Unleashing the Potential of AutoML in Understanding and Addressing Poverty

Inês Magessi
NOVA Information Management School, Lisbon, Portugal

June, 2023

Abstract

This study explores the application of Automated Machine Learning (*AutoML*) techniques in understanding and addressing poverty. Poverty extends beyond income insufficiency and encompasses various challenges such as hunger, malnutrition, limited education, and social exclusion. The objective of this work is to develop predictive models using Supervised Machine Learning to identify individuals at a higher risk of poverty. Two AutoML approaches were utilized: a Grid Search method to evaluate different machine learning model configurations within predefined hyperparameters, and TPOT, an *AutoML* tool employing genetic programming for automating the optimization of machine learning pipelines. By leveraging *AutoML*, this project aims to facilitate proactive poverty prevention and intervention measures, mitigating the challenges associated with it and empowering individuals to overcome its adverse effects. The results demonstrate the effectiveness of *AutoML* in generating accurate predictive models for identifying individuals at greater risk of poverty.

Keywords: AutoML, Poverty, Predictive Models

1 Introduction

Poverty represents an enduring and persistent challenge that results in more than just income insufficiency. Its consequences comprise hunger, malnutrition, constrained educational opportunities and social exclusion, among many others [4]. In light of these ramifications, researchers, policymakers, and governments have the collective responsibility of delving deeper into the roots of poverty to identify vulnerabilities and recognize those at greater risk. By prioritizing prevention and intervention measures, we can work towards reducing the significant challenges associated with poverty and empowering individuals to overcome its adverse effects.

This work focuses on demonstrating the power of using Automated Machine Learning (*AutoML*) to obtain predictive models capable of identifying those at greater risk of poverty by means of Supervised Machine Learning. Two *AutoML* approaches were used: a Grid Search to train and test an extensive set of machine learning model configurations within a pre-defined hyperparameters search space; and an *AutoML* tool that uses genetic programming to automate the optimization of machine learning pipelines, called *TPOT*.

2 Data

2.1 Data Collection

To accomplish this project goals, it was crucial to collect detailed and extensive individual-level data, which would subsequently serve as input for machine learning models to infer the poverty risk of each individual.

The data selected for this project was collected by the National Financial Well-Being Survey [1], conducted by the Consumer Financial Protection Bureau, in the United States in 2017. This dataset includes an extensive set of variables regarding 6,394 members of the GfK's Knowledge Panel [2] who agreed to respond to the survey, providing answers that would be helpful in assessing their financial well-being and the development. The 217 collected variables include social-demographic information, psychological profile, housing situation and financial knowledge, behavior and status. The target variable in this analysis is the poverty status, which

is determined based on the Federal Poverty Level. This indicator serves as a guideline to assess an individual's income relative to the poverty threshold. This way, the target variable has three possible values, representing different income categories based on the Federal Poverty level. A more detailed, yet still resumed, table with the variables can be analysed in Appendix A.

The survey aimed at representing the non-institutionalized adult population in the 50 U.S. states and Washington, D.C. To achieve this, the panel recruitment employed random digit dialing (RDD) and address-based sampling (ABS) methods, with a focus on maintaining representativeness. Additionally, the survey ensured adequate representation of the Latino population by selecting elements from the Latino KnowledgePanel. Furthermore, an oversample of 999 adults aged 62 and older was included in the sample. This oversampling is reflected in the *Sample* feature of the dataset.

Lastly, to ensure the sample's accuracy in reflecting the larger population, the survey organization compared the collected data to geodemographic benchmarks, including variables such as sex, age groups, race/ethnicity categories, education levels, regions, household income ranges, home ownership status, and metropolitan area residence. Adjustments were made to the sample's weights and selection probabilities to achieve representativeness, which is featured in the variable *Sample Weight*. Aligning the sample with these benchmarks enhances the generalizability of the survey results to the broader population.

2.2 Data Cleaning

The *National Financial Well Being Survey* was conducted online, with a significant number of participants from the Knowledge Panel. In such surveys, it is common for individuals to not respond to certain questions, either by choice or due to a lack of recollection. Additionally, misinterpretation and lack of support in online surveys can lead to incorrect responses. Therefore, a thorough analysis was conducted on the dataset to exclude inconsistent data and address missing values.

To handle missing values, a decision was made to focus on reliable data for training the models. Consequently, any observation with five or more missing values was discarded, resulting in a dataset with 297 fewer observations. On the other hand, for the remaining observations, features with fewer than 100 missing values were imputed using central tendency measures. Depending on the feature's distribution and data type, the missing values were filled with either the median or the mode. The mean was not used since no continuous feature was among these particular features.

For features with more than 100 missing responses, the *Nearest Neighbors Imputation* method was employed. This method involved analyzing the feature with missing values in relation to all other features to identify the strongest correlations. By using only these discriminating features to find the five nearest neighbors, a more accurate estimation of the missing values was ensured. All these features with missing values were either ordinal or nominal, therefore customized *KNNImputers* were implemented to handle these types of data. These imputations utilized a custom distance measure and filled the missing values using the mode or median value of the feature within the neighbors' values. Prior to imputation, scaling was applied to the features used to determine the nearest neighbors, ensuring equal weighting, regardless of their value range.

Additionally, redundant features were eliminated during the cleaning stage. This included removing answers to questions used for calculating financial knowledge scores, as their information was already captured in features such as *KHscore* and *LMScore*. Furthermore, one-hot-encoded features indicating the absence of any other selected one-hot feature were also removed. Removing these features was crucial to avoid perfect collinearity among the variables. Additionally, features like *KIDS_X*, which denoted the number of children within specific age ranges, were eliminated due to numerous inconsistencies likely caused by misinterpretation.

Lastly, it is important to note that some questions with categorical answers represented by numbers were originally stored as a single variable. However, it is necessary to transform this variables before utilizing them in machine learning models. These models consider the order between numbers, which is not applicable in this context where the distance or numerical relationship between the potential answers is irrelevant. Consequently, categorical variables were encoded into one-hot variables to ensure accurate representation and interpretation by the models.

By the end of the process, a total of 55 original features and 316 observations were removed from the dataset. Following the one-hot encoding step, the cleaned dataset consisted of 188 variables and 6,078 observations. Among these final features, 64 were either numerical or ordinal, while the remaining variables comprised the original binary variables or the one-hot encoded representations of the categorical variables.

2.3 Data Preprocessing

After cleaning the data and converting categorical features into one-hot encodings, the data needs to undergo a preprocessing stage before being used to train machine learning models. This stage typically involves scaling numeric features and may optionally include a feature engineering step.

Feature engineering is a powerful machine learning technique that involves creating new variables from existing data to enhance model performance. It is typically applied after data cleaning and before scaling, ensuring that the newly engineered features are also appropriately scaled. The purpose of feature engineering is to simplify and expedite data transformations while improving the accuracy of the models.

In this project, the specific feature engineering technique employed is known as *Feature Extraction*. During this process, new features are derived by compressing existing features into more manageable quantities without distorting the original relationships or losing significant information. For instance, multiple features related to the same variable group, such as subjective well-being questions, can be consolidated into a single variable by calculating the mean and storing the aggregated values. This technique was utilized in one of the two runs of the Hyperparameter Grid Search, as explained in Section 3.1. For further details, please refer to the corresponding section.

After optionally performing feature selection (discussed in the next section) , an essential next step is scaling the numeric features. Scaling is a critical step in the AutoML pipeline as it brings all feature values to a similar range. This ensures that machine learning models treat all numerical features as equally important, regardless of their original scales. Additionally, when features are scaled to a similar range, models tend to perform better and converge faster.

There are several scaling techniques available for numeric features. This project focuses on two commonly used ones: *Standard Scaling* and *Min-Max Scaling*. The *Standard Scaler* was applied to features that follow a Gaussian distribution, transforming them to have a mean of 0 and a standard deviation of 1. On the other hand, *Min-Max Scaling* subtracts the minimum value in the feature and then divides by the range, which is the difference between the original maximum and original minimum values. *Min-Max Scaling* preserves the shape of the original distribution, ensuring that the information embedded in the data remains unchanged. This last scaler was applied to the remaining numeric features.

2.4 Feature Selection

Feature selection techniques are employed to reduce the number of input variables by eliminating redundant or irrelevant features and narrowing down the set of features to those most relevant to the machine learning models. This results in simpler and usually more accurate models, that require less time to train and that help avoid the the curse of high dimensionality [5].

However, one must also consider that feature selection can potentially remove important information and lead to reduced model performance. To account for this, all models trained during the Grid Search were evaluated with and without feature selection. This way, all model's configurations were evaluated, not only to compare performances but also to observe the impact of feature selection on the identical model setup.

There are several methods of feature selection and two were employed in this work. The first one is a univariate feature selection method that chooses features based on their correlation with the target. The correlation based method can only be applied to numeric or ordinal features, excluding categorical ones. This is because correlation measures the linear or non-linear relationship between two variables, which implies that they are continuous or at least ordinal, having a relationship of order between them. Categorical features are discrete, therefore they do not have a natural ordering or numeric representation that allows for meaningful correlation calculations. This method showed to be insufficient to perform feature selection as only one variable (Household income) showed a considerable value of correlation (0.7). However, it was useful to find highly correlated redundant variables, which correlations can be seen in Table 1. The four features highlighted in grey were removed from the dataset due to their high correlation with other variables and their perceived lower relevance. This step removed some evident redundancies on the dataset.

However, the objective of selecting features based on their predictive information for the target variable was not fully achieved. To address this limitation, a more robust feature selection technique called Recursive Feature Elimination (RFE) was employed. RFE systematically eliminates less important features from the dataset, identifying the most relevant ones. Initially, a predefined model (e.g., Decision Tree) is trained on all features, ranking them by importance. The least important feature is then eliminated, and the model is

Feature	*socsec1*	*mortgage*	*valuerange*	*agecat*	*employ_8*	*ppeduc*	*pphhsize*	*hheduc*	*ppt18ov*
socsec1	1	-	-	0.762	0.787	-	-	-	-
mortgage	-	1	0.819	-	-	-	-	-	-
valuerange	-	-	1	-	-	-	-	-	-
agecat	-	-	-	1	0.714	-	-	-	-
employ_8	-	-	-	-	1	-	-	-	-
ppeduc	-	-	-	-	-	1	-	0.787	-
pphhsize	-	-	-	-	-	-	1	-	0.802
hheduc	-	-	-	-	-	-	-	1	-
pp18ov	-	-	-	-	-	-	-	-	1

Table 1: Highly correlated variables

retrained on the reduced feature set. This iterative process continues until a minimum number of features is reached. RFE reduces dimensionality, improves model performance, and preserves the subset of features contributing most to predictive power.

In this project, a Decision Tree was utilized as the feature ranking model, employing an entropy-based metric suitable for tabular data with a discrete target variable. The optimal number of features, ranging from 70 to the maximum available, was determined through performance evaluation during cross-validation. Interestingly, on all cross-validation splits in the hyperparameters GridSearch pipeline, it was observed that 70 features, the minimum number, consistently provided the best results. This observation suggests that a small subset of features has an equivalent impact on performance compared to using all the available features, highlighting the potential for simplification of the model without sacrificing predictive ability.

RFE was employed in each cross-validation fold during one of the runs of the hyperparameter Grid Search, providing insights into the features that were consistently chosen across the dataset splits. This allows us to infer which features are generally considered the most important. Interestingly, there were 23 features that were selected in every cross-validation split, indicating their consistent significance. For a comprehensive description of these features, please refer to Table 2.

3 Modeling

Finding the most suitable Machine Learning Pipeline for a given task can be daunting. The *No Free Lunch* theorem highlights the need to consider the problem's specific characteristics when selecting the optimal preprocessing steps and model hyperparameters for a machine learning algorithm. Consequently, it is common practice to conduct a comprehensive search within the space of machine learning pipelines. This involves systematically exploring different combinations of preprocessing steps and model hyperparameters while evaluating their impact on the model's performance.

In the subsequent sections, we delve into the details of the two *AutoML* approaches employed in this project, providing a comprehensive explanation of their methodologies.

3.1 Hyperparameter Grid Search

One commonly used method for exhaustive model training and comparison is Hyperparameter Grid Search, aiming to identify the most robust and high-performing model, searching among a set of predefined algorithms and corresponding hyperparameters search space. This *AutoML* approach does not include the optimization of preprocessing steps on the Machine Learning pipeline, and focus solely on optimizing the models' parameters. In this project, a Grid Hyperparameters Search was conducted using both the original cleaned dataset and a reduced version obtained through feature selection and engineering, as described in sections 2.3 and 2.4.

To ensure a more reliable performance assessment of each model, a Stratified 10-Fold Cross Validation was applied. The results from the training and validation stages were averaged across the different folds to obtain the final performance metrics. This results were later analysed and compared using statistical measures, in order to asses the quality and reliability of the best models.

Considering the target variable's imbalance and to effectively evaluate each model, the assessment relied

Feature	*Description*
FWBscore	Financial well-being scale score
FSscore	Financial skill scale score
KHscore	Knoll and Houts financial knowledge scale score
SUBKNOWL1	Subjective Financial Knowledge
SAVEHABIT	Saving habit
FRUGALITY	Willingness to re-use an item instead of buying a new one
AUTOMATED_1	Have money automatically transferred into a Retirement Savings Account
CHANGEABLE	Belief that ability to manage money is not changeable
HOUSERANGES	Monthly mortgage
CONSPROTECT2	Familiarity w/ agencies and orgs to resolve problems w/ financial services
agecat	Age category
PPHHSIZE	Household Size
PPINCIMP	Household Income
PPREG9	Census Division
PROPPLAN	Budget planning (Aggregation Engineered Feature)
FINSOC	Financial education while growing up (Aggregation Engineered Feature)
MATERIALISM	Materialism (Aggregation Engineered Feature)
SNAP	Any household member received SNAP benefits
PAREDUC	Highest level of education by person/people who raised respondent
CONNECT	Psychological connectedness
HEALTH	Health status
HOUSESAT	Satisfaction with current housing
LIFEEXPECT	Likelihood of living beyond age 75

Table 2: Features selected in every CV split

not only on the accuracy score but also on metrics such as the *F1-measure*, *Precision*, and *Recall*, which were weighted across all target classes.

The Hyperparameter Grid Search was performed on a diverse range of Machine Learning Classification algorithms, including Logistic Regression, Decision Tree, Random Forest, Gradient Boosting, Adaptive Boosting, Support Vector Machines, K-Nearest Neighbors (KNN), and Multi-Layer Perceptron (MLP). This resulted in a total of 368 different models being trained and tested.

All models were trained taking into account the weight of each sample (for further details consult Section 2.1). The term "model", in this context, refers to the result of using a specific machine learning algorithm with a set of hyperparameters and set of features obtained with or without feature selection and engineering.

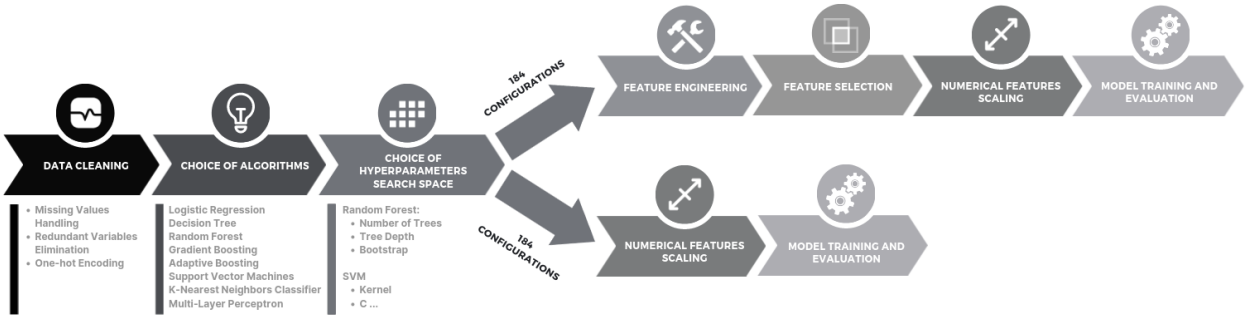

Figure 1: Hyperparameter Grid Search Overview

3.2 TPOT

This project also assessed the power of a different AutoML tool called TPOT, which stands for Tree-based Pipeline Optimization Tool [3]. This tool was used to identify the Machine Learning Pipeline with greatest potential in correctly predicting the risk of poverty, by using genetic programming to iteratively find the pipelines that result in the highest fitness, which, in our case, was the highest weighted *F1-Measure*.

The greatest advantage of using this tool compared to using the Hyperparameter Grid Search, is that it automates the most tedious part of machine learning by intelligently exploring thousands of possible pipelines to find the best one, including stages like feature selection, preprocessing and engineering, model selection, while also optimizing hyperparameters.

A notable advantage is that TPOT doesn't require a predefined hyperparameter search space. Instead, it autonomously discovers the most suitable pipeline configurations through genetic programming. Additionally, the dataset used in this project was not preprocessed or subjected to feature selection before being fed into TPOT, as these tasks are integral parts of the optimization pipeline. The only requirement is that the data is clean.

To ensure a fair comparison with the Grid Search method, TPOT employed the same Stratified 10-Fold Cross Validation split. This approach allows for a reliable and accurate comparison of results between the two AutoML tools.

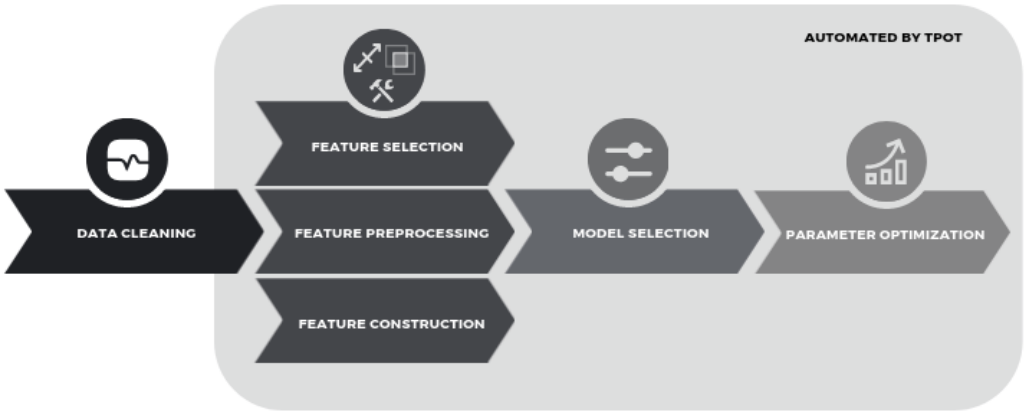

Figure 2: Overview of the TPOT Pipeline Search

4 Experimental Results

4.1 Grid Search Results

After executing the Grid Search, a total of 368 model configurations were trained and evaluated using 10-Fold cross-validation. The performance of each model configuration was recorded for metrics such as *Accuracy*, *Precision*, *Recall*, and *F1-Measure* in each fold. To gain insights into the overall performance of different algorithms, the median performance of each type of algorithm across the 10 evaluation runs was calculated.

Attempting to individually analyze the performance of all tested models would be exhaustive, impractical, and not very informative. Therefore, the primary focus of this initial high-level analysis is to identify algorithms that demonstrated significant potential. The evaluation revealed that Multilayer Perceptrons (MLPs) and Random Forests exhibited the most promising performance across all metrics. Thirty different configurations of each of these algorithms underwent training and evaluation. In one set of configurations, feature engineering and feature selection techniques were applied, while in the other set, only the cleaned data was utilized. The median *F1-Measures* achieved by the total of 60 models of each of these algorithms on the validation set were approximately 0.93 and 0.92, respectively.

In the following sections we delve deeper in the analysis of the performance of the best algorithms.

4.2 TPOT Results

The optimization process conducted by TPOT resulted in the discovery of an efficient pipeline comprising of three main steps: preprocessing, feature construction and the final model.

For the preprocessing step, TPOT determined that the Maximum Absolute Scaler was the optimal method for scaling numerical features. This technique is commonly employed in machine learning to normalize features within the range of [-1, 1], based on the maximum absolute value of each feature. It achieves this by dividing each feature by its maximum absolute value.

In the feature construction step, TPOT employed a Stacking Estimator as a meta-transformer. This component combines base estimators to generate additional features. Specifically, it utilized a Random Forest as the base estimator of the Stacking Estimator. Random Forest is an ensemble method that constructs multiple decision trees and combines their predictions. The Stacking Estimator leverages the base estimator to create new features, such as predictions or class probabilities, which are subsequently used as input for subsequent modeling stages.

Finally, the pipeline included a Logistic Regression model as the final component. This model predicts class labels for the given data by modeling the probability of the outcome based on new features, which are the outputs of the base estimator. To prevent overfitting, the Logistic Regression model employs a Ridge penalty.

In summary, TPOT identified a relatively complex pipeline that would have been challenging to discover intuitively through more manual methods like Grid Search.

4.3 Algorithms Best Configurations

After conducting a high-level analysis of the performance of the Grid Search model and the pipeline generated by TPOT, our attention turns to examining the optimal configurations of each algorithm within the Grid Search. We also compare these configurations to the pipeline produced by TPOT.

Figure 3 provides valuable insights into the comparative performance of the various algorithms. Notably, both MLP and TPOT demonstrate exceptional *F1-Measure* scores in both the training and validation stages, while exhibiting minimal overfitting. This indicates their robustness and generalization capability.

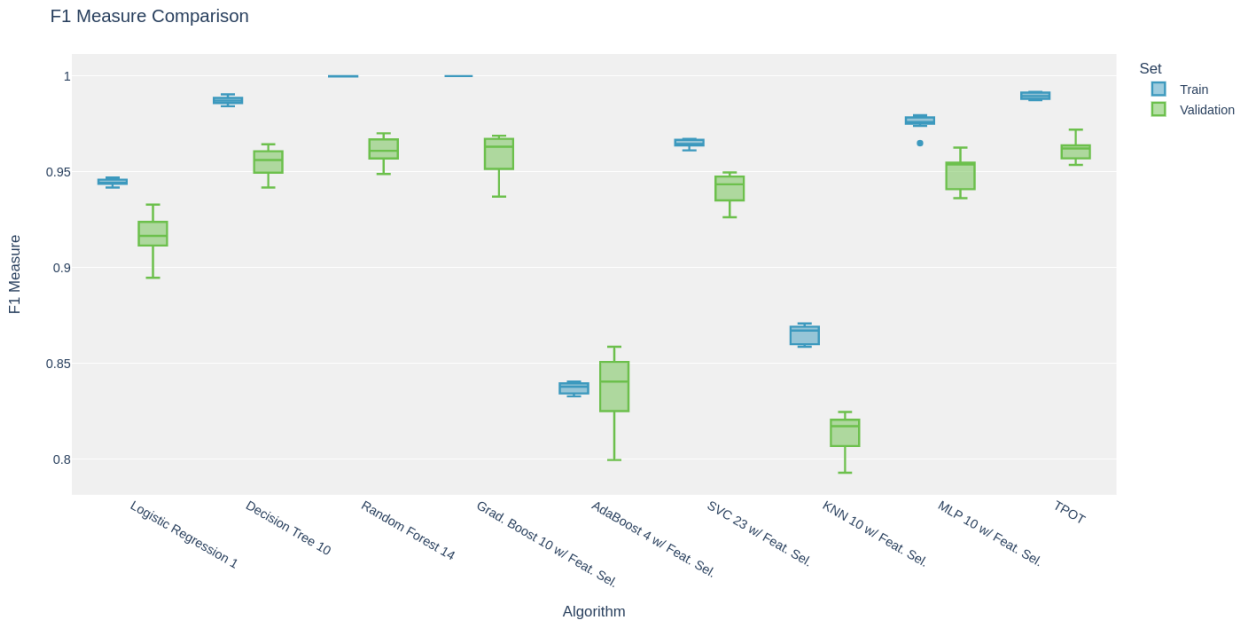

Figure 3: F1-Measure - Best Models

To delve deeper into the evaluation, we refer to Table 3, which presents a detailed breakdown of additional metrics. Based on the results showcased in the table, we conclude that the Gradient Boosting Classifier, whose

hyperparameters included a learning rate of 0.1, 350 estimators, and a maximum tree depth of 10, and that was trained using the reduced dataset after feature selection and engineering, attains the highest performance in all metrics in the validation set. This outcome aligns with expectations, as tree-based ensemble models are well-known for their efficacy in handling tabular data. The Gradient Boosting algorithm excels at rectifying errors from previous iterations through gradient-based optimization.

However, it is crucial to note that the performance of the TPOT-generated pipeline remains remarkably competitive with the results achieved by the Gradient Boosting Classifier. Furthermore, the TPOT pipeline exhibits less overfitting, implying a superior ability to generalize to unseen data. This highlights the efficacy of the TPOT optimization process in discovering highly competitive pipelines for model construction.

Model/Metric	Train Accuracy	Train F1-Measure	Train Precision	Train Recall	Val Accuracy	Val F1-Measure	Val Precision	Val Recall
Logistic Regression 1	0.941	0.944	0.952	0.941	0.912	0.917	0.928	0.912
Decision Tree 10	0.986	0.987	0.987	0.987	0.956	0.956	0.958	0.956
Random Forest 14	1.000	1.000	1.000	1.000	0.961	0.961	0.962	0.961
Gradient Boosting 10 w/ Feat. Sel.	1.000	1.000	1.000	1.000	0.963	0.963	0.964	0.963
AdaBoost 4 w/ Feat. Sel.	0.853	0.838	0.864	0.853	0.857	0.840	0.871	0.857
SVC 23 w/ Feat. Sel.	0.964	0.965	0.966	0.964	0.942	0.943	0.946	0.942
KNN 10 w/ Feat. Sel.	0.877	0.867	0.867	0.877	0.834	0.817	0.810	0.834
MLP 10 w/ Feat. Sel.	0.976	0.976	0.976	0.976	0.954	0.954	0.954	0.954
TPOT	0.990	0.990	0.990	0.990	0.962	0.962	0.962	0.962

Table 3: Best Models Performance

4.4 Best Models Comparison

The statistical significance of the best models from each algorithm was evaluated using the *Wilcoxon* signed-rank test. This test allows us to determine if there is a significant difference in performance between two models. The null hypothesis assumes that both models have equal performance on the classification task, while the alternative hypothesis suggests that they have different levels of performance.

To assess the significance, we calculate the p-value, which measures the probability of obtaining the observed difference in performance between the two models if the null hypothesis is true. The default significance threshold of 0.05 was chosen for the test. If the p-value is lower than this threshold, it indicates that the observed difference in performance is highly unlikely to occur by chance alone. Consequently, for every p-value below 0.05, we conclude that there is a significant difference in performance between the two models being compared.

Based on the results of the test, we can draw several conclusions. Firstly, the Decision Tree model does not exhibit a statistically significant difference in performance compared to Gradient Boosting, MLP, and TPOT. Similarly, Random Forest's performance is not significantly different from that of Decision Tree, Gradient Boosting and TPOT. Additionally, the performance of Gradient Boosting is not significantly different from Decision Tree, Random Forest, MLP, and TPOT. Moreover, MLP's performance does not significantly differ from Decision Tree and Gradient Boosting. Lastly, TPOT's performance is not significantly different from Decision Tree, Random Forest, and Gradient Boosting.

On the other hand, the models Logistic Regression, AdaBoost, KNN (K-Nearest Neighbors), and SCV (Support Vector Classifier) consistently demonstrate significant differences when compared to all other models. These results suggest that those perform noticeably worse than the remaining algorithms.

This information helps us better understand the relative strengths and weaknesses of the models and their suitability for the classification task at hand.

5 Discussion and Final Thoughts

Throughout the course of this project, several significant conclusions have been drawn. One crucial finding pertains to the identification of key variables that show a strong influence on an individual's poverty status. This discovery deepens our understanding of the intricate dynamics that contribute to poverty, enabling us to design targeted interventions that address underlying causes and allocate resources effectively for maximum impact.

Moreover, it is worth noting that despite the similarity and lack of statistical significance between the best machine learning pipeline identified by TPOT and the best algorithms determined by Grid Search, the

utilization of TPOT yielded a notably smoother and more efficient process, resulting in excellent outcomes. The design process with TPOT proved to be intuitive, requiring minimal expertise beyond data cleaning. This advantage is particularly significant for individuals such as policymakers and government workers who may lack extensive data expertise but still need accessible results to inform decision-making and prompt effective actions. TPOT's nearly ready-to-use API facilitates a seamless workflow, enabling reliable risk assessment predictions and contributing to preventive measures against poverty.

Most importantly, this project successfully demonstrated the potential of leveraging AutoML and other Machine Learning tools to accurately predict poverty risk and identify the most influential contributing factors. It is our hope that this project serves as inspiration for government entities and non-profit organizations to harness these invaluable tools, empowering them to gain a deeper understanding of poverty risks and take proactive measures to mitigate them effectively. By embracing these powerful technologies, we can foster a more comprehensive and informed approach to poverty alleviation.

A Dataset Features

Feature Name/Category	**Data Type**	**Group**
Public Use File ID	Numeric	N/A
Sample	Categorical	N/A
Social information	Categorical & Ordinal	Social-Demographic
Demographic information	Categorical	Social-Demographic
Household information	Categorical & Ordinal	Social-Demographic
Health	Ordinal	Social-Demographic
% of people with income < FPL	Categorical	Social-Demographic
Optimism about life	Ordinal	Psychological Profile
Self-Control	Categorical & Ordinal	Psychological Profile
Connections	Ordinal	Psychological Profile
Stress	Ordinal	Psychological Profile
Subjective Well Being	Ordinal	Psychological Profile
Materialism	Ordinal	Psychological Profile
Household Income	Ordinal	Financial Situation
Savings	Ordinal	Financial Situation
Financial Benefits	Binary	Financial Situation
Debt Collection	Binary	Financial Situation
Financial Products	Ordinal	Financial Situation
Potential Financial Setbacks	Binary	Financial Situation
Victim of Fraud	Binary	Financial Situation
Credit Acceptance	Binary	Financial Situation
Financial Knowledge Scores	Ordinal	Financial Knowledge
Subjective Financial Knowledge	Ordinal	Financial Knowledge
Financial Education	Binary	Financial Knowledge
Financial Planning and Goals	Ordinal & Binary	Financial Behaviour
Financial Responsibility	Ordinal	Financial Behaviour
Saving Habits	Ordinal & Binary	Financial Behaviour
Frugality	Ordinal	Financial Behaviour
Action in case of Money Need	Ordinal & Binary	Financial Behaviour
Financial Management	Ordinal & Binary	Financial Behaviour
Relationship with Financial Entities	Ordinal	Financial Behaviour
Sample weight	Numeric	N/A
Poverty status (Target)	Categorical	N/A

References

[1] Consumer Financial Protection Bureau. *Financial Well-Being Survey Data*. en. https://www.consumerfinance.gov/data-research/financial-well-being-survey-data/. June 2023.

[2] Ipsos. *GfK Knowledge Panel*. en-us. 2023.

[3] *TPOT*. URL: http://epistasislab.github.io/tpot/.

[4] United Nations. *Poverty Eradication*.

[5] Wikipedia. *Curse of Dimensionality*. en. May 2023.