CS3033 - CS6405 - Data Mining

Second Assignment

Lecturer: Dr Andrea Visentin

TA: Andrea Balogh

Submission

This assignment is due on the 07 of April at 23:59. You should submit a single .ipnyb file with your python code and analysis electronically via Canvas.

The deadline to participate in the competition is the 07 of April at 08:00. You should submit a single .ipnyb file with your python code and analysis electronically via Canvas. Participating in the competition won't grant you additional points, but you might win a nice prize.

Please note that this assignment will account for **25 Marks** of your module grade.

Declaration

By submitting this assignment. I agree to the following:

"I have read and understand the UCC academic policy on plagiarism, and agree to the requirements set out thereby in relation to plagiarism and referencing. I confirm that I have referenced and acknowledged properly all sources used in the preparation of this assignment.

I declare that this assignment is entirely my own work based on my personal study. I further declare that I have not engaged the services of another to either assist me in, or complete this assignment"

Marking Scheme

Program Correctness: Your program should work correctly on all inputs (including new datasets). Also if there are any specifications about how the program should be written, or how the output should appear, those specifications should be followed.

Readability: Variables functions should have meaningful names. Code should be organised into functions/methods where appropriate.
There should be an appropriate amount of white space so that the code is readable, and the indentation should be consistent.

Documentation: your code and functions/methods should be appropriately commented. However, not every line should be commented because that makes your code overly busy. Think carefully about where comments are added.

Code Elegance: There are many ways to write the same functionality into your code, and some of them are needlessly slow or complicated. For example, if you are repeating the same code, it should be inside creating a new method/function or for loop.

Code efficiency: The implementation is logically well designed without inappropriate design choices (e.g., unnecessary loops).

Objective

The Boolean satisfiability (SAT) problem consists in determining whether a Boolean formula F is satisfiable or not. F is represented by a pair (X, C), where X is a set of Boolean variables and C is a set of clauses in Conjunctive Normal Form (CNF). Each clause is a disjunction of literals (a variable or its negation). This problem is one of the most widely studied combinatorial problems in computer science. It is the classic NP-complete problem. Over the past number of decades, a significant amount of research work has focused on solving SAT problems with both complete and incomplete solvers.

One of the most successful approaches is an algorithm portfolio, where a solver is selected among a set of candidates depending on the problem type. Your task is to create a classifier that takes as input the SAT instance's features and identifies the class.

In this project, we represent SAT problems with a vector of 327 features with general information about the problem, e.g., number of variables, number of clauses, the fraction of horn clauses in the problem, etc. There is no need to understand the features to be able to complete the assignment.

The original dataset is available at:

https://raw.githubusercontent.com/andvise/DataAnalyticsDatasets/main/train_dataset.csv

Use the following template:

https://colab.research.google.com/drive/1_AX2wTRe6RoCX36HvP0UKSb13X3Gr77W?usp=sharing

Tasks

Basic models and evaluation (5 Marks)

Using Scikit-learn, train a decision tree classifier using 70% of the dataset from training and 30% for testing. For this part of the project, we are not interested in optimising the parameters; we just want to get an idea of the dataset.

Robust evaluation (10 Marks)

In this section, we are interested in more rigorous techniques by implementing more sophisticated methods, for instance:

- Hold-out and cross-validation.
- Hyper-parameter tuning.
- Feature reduction.
- Feature normalisation.

Your report should provide concrete information about your reasoning; everything should be well-explained.

The key to getting good marks is to show that you evaluated different methods and that you correctly selected the configuration.

New classifier (10 Marks)

Replicate the previous task for a classifier that we did not cover in class. So different from K-NN and decision trees. Briefly describe your choice.

Try to create the best model for the given dataset.

Save your best model into your GitHub. And create a single code cell that loads it and evaluates it on the following test dataset:

https://github.com/andvise/DataAnalyticsDatasets/blob/main/test_dataset.csv

This link currently contains a sample of the training set. The real test set will be released after the submission. I should be able to run the code cell independently and load all the libraries you need as well.