# ML ALGORITHM SELECTION

**Ali Torabi**
Computer Science Department
University of Wyoming
WY, 82071
atorabi@uwyo,edu

November 2, 2023

## 1 Introduction

Machine learning is changing our lives swiftly. While a simple task like image recognition seems so trivial for humans, there is a lot of work to do in the machine learning pipeline, such as data cleaning, feature engineering, finding which models best fit for the specific problem, and hyperparameters tuning, among many other tasks. A lot of these tasks are still not automated and need an expert to do some of these trials and errors. Automated Machine Learning (AutoML) provides a process to discover the best machine learning model for a given task according to the dataset automatically with very little need expert need. In this experiment, a sample dataset is used to see how it works with different machine learning models. The aim os to find the best model for this particular dataset.

## 2 Dataset Description

The dataset chosen for this experiment is Wine Quality dataset [1]. It is related to red wine samples from the north of Portugal. It comprises with 1599 instances and 11 different features. The label is the quality of the wine that make the data suitable for Classification and Regression tasks. The task of each of these algorithms would be used to detect the quality of wine ranges from poor to excellent. As mention in the description of the dataset itself, it has no missing value. The features are: fixed_acidity, volatile_acidity, citric_acid, residual_sugar, chlorides, free_sulfur_dioxide, total_sulfur_dioxide, density, pH, and sulphates. The output variable is quality which is scored between 0 and 10.

## 3 Experimental Setup

For this experiment, the Python programming language (version 3.10) has been chosen. Also, for the sake of simplicity and to show how AutoML simplifies things, the AutoGluon [2] library has been used. In this libray, you choose which models you want your dataset to be trained from in dictionary format. You can choose as much as models you want. After training on each of those models, then it can predict on testset on those models. Also, for reading dataset and splitting them into separate train and test data, we use pandas and sklearn, respectively. Autogluon uses TabularPredictor method, which it takes label, train dataset, and a set of models as their parameters. Therefore, it uses predict method to predict on test set. The result would be leaderboad, which say, the one with highest value would be the best model.

The algorithms that I choose for algorithm selection are RandomForest, XGBoost, CATBoost, WeightedEnsemble, Linear Regression, KNN, ExtraTrees, and GBM. As AutoGluon implies, the whole data set partitioning to train, validate, and test has been done implicitly inside the calling function. After specifying the algorithms, each one of them has gone through the training and testing phase to get their accuracies.

Table 1: Leaderboard for Algorithm Selection

| model | score_val | pred_time_val | fit_time | pred_time_val_marginal | fit_time_marginal | stack_level | can_infer | fit_order |
|---|---|---|---|---|---|---|---|---|
| WeightedEnsemble_L3 | 0.73 | 0.69 | 168.26 | 0.0008 | 0.51 | 3 | True | 15 |
| CatBoost_BAG_L2 | 0.73 | 0.66 | 145.27 | 0.03 | 77.15 | 2 | True | 11 |
| LightGBM_BAG_L1 | 0.72 | 0.09 | 13.79 | 0.09 | 13.79 | 1 | True | 2 |
| WeightedEnsemble_L2 | 0.72 | 0.09 | 14.39 | 0.0009 | 0.59 | 2 | True | 8 |
| LightGBM_BAG_L2 | 0.72 | 0.66 | 100.52 | 0.036 | 32.40 | 2 | True | 9 |
| XGBoost_BAG_L2 | 0.71 | 0.66 | 90.59 | 0.032 | 22.47 | 2 | True | 13 |
| RandomForest_BAG_L2 | 0.71 | 0.83 | 70.55 | 0.20 | 2.43 | 2 | True | 10 |
| ExtraTrees_BAG_L2 | 0.71 | 0.86 | 68.90 | 0.23 | 0.77 | 2 | True | 12 |
| CatBoost_BAG_L1 | 0.71 | 0.02 | 38.02 | 0.02 | 38.02 | 1 | True | 4 |
| RandomForest_BAG_L1 | 0.70 | 0.15 | 1.56 | 0.15 | 1.56 | 1 | True | 3 |
| XGBoost_BAG_L1 | 0.70 | 0.06 | 8.9 | 0.06 | 8.99 | 1 | True | 6 |
| ExtraTrees_BAG_L1 | 0.70 | 0.16 | 1.0 | 0.16 | 1.01 | 1 | True | 5 |
| LinearModel_BAG_L2 | 0.70 | 0.81 | 71.7 | 0.18 | 3.65 | 2 | True | 14 |
| LinearModel_BAG_L1 | 0.6 | 0.08 | 4.71 | 0.08 | 4.71 | 1 | True | 7 |
| KNeighbors_BAG_L1 | 0.51 | 0.047 | 0.005 | 0.04 | 0.005 | 1 | True | 1 |

## 4 Results

After training those models, we have the output like this: AutoGluon training complete, total runtime = 219.21s ... Best model: WeightedEnsemble_L3

So, the final output would be like Table 1, where their columns are:

score_val: The validation score of the model

pred_time_val: The inference time required to compute predictions on the validation data end-to-end, which is equivalent to the sum of all

pred_time_val_marginal: values for the model and all of its base models

fit_time: The fit time required to train the model end-to-end (Including base models if the model is a stack ensemble), which is quivalent to the sum of all fit_time_marginal values for the model and all of its base models

pred_time_val_marginal: The inference time required to compute predictions on the validation data (Ignoring inference times for base models)

fit_time_marginal: The fit time required to train the model (Ignoring base models)

stack_level: The stack level of the model. A model with stack level N can take any set of models with stack level less than N as input, with stack level 1 models having no model inputs

can_infer: If model is able to perform inference on new data. If False, then the model either was not saved, was deleted, or an ancestor of the model cannot infer

fit_order: The order in which models were fit. The first model fit has fit_order=1, and the Nth model fit has fit_order=N. The order corresponds to the first child model fit in the case of bagged ensembles

## References

[1] Wine Quality, UC Irvine Machine Learning Repository https://archive.ics.uci.edu/dataset/186/wine+quality

[2] AutoGluon library for Algorithm Selection https://auto.gluon.ai/stable/api/autogluon.tabular.TabularPredictor.leaderboard.html