Coursework #1: Applied Artificial Intelligence

Overview

The goal of this coursework is to provide a realistic setting for machine learning application and develop critical thinking of each step of the application. It is part of the assignment to encode/pre-process the features, deal with potential missing data, identify potential models to solve the machine learning task, apply them appropriately and interpret the results.

Submission: Your submission should contain three components:

1. A short written report (**max 3 sides**), in pdf format, documenting what you have done.
2. A second pdf document containing figures. The figures must all be numbered and referred to from the report. Each figure must have a short caption explaining what it is.
3. A jupyter notebook with the code used, which the report should also refer to enabling the marker to find which code corresponds to which task if necessary (the code should be well commented). However, do not expect the code to be checked in depth. A well-written report should mean we don't have to check the code to understand what was done.

Dataset description

The dataset for the course work can be found in Moodle and in the UCI Machine Learning repository (https://doi.org/10.24432/C5230J). The dataset includes over 40 features representing patient and hospital outcomes. The aim is to predict which patients will need hospital readmission. Although there are three classes in the dataset ("<30" if the patient was readmitted in less than 30 days, ">30" if the patient was readmitted in more than 30 days, and "No" for no record of readmission) we ask you to simplify the problem to classify patients with no record of readmission versus patients with record of readmission (combing the classes labelled as "<30" and ">30").

Note

This is an individual coursework, and we will investigate any submissions that seem to have been plagiarised.

Task 1: Dataset description (5%)

- Describe the characteristics of the dataset (data types, sample-to-features ratio, etc) and use descriptive statistics and figures/plots to show the dataset characteristics. (2.5%)
- Describe the challenges with the dataset (missing data, unbalanced classes, etc). (2.5%)

Task 2: Data assembling and initial pre-processing (15%)

- Assemble a dataset consisting of features and labels (e.g. X and y). **You can create a balanced dataset and use a smaller subset of the data to decrease the computation**

load (you can make a choice about the subset size depending on your computational resources). Describe the procedure used for assembling the data. (2.5%)

- Apply the pre-processing steps you consider necessary at this stage. Describe which data cleaning and pre-processing steps are needed (dropping features, encoding features, data imputation, etc). Keep in mind that some pre-processing steps need to be embedded in the cross-validation framework to avoid data leaking. (10%)
- Discuss which strategy could be used to better encode the diagnoses features (diag_1, diag_2, diag_3). (2.5%)

Task 3: Design and build a machine learning pipeline (40%)

Based on the machine learning task and dataset characteristics:

- Describe a set of metrics chosen to quantify the models' performance. Justify your choices. (2.5%)
- Use a linear Support Vector Machine (SVM) as a baseline model and choose three additional models to compare. Justify your choices. (2.5%)
- Implement a cross-validation (CV) pipeline to optimize the models' hyperparameters considering the dataset properties (e.g. potential need for stratified CV) and measure the models' performance on a test set. Include any pre-processing steps you consider necessary at this stage. Describe the cross-validation pipeline in a way that would enable someone to implement/reproduce it. (15%)
- Create plots to show for each model how the performance varies as function of the hyper-parameter values and describe what you observed in the plots for the different models. (10%)
- Create a table or plot to show **the mean cross-validation performance with standard deviation** (e.g. table with mean and std for the different metrics or violin plots for the different metrics) as well as the **test performance** of the different models. (5%)
- Briefly describe the results. (5%)

Task 4: Model Interpretation (10%)

- Create figures/plots to show the relative importance of different features for the four models. You can plot the models' coefficients (for linear models) or the feature importance (for tree-based models). Alternatively, use other potential strategies to identify relevant features to the model. (5%)
- Describe the similarities and differences of the features identified by the different models. (2.5%)
- Discuss if the features identified by the different models make sense (or not) with respect to the task. (2.5%)

Task 5: Alternative machine learning pipeline (20%)

- Implement an alternative machine learning pipeline that optimizes the models' hyperparameters and provides **multiple estimation of test performance**. Describe the pipeline in a way that would enable someone to implement/reproduce it. (10%)
- Apply the alternative pipeline to the same models as in previous tasks and a create table or plot to show the **mean test performance with standard deviation** (e.g. table with mean and std for the different metrics or violin plots for the different metrics). (5%)

- Briefly describe the results. (5%)

Task 6: Identify limitations and propose potential solutions (10%)

- Identify and describe potential limitations with the dataset that could limit the models' performance in a real application. (2.5%)
- Discuss what could be improved in the dataset to improve the models' performance. (2.5%)
- Identify and describe limitations with the pipelines and models implemented. (2.5%)
- Propose potential solutions for the pipelines and models' limitations (e.g. alternative models or pipelines for the considered machine learning task or new methodological developments). (2.5%)