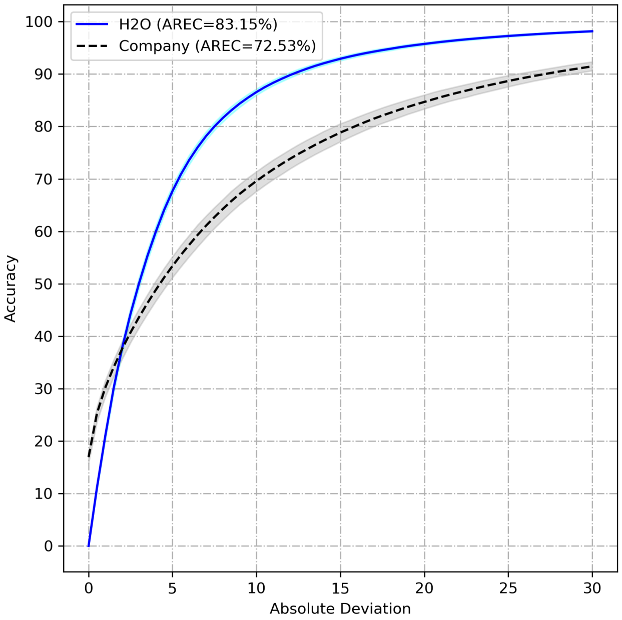

Fig. 1: Median REC curves with Wilcoxon 95% confidence intervals for RW.

For demonstration purposes, we analyzed the ML model selected by the H2O algorithm in the first RW iteration, which was XGBoost, with a total of 50 trees. Although it had a training time of 30 minutes in the first iteration, which corresponds to the established limit, its retraining on the remaining iterations was very fast. Specifically, the median training time for those iterations was around 25 seconds. The H2O tool includes an XAI module based on the SHAP method [13]. Using such XAI, the top of Fig. 2 presents the 5 most relevant inputs extracted from the XGBoost model on the last RW iteration. The waiting time of the last similar ticket is the predominant attribute, with a relative importance superior to 70%. In second place appears the number of people in front in the queue, with less than 10%, followed by the averaged waited time in the current day (4%), the number of times that a ticket was forwarded (subID, 4%), hour of the day (2%) and the ticket type ID (2%). Overall, these results demonstrate the importance of data preparation stage, since 2 of the newly engineered input variables are among the 4 top relevant inputs of the model. As for the bottom XAI graph of Fig. 2, it shows the overall impact of an input in the predicted responses. For instance, any decrease of the top three inputs (e.g., waiting time) produces also an average decrease on the estimated time (as shown by the blue colored dots).

4 Conclusions

In this paper, we demonstrate the execution of the CRISP-DM methodology to predict a challenging task: multiple domain queue waiting times for printed tickets of physical stores. Working in collaboration with a ticket management software company, we have analyzed millions of records, aiming to compare a ML approach with the current estimation method adopted by the company. Using a one-year dataset (related with the year of 2022), the data was first analyzed and preprocessed. Then, five ML regression algorithms and four input selection scenarios were compared, using a robust cross-validation procedure and several predictive and computational measures. The best modeling results were obtained by an AutoML algorithm fed with newly engineered attributes (scenario C). In the deployment phase, we applied a RW procedure to realistically simulate the predictive performance of the selected ML approach. The RW experiments were executed over a larger two-year dataset (collected from January 2021 to December 2022), assuming a total of 26 training and testing iterations over time (one year of predictions). Overall results, competitive results were obtained by the AutoML method, both in the evaluation and deployment phases, show a high level of consistency and outperforming the current company estimation system. Furthermore, the selected AutoML tool requires a reasonable computational effort and very fast inference times, thus being feasible for real-time responses. Finally, we used a XAI approach to demonstrate the extraction of explanatory knowledge from a trained predictive model.

The obtained results were shown to the ticket management software company, which provided a positive feedback. Indeed, in future work, we intend to implement our approach in the company real-world environment and further assess the quality of its predictions. Furthermore, we plan to create additional engineered features (e.g., average waiting time for specific time periods) and also include external features (e.g., meteorology data) in the next CRISP-DM iterations.

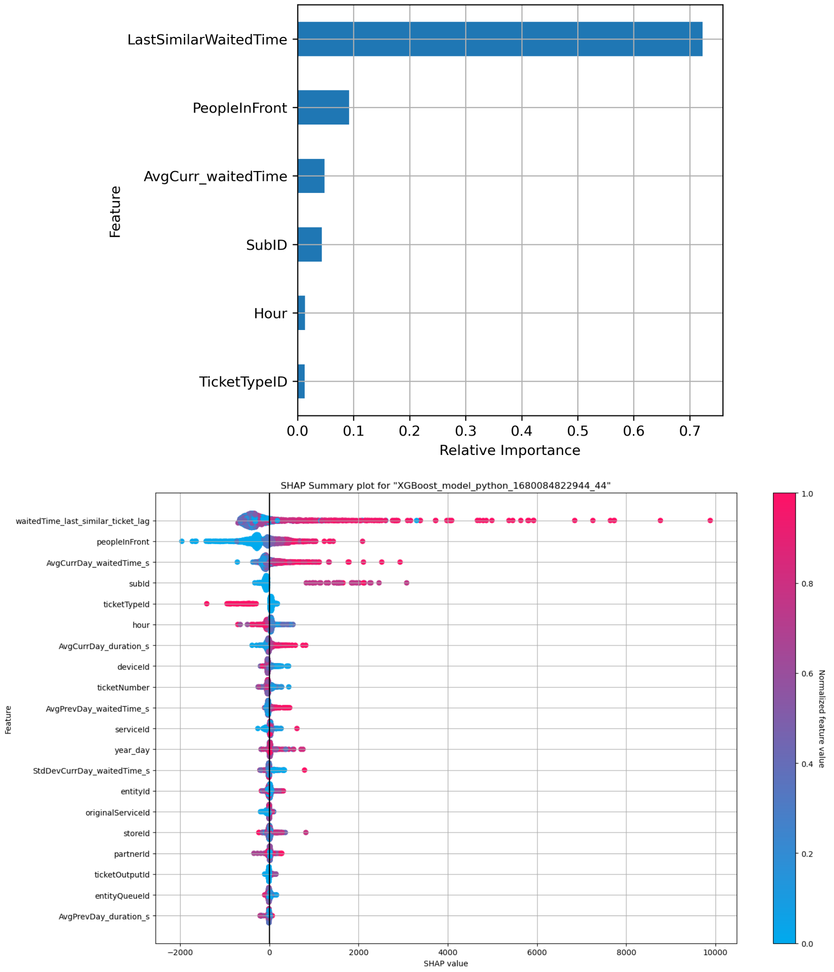

Fig. 2: Input importance for H2O best model on the last iteration of RW (top) and overall impact of an input in the predicted responses (bottom).

Acknowledgments

This work has been supported by FCT - Fundação para a Ciência e Tecnologia within the R&D Units Project Scope: UIDB/00319/2020 and the project "QOMPASS .: Solução de Gestão de Serviços de Atendimento multi-entidade, multiserviço e multi-idioma" within the Project Scope NORTE-01-0247-FEDER-038462.

References

1. Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., Zheng, X.: TensorFlow: Large-scale machine learning on heterogeneous systems (2015), `https://www.tensorflow.org/`, software available from tensorflow.org
2. Azevedo, J., Ribeiro, R., Matos, L.M., Sousa, R., Silva, J.P., Pilastri, A.L., Cortez, P.: Predicting yarn breaks in textile fabrics: A machine learning approach. In: Cristani, M., Toro, C., Zanni-Merk, C., Howlett, R.J., Jain, L.C. (eds.) Knowledge-Based and Intelligent Information & Engineering Systems: Proceedings of the 26th International Conference KES-2022, Verona, Italy and Virtual Event, 7-9 September 2022. Procedia Computer Science, vol. 207, pp. 2301–2310. Elsevier (2022). https://doi.org/10.1016/j.procs.2022.09.289
3. Benevento, E., Aloini, D., Squicciarini, N.: Towards a real-time prediction of waiting times in emergency departments: A comparative analysis of machine learning techniques. International Journal of Forecasting **39**(1), 192–208 (2023). https://doi.org/10.1016/j.ijforecast.2021.10.006
4. Caetano, N., Cortez, P., Laureano, R.M.S.: Using data mining for prediction of hospital length of stay: An application of the CRISP-DM methodology. In: Cordeiro, J., Hammoudi, S., Maciaszek, L.A., Camp, O., Filipe, J. (eds.) Enterprise Information Systems - 16th International Conference, ICEIS 2014, Lisbon, Portugal, April 27-30, 2014, Revised Selected Papers. Lecture Notes in Business Information Processing, vol. 227, pp. 149–166. Springer (2014). https://doi.org/10.1007/978-3-319-22348-3_9
5. Core, T.: Overfit and Underfit, `https://www.tensorflow.org/tutorials/keras/overfit_and_underfit`, accessed: 2023-03-28
6. Ferreira, L., Pilastri, A.L., Martins, C.M., Pires, P.M., Cortez, P.: A comparison of automl tools for machine learning, deep learning and xgboost. In: International Joint Conference on Neural Networks, IJCNN 2021, Shenzhen, China, July 18-22, 2021. pp. 1–8. IEEE (2021). https://doi.org/10.1109/IJCNN52387.2021.9534091
7. Gonçalves, F., Pereira, R., Ferreira, J., de Vasconcelos, J.B., Melo, F., Velez, I.: Predictive analysis in healthcare: Emergency wait time prediction. In: Novais, P., Jung, J.J., Villarrubia-González, G., Fernández-Caballero, A., Navarro, E., González, P., Carneiro, D., Pinto, A., Campbell, A.T., Durães, D. (eds.) Ambient Intelligence - Software and Applications -, 9th International Symposium on Ambient Intelligence, ISAmI 2018, Toledo, Spain, 20-22 June 2018. Advances in Intelligent Systems and Computing, vol. 806, pp. 138–145. Springer (2018). https://doi.org/10.1007/978-3-030-01746-0_16
8. Hastie, T., Tibshirani, R., Friedman, J.H.: The Elements of Statistical Learning: Data Mining, Inference, and Prediction, 2nd Edition. Springer Series in Statistics, Springer, NY, USA (2009). https://doi.org/10.1007/978-0-387-84858-7
9. Hollander, M., Wolfe, D.A., Chicken, E.: Nonparametric statistical methods. John Wiley & Sons, NJ, USA (2013)
10. Kuo, Y., Chan, N.B., Leung, J.M.Y., Meng, H., So, A.M., Tsoi, K.K., Graham, C.A.: An integrated approach of machine learning and systems thinking for waiting time prediction in an emergency department. Int. J. Medical Informatics **139**, 104143 (2020). https://doi.org/10.1016/j.ijmedinf.2020.104143

11. Kyritsis, A.I., Deriaz, M.: A machine learning approach to waiting time prediction in queueing scenarios. In: Second International Conference on Artificial Intelligence for Industries, AI4I 2019, Laguna Hills, CA, USA, September 25-27, 2019. pp. 17–21. IEEE (2019). https://doi.org/10.1109/AI4I46381.2019.00013
12. LeDell, E., Poirier, S.: H2O AutoML: Scalable automatic machine learning. 7th ICML Workshop on Automated Machine Learning (AutoML) (July 2020), https://www.automl.org/wp-content/uploads/2020/07/AutoML_2020_paper_61.pdf
13. Lundberg, S.M., Lee, S.: A unified approach to interpreting model predictions. In: Guyon, I., von Luxburg, U., Bengio, S., Wallach, H.M., Fergus, R., Vishwanathan, S.V.N., Garnett, R. (eds.) Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA. pp. 4765–4774 (2017)
14. Matos, L.M., Cortez, P., Mendes, R., Moreau, A.: Using deep learning for mobile marketing user conversion prediction. In: International Joint Conference on Neural Networks, IJCNN 2019 Budapest, Hungary, July 14-19, 2019. pp. 1–8. IEEE (2019). https://doi.org/10.1109/IJCNN.2019.8851888
15. Pereira, P.J., Gonçalves, C., Nunes, L.L., Cortez, P., Pilastri, A.: AI4CITY - An Automated Machine Learning Platform for Smart Cities. In: SAC '23: The 38th ACM/SIGAPP Symposium on Applied Computing, Tallinn, Estonia, March 27 - 31, 2023. pp. 886–889. ACM (2023). https://doi.org/10.1145/3555776.3578740
16. Ribeiro, R., Pilastri, A.L., Moura, C., Rodrigues, F., Rocha, R., Cortez, P.: Predicting the tear strength of woven fabrics via automated machine learning: An application of the CRISP-DM methodology. In: Filipe, J., Smialek, M., Brodsky, A., Hammoudi, S. (eds.) Proceedings of the 22nd International Conference on Enterprise Information Systems, ICEIS 2020, Prague, Czech Republic, May 5-7, 2020, Volume 1. pp. 548–555. SCITEPRESS (2020). https://doi.org/10.5220/0009411205480555
17. Saaty, T.L.: Elements of queueing theory: with applications, vol. 34203. McGraw-Hill New York (1961)
18. Sanit-in, Y., Saikaew, K.R.: Prediction of waiting time in one stop service. International Journal of Machine Learning and Computing **9**(3), 322–327 (2019)
19. Tashman, L.J.: Out-of-sample tests of forecasting accuracy: an analysis and review. International journal of forecasting **16**(4), 437–450 (2000)
20. Wirth, R., Hipp, J.: Crisp-dm: Towards a standard process model for data mining. In: Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining. vol. 1, pp. 29–39. Manchester (2000)
21. Witten, I.H., Frank, E., Hall, M.A., Pal, C.J.: Data mining: practical machine learning tools and techniques, 4th Edition. Morgan Kaufmann (2016)