A simple AutoML system

In this task, your group has to implement a simplified AutoML system.

Your task is to develop a program that is able to predict whether a flight will arrive delayed or not. This program should be sold to various airport companies. Unfortunately, each airport company is very secretive and they do not share their data with other airports. Therefore, they want a program that they can train on their own data and that will predict delays for new arrivals as well as possible.

However, you are lucky because the data from all airports is in the same format and you already got your hands on the datasets of 20 different airports. This will allow you to simplify the training procedure of your AutoML system and it will allow you to give estimates on how well your system will work for potential clients.

Upload your solution as a zip file in moodle under *Final Exercise*.

The points below have no meaning but to indicate the balance of work between both tasks.

1. **Programming** [3 points]

   Your AutoML system should consist of a single object. Helper functions are allowed. The object should have a `train` method that finds the optimal learner and/or hyperparameters for the given dataset and then trains it accordingly. The obtained model should be stored within the object. After `train` has been called, it should be possible to call the `predict` method that receives unlabeled data and returns a vector with the predictions.

   (a) If you use R: Write an R6 class that has a train method `automl$train(data)` and a method to obtain predictions on new data `predictions = automl$predict(data[,-"Delay"])`.

   (b) If you use Python: Write a class that has a train method `automl.train(data)` and a method to obtain predictions on new data `predictions = automl.predict(data.loc[:, data.columns != 'Delay'])`.

   You also have to write code to evaluate the performance of your AutoML system correctly.

   Hint: In `Automl.R` you will find the abstract `Automl` base class. You can implement your method as a child class. A stub is given as `AutomlCustom`. `make_mlr3_learner.R` contains a function that generates an mlr3 learner from your Automl class object. `test.R` contains some code that can help you to check that your code runs. If you want to use Python: `Automl.py` contains an `AutomlClassfier` class that can be used to implement an sklearn classifier.

2. **Report** [7 points]

   Finally, you have to write a report.

   (a) Explain how your AutoML system works.

   (b) Explain why you chose a specific ML algorithm, optimization method and search space.

   (c) Demonstrate, how your method performs in comparison to a naive solution (e.g. an untuned random forest).

   (d) Estimate how well your AutoML system will work for data from new airports.

   (e) (Bonus) Analyze the hyperparameter configurations that your AutoML system considered and show which performed best.

Please see `practical_tips.pdf` for further requirements and recommendations for your report.