uc3m
2024-25. ADVANCED PROGRAMMING

SECOND ASSIGNMENT: PREDICTING EMPLOYEE ATTRITION/BURNOUT (4 POINTS)

INDEX OF CONTENTS

INTRODUCTION

The purpose of the first assignment is to practice with machine learning methods, both basic and advanced, including hyper-parameter tuning, some preprocessing (categorical encoding, scaling, constant features, etc.).

The topic is employee attrition: a company is worried about employee attrition/burnout (employees leaving the company) and would like to create a model that predicts whether employees are likely to resign based on a dataset collected by the human resources department.

GENERAL CONSIDERATIONS

1. Results **must be reproducible**. Therefore, set the seed (the student ID of one of the members of the group) at the appropriate places.

2. Estimation of future performance will be done with **Holdout** (train/test). The main metrics will be **Accuracy and Balanced Accuracy**, and **confusion matrices** can be used as well to report results.

3. **Execution time** of the training process for all methods (fit) should also be reported.

4. **Preprocessing** should be conducted using **pipelines when appropriate** and using the required preprocessing steps for each of the chosen methods.

5. Presenting results in a **clear way** will be valued. Tables can be used to present multiple results and compare alternatives.

6. Each group will use **two files**: "attrition_availabledata_xx.csv" and "attrition_competition_xx.csv". xx = a + b, where "ab" are the last two digits of the NIA of one of the group's members. "available_data" contains the data for doing most tasks in the assignment (training models, doing hyper-parameter optimization, estimating future performance, etc). "competition_data" contains data that simulates a competition, and therefore, it does not contain the response variable ("Attrition"). The final model will be used to make predictions for the "competition_data".

STEPS TO FOLLOW

1. SIMPLIFIED EDA (0.5 POINTS)

Do a **simplified EDA**, mainly to determine how many features and instances there are, which variables are categorical/numerical, if there are categorical variables with high cardinality, which features have missing values and how many, whether there are constant columns or ID columns, and whether it is a regression or classification problem. If the latter, is it imbalanced? You can explore other issues in the data you find useful.

2. SETUP (0.25 POINTS)

1. Split your data into train and test.
2. Decide on how the **inner evaluation** is going to be carried out. The inner evaluation is used when doing hyper-parameter optimization/tuning, but it is also used to evaluate and compare different alternatives. This means that if for example, you decide to use 3-fold crossvalidation for the inner evaluation, all alternatives that you try should be evauated with 3-fold crossvalidation, in order to be compared.

3. BASIC METHODS: TREES AND KNN (1.75 POINTS)

1. Train, evaluate and **compare the two basic methods** with default hyperparameters, and also a **dummy method**. For KNN, you should use a pipeline with preprocessing included (scaling, at least). You should compare 2 scaling methods for KNN and determine which one works better in this problem.

2. Do **hyperparameter tuning for the two basic methods,** using GridSearch and/or Random Search. Does HPO improve results over default hyper-parameter values? At what computational cost? Which HPO technique obtains the best results?

4. RESULTS AND FINAL MODEL (0.5 POINTS)

- **Report your results**: report the inner evaluation of all alternatives tested and select the best one according to the inner evaluation.

- **Evaluate the best alternative on the test set** in order to estimate what the performance of your model could be at the competition (outer evaluation **== estimation of future performance**).

- **Using the best method**, train the final model and use it to **make predictions on the competition dataset**. Save both the **final model** in an appropriate ML format and the **competition predictions** in a a csv file.

5. FEATURE SELECTION FOR KNN (0.5 POINTS)

- Using now the optimal number of neighbors found when doing HPO, use grid-search for selecting the optimal number of features (using SelectKBest). Is the number of features selected much smaller than the original number? Which are the most important features?

6. OPEN-CHOICE TASK (0.5 POINTS)

- **Decide on your own some additional task,** either because it could improve results or because you find it particularly interesting to explore. Justify your selection**.**

WHAT TO HAND IN

- A **jupyter notebook** with the code in the proposed order of steps. Please **use some of the cells to comment** about what you are doing and your results. In particular, emphasise your

conclusions after each step with short arguments based on your results. If it is more convenient, you can also hand in a file with Python code instead and a separate report.
If you decide to use any AI chatbot, **briefly explain in those commented cells what purpose you used it for and how you used it (for instance, you can quote the prompt and the output used, in case they are short. Otherwise, you can give a brief summary**).
Please **write the names of the components of your group at the beginning of the notebook**.

- A file containing your **final trained model** in an appropriate ML format (joblib).

- A .csv file containing **your final model's predictions (values of your model's predictions** in the competition set).