School of Computer Science, University of St Andrews, CS5011, 2023-2024

CS5011 - Practical 2: Machine Learning

Deadline: 13th of March, 2024

Weighting: 33% of module mark

Please note that MMS is the definitive source for deadline and credit details. You are expected to have read and understood all the information in this specification and any accompanying documents at least a week before the deadline. You must contact the lecturer regarding any queries well in advance of the deadline.

---

1 Introduction

This practical makes use of the dataset of a competition provided by DataDriven [1]: `https://www.drivendata.org/competitions/7/pump-it-up-data-mining-the-water-table/`. The dataset consists of information (i.e., features) about water pumps across Tanzania and their status (`functional`, `functional needs repair`, or `non functional`). The main task is to build a machine learning model that can predict the status of a water pump based on their features. A full list of features is provided on the competition website. [2].

In this practical, we will evaluate different machine learning models on the dataset, and look into hyper-parameter optimisation of those models. The practical must be implemented in Python. We will use the `scikit-learn` library [3] for the implementation of all machine learning models. Additionally, the `pandas` [4] and `numpy` [5] can be used for data reading and preprocessing.

2 Part 1

Firstly, you need to register an account and download the data from the competition website [6]. The data consists of training data (input features and labels), and test data (input features only). You will build and evaluate various machine learning models using the provided training data.

There are various pre-processing steps that you may want to conduct before training your machine learning models, including:

- Dealing with categorical features. Note that some categorical features are with high cardinality.
- Dealing with missing values.
- Scaling numerical values.
- Dealing with datetime features.

[1] `https://www.drivendata.org/`
[2] `https://www.drivendata.org/competitions/7/pump-it-up-data-mining-the-water-table/page/25/`
[3] `https://scikit-learn.org/`
[4] `https://pandas.pydata.org/`
[5] `https://numpy.org/`
[6] `https://www.drivendata.org/competitions/7/pump-it-up-data-mining-the-water-table/data/`

For categorical features, we will consider three types of encoding provided by `scikit-learn`: `OneHotEncoder`, `OrdinalEncoder` and `TargetEncoder`. For numerical features, we will consider two options: (i) no scaling, and (ii) `StandardScaler`.

In this part, you will study the performance of five families of machine learning models: `LogisticRegression`, `RandomForestClassifier`, `GradientBoostingClassifier`, `HistGradientBoostingClassifier` and `MLPClassifier`. Note that each family of model may have specific requirements for data preprocessing. You would need to identify them based on the library documentation to make sure the training does not produce error messages.

To ensure a statistically sound comparison between different models, we recommend using 5-fold cross validation for evaluating each combination of data preprocessing and machine learning model. This can be done via applying the `KFold` function [7] of `scikit-learn` on the provided training data. Following the competition rules, we will use the classification accuracy as the performance metric.

Your task is to study the performance of different combinations of data preprocessing methods and machine learning models. Consider the following points in your report:

- Explain the various preprocessing steps you have done.
- Which design decisions (parameters of preprocessing methods, hyper-parameters of machine learning models) you have to make while implementing this part? How did you set them? To make it simple, you can start your analysis with the default setting, and investigate a few hyper-parameters of one or two machine learning models of your choice afterwards.
- Analyse the results of your experiments based on the provided training set (via cross-validation mechanism). Summarise the key insights and findings of your analysis.

Code submission: the submission must include a Python script named `part1.py` that works with the following syntax:

```
python part1.py <train-input-file> <train-labels-file> <test-input-file><numerical
    -preprocessing> <categorical-preprocessing> <model-type> <test-prediction-
    output-file>
```

where:

- `<train-input-file>`, `<train-labels-file>`, `<test-input-file>` are the paths to the `.csv` data files provided by the competition.
- `<numerical-preprocessing>` represents the type of scaling method for numerical features. Valid values include: `None` and `StandardScaler`.
- `<categorical-preprocessing>` represents the type of encoding for categorical features. Valid values include: `OneHotEncoder`, `OrdinalEncoder`, and `TargetEncoder`.
- `<model-type>` represents the model type. Valid values include: `LogisticRegression`, `RandomForestClassifier`, `GradientBoostingClassifier`, `HistGradientBoostingClassifier` and `MLPClassifier`.
- `<test-prediction-output-file>` consists of the predictions on the test dataset of the competition. This must follow the `.csv` submission format required by the competition. [8]

[7] `https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.KFold.html`

[8] `https://www.drivendata.org/competitions/7/pump-it-up-data-mining-the-water-table/page/25/`

The script should train a model with the corresponding choices of pre-processing method and machine learning model, print out the (cross-validation) classification accuracy score, and produce the output prediction file (on the test data) as required.

Note that you should not include any hyper-parameter optimisation and plotting functions in that script. Any investigation on hyper-parameter optimisation should be done in a separate script, and the best configuration should be used by `part1.py`.

3 Part 2

In this part, we will look into optimising the design choices and hyper-parameters of your models using an automated hyper-parameter optimisation (HPO) tool. There are various free and open-source HPO tools available. Two examples include SMAC3 [9] and Optuna [10].

This part is an open-ended question. You are free to choose the libraries and tools that you want to use, and it is up to you to define what the configuration space of your machine learning pipeline is. For example, you can pick one family of model and optimise its hyper-parameters, or you can include multiple families of models in your configuration space.

The report should include a clear explanation of your HPO setup and a good analysis of the results.

4 Submission

Your submission to MMS must include the following items in a single ZIP file. Submissions in any other format will be rejected.

- A report in PDF format.
- The Python scripts of your implementation.
- A `requirements.txt` file, listing all the libraries used by your implementation.

There is an **advisory limit of 10 pages for your report**. This includes all text, figures and tables. **Note that the limit is not a target**, please keep your report compact and precise by focusing on the important points. Whenever possible, try to use summarised plots/tables and statistical measures to make your analysis clear and easy to follow. Note that you can skip the Testing section for this practical.

In addition to the report and the Python scripts, you can include some additional files only if they are lightweight and are important to support your analysis. **Please do not include the competition data, your scikit-learn trained models or other unnecessary files**.

5 Assessment Criteria

Marking will follow the guidelines given in the school student handbook. Some guideline descriptors for this practical are given below:

- For a mark up to 13: the submission implements part 1 to a good standard, with reasonable attempt at the evaluation and analysis.
- For a mark up to 16: the submission implements part 1 excellently with a thorough evaluation and insightful analysis.
- For a mark up to 20: the submission implements both parts excellently with high quality analysis on the results.

[9] `https://github.com/automl/SMAC3`

[10] `https://optuna.org/`

6 Policies and Guidelines

Marking: See the standard mark descriptors in the School Student Handbook

`https://info.cs.st-andrews.ac.uk/student-handbook/learning-teaching/feedback.html#Mark_-Descriptors`

Lateness Penalty: The standard penalty for late submission applies :

`https://info.cs.st-andrews.ac.uk/student-handbook/learning-teaching/assessment.html#latenesspenalties`

Good Academic Practice: The University policy on Good Academic Practice applies:

`https://www.st-andrews.ac.uk/students/rules/academicpractice/`

Nguyen Dang, Alice Toniolo, Fahrurrozi Rahman

cs5011.staff@st-andrews.ac.uk

February 14, 2024