

Machine Learning

Programming Assignment II-b

Ujjwal Sharma, Athanasios Efthymiou and dr. Stevan Rudinac

The following assignment will test your understanding of topics covered in the first five weeks of the course. This assignment will count towards your grade and should be submitted through Canvas by **01.12.2022 at 08:59 (CET)**. You must submit this assignment in groups (as registered on Canvas). You can get at most 5 points for this assignment, which is 5% of your final grade.

Instructions

- Alongside the code for your experiments, you are also required to present a report summarizing the observations and results of each of the experiments. You can use text and graphs/plots (`matplotlib`) for these reports. You should place these report blocks within the Jupyter Notebook in separate text cells. Plots can be appropriately placed near the text explanation. Your final submission should be a single Jupyter Notebook with code and report blocks.
- While it is perfectly acceptable to brainstorm and discuss solutions with other colleagues, please do not copy code. We will check all submissions for code similarity with each other and with openly-available solutions on the web. Submissions with high similarity will be summarily rejected and no points will be awarded.
- Please ensure that all code blocks are functional before you finalize your submission. Points will NOT be awarded for exercises where code blocks are non-functional.

Submission

You can submit your solutions within a Jupyter Notebook (*.ipynb). To test the code we will use Anaconda Python (3.9). Please state the names and student ids of the authors at the top of the submitted file.

1 Data

For this assignment, you will continue to use the dataset provided with Programming Assignment II-A. If you are attempting this assignment without attempting part II-A, we advise you to go back and look at the previous assignment for more information on the data.

2 Models

The model structure introduced in Assignment II-A uses a **Pipeline** to wrap a *scaler* and a *classifier*. Since scaling is not needed for tree-based models, the resulting pipeline will end up containing only the classifier. Thus, for the tree-based models specified in this assignment, the usage of a pipeline is NOT required.

In this assignment, you are asked to use the `DecisionTreeClassifier` and `RandomForestClassifier` as the classifiers. With these models, you must perform the following experiments:

✉ u.sharma@uva.nl, a.efthymiou@uva.nl, s.rudinac@uva.nl

1. Initialize a **DecisionTreeClassifier** model and perform the following tasks:
 - (a) Fit the classifier on the data and report the model accuracy.
 - (b) Report all evaluation metrics listed in Section 3.
 - (c) Answer the question : Are these models resilient to overfitting when model hyperparameters have not been carefully selected? Supplement your explanation with suitable figures/tables if necessary. Please limit your explanation to a maximum of 50 words.
 - (d) Perform a grid search on the decision tree model parameters [**'max_depth'**, **'max_features'**] to evaluate the optimal values for these parameters. Report model hyperparameters for the best classifier model.

In less than 50 words, explain your observations and your assessment of the classifier. You can use a text box (i.e., Markdown Cell) in Jupyter to write down your analysis.

2. Initialize a **RandomForestClassifier** model and perform the following tasks:
 - (a) Fit the classifier on the data and report the model accuracy.
 - (b) Report all evaluation metrics listed in Section 3.
 - (c) Answer the following questions:
 - i. How are decision tree classifiers different from random forests on a structural level? (max. 50 words)
 - ii. Where would you choose decision trees over random forests and vice-versa? Demonstrate this using an appropriate example from your data. (max. 50 words)
 - iii. Is accuracy an appropriate evaluation metric for this classification task? Justify your answer in less than 20 words.
 - (d) Perform a grid search on the random forest model parameters [**'max_depth'**, **'max_features'**, **'n_estimators'**] to evaluate the optimal values for these parameters. Report model parameters for your best classifier model.

In no more than 50 words, explain your observations and your assessment of the classifier. You can use a text box (i.e., Markdown Cell) in Jupyter to write down your analysis.

3 Model Evaluation

For all models in Assignments II-A and II-B, you must report:

1. Accuracy
2. Macro and Micro-Averaged Precision and Recall
3. F1 Score

Additionally, present your observations on what these scores mean for the models under consideration.

4 Grading

Component	Points
Decision Trees + Analysis	2
Random Forests + Analysis	2
Evaluation Metrics	1