Universidade do Minho
Departamento de Informática
Mestrado [Integrado] em Engenharia Informática
Mestrado em Matemática e Computação

Dados e Aprendizagem Automática
1°/4° Ano, 1° Semestre
Ano letivo 2024/2025

Practical Exercise no. 5

Theme

Feature Engineering

Decision Tree Pruning and Hyperparameter Tuning with GridSearch

Exercise

Feature engineering in ML is the process of transforming raw data into meaningful features that improve model performance. It highlights important information to the algorithm, helping it to detect patterns and make more accurate predictions. Techniques include scaling, encoding, and creating new features from existing data.

Decision tree pruning is the process of trimming down a decision tree to reduce its complexity and prevent overfitting. Pruning is applied to prevent overfitting by removing unnecessary branches, making the model simpler and better at generalizing to new data. This improves accuracy, efficiency, and interpretability. It can be done by setting a maximum depth, a minimum samples per leaf, or using cost-complexity pruning.

Hyperparameter tuning with GridSearch is a method of optimizing a model's hyperparameters by systematically testing different combinations over a specified range. It ensures that the model is neither underfitting nor overfitting, maximizing its performance and generalization to new data. This process typically uses cross-validation to find the best set of hyperparameters for improved accuracy.

Tasks

The aim of this practical statement is for you to carry out a series of tasks that will help you to understand these concepts. With the Titanic dataset, it is intended to:

T1. Load the dataset using the *pandas.read_csv(...)* function;

T2. Apply feature engineering and exploratory data analysis procedures;

T3. Define the set of input and output variables of the model;

T4. Prepare and organise the set of case studies from the dataset into training and test data, using the *sklearn.model_selection.train_test_split* function;

T5. Train the decision tree classifier model (*sklearn.tree_DecisionTreeClassifier*) using the training data;

T6. Plot the resulting tree and evaluate the model using the classification report and the confusion matrix;

T7. Tune the model using GridSearchCV *(sklearn.model_selection.GridSearchCV)*. Test hyperparameters like as *criterion*, *max_depth* and *min_samples_leaf*;

T8. Evaluate the best developed model and perform the corresponding critical analysis;

T9. Apply the pruning technique to the first model. Try both pre- and post-pruning (use *max_depth* and *ccp*_alpha respectively);

T10. Repeat **T8** for the best resulting models of each pruning process;

T11. What conclusions have you drawn from the results observed in **T6**, **T8** and **T10**, what conclusions did you draw? In which situations does the model perform better? Which is the best model (set of parameters)? Which setting of the model should be improved?