AI4CITY - An Automated Machine Learning Platform for Smart Cities

Pedro José Pereira
EPMQ - IT Engineering Maturity and Quality Lab, CCG ZGDV Institute
ALGORITMI Centre/LASI
University of Minho
Guimarães, Portugal
pedro.pereira@ccg.pt

Carlos Gonçalves
EPMQ - IT Engineering Maturity and Quality Lab, CCG ZGDV Institute
University of Minho
Guimarães, Portugal
carlos.goncalves@ccg.pt

Lara Lopes Nunes
NOS SGPS, S.A
Lisboa, Portugal
lara.nunes@parceiros.nos.pt

Paulo Cortez
ALGORITMI Centre/LASI
Dep. Information Systems
University of Minho
Guimarães, Portugal
pcortez@dsi.uminho.pt

André Pilastri
EPMQ - IT Engineering Maturity and Quality Lab, CCG ZGDV Institute
Guimarães, Portugal
andre.pilastri@ccg.pt

ABSTRACT

Nowadays, the general interest in Machine Learning (ML) based solutions is increasing. However, to develop and deploy a ML solution often requires experience and it involves developing large code scripts. In this paper, we propose AI4CITY, an automated technological platform that aims to reduce the complexity of designing ML solutions, with a particular focus on Smart Cities applications. We compare our solution with popular Automated ML (AutoML) tools (e.g., H2O, AutoGluon) and the results achieved by AI4CITY were quite interesting and competitive.

CCS CONCEPTS

• **Computing methodologies** → **Learning settings**; **Machine learning algorithms**;

KEYWORDS

Automated Machine Learning, Smart Cities, Supervised Learning

ACM Reference Format:

Pedro José Pereira, Carlos Gonçalves, Lara Lopes Nunes, Paulo Cortez, and André Pilastri. 2023. AI4CITY - An Automated Machine Learning Platform for Smart Cities. In *The 38th ACM/SIGAPP Symposium on Applied Computing (SAC '23), March 27 - March 31, 2023, Tallinn, Estonia.* ACM, New York, NY, USA, Article 4, 4 pages. https://doi.org/10.1145/3555776.3578740

1 INTRODUCTION

Due to advances in Information Technology (IT), nowadays it is more easy to collect, store and process data that reflects multiple aspects of our daily lives, giving rise to the concept of Smart Cities[4]. All this data hold potentially valuable knowledge that can be extracted to better support decision-making. Thus, there has been a growing need to rely on data-driven systems, such as Artificial Intelligence (AI) and Machine Learning (ML) tools to make sense of the constant inputted data. As ML becomes so vast, in a way that is hard to keep up with, there is a shortage of experts that can effectively take advantage of state-of-art ML solutions. Thus, there has been a growing focus on the development of Automated ML (AutoML) solutions, which automate the search for the best ML algorithm and its hyperparameter setup [2, 9].

The paper proposes the AI4CITY technological platform, consisting of an AutoML tool that facilitates the application of ML algorithms to solve smart cities tasks. The goal is to reduce the complexity of the ML design code for smart cities applications, allowing both non-expert users and expert users to benefit from the whole ML workflow by using just a few lines of code. Our platform works with supervised learning tasks (classification, regression, and time series forecasting). In particular, it assumes a step-by-step approach architecture that includes the typical ML workflow, such as task detection, data preprocessing stage (e.g., missing data imputation), model and hyperparameter selection and pipeline deployment.

2 PLATFORM ARCHITECTURE

This work is inserted in a large R&D project, termed CityCatalyst, related to the Smart Cities context, which involves multiple large Portuguese companies from different domains (e.g., Telecommunications, Electric Power) and different technological subjects (e.g., AI, Data Warehousing, Interoperability). Considering the lack of ML specialists working in some of these companies and the diversity of domains involved, the main outcome of this R&D project, in terms of AI, is an agnostic automated ML solution that can be used by both experts and non-ML-experts with just a few lines of code.

In this paper, we propose AI4CITY, an automated smart city solution for predictive analytics, specifically targeting supervised learning tasks with tabular data. The solution builds default full ML pipelines that include most of the usual ML workflow steps and only

SAC '23, March 27 - March 31, 2023, Tallinn, Estonia

ACM ISBN 978-1-4503-9517-5/23/03.
https://doi.org/10.1145/3555776.3578740

require two input parameters: data and the target column. Then, the solution automatically detects the ML task, the necessary data preprocessing steps and then fits an AutoML algorithm. Moreover, if needed, the AI4CITY platform also allows ML-experts to decide which steps should be included in the ML pipeline, as well as the algorithm to be used in each step. In both cases, the AI4CITY output is an ML pipeline ready to be used in new and not-yet-transformed data, applying all the defined steps and returning predictions.

The proposed solution was written using Python programming language and returns a pipeline that relies on imblearn pipeline logic, which is similar to the scikit-learn's, with the particularity of applying some of the steps only to the training data (e.g., balancing techniques). In fact, the proposed solution works as a wrapper of an extensive set of ML techniques from several other Python modules (e.g., scikit-learn, H2O, CANE). The main advantages of the AI4CITY solution can be expressed in terms of: **usability**, since it requires considerably less coding when compared to other tools, improving the usage of for non-ML-expert users; **flexibility**, since all pipeline steps are highly configurable, useful for advanced ML users; and **automation**, considering that it assumes a predefined set of steps and only requires data and a target output to be provided. Figure 1 summarizes the overall AI4CITY architecture.

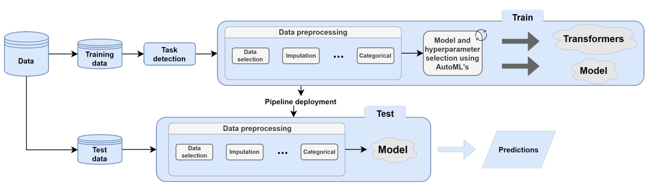

Figure 1: The AI4CITY platform architecture.

2.1 Pipeline Steps

ML task detection: the solution detects automatically the supervised ML task, according to the target column data type. If needed, the user can also explicitly define the ML task. The possible choices are: binary and multi-class classification, regression and TSF.

Data selection: this pipeline step allows expert users to filter data records based on a query. By default, this step is not adopted.

Imputation: the purpose is to replace missing values in the dataset and it uses the scikit-learn Simple Imputer to do so. By default, whenever missing data is detected in the training data, this step is added to the pipeline, using mean values for the numerical attributes and most frequent values for the categorical ones.

Categorical Transformation: we provide 4 implementations: scikit-learn ordinal encoding and CANE one-hot encoding, Inverse Document Frequency (IDF) and Percentage Categorical Pruning (PCP) [5]. By default, AI4CITY applies one-hot encoding for datasets with a maximum cardinality of 50 levels. Otherwise, based on recent studies[6–8], IDF is applied. Nevertheless, experts can explicitly select any of the mentioned techniques.

Numerical Transformation: for experts, we provide 6 scikit-learn numerical transformation options to experts: min-max, standard, maximum absolute, robust, quantile and power. By default, standard scaling is applied to all numerical features.

Data Balancing: in a recent study [10], we have compared several data balancing techniques, namely Gaussian Copula (GC), Synthetic Minority Over-sampling Technique (SMOTE), Random Undersampling and Tomek Links. The results achieved presented advantages in using GC when compared with the remaining options. Thus, GC is used in terms of AI4CITY default option for unbalanced classification data. Yet, any of the mentioned methods can be used.

Feature Selection: AI4CITY wraps all of the scikit-learn implementations and, if needed, the user can explicitly select the relevant input features. However, this step is not yet automated.

Modeling: for expert users, in this step, AI4CITY provides a large set of options. In particular, from scikit-learn we have wrapped 22 classifiers and 29 regressors. Furthermore, for both classification and regression tasks, we have also wrapped 4 AutoML tools (H2O, AutoGluon, AutoKeras and FEDOT) and 3 popular ensemble alternatives (XGBoost, CatBoost and LightGBM). Concerning TSF tasks, we provide 2 AutoML options: FEDOT and AutoTS. In terms of default choices (for non experts), based on[2, 9], we chose the AutoGluon for both regression and multi-class classification tasks, H2O for the binary classification ones and FEDOT for TSF tasks.

Evaluation: although this is not a step of the pipeline, the proposed solution allows to easily compute several measures related to the predictive performance. These include 15 known scikit-learn's metrics (e.g., Mean Absolute Error (MAE), Area Under the Curve (AUC) of the ROC analysis) and a complementary metric that is useful for regression and TSF tasks [2, 9]: the Normalized Mean Absolute Error (NMAE). Furthermore, the AI4CITY solution implements a set of 14 scikit-learn cross-validation procedures and two realistic procedures: Rolling Window and Growing Window.

Pipeline deployment: the AI4CITY solution allows to import and export of all steps of the pipeline (e.g., transformers, models) as a whole, in the joblib format. This way, when users need to perform predictions, they only need to load the pipeline object and feed it with new unseen data. Then, our solution ensures that the new data goes through the same data processing executed for the training.

2.2 Usability

In addition to the ML workflow automation provided by our platform, its ease of use by both expert and non-expert users is a major advantage. With a few lines of code, non-ML-experts can easily implement a whole pipeline with multiple data preprocessing steps that intend to improve the model's predictive performance. In fact, most of the choices performed internally in an automatic way are based on recent studies that proved the value of using each of these steps in the ML workflow. On the other hand, since there is not a single solution that can solve all the problems, we propose a flexible solution that allows expert ML users to select each of the steps and techniques they intend to use for their problem. As far as we know, this is the first work providing high flexibility, automation, variety of techniques and ease of use in a single solution for the creation of predictive ML pipelines within the context of Smart Cities. Furthermore, for comparison purposes, we performed exactly the same ML workflow in two different ways: writing the required code from the used libraries versus the code required by our solution. With AI4CITY, we could verify a considerable decrease in the required code, from 15 lines to 5 lines.

3 MATERIALS AND METHODS

3.1 AutoML Tools

Section 2.1 presents all the steps and methods provided by AI4CITY, including the default pipeline choices for non-expert users. However, using predefined pipeline steps based only on data and without any domain context, can lead to wrong assumptions and, consequentially, to low quality predictive performance. Therefore, in this work we perform a comparison between two pipeline usage modes of our platform: **auto** and **none**. The former mode automatically builds the pipeline with the preprocessing steps mentioned in section 2.1, while the latter inputs the data directly to the AutoML tool, executing the data preprocessing step only if directly assumed by the AutoML tool. The purpose of this comparison is to assess the influence of our data preprocessing steps in the ML workflow. Based on AutoML benchmarking studies [2, 8], the chosen AutoML tools are H2O, AutoGluon and FEDOT.

3.2 Data

The datasets used in this work for testing the proposed solution were divided into three major predictive ML tasks: regression, binary and multi-class classification. Although the AI4CITY platform is able to deal with TSF tasks, it implements the FEDOT tool without any additional preprocessing steps. Thus, no time series data was selected for experimentation in this paper. The selection of public datasets was based on recent AutoML tools benchmark studies [1–3]. Excepting the counts datasets, which are private and were provided by a company related to CityCatalyst project, all datasets are publicly available in OpenML, Kaggle and UCI platforms.

Table 1 summarizes the characteristics the adopted datasets in terms of: ML task (bin. – binary classification, multi. – multi-class classification and reg. – regression); dataset name; Rows (total number of instances); Num. cols (number of numerical attributes); Cat. cols (number of categorical attributes); #Nulls (sum of null values from all columns); #Levels (sum of different levels/categories from all columns); %Min. class (percentage of instances from the minority class - only for classification tasks); and Classes (number of different levels/values from the target data).

Table 1: Datasets Characteristics.

Task	Dataset	Rows	Num. cols	Cat. cols	#Nulls	#Levels	%Min class	Classes
bin.	PMAI4I	10K	5	1	0	3	3.39	2
	creditcard	285K	30	0	0	0	0.17	2
	machine17	9K	4	0	0	0	0.17	2
	machine22	9K	4	0	0	0	0.17	2
	machine83	9K	4	0	0	0	0.16	2
	machine98	9K	4	0	0	0	0.18	2
	machine99	9K	4	0	0	0	0.22	2
	churn	5K	20	0	0	0	14.14	2
	diabetes	0.8K	8	0	0	0	34.9	2
	hotel	119K	19	11	13K	228	37.04	2
	road_safety	100K	4	1	0	99K	15.16	2
multi.	machines	969K	5	0	0	0	0.01	5
	cmc	1K	9	0	0	0	22.61	3
	dmft	0.8K	2	2	0	5	15.43	6
	mfeat	2K	6	0	0	0	10.00	10
reg.	automobile	0.2K	10	15	0	234	-	186
	cholesterol	0.3K	13	0	6	0	-	152
	cloud	0.1K	4	2	0	6	-	94
	disorders	0.3K	5	0	0	0	-	16
	life	3K	19	2	2K	185	-	362
	counts(00-06)	5K	6	0	0	0	-	1544
	counts(06-12)	5K	6	0	0	0	-	3868
	counts(12-14)	5K	6	0	0	0	-	2570
	counts(14-19)	5K	6	0	0	0	-	4361
	counts(19-00)	5K	6	0	0	0	-	3266
	counts(day)	5K	6	0	0	0	-	6841

3.3 Evaluation

For evaluation purposes and in order to present a more robust set of results, a five-fold cross-validation approach was assumed, where in each of the five runs, 80% of data is used in training and 20% for the predicting phase (test data). We note that the preprocessing techniques were only applied in the training phase, with the predicting phase assuming the previous data preprocessing. For classification tasks, a stratified 5-fold cross-validation was employed. In order to measure the performance of each task, we adopt the same predictive performance measures assumed in a recent benchmark study [2]: the AUC for binary classification, the macro F-score for multi-class classification problems and the NMAE for regression tasks.

4 RESULTS

In this section, we compare the two mode usages presented in Section 3.1: **auto** and **none**. Tables 2 and 3 present median values from the 5 runs for each mode regarding all predictive measures (in percentage) and computational effort in terms of training (**Train Time**) and inference (**Prediction Time**).

Table 2 summarizes the results for the classification tasks. In terms of binary classification, the "auto" and "none" execution modes resulted each in five best results and one tie. Several of the obtained predictive results are of quality. Concerning computational effort, similar values were obtained by both modes, except for the road_safety dataset. Regarding the multi-class classification task, overall, the "auto" mode produces two best results, while the "none" mode excels for the other two datasets. The averaged F-score values are similar, except for the machines dataset, where the "auto" mode produces a substantial improvement. As for computational effort, both modes tend to produce similar training and inference results.

Finally, Table 3 shows the regression results. For this type of ML task, the "auto" mode produces 7 best results, while the "none" mode obtains 4 best NMAE values. We particularly highlight the counts dataset, which relates to a real-world smart city task. For the six counts datasets, the "auto" mode consistently provided the best results, which are of quality (e.g., NMAE lower than 1%). As for the computational effort, both modes tend to require similar training and inference times.

5 CONCLUSIONS

In this paper, we present AI4CITY, a novel integrated AutoML platform that is capable of handling predictive tasks related with Smart Cities. Working as a Python wrapper that makes use of several ML libraries and frameworks, AI4CITY works as a single module that is easy to install and use. To assess the utility of its

Table 2: Results for classification tasks (best values in bold).

Measure	Dataset	Usage	Value	Train Time (s)	Prediction Time (ms)
AUC	PMAI4I	auto	93.72	313.51	0.86
		none	**97.46**	309.77	0.31
	churn	auto	**92.34**	310.80	0.78
		none	91.58	310.98	0.69
	creditcard	auto	**99.00**	353.04	0.15
		none	98.54	335.31	0.12
	diabetes	auto	**84.00**	308.05	3.40
		none	82.34	309.66	3.20
	hotel	auto	**100**	320.04	0.11
		none	**100**	313.88	0.08
	machine17	auto	**67.64**	311.27	0.58
		none	66.77	311.50	0.31
	machine22	auto	72.35	311.23	0.57
		none	**76.45**	319.92	0.31
	machine83	auto	67.20	310.67	0.61
		none	**72.32**	311.30	0.32
	machine98	auto	58.69	311.97	0.55
		none	**64.54**	310.90	0.30
	machine99	auto	**71.22**	310.98	0.57
		none	65.27	311.46	0.37
	road_safety	auto	98.72	825.81	0.09
		none	**98.75**	310.91	0.08
	Median	auto	84.13	311.86	0.57
		none	**85.47**	311.42	0.31
F1	machines	auto	**30.09**	315.87	0.08
		none	19.99	292.77	0.01
	cmc	auto	51.55	31.34	0.63
		none	**53.35**	28.58	0.26
	dmft	auto	19.06	25.86	0.26
		none	**19.62**	26.25	1.13
	mfeat	auto	**71.96**	59.47	0.54
		none	71.77	57.76	0.51
	Median	auto	**41.73**	45.50	0.23
		none	37.09	44.20	0.21

Table 3: Results for regression tasks (best values in bold).

Dataset	Usage	NMAE	Train Time (s)	Prediction Time (ms)
automobile	auto	5.08	14.05	3.16
	none	**4.85**	21.63	3.85
cholesterol	auto	15.59	9.37	0.80
	none	**15.35**	9.26	0.58
cloud	auto	**7.24**	10.20	3.45
	none	7.59	10.03	5.48
disorders	auto	14.43	8.77	1.62
	none	**14.29**	8.62	0.94
counts(00-06)	auto	**0.56**	308.58	0.14
	none	1.71	305.14	0.03
counts(06-12)	auto	**0.51**	308.16	0.12
	none	1.86	305.44	0.03
counts(12-14)	auto	**0.49**	307.99	0.08
	none	1.95	304.75	0.02
counts(14-19)	auto	**0.46**	308.25	0.11
	none	1.96	305.44	0.03
counts(19-00)	auto	**0.50**	308.00	0.08
	none	1.90	305.34	0.03
counts(day)	auto	**0.61**	309.57	0.09
	none	2.11	305.92	0.03
life	auto	2.15	194.32	0.40
	none	**2.01**	216.35	0.31
Median	auto	**0.61**	307.69	0.14
	none	2.05	304.61	0.04

automation, we performed a substantial computational experiment, using a total of 26 distinct datasets. Overall, interesting results were provided by the AI4CITY "auto" default mode, particularly for multi-class classification and regression tasks.

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 was carried out within the project "City Catalyst - Catalisador para Cidades Sustentáveis" reference POCI/LISBOA-01-0247-FEDER-046119, co-funded by Fundo Europeu de Desenvolvimento Regional (FEDER), through Portugal 2020 (P2020).

REFERENCES

[1] Luís Ferreira, André Luiz Pilastri, Carlos Martins, Pedro Santos, and Paulo Cortez. 2020. An Automated and Distributed Machine Learning Framework for Telecommunications Risk Management. In *ICAART (2)*. SCITEPRESS, 99–107.

[2] Luís Ferreira, André Luiz Pilastri, Carlos Manuel Martins, Pedro Miguel Pires, and Paulo Cortez. 2021. A Comparison of AutoML Tools for Machine Learning, Deep Learning and XGBoost. In *IJCNN*. IEEE, 1–8.

[3] Gonçalo Fontes, Luís Miguel Matos, Arthur Matta, André Luiz Pilastri, and Paulo Cortez. 2022. An Empirical Study on Anomaly Detection Algorithms for Extremely Imbalanced Datasets. In *AIAI (1) (IFIP Advances in Information and Communication Technology)*, Vol. 646. Springer, 85–95.

[4] Luminita Hurbean, Doina Danaiata, Florin Militaru, Andrei-Mihail Dodea, and Ana-Maria Negovan. 2021. Open Data Based Machine Learning Applications in Smart Cities: A Systematic Literature Review. *Electronics* 10, 23 (2021). https://doi.org/10.3390/electronics10232997

[5] Luís Miguel Matos, João Azevedo, Arthur Matta, André Luiz Pilastri, Paulo Cortez, and Rui Mendes. 2022. Categorical Attribute traNsformation Environment (CANE): A python module for categorical to numeric data preprocessing. *Softw. Impacts* 13 (2022), 100359.

[6] Luís Miguel Matos, Paulo Cortez, Rui Mendes, and Antoine Moreau. 2019. Using Deep Learning for Mobile Marketing User Conversion Prediction. In *IJCNN*. IEEE, 1–8.

[7] Nikolay O. Nikitin, Pavel Vychuzhanin, Mikhail Sarafanov, Iana S. Polonskaia, Ilia Revin, Irina V. Barabanova, Gleb Maximov, Anna V. Kalyuzhnaya, and Alexander Boukhanovsky. 2022. Automated evolutionary approach for the design of composite machine learning pipelines. *Future Gener. Comput. Syst.* 127 (2022), 109–125.

[8] Pedro José Pereira, Paulo Cortez, and Rui Mendes. 2021. Multi-objective Grammatical Evolution of Decision Trees for Mobile Marketing user conversion prediction. *Expert Syst. Appl.* 168 (2021), 114287.

[9] Pedro José Pereira, Nuno Costa, Margarida Barros, Paulo Cortez, Dalila Durães, António Silva, and José Machado. 2022. A Comparison of Automated Time Series Forecasting Tools for Smart Cities. In *Progress in Artificial Intelligence - 21st EPIA Conference on Artificial Intelligence, EPIA 2022, Lisbon, Portugal, August 31 - September 2, 2022, Proceedings (Lecture Notes in Computer Science)*, Vol. 13566. Springer, 551–562.

[10] Pedro José Pereira, Adriana Pereira, Paulo Cortez, and André Luiz Pilastri. 2021. A Comparison of Machine Learning Methods for Extremely Unbalanced Industrial Quality Data. In *EPIA (Lecture Notes in Computer Science)*, Vol. 12981. Springer, 561–572.