AM41ML Coursework - 2023/2024

Hand-in date: December 4^{th} 2023.

Deadline: January 8^{th} 2024.

1. Marking Scheme

This assignment counts 100% towards your assessment for AM41ML. This assignment is an **individual assignment**, no cooperation in any form, eg. no sharing of files, is permitted.

You will analyse the dataset "**Coursework_dataset.csv**" available from the course's Blackboard. It consists of 5000 entries and 4 columns. Your **goal is to build a binary classifier** to make predictions of the label on test data.

You will produce a **typed report** in **pdf** format describing the analyses you carry out, the results of your analyses, and discussion and interpretation of the results. The report should be structured by following the "**Coursework Guide and Questions**" section below; **feel free to expand upon this in any direction that you wish**. As usual, your code, plots, reasoning and interpretations should be coherently and clearly presented. You may present some of your code as part of your report if deemed relevant, and your entire code should be sent as an additional file, separately.
The report (including text, relevant code and figures) should not exceed 20 A4 pages (font size 10 to 12). Marks will be given for technical content, implementation and written presentation. See the following breakdown of marks in the different categories.

Technical content (40%)	Choice of appropriate methods, implementation.
Interpretation (40%)	Interpretation of the methods used and the results.
Presentation (20%)	Overall clarity and coherence.

The marks displayed in the left margin add up to 80, covering the 40% +40% of the first 2 criteria of the marking scheme. The remaining 20 marks to obtain relate to the overall clarity and coherence of the report. It is recommended to use either Jupyter Notebooks or LaTeX for the write up, similar to what was done during the Labs. Please use titles and subtitles when appropriate to separate sections, articulating your approach, showing relevant code and figures, and interpreting your findings. You must explain each step of the process such that each part of the report links to the next one.

2. Coursework Guide and Questions

Let us assume that you are a data scientist. Your boss comes to you with a dataset "**Coursework_dataset.csv"** (available on the course's Blackboard) and has the following request:"**I want you to build a model that will make less than** 5% **errors in predicting labels on new data**".

Your job is now to try and meet this request and give a report of your research and findings. Remember that you are giving this report back to your boss, who wants details as well as clarity; you have to show that you have done the best that you could, given the data provided, and that every decision taken is justified. Your report should include a coherent structure (logical steps that motivate your research), useful plots and interpretations, and made easy to read and understand.

To help you in your task, a Senior Data Scientist is giving you a guide and a set of questions that you are expected to address to help you structure your report.

[5 marks] 1. **Data Cleaning**:

The first step of any data science or machine learning project is to make sure that the data that you will feed to your model is appropriate ("Garbage in, garbage out").
Clean your data by taking care of missing values, useless columns and outliers, as well as checking that numerical features do not contain any other type of data (eg, strings). Do not forget to set your labels as factors if required.

[5 marks] 2. **Data Exploration and Visualisation**.

Now that your data is cleaned up, a good idea is to visualise it in order to find possible patterns that you will want your model to learn. This step will also help you in spotting any outliers. Show a plot of the data, where each class is represented by a different colour.

[60 marks] 3. **Model Selection**:

With a better understanding of the data, you are able to think about which algorithm might lead you to a good model. The following points are to be implemented.

(a) Split your data into a training set and a test set of appropriate sizes. Remember that the test set cannot be used for training, it is only used at the very end to evaluate how well your model generalises to new data. [5 marks]

(b) Run a SVM algorithm on the training set with a linear kernel and default values for the hyperparameters (C (or λ) = 1).

Apply this model to make predictions on the training data itself and show the plot of the data and the decision boundary given by this model.

Relate the training error (or accuracy) for this model to how the decision boundary looks. Which type of regime do we have here (high-variance? high-bias?) and how could you decrease it? [20 marks]

(c) Repeat the previous step, this time with a RBF kernel and 3 different values of γ: 1, 10, and 1000 (with C (or λ) fixed to 1). Explain what is happening in terms of bias and variance as γ increases (you may wish to illustrate your explanations with relevant plots). [15 marks]

(d) To get a better model, we need to find the hyperparameters that will minimise the validation error. With a RBF kernel, there are 2 such hyperparameters. The γ parameter as seen previously, as well as the degree of regularisation (or cost C ($= \frac{1}{\lambda}$)).

Perform Cross-Validation (either making your own loop, or using any function) on both parameters (either via grid-search, or narrowing down γ and then the cost) to find the value that will minimise the validation error (or maximise the accuracy). Note that the computations at this stage might take a few minutes.

You may show plots such as "Validation error rate (or accuracy) vs γ" and "Validation error rate (or accuracy) vs cost", or a grid displaying accuracy (or error) for each (γ, C) pair, to justify your choice of γ and C for your final model. [20 marks]

[10 marks] 4. **Testing and Interpretations.**

Now that the model has been trained and fine-tuned, it is time to test it (only once! Don't use the test set to tune your model as you might overfit on the test set itself and make your model less generalisable).

Apply your final model to the test data and report your results (plots, error rates, etc).

Did you manage to get the test error below the threshold defined by your boss? If not, how would you explain it? Could anything be done to decrease the test error further?