Available online at www.ijournalse.org

Emerging Science Journal

(ISSN: 2610-9182)

Vol. 6, No. 6, December, 2022

The Benefits of Automated Machine Learning in Hospitality: A Step-By-Step Guide and AutoML Tool

Mauro Castelli [1*], Diego Costa Pinto [1], Saleh Shuqair [1], Davide Montali [1], Leonardo Vanneschi [1]

[1] *NOVA Information Management School (NOVA IMS), Universidade Nova de Lisboa, Campus de Campolide, 1070-312 Lisboa, Portugal.*

Abstract

The manuscript presents a tool to estimate and predict data accuracy in hospitality by means of automated machine learning (AutoML). It uses a tree-based pipeline optimization tool (TPOT) as a methodological framework. The TPOT is an AutoML framework based on genetic programming, and it is particularly useful to generate classification models, for regression analysis, and to determine the most accurate algorithms and hyperparameters in hospitality. To demonstrate the presented tool's real usefulness, we show that the TPOT findings provide further improvement, using a real-world dataset to convert key hospitality variables (customer satisfaction, loyalty) to revenue, with up to 93% prediction accuracy on unseen data.

Keywords:

Artificial Intelligence;
Automated Machine Learning;
Behavioral Research; Hospitality.

Article History:

Received:	31	May	2022
Revised:	06	August	2022
Accepted:	17	August	2022
Available online:	13	September	2022

1- Introduction

Machines increasingly perform "intelligent" tasks. For marketers, artificial intelligence (AI) provides algorithms that support decision-making and guide sales teams to more profitable leads by predicting a customer's lifetime value [1]. Indeed, machine learning (ML) tools have gone beyond commercial enterprises and addressed the needs of social science research in general [2, 3] and hospitality research specifically [4–6]. ML can use standard estimation methods, including test theories [7, 8], or make predictions, enabling researchers to answer questions previously beyond the reach of science [9]. Advancements in data analytics span many domains, including business [9], psychology [10], and economics studies [8]. Researchers today rely on big data methods [11, 12], text analysis [5], and single-paper meta-analysis [13, 14] to increase the generalizability of findings from traditional methods (e.g., surveys, lab experiments). Such practices allow researchers to investigate further the differences among studies by quantifying the impact of study-level covariates or the degree of between-study variation. Indeed, business studies have concerned causal mechanisms that better explain consumer behavior [5, 13]. However, hospitality practice is ahead of academic research when it comes to applications of AI and ML, and there is a need to reduce the gap between scholarship and marketing practice [15]. AI is quite beneficial for hospitality providers [16, 17], and prior work has undoubtedly expanded our understanding of how AI can help hospitality providers solve key problems, such as predicting hotel cancellations [17] and identifying factors that contribute to customer value in the restaurant context [18], analyzing online reviews [16], and marketing segmentation [19]. It is still unclear what tool, if any, would help academic researchers and practitioners use AI to predict performance or consumer behavior (e.g., loyalty or satisfaction). In response to this gap in the literature, the current study presents a

[*] **CONTACT**: mcastelli@novaims.unl.pt

DOI: http://dx.doi.org/10.28991/ESJ-2022-06-06-02

descriptive application of automated machine learning (AutoML), with the aim of enhancing scholars' and practitioners' ability to predict future consumer behaviors accurately. Additionally, AutoML provides the ability to explain behavior (i.e., to accurately describe causal underlying processes) and predict consumer behavior (i.e., to forecast behaviors accurately that have not yet been observed). Therefore, it would be beneficial for marketers to broaden their horizons by adopting AutoML tools and algorithms.

This study makes a number of contributions. We build on prior research on ML in hospitality (i.e., [17, 18]). In particular, we propose the use of an AutoML tool that can predict consumers' responses to providers or to predict important outcomes, such as revenue. Notably, this tool is beneficial for managers' "practitioners" and academic research, as it enables hospitality providers to test, examine, and predict important outcomes (e.g., loyalty).

The study also provides a very important contribution in the technical area of auto ML. The particular Auto ML system that we present, the tree-based pipeline optimization tool (TPOT) [20], is quite new: it was introduced in 2016, and it became widely known in the community only 2 or 3 years ago. The application areas in which the TPOT has been applied are rather few and of a completely different nature (mainly medical data). Although researchers have reported promising results, new validations on more, different, and possibly more challenging applications are needed. Applying the TPOT for the first time in hospitality and obtaining predictive models of high accuracy is impactful. The datasets we use are of vast size, and the relationship between dependent and independent variables has no visible or expected shape. These conditions represent a new and interesting challenge for the TPOT, which we expect to be appropriate in the presence of large amounts of data whose mutual relationships have no predefined shape.

2- Literature Review

AutoML is an AI tool tailored to consumer behavior research and is well-established as an efficient computer science research tool due to its relative effectiveness compared to computational effort [2]. However, few studies have relied on AutoML to estimate consumer behavior outcomes. The work of Ferreira [4] is an exception and uses unsupervised ML algorithms, such as self-organizing maps and agglomerative hierarchical clustering, to construct descriptive models capable of finding clusters of over-indebted consumers. AutoML enabled the authors to test and compare 32,730 predictive models to produce an algorithm that predicted an external dataset's indebtedness with up to 89.5% accuracy. AutoML offers hospitality scholars and practitioners several benefits. It helps gauge out-of-sample accuracy to provide a hedge against overfitting. It also provides high levels of predictive accuracy [4], building simplified models that provide viable alternatives to logit models [21]. Although ML research in general has grown recently in business research and practice (for a detailed review, please see [4]), researchers have rarely explored these benefits of AutoML [22].

We aim to illustrate a step-by-step approach to using AutoML in consumer and market prediction studies; the illustration offers insights into how to convert loyalty and customer satisfaction into revenue. AutoML requires only a few steps and limited programming knowledge from researchers. The tool we discuss can be widely applied with only basic familiarity with ML algorithms. Researchers can download conveniently referenced packages in R or Python to implement and compare multiple ML algorithms including decision trees, random forests, artificial neural networks, LASSO (least absolute shrinkage and selection operator), and regression models. However, although we acknowledge that R and Python are widely applied in data science, this manuscript focuses on how to produce results through Python for two reasons: (a) Python provides a more general approach to data science, and it (b) provides a TPOT library for AutoML, based on genetic programming (GP) (e.g., [23]). GP [24, 25] is the most advanced technique in the area of evolutionary computation and since its inception, it has successfully addressed optimization tasks in various domains. The use of Python as the programming language allows for the integration of the AutoML tool with other packages that are commonly used in the ML field (i.e., NumPy, Pandas, Matplotlib, Scikit-learn, and Tensorflow). Therefore, the combination of an advanced evolutionary-based optimization process and the availability of a complete package ecosystem make the tool a suitable option in the AutoML context.

2-1- From ML to AutoML

Four basic steps characterize a standard ML workflow: (1) ingestion, (2) cleaning, (3) feature engineering, and (4) modeling. Ingestion consists of collecting data to be used in the subsequent phase. The ingestion phase can rely on a specific platform (i.e., collecting data related to a Twitter user) or specific devices (such as sensors or video cameras), or it can be performed in other simple ways with data stored in a spreadsheet. Ingestion does not benefit from the use of AutoML and therefore it is not a focus of the paper. All the TPOT tool needs is to presume that a user was able to collect data for a particular problem with the potential to be addressed by means of an ML algorithm.

2-1-1- Data Cleaning

Cleaning is a fundamental task in ML that may significantly impact model performance. In particular, the cleaning phase aims at removing or modifying data (collected in the ingestion phase) that is incorrect, irrelevant, or duplicated. In other words, the cleaning phase prepares the data for subsequent steps, ensuring that ML models are built only with high-quality data. Despite this step's apparent fundamental role, ML users who fully rely on the ML algorithm often ignore cleaning. Not even the best ML algorithm for a given instance of a problem can produce a satisfactory performance if fed poor-quality data ("garbage in, garbage out"). The cleaning phase presents an important and challenging task. Several data cleaning techniques can be considered for various purposes and according to the available data [26]. However, these steps could be considered a standard cleaning procedure: (1) removing irrelevant data, (2) detecting and fixing structural errors, (3) outlier detection, and (4) handling missing values.

In the first step, two analyses should be performed: removal of duplicate observations and removal of irrelevant observations. The presence of duplicate observations is quite common because big data are typically ingested from multiple sources. Removing irrelevant observations reduces the dataset's size and mitigates the possibility of data irrelevant to the problem at hand. For instance, if the ML task aims at analyzing the behavior of a supermarket's female clients, all observations related to male clients must be screened out.

The second step (handling structural errors) aims at increasing data. For instance, a variable reporting country may contain "Portugal" or "PT." Routines need to search for and resolve such inconsistencies. Other factors may render a data point structurally irrelevant. For instance, a variable that is invariant across all observations provides no information. The further reduction in data volume further facilitates subsequent ML steps.

Third, outlier detection aims to identify abnormal data patterns. Data analysts have multiple tools at their disposal for outlier detection [27]. A review is beyond this study scope but is presented in [28]. Removing outliers is critical because most of the ML techniques are sensitive to the presence of outliers.

Fourth, several strategies can be used for handling missing values [29]. The nature of the data (e.g., text, numbers, point in a time series) becomes a strong consideration in how to proceed. Many approaches to data cleaning exist, so selecting an appropriate procedure can be difficult [30, 31]. Many data cleaning techniques can be easily automatized using the Scikit-learn ML tool [32].

2-1-2-Feature Engineering

Feature engineering aims to improve ML models' performance of ML models by creating new and possibly more informative variables, combining existing ones, or representing them in a more convenient manner. Feature engineering is a complex task that influences the resulting ML model's performance. Feature engineering most commonly uses binning, logarithmic transformation, one-hot encoding, grouping operations, feature split, and scaling. A description of the feature-engineering techniques is beyond this paper's scope, but we refer the reader to Kuhn and Johnson [29] for a complete review. What should be clear is that feature engineering is characterized by a large number of methods and that each method may have its own parameters. Feature engineering is more of an art than a science, and it is perhaps the most common source of mistakes in deploying the ML workflow.

2-1-3- Modeling

Modeling is the final task in the ML workflow. It involves the choice of an algorithm and its hyperparameter configuration. Modeling requires some experience in the ML area because no formal procedure is available for selecting the "best" algorithm to address a specific optimization problem. In fact, the no free lunch theorem states that any two ML algorithms are equivalent when their performance is averaged across all possible problems [33]. Additionally, algorithm selection is only as good as the optimization of its hyperparameters. That is, algorithms are characterized by the presence of specific parameters whose values are fundamental for achieving a robust model. For this reason, the modeling phase is a critical step that, due to the choice of the ML algorithm and its parameterization, is characterized by a high level of complexity.

Fortunately, ML practitioners can take advantage of the existing AutoML tools for building a successful ML pipeline without being a ML expert. The scientific literature reports several definitions of AutoML, which is a process designed "to reduce the demand for data scientists by enabling domain experts to build ML applications automatically without extensive knowledge of statistics and ML" [34]. In Yao et al. [35], AutoML is defined as the process that allows for the automated construction of an ML pipeline. Therefore, AutoML is particularly promising considering the hardware that is available today and the need to extract insights from a vast amount of data. Domain experts without extensive knowledge of ML can easily build a pipeline that takes care of all the steps previously discussed, from data cleaning to the definition of the final model.

2-2- Tree-Based Pipeline Optimization Tool (TPOT)

Among the AutoML packages, we adopted the TPOT to build an ML model able to achieve satisfactory performance in a complex task. The TPOT is an open-source library for performing AutoML in Python. It is based on GP [24, 25, 36] and provides a stochastic global search procedure to discover a top-performing model pipeline efficiently for a given dataset. In particular, one of the main advantages of the TPOT is its focus on the cleaning and feature engineering steps, which other tools often ignored, as they are only focused on the model's optimization. The choice of the TPOT in this study is related to the analysis presented in Balaji and Allen [37]. In the study, the authors benchmarked the most commonly used open-source AutoML tools, including auto-sklearn [2], TPOT, Auto ML (https://github.com/ClimbsRocks/auto_ml), and H2o [38]. In particular, they compared the resulting models' performance on regression and classification dataset and found that auto-sklearn and the TPOT are the best performers. Therefore, we selected the TPOT for its advanced optimization process and the possibility of specifying many parameters. Expert ML practitioners and non-experts can proficiently use the tool; non-experts may rely on the default parameters, whereas ML experts may set each parameter's value. In particular, the TPOT automates the most challenging part of the ML workflow by intelligently exploring thousands of possible pipelines to find the best one for the data at hand. Figure 1 (taken from http://epistasislab.github.io/tpot/) shows the parts of the ML workflow the TPOT automates.

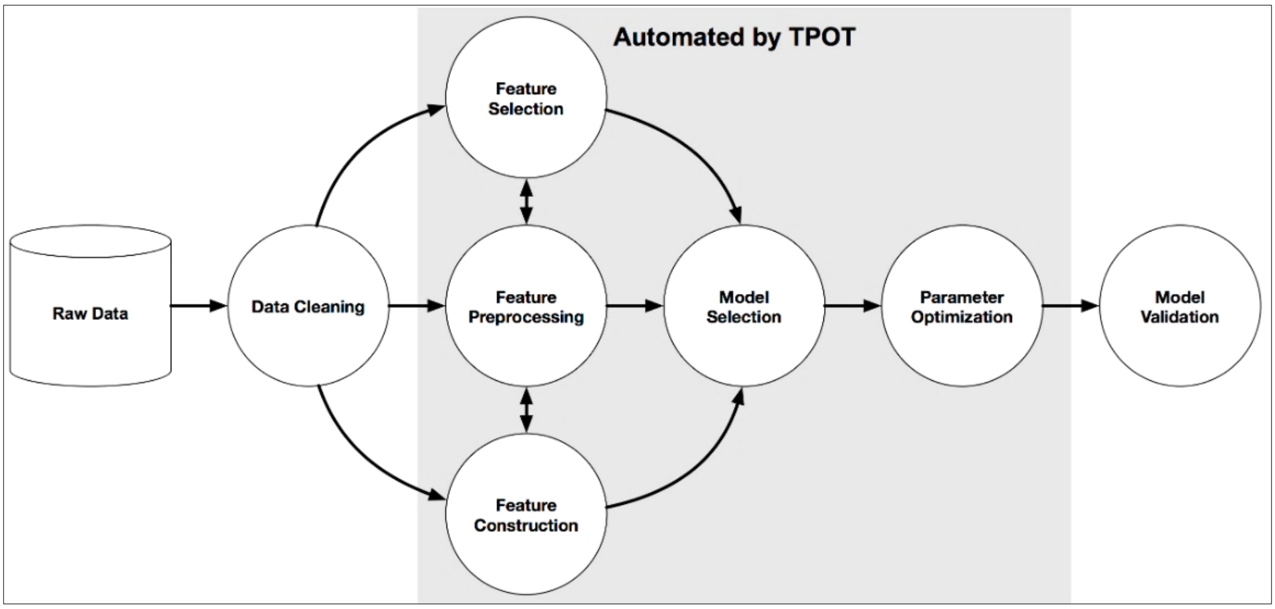


Figure 1. The ML pipeline automated by TPOT

2-3- Advantages of Automated ML

The previous section outlined the process that data scientists need to implement whenever a business identifies a problem that can be solved with ML. Hundreds of algorithms exist for every step of data cleaning, feature engineering, and modeling. Typically, each algorithm depends on a large set of parameters, and finding the correct setting for those parameters becomes crucial for an algorithm to perform accurately. The "No Free Lunch" theorem [33] guarantees that no formal process can exist to make an optimal choice among this huge variety of algorithms and configurations. Data scientists face an extremely difficult and time-consuming process in which they must consider many algorithms and configurations. Consequently, turnaround can be slow and often take too long to capitalize on a business opportunity. AutoML is a technique that will revolutionize ML-based solutions and the way they are obtained, enabling business analysts and developers to generate ML models that can address complex scenarios [8]. The AutoML platform will abstract the steps after data ingestion. Essentially, the AutoML tool allows users to upload data, identify the labels, and "push the play button" to generate a thoroughly trained and optimized model capable of accurate prediction. When dealing with AutoML, business analysts stay focused on the business problem instead of dwelling on the ML process and workflow. AutoML handles all the steps involved in preparing the data "behind the scenes", choosing the right algorithms and optimizing and tuning the hyperparameters.

For each phase of the ML pipeline, AutoML identifies a large set of (algorithm, hyperparameter configuration) pairs. It then combines all the existing pairs in all possible combinations in an exhaustive grid-search fashion. The pipeline configuration then returns the best results. Put simply, AutoML automatizes the process that data scientists perform manually to generate a predictive model. By doing so, AutoML can test many more possible combinations than any human would ever be able to consider manually. The big benefit of AutoML is the fact that it is easy to use and does not require advanced data science skills. The only drawback is a typically large learning curve, which makes the use of powerful computational resources particularly appropriate for AutoML. AutoML represents one of the most concrete and effective efforts toward the democratization of ML, de facto representing the future of AI. It puts the power of AI into the hands of business analysts and technology decision makers.

3- AutoML Tool Using TPOT

In this tutorial, we explore how to use TPOT to search for the optimal ML pipeline automatically and how to use the resulting model to predict target variables. TPOT is described as a "genetic search algorithm" and enables researchers to find the best parameters of model ensembles [39]. As previously mentioned, TPOT relies on GP, which is a method that, applying a process that mimics the Darwinian theory of evolution, allow us to "evolve" (and thereby improve in a stepwise refinement fashion) the ML configuration until the most suitable pipeline (i.e., for instance, the one that produces the most accurate model) is found. The process is repeated for a specified number of generations before settling on a final optimal pipeline that produces the highest possible accuracy and has low complexity (e.g., the smallest number of pipeline operators [39]). The TPOT allows behavioral science researchers to predict targets using AI. Plus, it is open source (free) and runs on Python, which makes it accessible for a wide audience. The tool allows researchers to perform sophisticated analyses without the need to download Python. Furthermore, the main advantage of using the proposed tool is that researchers do not need to write Python code, as the tool is already coded for data prediction. In this section, we illustrate the AutoML methodology, by applying it to a field dataset. The section demonstrates how to use AutoML methodology and interpret the estimates. First, the section describes the input data the AutoML methodology requires and demonstrates the additional benefits of using AutoML as an alternative and effective approach to data analysis. In our applications, the TPOT library will be programmed to minimize the error for the ML pipeline. The application is publicly available as a “Google Colab” Python notebook. Google Colab allows users to write and execute Python code through the browser and is especially well suited for ML, data analysis, and education (e.g., [40]). This setup is easy to use, is user-friendly and requires only a few steps as described below. In this tutorial, we will explore how to use the TPOT to tune ML models automatically to the considered data. In the first part of the tutorial, we develop classification models, and in the second part, we develop a model for regression.

3-1- Using AutoML to Predict Customer Loyalty, Satisfaction and Revenue

We will use AutoML to predict customer loyalty (classification tool), satisfaction (regression tool), and revenue (regression tool) (Table 1). We provide a step-by-step guide on how to use the AutoML tool in the Appendix I. With the field data, we aim to investigate consumers’ evaluations of a major European hotel chain, including key variables, such as overall evaluation and loyalty. The hotel chain has several branches operating across Europe and America and offers high-end hospitality services. The survey data represent all branches and consist of 284,229 samples from consumers who stayed in the hotel between 2009 and 2019. Consumers measured satisfaction with multiple attributes on a 5-point Likert scale (i.e., “staff friendliness”, "cleanliness”, “location”, “room quality”, “sports facilities”, “price”, “comfort”, “food quality, and “room amenities”. The attribute evaluation of the hotel experience shows good reliability (α=.80). The dataset also includes overall satisfaction in a single item (“Overall Satisfaction” 5 □, 4 □, 3 □, 2 □, 1 □, N/A).

Table 1. Survey measures

Variable	Operationalization	Actual Measures
Personal Information	Hotel System	Customer ID, loyalty card, hotel, booking number, room, stay dates, purpose of stay (business, tourism)
Service Experience Evaluation (α=.80)	Likert scale 1=strongly disagree 5=strongly agree	Staff friendliness, hotel cleanliness, hotel location, room quality, sports facilities, price, comfort, food quality 5 □, 4 □, 3 □, 2 □, 1 □, N/A
Satisfaction	Likert scale (1=strongly disagree, 5=strongly agree)	Satisfaction 5 □, 4 □, 3 □, 2 □, 1 □, N/A
Loyalty	Hotel System	Loyal guests who have booked again with the same hotel after the first experience (0 = no new reservations / hotel stays 1 = new reservations / hotel stays)
Estimated Revenue*	Hotel System	Estimated revenue was calculated by the median hotel price, based on season, room category, and the length of stay.

* For reviewers only. For confidentiality purposes, we cannot display actual revenue results, as they are part of our non-disclosure agreement (NDA) with the company.

Following the steps described in Section 2-1, the flowchart in Figure 2 summarizes the research methodology.

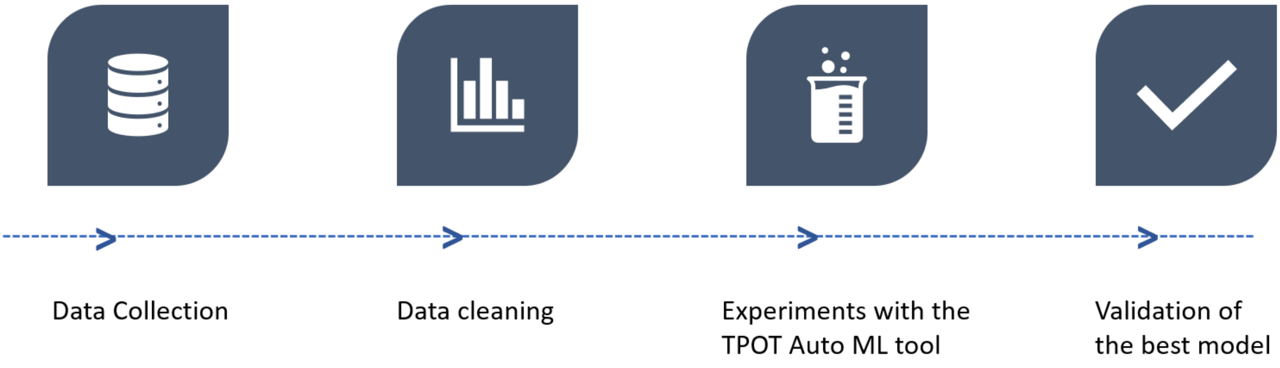

Data Collection
Data cleaning
Experiments with the TPOT Auto ML tool
Validation of the best model

Figure 2. Flowchart of the methodology

We started by retrieving the data from the surveys the considered European hotel chain provided. The dataset contains all the available features and the target variables (i.e., loyalty, customer satisfaction, and estimated revenue). Subsequently, in the data cleaning step, we analyzed the data to detect outliers and handle missing values. The data did not present outliers and missing values, so we maintained all the samples. In the experimental phase, we employed the TPOT AutoML framework to construct a machine-learning pipeline automatically. In particular, as described in Section 4, we used the TPOT to build a classification model and two regression models. Finally, we validated the best pipeline the TPOT returned, following a 3-fold cross-validation procedure. We discuss all the details concerning the experiments performed with the TPOT, the results achieved, and the respective validation in Section 4.

4- Interpreting the AutoML Tool Results

In this section, we present and analyze the ML models that the TPOT produced. In particular, a classification model for predicting customer loyalty (described in Section 4.1) and two regression models (Section 4.2), one for predicting customer satisfaction and the other for revenue prediction, were built. Given that this work represents the first effort to apply AutoML to this type of application, no comparison with previous results is possible. However, we aimed to obtain the generated models' greatest possible predictive accuracy.

4-1- AutoML Classification Tool Results

Here, we describe the ML model the AutoML tool produced as a result of the TPOT's evolutionary search process. The target variable represents the consumer's loyalty to the hotel. The TPOT allows us to export the best model directly as a Python script. The Python script includes the whole pipeline for the dataset it was trained on and all the hyperparameters for the model. Focusing on the analysis of the best model, the AutoML returned the following best ML pipeline:

ExtraTreesClassifier(input_matrix, bootstrap=False, criterion=entropy, max_features=0.3, min_samples_leaf=20, min_samples_split=16, n_estimators=100).

The analyses of the model reveal that the best modeling algorithm to predict customer loyalty is an ExtraTrees classifier, which is an ensemble of decision trees. A decision tree [41] is a tree-like structure composed of multiple nodes. Each of the tree's internal nodes contains a decision variable, and it has a number of branches that is equal to the number of values the decision variable can assume. Leaf nodes (also called external nodes) contain the class labels. A decision tree works by splitting the decision variable space sequentially in a set of partitions and subpartitions to form homogenous classes in terms of target variables. Practically, decision trees have roots in familiar tools, such as chi-square automatic interaction detection (CHAID). Instead of relying on a single predictor's output, the ensemble of decision trees considers the predictions of multiple models (called "weak learners") to obtain a more robust and reliable prediction. The ensemble predicts the class label through a majority voting procedure: the class that the majority of the weak learners predicts is the ensemble model's prediction. The weak learners the ExtraTrees ensemble model uses (i.e., the decision trees) are built by considering various sub-samples of the training set, thus ensuring diversity among the weak learners and controlling overfitting. To build a decision tree, the following iterative process is applied starting at the root node: 1. Let "A" be the best decision attribute for the next node. 2. Use "A" as a decision attribute for the current node. 3. For each value of "A", create a new branch with a descendant node. 4. Sort training examples to the leaf node according to the branch's attribute value. 5. If all training examples are perfectly classified or there are not enough training examples for a further splitting, stop; else, iterate over new descendant nodes. To select the "best" decision attribute for the split, Gini impurity or information gain can be used [42]. Despite the selected splitting criterion, the main idea is to consider the attribute that allows for better separation of the samples belonging to different classes. The set of hyperparameters includes the splitting criterion, the tree's maximum depth, the minimum number of samples required to be at a leaf, and the minimum number of samples required to split an internal node. For full reference, we refer the reader to the Scikit-learn documentation [32]. Once the decision tree is built, a new sample can be classified following the path from the root of the tree to a leaf node and assigning the new sample the corresponding label.

To classify customer loyalty, the model obtained consists of 100 decision trees (i.e., estimators). The model's parameters included at least 20 samples per leaf and 16 samples minimum per split. The number of features to consider when looking for the best split (max_features parameter) includes 30% of the attributes. Entropy was the splitting criterion (i.e., each split maximizes the information gain). The model yielded an accuracy of 0.82 and an F1 score of 0.78. The F1 score (a combined metric of precision and recall) indicates that the precision and recall values are satisfactory. In particular, the F1 score suggests that the model is precise and robust in predicting loyalty, as it correctly classified 82% of the samples.

4-2- AutoML Regression Tool Results

In this section, we describe the best pipeline the TPOT produced for the problem of predicting customer satisfaction. In this case, the considered task is a regression problem. The pipeline's fitness (i.e., objective function) is defined by its complexity (i.e., number of steps) and a metric that quantifies the difference between the predicted and actual satisfaction (in this case, given that more than one criterion are optimized at a time, we say that GP is working as a multi-objective optimization framework). We applied the TPOT to the data for 10 generations (with a population size of 10) to find the best-fitted ML pipeline. We subsequently evaluated the optimal pipeline the TPOT suggested on a validation dataset, using 3-fold cross-validation. Using the AutoML tool on the data considered, the best pipeline returned is the following:

```
ExtraTreesRegressor(LinearSVR(GradientBoostingRegressor(input_matrix, alpha=0.85,
learning_rate=0.1, loss=quantile, max_depth=2, max_features=0.3,
min_samples_leaf=1, min_samples_split=14, n_estimators=100, subsample=0.8),
C=20.0, epsilon=0.001, loss=squared_epsilon_insensitive, tol=0.01),
bootstrap=False, max_features=0.4, min_samples_leaf=8, min_samples_split=14,
n_estimators=100).
```

In this case, the pipeline consists of an ExtraTree regressor, a linear SVR, and a gradient boosting regressor (GB) [43]. The first ML model applied to the training set is the GB, which is a particular type of ensemble model. The working principle is simple yet effective: the idea is to combine multiple models' predictions so that the best possible subsequent model, when combined with the previously defined models, minimizes the prediction error. To implement this idea, the GB builds an additive model in a forward stage-wise manner, allowing for the optimization of arbitrary differentiable loss functions [32]. In particular, in each iteration, the GB fits a regression tree on the negative gradient of the given loss function. Therefore, at each iteration, the newly created model aims to minimize the error resulting from the application of the previously created models. The construction of the regression trees follows the same procedure explained in Section 4.1 for the construction of a decision tree. However, in this case, the sequence of splits is stopped when a further sub-partition is believed to decrease the mean square error of the target variables non-significantly. The leaf nodes contain the decision rules on which the target variable predictions are based. The GB model uses 100 predictors, and the specified loss (quantile) refers to quantile regression (i.e., an extension of linear regression used when the conditions of linear regression are not met), with alpha=0.85. The learning rate shrinks each tree's contribution by 0.01 (i.e., this value is related to the choice of using 100 estimators), the maximum depth for each estimator (i.e., regression tree) is two, and the fraction of samples to be used to fit the regression trees is 0.8. For a complete description of the parameters, we refer the reader to the Scikit-learn documentation [32].

The subsequent step in the pipeline is to provide the GB's output to the linear SVR, a linear support vector regression [44]. An SVR maps the data into a k-dimensional feature space through a nonlinear mapping. The idea is to allow a linear regression model to be fitted to the data points in the k-dimensional feature space. The obtained linear model is then used to make predictions in the new feature space. The kernel function defines the mapping from the input space into the new feature space [45]. One feature of SVR is related to the model errors: instead of minimizing the observed training error, SVR minimizes a combination of the training error and a regularization term and is meant to improve the model's generalization ability [46]. The pipeline considers an SVR with the following parameters: the regularization term (C) is equal to 20, the loss is the L2 loss (indicated as '*squared_epsilon_insensitive*') [47] with epsilon=0.001, and the tolerance (tol) for the stopping criteria is 0.01.

As a final step in the pipeline, an ExtraTree regressor is used to produce the final prediction. The ExtraTree takes as input the outputs of the SVR model and, as discussed in Section 4.1, builds an ensemble of randomized trees. In this case, different from Section 4.1, in which the ensemble outputs the predicted class label, the ensemble predicts a real value that corresponds to the predicted customer satisfaction. The ExtraTree regressor contains 100 estimators, eight samples minimum per leaf, and 14 samples minimum per split. The model yielded an R^2 of 0.50, revealing that only 50% of the data fits the regression model. Table 2 lists the values of the metrics referring to the best model. In this case, the model produced a poor performance due to the lack of predictive attributes in the considered dataset. In other words, the collected data are not sufficient to allow the model to produce an accurate prediction of customer satisfaction. This is evident when we consider the R^2 and the explained variance in the model.

Table 2. **Regression outputs (satisfaction)**

R^2:	0.50
Mean Squared Error:	0.55
Root Mean Squared Error:	0.74
Mean Absolute Error:	0.43
Explained Variance:	0.50
Max Error:	4.93
Median Absolute Error:	0.22

As a final experiment, to show that the AutoML tool can help predict revenue from loyal customers, we created a new variable in the dataset (estimated revenue). We took the median hotel price according to season, room category, and length of stay. As we did to predict customer satisfaction, we ran the TPOT for 10 generations (with a population size of 10) to find the best-fitted ML pipeline. We subsequently evaluated the optimal pipeline using 3-fold cross-validation. Using the AutoML tool on the considered dataset, the best pipeline that was returned is the following:

```
ExtraTreesRegressor(PCA(input_matrix, iterated_power=2, svd_solver=randomized),
bootstrap=True,  max_features=0.85,  min_samples_leaf=6,  min_samples_split=15,
n_estimators=100)
```

The pipeline performs an initial principal component analysis [48] to reduce the dataset's dimensionality and remove the less predictive attributes. In particular, PCA relies on the singular value decomposition of the data to project it to a lower-dimensional space. After the execution of PCA, the resulting dataset was provided as input to an ExtraTree regressor to produce the final prediction. The ExtraTree model consists of 100 estimators. In each estimator (i.e., regression tree), 85% of the attributes are considered when looking for the best split. The minimum number of samples required to split an internal node is 15, and the minimum number of samples required to be at a leaf node is six. The model yielded an R^2 of 0.93, suggesting that 93% of estimated revenue data fits the regression model. This result is a clear indication of the obtained model's suitability in providing an accurate prediction of the revenue. This result is further strengthened by the explained variance, which is 0.938. Therefore, we have clear evidence of the model's robustness and ability to address the problem at hand.

5- Discussion

The current practice of using AutoML in theory testing has significantly increased across various scientific fields, including financial economics [49], educational research [50], clinical research [51], and neuroscience [39]. To this end, the application of AutoML in business research is still in its infancy [4, 52]. Furthermore, only a few of the newly developed methods have been implemented in consumer research [52, 53]. Notably, recent technological advancements in automation and ML are essential for revolutionizing traditional consumer research and development in industry and academia, allowing researchers to generate ML models and solve complex scenarios [53]. We examine how loyalty and satisfaction can be predicted using AI tools. Overall, our findings extend prior studies on consumer loyalty in hospitality [54, 55] by showing how loyalty can be predicted. Therefore, the current study contributes to emerging studies on AI's impact on data prediction [56, 57]. However, unlike prior studies, our article provides a tool that can benefit academic research in solving complex problems. Furthermore, we contribute to recent hospitality research on the use of AI [18]. Prior hospitality research has mainly focused on the factors that predict hotel booking cancelations [58] or other factors, such as occupancy rates [59]. By introducing the AutoML tool, we make valuable and important contributions. AI tools have proven abilities to tackle complex data empirically [60], provide accurate estimations, and solve complex phenomena [4]. One of the difficulties in using AI is that a simple task is extremely difficult to program [8, 60], as it requires training and expertise in ML. Therefore, AutoML offers the hospitality field (in academia and industry) a simple method that enables researchers to take advantage of ML without being ML experts.

The proposed tool implements AutoML in one easy-to-use package. It is user friendly, widely accessible, practical, and tailored to provide more accurate estimations. We expect hospitality and behavioral researchers to start relying on novel applications and advanced tools in their future research. Ultimately, we believe that AutoML tools may increase the scope of hospitality research, not just by providing powerful data analysis but also by addressing complex research questions. AutoML can abstract all the steps after data ingestion. Essentially, users bring their data, identify the labels, and "push a button" to generate a thoroughly trained and optimized model ready to make an accurate prediction. When dealing with AutoML, business analysts often remain focused on the business problem instead of dwelling on the ML process and workflow. AutoML handles the "behind the scenes" steps involved in preparing the data, choosing the right algorithms, and optimizing and tuning the hyperparameters. For each ML phase, AutoML identifies a large set of (algorithm, hyperparameters, and configuration) tuples. It then analyzes the space of all the existing tuples in all possible combinations, searching for the best one. Technologically advanced methods, such as GP allow AutoML not to have to consider the huge space of all possible tuples but to navigate it intelligently. The outcome is the pipeline configuration that was able to return the best results. Put in simple terms, AutoML simply automates the process that data scientists perform manually when they have to generate a predictive model.

In sum, the study presents practical ways to analyze data using AutoML, based on simple steps, including a practical research tool. We believe that behavioral and social science researchers can conduct robust and powerful analyses without the need for programming. Therefore, we encourage researchers in the behavioral and social science fields to use AutoML for their benefit. Relying on AutoML would not only supplement the data analysis but also help synthesize the results and provide more accurate outcomes. Therefore, this tutorial can be used as a starting point for behavioral researchers who aim to benefit from the use of ML. Table 3 highlights the AutoML tool's key advantages and disadvantages.

Table 3. Key advantages and disadvantages of AutoML

AutoML Tool's Advantages	AutoML Tool's Disadvantages
No need to download or install Python or any dependencies locally. Researchers can easily share their notebooks with other researchers	More restricted to Python than relying on R modules
The TPOT uses genetic programming to explore the search space, and this can relieve the system from an exhaustive analysis of all the possible combinations.	Only for numerical data and restricted to quantitative research
No need to program to create a specific algorithm or coding.	It is only designed for data analysis purposes
Relies on specialized libraries	Analyzing larger generations will take a longer time
The TPOT can be employed to execute fast models and potentially increase the productivity of analyses without hampering prediction performance	It only performs regressions and builds classification models

6- Conclusion

This study contributes to the field by presenting an AutoML tool for hospitality. Given AI's prevalence in today's business environment, AutoML's main practical value is its ability to predict consumers' responses (e.g., satisfaction, loyalty) and forecast revenue from loyal customers. Indeed, the application of AutoML may enhance future firms' performance. One key issue that managers face in most industries is their insufficient understanding of AI and how to implement it in their daily practice [61, 62]. Furthermore, one of the complexities that faces managers is the difficulty that is associated with analyzing large and complex datasets [61]. Notably, this tool is beneficial for practitioners. For example, companies and service providers, including but not limited to hotels, restaurants, and retailers, can benefit from using this tool to test and predict their data (e.g., surveys, observations, etc.).

However, this study has some limitations that might inspire further research. The use of AI in businesses has been a challenge for many practitioners [63]. Many firms may not have the capacity or resources to build or implement AI models, and in most cases, they rely on data scientists or consulting firms to perform complex data analysis [63]. AutoML facilitates the application of AI in the area of hospitality by offering a free, open, and easy-to-use tool. Despite its importance, many hospitality providers might not be able to use this tool. Future research can be conducted on the implementation of AutoML tools in the hospitality domain.

The key results of this study are the 93% predicting power of estimated revenue data, which suggests that by using the AutoML tool, practitioners can determine in advance when their businesses will have more success and when that is not the case. This knowledge can help hospitality and business managers create effective seasonal offers to boost occupation in the periods in which they are expected to have lower demand. More importantly, we contribute to new knowledge in the domain, bridging the gap between advanced ML models and hospitality/business practice. By doing so, we contribute to recent research in the field (cite), providing practitioners with more technological development. Importantly, none of these prior studies in hospitality employed an AutoML tool to predict revenue, which would contribute to knowledge in this area. Moreover, the high accuracy obtained with this application represents an important contribution to Auto ML, confirming the TPOT's power in a domain that is completely different from the ones in which it has been used.

7- Declarations

7-1- Author Contributions

Conceptualization, M.C., D.C.P., S.S., and L.V.; methodology, M.C., D.C.P., S.S., and L.V.; software, D.M.; validation, M.C., D.C.P., S.S., and L.V.; investigation, M.C., D.C.P., S.S., D.M., and L.V.; writing—original draft preparation, M.C., D.C.P., S.S., D.M., and L.V.; writing—review and editing, M.C., D.C.P., S.S., and L.V.; visualization, M.C., D.C.P., S.S., D.M., and L.V.; supervision, L.V.; funding acquisition, L.V., and D.C.P.; All authors have read and agreed to the published version of the manuscript.

7-2- Data Availability Statement

Data is not publicly available given an NDA (non-disclosure agreement) between the company and the authors. However, it is available for editorial consultation, in case of need.

7-3- Funding

This study was supported by grant DSAIPA/DS/0113/2019 from FCT (Fundação para a Ciência e a Tecnologia), Portugal. This work was also supported by national funds through FCT (Fundação para a Ciência e a Tecnologia), under the project - UIDB/04152/2020 - Centro de Investigação em Gestão de Informação (MagIC)/NOVA IMS.

7-4- Institutional Review Board Statement

The project AutoML tool for hospitality was approved in 2022 by the Ethics Committee and Institutional Review Board, code DSCI2022-6-213250.

7-5- Informed Consent Statement

Not applicable.

7-6- Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this manuscript. In addition, the ethical issues, including plagiarism, informed consent, misconduct, data fabrication and/or falsification, double publication and/or submission, and redundancies have been completely observed by the authors.

8- References

[1] Columbus. L., (2018). 10 Ways Machine Learning Is revolutionizing Sales. Available online: https://www.forbes.com/sites/louiscolumbus/2018/12/26/10-ways-machine-learning-is-revolutionizing-sales/?sh=4c9cc92e3fd1 (accessed on April 2022).

[2] Feurer, M., Klein, A., Eggensperger, K., Springenberg, J., Blum, M., & Hutter, F. (2015). Efficient and robust automated machine learning. Advances in neural information processing systems, NeurIPS Proceedings, 28, 1-9.

[3] Caputo, F., & Walletzký, L. (2017). Investigating the users' approach to ict platforms in the city management. Systems, 5(1), 1–15. doi:10.3390/systems5010001.

[4] Boto Ferreira, M., Costa Pinto, D., Maurer Herter, M., Soro, J., Vanneschi, L., Castelli, M., & Peres, F. (2021). Using artificial intelligence to overcome over-indebtedness and fight poverty. Journal of Business Research, 131, 411–425. doi:10.1016/j.jbusres.2020.10.035.

[5] Humphreys, A., & Wang, R. J. H. (2018). Automated text analysis for consumer research. Journal of Consumer Research, 44(6), 1274–1306. doi:10.1093/jcr/ucx104.

[6] Shrestha, Y. R., Krishna, V., & von Krogh, G. (2021). Augmenting organizational decision-making with deep learning algorithms: Principles, promises, and challenges. Journal of Business Research, 123, 588–603. doi:10.1016/j.jbusres.2020.09.068.

[7] Melidis, C., Denham, S. L., & Hyland, M. E. (2018). A test of the adaptive network explanation of functional disorders using a machine learning analysis of symptoms. BioSystems, 165, 22–30. doi:10.1016/j.biosystems.2017.12.010.

[8] Mullainathan, S., & Spiess, J. (2017). Machine Learning: An Applied Econometric Approach. Journal of Economic Perspectives, 31(2), 87–106. doi:10.1257/jep.31.2.87.

[9] Ciechanowski, L., Jemielniak, D., & Gloor, P. A. (2020). Tutorial: AI research without coding: The art of fighting without fighting: Data science for qualitative researchers. Journal of Business Research, 117, 322–330. doi:10.1016/j.jbusres.2020.06.012.

[10] Yarkoni, T., & Westfall, J. (2017). Choosing Prediction over Explanation in Psychology: Lessons from Machine Learning. Perspectives on Psychological Science, 12(6), 1100–1122. doi:10.1177/1745691617693393.

[11] Chong, A. Y. L., Ch'ng, E., Liu, M. J., & Li, B. (2017). Predicting consumer product demands via Big Data: the roles of online promotional marketing and online reviews. International Journal of Production Research, 55(17), 5142–5156. doi:10.1080/00207543.2015.1066519.

[12] Liu, X., Shin, H., & Burns, A. C. (2021). Examining the impact of luxury brand's social media marketing on customer engagement: Using big data analytics and natural language processing. Journal of Business Research, 125, 815–826. doi:10.1016/j.jbusres.2019.04.042.

[13] Mcshane, B. B., & Böckenholt, U. (2017). Single Paper Meta-analysis: Benefits for Study Summary, Theory-testing, and Replicability. Journal of Consumer Research, ucw085. doi:10.1093/jcr/ucw085.

[14] Smith, L. W., & Rose, R. L. (2020). Service with a smiley face: Emojional contagion in digitally mediated relationships. International Journal of Research in Marketing, 37(2), 301–319. doi:10.1016/j.ijresmar.2019.09.004.

[15] Moorman, C., van Heerde, H. J., Moreau, C. P., & Palmatier, R. W. (2019). JM as a Marketplace of Ideas. Journal of Marketing, 83(1), 1–7. doi:10.1177/0022242918818404.

[16] Guo, X., Ye, Q., Law, R., Liang, S., & Zhang, Y. (2022). Power of apologetic responses in online travel community. International Journal of Hospitality Management, 103, 103208. doi:10.1016/j.ijhm.2022.103208.

[17] Sánchez-Medina, A. J., & C-Sánchez, E. (2020). Using machine learning and big data for efficient forecasting of hotel booking cancellations. International Journal of Hospitality Management, 89, 102546. doi:10.1016/j.ijhm.2020.102546.

[18] Kwon, W., Lee, M., & Back, K. J. (2020). Exploring the underlying factors of customer value in restaurants: A machine learning approach. International Journal of Hospitality Management, 91, 102643. doi:10.1016/j.ijhm.2020.102643.

[19] Ahani, A., Nilashi, M., Ibrahim, O., Sanzogni, L., & Weaven, S. (2019). Market segmentation and travel choice prediction in Spa hotels through TripAdvisor's online reviews. International Journal of Hospitality Management, 80, 52–77. doi:10.1016/j.ijhm.2019.01.003.

[20] Olson, R.S., Moore, J.H. (2019). TPOT: A Tree-Based Pipeline Optimization Tool for Automating Machine Learning. Automated Machine Learning, The Springer Series on Challenges in Machine Learning, Springer, Cham, Switzerland. doi:10.1007/978-3-030-05318-5_8.

[21] Hindman, M. (2015). Building Better Models. The ANNALS of the American Academy of Political and Social Science, 659(1), 48–62. doi:10.1177/0002716215570279.

[22] Wang, W. M., Wang, J. W., Barenji, A. V., Li, Z., & Tsui, E. (2019). Modeling of individual customer delivery satisfaction: an AutoML and multi-agent system approach. Industrial Management and Data Systems, 119(4), 840–866. doi:10.1108/IMDS-07-2018-0279.

[23] Gijsbers, P., Vanschoren, J., & Olson, R. S. (2018). Layered TPOT: Speeding up tree-based pipeline optimization. arXiv preprint arXiv:1801.06007. doi:10.48550/arXiv.1801.06007

[24] Koza, J. R. (1992). Genetic programming: On the programming of computers by means of natural selection (1st Ed.). A Bradford Book, Pennsylvania, United States.

[25] Poli, R., Langdon, W. B., & McPhee, N. F. (2008). A Field Guide to Genetic Programing. Lulu Enterprise, Morrisville, United States.

[26] Osborne, J. W. (2013). Best practices in data cleaning: A complete guide to everything you need to do before and after collecting your data. SAGE Publications, Newbury Park, United States. doi:10.4135/9781452269948.

[27] Suri, N. M. R., Murty, M. N., & Athithan, G. (2019). Outlier detection: techniques and applications. Springer Nature, Cham, Switzerland doi:10.1007/978-3-030-05127-3.

[28] Zhang, Y., Meratnia, N., & Havinga, P. (2010). Outlier detection techniques for wireless sensor networks: A survey. IEEE communications surveys & tutorials, 12(2), 159-170. doi:10.1109/SURV.2010.021510.00088.

[29] Kuhn, M., & Johnson, K. (2019). Feature engineering and selection: A practical approach for predictive models (1st Ed.). CRC Press, New York, United States. doi:10.1201/9781315108230.

[30] Antonopoulou, H., Mamalougou, V., & Theodorakopoulos, L. (2022). The Role of Economic Policy Uncertainty in Predicting Stock Return Volatility in the Banking Industry: A Big Data Analysis. Emerging Science Journal, 6(3), 569-577. doi:10.28991/ESJ-2022-06-03-011.

[31] Hossen, M. I., Goh, M., Hossen, A., & Rahman, M. A. (2020). A Study on the Aspects of Quality of Big Data on Online Business and Recent Tools and Trends towards Cleaning Dirty Data. 2020 11th IEEE Control and System Graduate Research Colloquium (ICSGRC). doi:10.1109/icsgrc49013.2020.9232648.

[32] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. the Journal of machine Learning research, 12, 2825-2830.

[33] Wolpert, D. H., & Macready, W. G. (1997). No free lunch theorems for optimization. IEEE Transactions on Evolutionary Computation, 1(1), 67–82. doi:10.1109/4235.585893.

[34] Zoller, M. A., & Huber, M. F. (2019). Benchmark and Survey of Automated Machine Learning Frameworks. arXiv. Learning, doi:10.48550/arXiv.1904.12054.

[35] Yao, Q., Wang, M., Chen, Y., Dai, W., Li, Y. F., Tu, W. W., ... & Yu, Y. (2018). Taking human out of learning applications: A survey on automated machine learning. arXiv preprint arXiv:1810.13306. doi:10.48550/arXiv.1810.13306.

[36] Banzhaf, W., Nordin, P., Keller, R. E., & Francone, F. D. (1998). Genetic programming: an introduction: on the automatic evolution of computer programs and its applications. Morgan Kaufmann Publishers Inc, Burlington, United States.

[37] Balaji, A., & Allen, A. (2018). Benchmarking automatic machine learning frameworks. arXiv preprint arXiv:1808.06492. doi:10.48550/arXiv.1808.06492.

[38] LeDell, E., & Poirier, S. (2020). H2o automl: Scalable automatic machine learning. 7th ICML Workshop on Automated machine Learning, 17-18 July, 2020, Vienna, Austria.

[39] Dafflon, J., Pinaya, W. H. L., Turkheimer, F., Cole, J. H., Leech, R., Harris, M. A., … Hellyer, P. J. (2020). An automated machine learning approach to predict brain age from cortical anatomical measures. Human Brain Mapping, 41(13), 3555–3566. doi:10.1002/hbm.25028.

[40] Bisong, E. (2019). Google Colaboratory. Building Machine Learning and Deep Learning Models on Google Cloud Platform. Apress, Berkeley, United States. doi:10.1007/978-1-4842-4470-8_7.

[41] Kingsford, C., & Salzberg, S. L. (2008). What are decision trees? Nature Biotechnology, 26(9), 1011–1012. doi:10.1038/nbt0908-1011.

[42] Raileanu, L. E., & Stoffel, K. (2004). Theoretical comparison between the Gini Index and Information Gain criteria. Annals of Mathematics and Artificial Intelligence, 41(1), 77–93. doi:10.1023/B:AMAI.0000018580.96245.c6.

[43] Friedman, J. H. (2002). Stochastic gradient boosting. Computational Statistics and Data Analysis, 38(4), 367–378. doi:10.1016/S0167-9473(01)00065-2.

[44] Smola, A. J., & Schölkopf, B. (2004). A tutorial on support vector regression. Statistics and Computing, 14(3), 199–222. doi:10.1023/B:STCO.0000035301.49549.88.

[45] Schölkopf, B., Mika, S., Burges, C. J. C., Knirsch, P., Müller, K. R., Rätsch, G., & Smola, A. J. (1999). Input space versus feature space in kernel-based methods. IEEE Transactions on Neural Networks, 10(5), 1000–1017. doi:10.1109/72.788641.

[46] Castelli, M., Clemente, F. M., Popovič, A., Silva, S., & Vanneschi, L. (2020). A Machine Learning Approach to Predict Air Quality in California. Complexity, 2020, 1–23. doi:10.1155/2020/8049504.

[47] Ho, C. H., & Lin, C. J. (2012). Large-scale linear support vector regression. Journal of Machine Learning Research, 13(1), 3323–3348.

[48] Wold, S., Esbensen, K., & Geladi, P. (1987). Principal component analysis. Chemometrics and Intelligent Laboratory Systems, 2(1-3), 37–52. doi:10.1016/0169-7439(87)80084-9.

[49] Agrapetidou, A., Charonyktakis, P., Gogas, P., Papadimitriou, T., & Tsamardinos, I. (2021). An AutoML application to forecasting bank failures. Applied Economics Letters, 28(1), 5–9. doi:10.1080/13504851.2020.1725230.

[50] Tsiakmaki, M., Kostopoulos, G., Kotsiantis, S., & Ragos, O. (2020). Implementing autoML in educational data mining for prediction tasks. Applied Sciences (Switzerland), 10(1), 90 2–27. doi:10.3390/app10010090.

[51] Orlenko, A., Kofink, D., Lyytikäinen, L.-P., Nikus, K., Mishra, P., Kuukasjärvi, P., … Moore, J. H. (2019). Model selection for metabolomics: predicting diagnosis of coronary artery disease using automated machine learning. Bioinformatics, 36(6), 1772–1778. doi:10.1093/bioinformatics/btz796.

[52] Liu, X. (2020). Analyzing the impact of user-generated content on B2B Firms' stock performance: Big data analysis with machine learning methods. Industrial Marketing Management, 86, 30–39. doi:10.1016/j.indmarman.2019.02.021.

[53] Cui, T. H., Ghose, A., Halaburda, H., Iyengar, R., Pauwels, K., Sriram, S., … Venkataraman, S. (2020). Informational Challenges in Omnichannel Marketing: Remedies and Future Research. Journal of Marketing, 85(1), 103–120. doi:10.1177/0022242920968810.

[54] Shuqair, S., Pinto, D. C., & Mattila, A. S. (2019). Benefits of authenticity: Post-failure loyalty in the sharing economy. Annals of Tourism Research, 78, 102741. doi:10.1016/j.annals.2019.06.008.

[55] Shuqair, S., Pinto, D. C., & Mattila, A. S. (2021). An empathy lens into peer service providers: Personal versus commercial hosts. International Journal of Hospitality Management, 99, 103073. doi:10.1016/j.ijhm.2021.103073.

[56] Mohamadou, Y., Halidou, A., & Kapen, P. T. (2020). A review of mathematical modeling, artificial intelligence and datasets used in the study, prediction and management of COVID-19. Applied Intelligence, 50(11), 3913–3925. doi:10.1007/s10489-020-01770-9.

[57] Peters, D. P. C., McVey, D. S., Elias, E. H., Pelzel‐McCluskey, A. M., Derner, J. D., Burruss, N. D., ⋯ Rodriguez, L. L. (2020). Big data–model integration and AI for vector‐borne disease prediction. Ecosphere, 11(6). doi:10.1002/ecs2.3157.

[58] Antonio, N., Almeida, A. de, & Nunes, L. (2017). Predicting hotel booking cancellations to decrease uncertainty and increase revenue. Tourism & Management Studies, 13(2), 25–39. doi:10.18089/tms.2017.13203.

[59] Caicedo-Torres, W., Payares, F. (2016). A Machine Learning Model for Occupancy Rates and Demand Forecasting in the Hospitality Industry. Advances in Artificial Intelligence - IBERAMIA 2016. Lecture Notes in Computer Science, 10022. Springer, Cham, Switzerland. doi:10.1007/978-3-319-47955-2_17.

[60] Xin, D., Wu, E. Y., Lee, D. J.-L., Salehi, N., & Parameswaran, A. (2021). Whither AutoML? Understanding the Role of Automation in Machine Learning Workflows. Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems. doi:10.1145/3411764.3445306.

[61] De Bruyn, A., Viswanathan, V., Beh, Y. S., Brock, J. K. U., & von Wangenheim, F. (2020). Artificial Intelligence and Marketing: Pitfalls and Opportunities. Journal of Interactive Marketing, 51, 91–105. doi:10.1016/j.intmar.2020.04.007.

[62] Mustak, M., Salminen, J., Plé, L., & Wirtz, J. (2021). Artificial intelligence in marketing: Topic modeling, scientometric analysis, and research agenda. Journal of Business Research, 124, 389–404. doi:10.1016/j.jbusres.2020.10.044.

[63] Kakatkar, C., Bilgram, V., & Füller, J. (2020). Innovation analytics: Leveraging artificial intelligence in the innovation process. Business Horizons, 63(2), 171–181. doi:10.1016/j.bushor.2019.10.006.

Appendix I: How to Perform the AutoML Tool Steps

- First, access one of the AutoML tools (classification, regression) online:

  (a) The classification AutoML tool is available online on Colab Google at this URL:
  https://colab.research.google.com/drive/1OJ5qjTZp3OnGzpaOOTRMD02WBX8dXyZf?usp=sharing

  (b) Tou can access the regression AutoML tool with this link:
  https://colab.research.google.com/drive/14OaJg9HyZVgZoogE55pYS2EhRAuOMZt9?usp=sharing

Please note that all steps are the same for classifier and regression tools. The only difference regards the outputs, which we will explain later.

- Make a copy of the tool on your Google Drive: Go to "File" and select "save a copy in the drive." For better tool performance, we strongly recommend saving a copy on your Google drive (see Figure A-1), which would allow you to perform the analysis on your own Google Drive account.

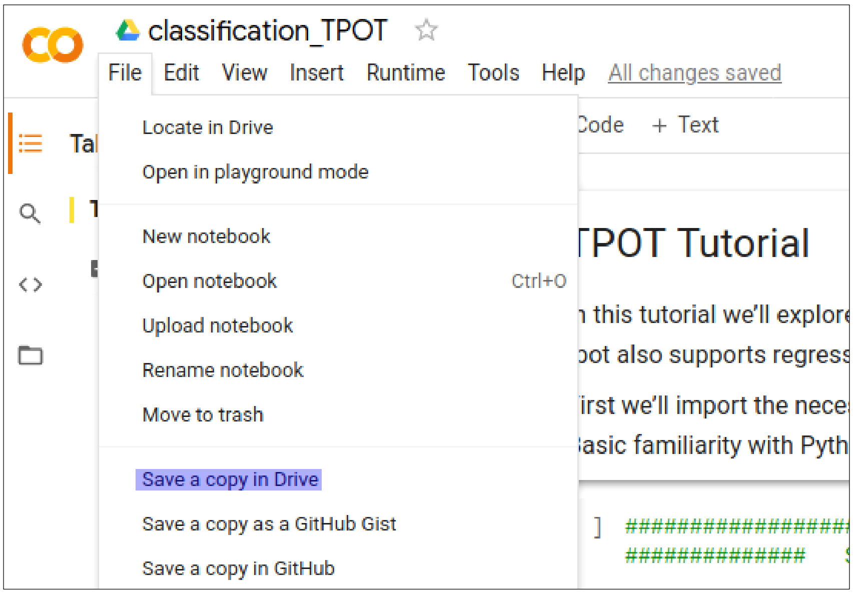


Figure A-1. Save a copy

- Once you save a copy on your drive, you should be able to run the following commands in sequential order. You can use the example dataset available for training the model, or you can upload your own CSV data file. To do so, go to "files" on the upper left corner and select the upload icon, 📄 as Figure A-2 shows.

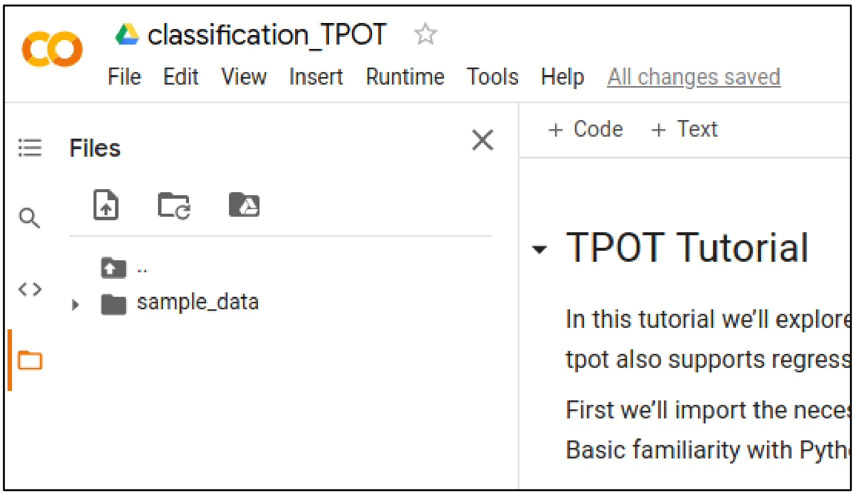
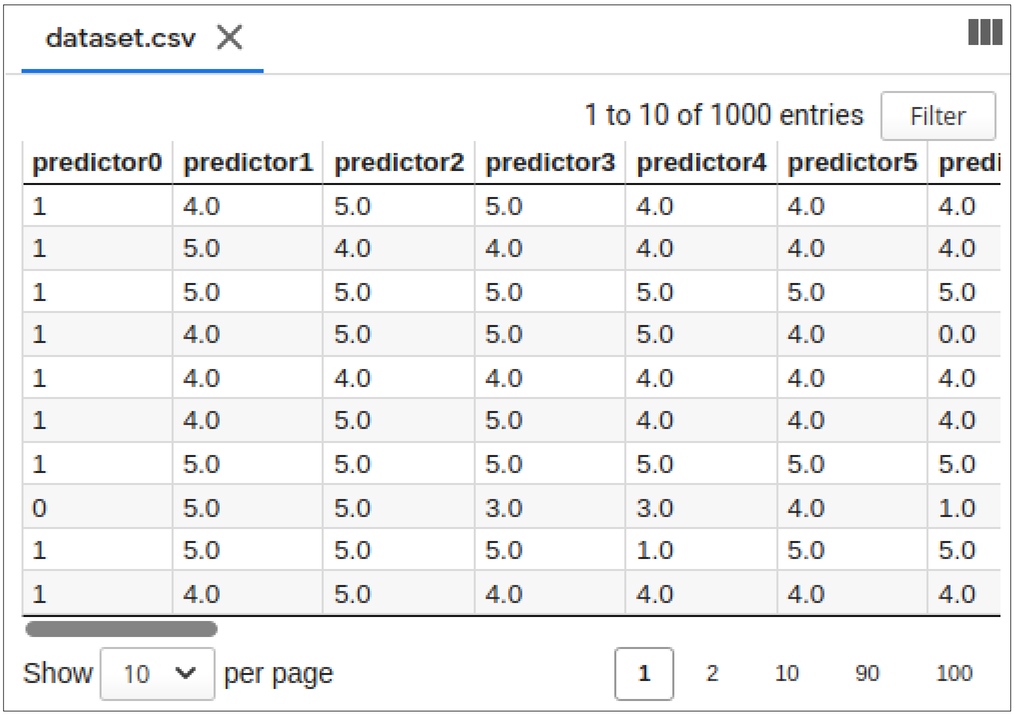


Figure A-2. Uploading dataset

The uploaded dataset must have the same format as the example dataset. In particular, the first row of the CSV file contains the variables' names. Each subsequent row of the file corresponds to a single observation (i.e., sample).

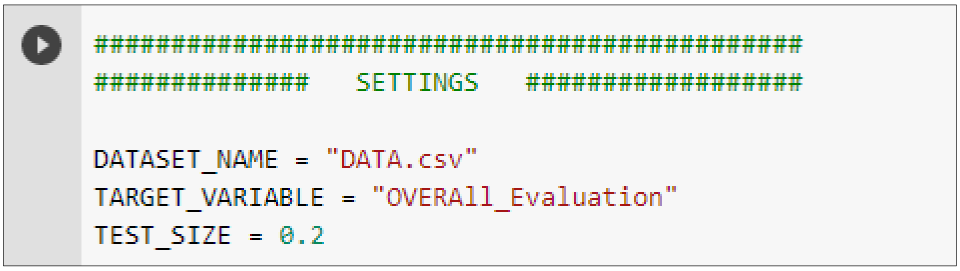
Important tip: Before running the AutoML tool, you must ensure that all your data is in a numeric format (e.g., 1, 2, 3 …). ML libraries are built to work with numeric matrices. Therefore, if the dataset contains categorical variables, code them in a numeric format (e.g., female = 1, male = 2). The file must be uploaded as a CSV file to ensure the code works correctly. To view your data, double-click on File sample data, and the dependent variables' values will appear on the right side (see Figure A-3).



predictor0	predictor1	predictor2	predictor3	predictor4	predictor5	predi
1	4.0	5.0	5.0	4.0	4.0	4.0
1	5.0	4.0	4.0	4.0	4.0	4.0
1	5.0	5.0	5.0	5.0	5.0	5.0
1	4.0	5.0	5.0	5.0	4.0	0.0
1	4.0	4.0	4.0	4.0	4.0	4.0
1	4.0	5.0	5.0	4.0	4.0	4.0
1	5.0	5.0	5.0	5.0	5.0	5.0
0	5.0	5.0	3.0	3.0	4.0	1.0
1	5.0	5.0	5.0	1.0	5.0	5.0
1	4.0	5.0	4.0	4.0	4.0	4.0

Figure A-3. **Data display**

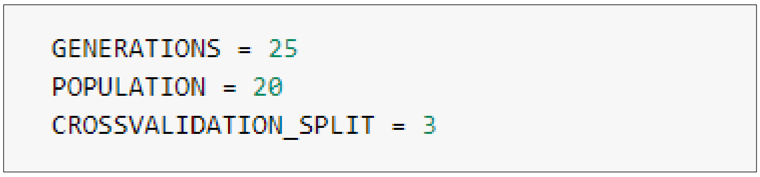
- Once you upload your data set, scroll down to the next segment of the coding. At this point, you will see DATASET_NAME and TARGET VARIABLE (see Figure A-4). Choose the name of your data and your target variable. The target variable is the variable that you want to predict. It can be any variable (e.g., satisfaction, purchasing intentions). In this demonstration, the target variable is "overall evaluation." The name of the target variable (e.g., "DV") should be equivalent to the name of the chosen variable in your dataset (e.g., column labeled "DV"). The DATASET_NAME variable should simply reflect your custom dataset's name.



```
##############################################
##############   SETTINGS   ##################

DATASET_NAME = "DATA.csv"
TARGET_VARIABLE = "OVERAll_Evaluation"
TEST_SIZE = 0.2
```

Figure A-4. **Choose a name for the target variable**

- Set parameter: The next step initializes the TPOT. At this stage, you need to set parameters. The most important parameters are the number of generations, the population size, and cross-validation. The TPOT will evaluate POPULATION_SIZE and GENERATIONS x OFFSPRING_SIZE pipelines in total. As Figure A-5 shows, we selected 25 generations and a population size of 20.

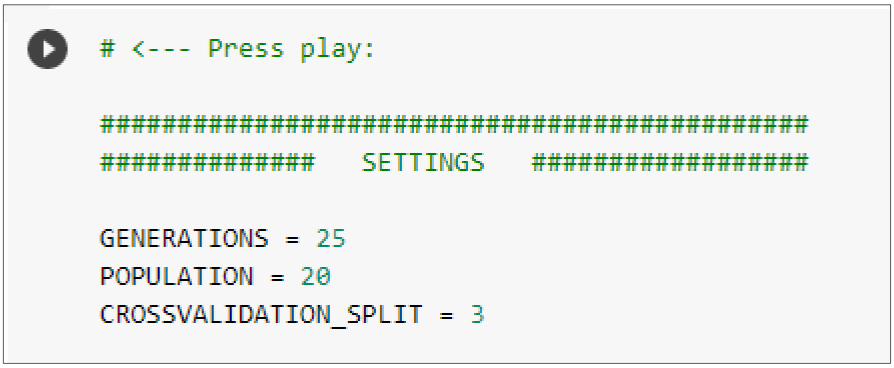


```
GENERATIONS = 25
POPULATION = 20
CROSSVALIDATION_SPLIT = 3
```

Figure A-5. **Set parameters**

A rule of thumb, the most important parameters are generations, population, and cross-validation split. The population size is the number of individuals to retain in the GP population in every generation. In this tutorial we set *N* as 20. In general, the TPOT works better when the sample size is bigger [31, 54]. The number of generations defines number of iterations of the pipeline optimization process.

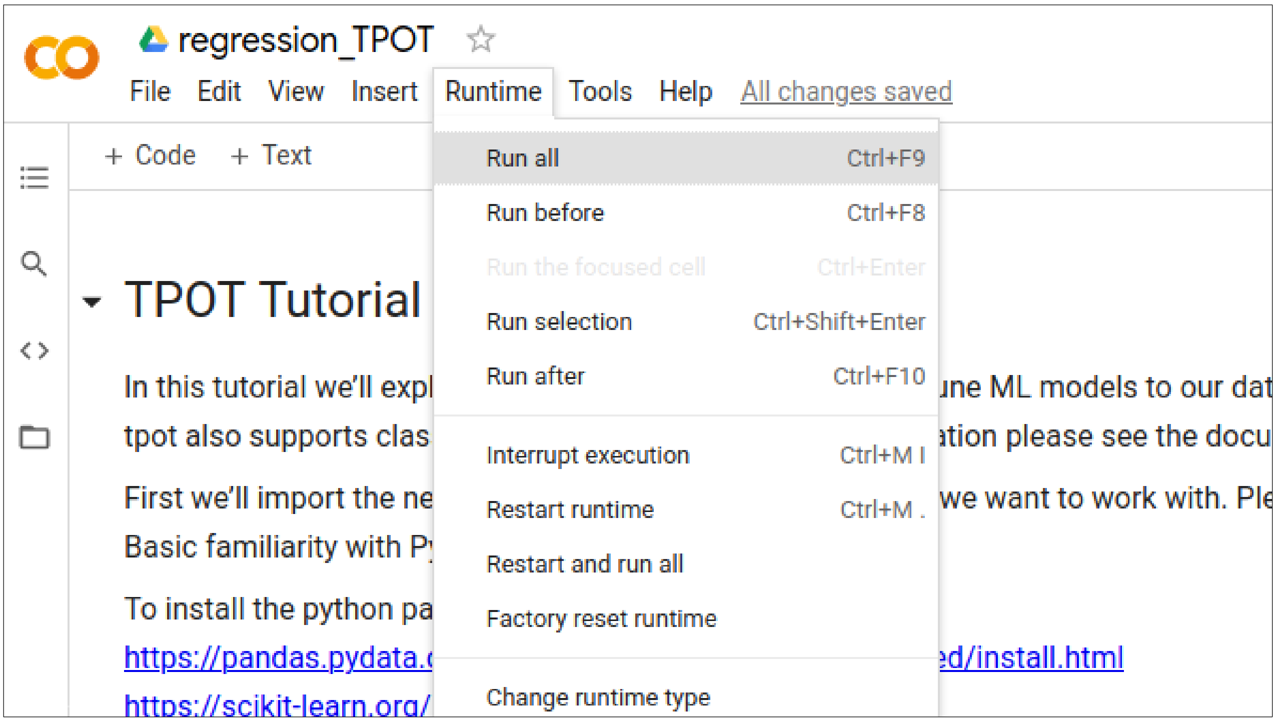
The TPOT takes longer with more data. By default, the TPOT settings include 100 generations and 100 as the population size, producing 10,000 model configurations to evaluate with 10-fold cross-validation, which means that 100,000 models are fitted and evaluated on the training data in one grid search [55]. Cross-validation affects how long TPOT takes to evaluate models, as each model will be trained k times, where k equals the number of cross-validation splits [56]. To initialize the TPOT library, you need to click on the "play button" (see Figure (A-6)).



```
# <--- Press play:

###############################################
##############   SETTINGS   ##################

GENERATIONS = 25
POPULATION = 20
CROSSVALIDATION_SPLIT = 3
```

Figure A-6. Select the number of generations

- Tune the tool: once you have defined your parameters, including the dataset and target variable as well as the number of generations, you can execute the whole notebook by either selecting “Run all” from the Runtime menu or pressing Ctrl+F9 (Figure A-7). The notebook will run through all its steps, as illustrated below, and print the outputs at the end.



TPOT Tutorial

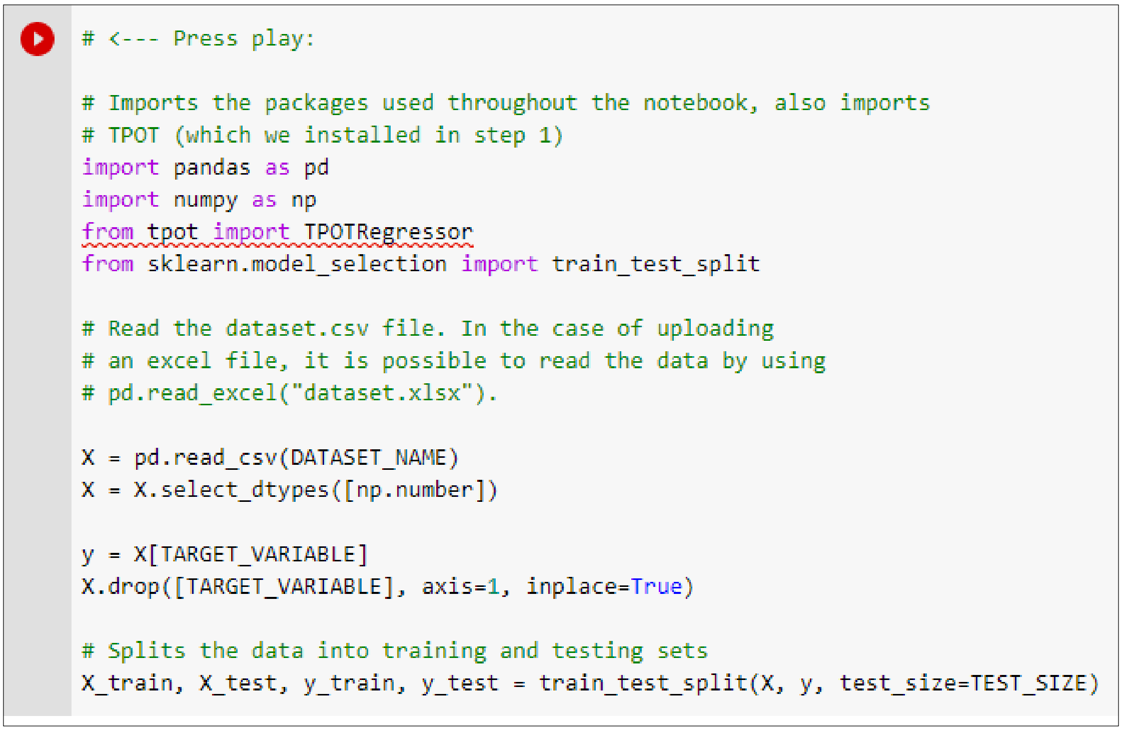
In this tutorial we'll expl … une ML models to our dat
tpot also supports clas … ation please see the docu

First we'll import the ne … we want to work with. Ple
Basic familiarity with P

To install the python pa
https://pandas.pydata.c … ed/install.html
https://scikit-learn.org/

Figure A-7. Run All Feature

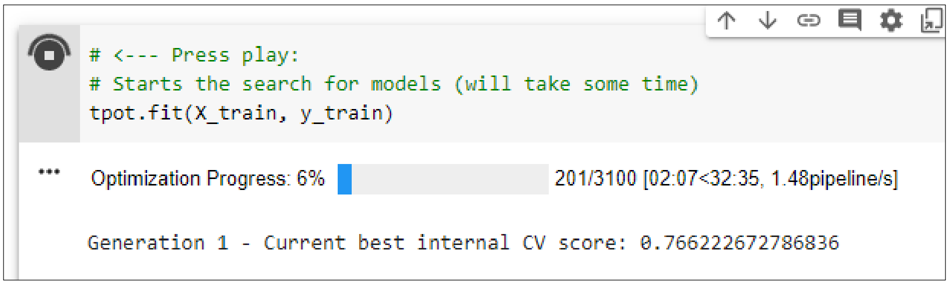
Import packages: The purpose of this test set is to later assess the model on data that has not been seen beforehand. To do so, press play as you see in the Figure A-8.



```
# <--- Press play:

# Imports the packages used throughout the notebook, also imports
# TPOT (which we installed in step 1)
import pandas as pd
import numpy as np
from tpot import TPOTRegressor
from sklearn.model_selection import train_test_split

# Read the dataset.csv file. In the case of uploading
# an excel file, it is possible to read the data by using
# pd.read_excel("dataset.xlsx").

X = pd.read_csv(DATASET_NAME)
X = X.select_dtypes([np.number])

y = X[TARGET_VARIABLE]
X.drop([TARGET_VARIABLE], axis=1, inplace=True)

# Splits the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=TEST_SIZE)
```

Figure A-8. **Import packages**

- Search model: To begin, scroll down to TPOT.fit and press (***play button***) as Figure A-9 shows. Please note that this step will take a few minutes to search for models.

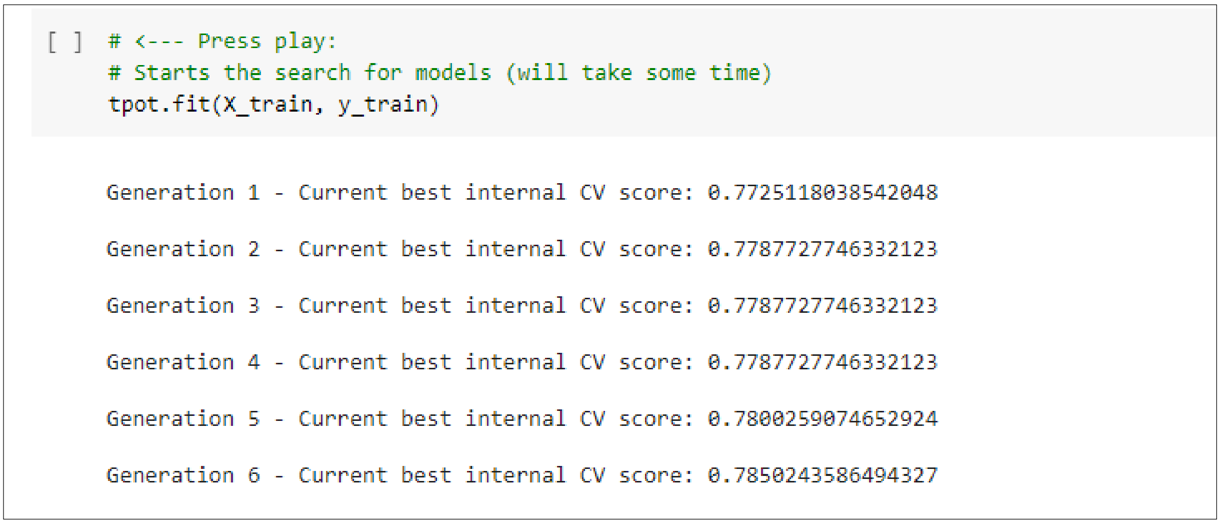


```
# <--- Press play:
# Starts the search for models (will take some time)
tpot.fit(X_train, y_train)
```

```
Optimization Progress: 6%   201/3100 [02:07<32:35, 1.48pipeline/s]

Generation 1 - Current best internal CV score: 0.766222672786836
```

Figure A-9. **Search models**

- Once it searches for the optimal model, TPOT displays the current generation and the cross-validated accuracy (see Figure A-10)



```
[ ]  # <--- Press play:
     # Starts the search for models (will take some time)
     tpot.fit(X_train, y_train)
```

```
Generation 1 - Current best internal CV score: 0.7725118038542048

Generation 2 - Current best internal CV score: 0.7787727746332123

Generation 3 - Current best internal CV score: 0.7787727746332123

Generation 4 - Current best internal CV score: 0.7787727746332123

Generation 5 - Current best internal CV score: 0.7800259074652924

Generation 6 - Current best internal CV score: 0.7850243586494327
```

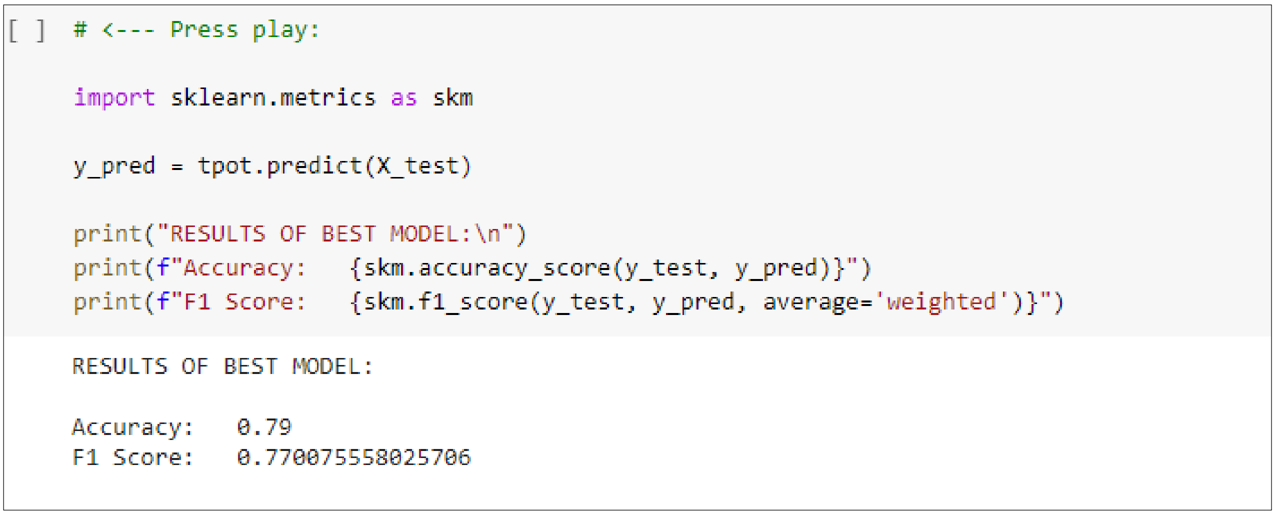
Figure A-10. **Results**

TPOT will run several algorithms, searching for the best pipeline model in terms of accuracy.

- To view the results, scroll down to (Results of the best model). Figure A-11 reports the model accuracy (0.79) and the F1 score (0.7700). The accuracy is calculated as the total number of correctly classified observations divided by the total number of observations (i.e., correctly and incorrectly classified). The F1 score is calculated as the weighted average of the precision and recall. Therefore, to understand the F1 score fully, it is necessary to define the concepts of precision and recall. Precision is defined as the ratio tp / (tp + fp), where tp is the number of true positives and fp the number of false positives. Recall is defined as the ratio tp / (tp + fn), where tp is the number of true positives and fn the number of false negatives. With the definition of precision and recall, it is possible to calculate the F1 score as

  F1 = 2 * (precision * recall) / (precision + recall)

  The F1 score reaches its best score at 1 and worst score at 0.

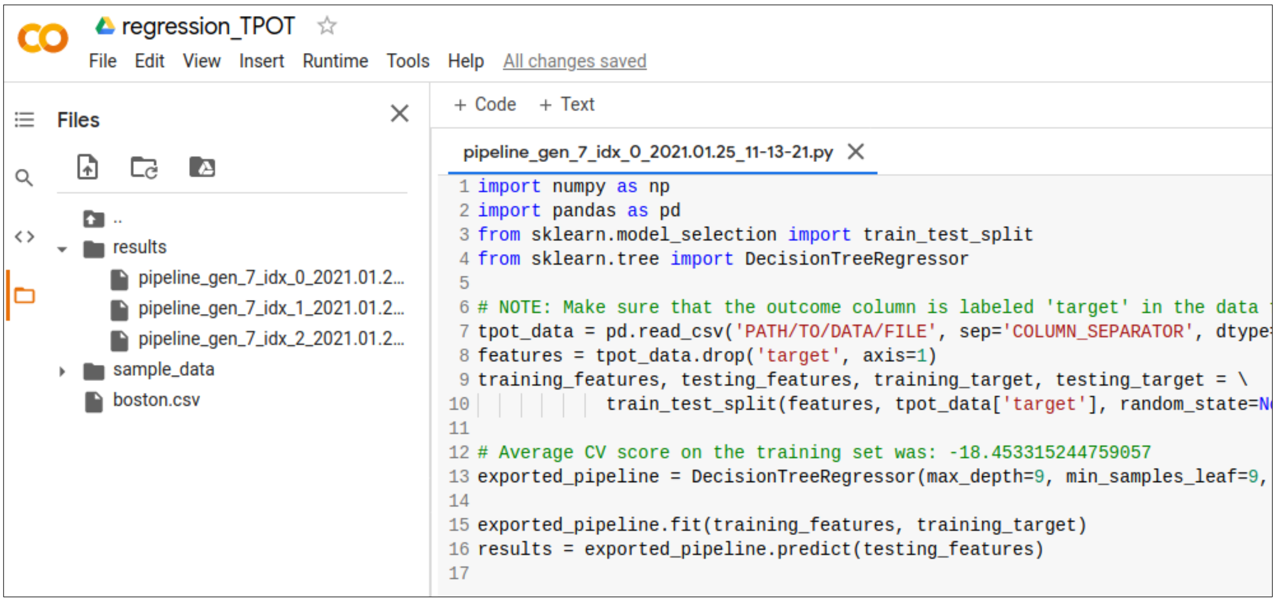


```
# <--- Press play:

import sklearn.metrics as skm

y_pred = tpot.predict(X_test)

print("RESULTS OF BEST MODEL:\n")
print(f"Accuracy:   {skm.accuracy_score(y_test, y_pred)}")
print(f"F1 Score:   {skm.f1_score(y_test, y_pred, average='weighted')}")
```

```
RESULTS OF BEST MODEL:

Accuracy:   0.79
F1 Score:   0.770075558025706
```

Figure A-11. Best model

- To export the Python pipeline, go to File and open results. You can then view the model's accuracy (see Figure A-12).

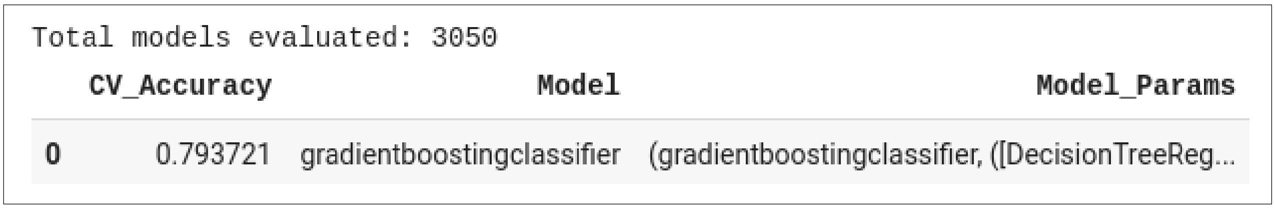


regression_TPOT
File
Edit
View
Insert
Runtime
Tools
Help
All changes saved
Files
+ Code
+ Text
pipeline_gen_7_idx_0_2021.01.25_11-13-21.py
..
results
pipeline_gen_7_idx_0_2021.01.2...
pipeline_gen_7_idx_1_2021.01.2...
pipeline_gen_7_idx_2_2021.01.2...
sample_data
boston.csv

```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeRegressor

# NOTE: Make sure that the outcome column is labeled 'target' in the data
tpot_data = pd.read_csv('PATH/TO/DATA/FILE', sep='COLUMN_SEPARATOR', dtype
features = tpot_data.drop('target', axis=1)
training_features, testing_features, training_target, testing_target = \
            train_test_split(features, tpot_data['target'], random_state=N

# Average CV score on the training set was: -18.453315244759057
exported_pipeline = DecisionTreeRegressor(max_depth=9, min_samples_leaf=9,

exported_pipeline.fit(training_features, training_target)
results = exported_pipeline.predict(testing_features)

```

Figure A-12. Export Pipeline

- The final step allows you to print the results. It will print the number of models the TPOT evaluated and show a table of the Pareto-front models
  - As a classification tool, the outcome will look like Figure 13-A

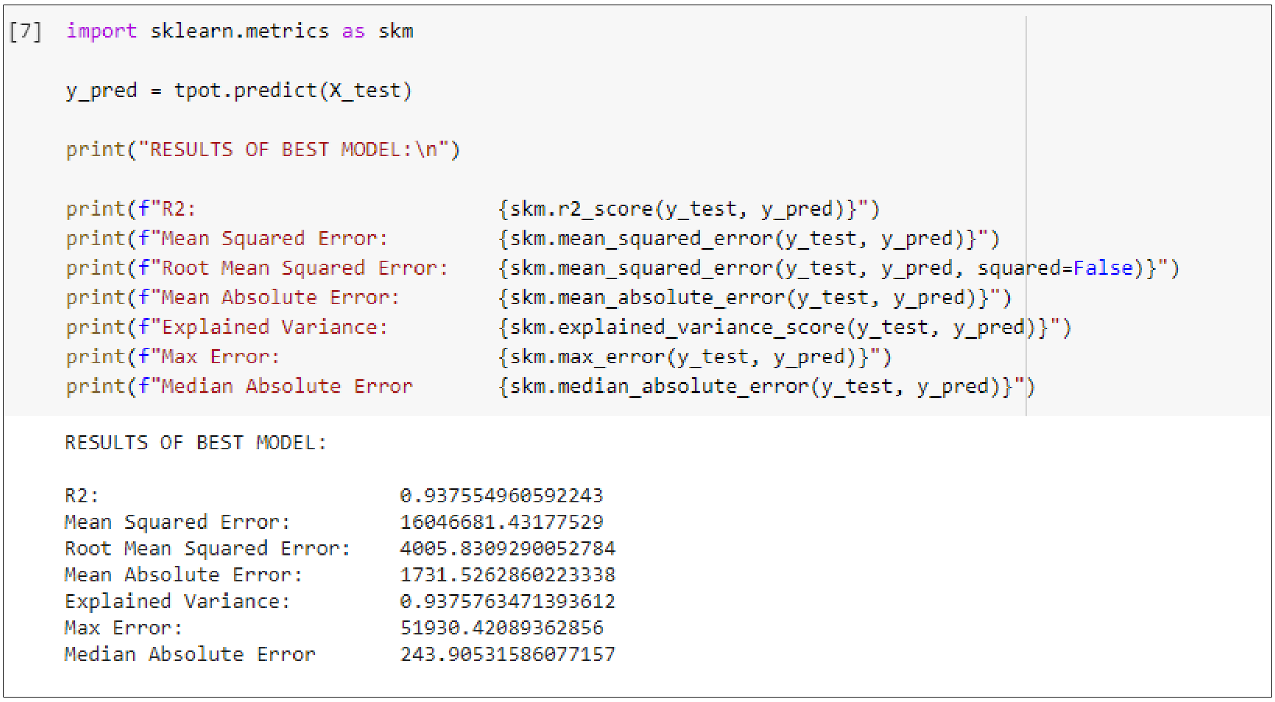


```
Total models evaluated: 3050
```

	CV_Accuracy	Model	Model_Params
0	0.793721	gradientboostingclassifier	(gradientboostingclassifier, ([DecisionTreeReg...

Figure A- 13. Results

– When you perform regression, the matrix output will be regression oriented (see Figure 14-A)



```
[7]  import sklearn.metrics as skm

     y_pred = tpot.predict(X_test)

     print("RESULTS OF BEST MODEL:\n")

     print(f"R2:                          {skm.r2_score(y_test, y_pred)}")
     print(f"Mean Squared Error:          {skm.mean_squared_error(y_test, y_pred)}")
     print(f"Root Mean Squared Error:     {skm.mean_squared_error(y_test, y_pred, squared=False)}")
     print(f"Mean Absolute Error:         {skm.mean_absolute_error(y_test, y_pred)}")
     print(f"Explained Variance:          {skm.explained_variance_score(y_test, y_pred)}")
     print(f"Max Error:                   {skm.max_error(y_test, y_pred)}")
     print(f"Median Absolute Error        {skm.median_absolute_error(y_test, y_pred)}")

     RESULTS OF BEST MODEL:

     R2:                         0.937554960592243
     Mean Squared Error:         16046681.43177529
     Root Mean Squared Error:    4005.8309290052784
     Mean Absolute Error:        1731.5262860223338
     Explained Variance:         0.9375763471393612
     Max Error:                  51930.42089362856
     Median Absolute Error       243.90531586077157
```

Figure A-14. Regression matrix