INST0060: Group Project Description
Multiclass Classification and Multicluster binary classification

Set by: Luke Dickens

November 16, 2022

Important Notice

This assessment forms part of your degree assessment. All work must be done by group members. In particular, **you must not:**

- collaborate or work with students outside of your group.
- send or show other groups your work.
- ask students outside of your group for help, or ask to see their work. As well as being against regulations, this is unfair to the other student concerned, since it may lead to them being accused of plagiarism.
- seek help from friends, relatives or others, other than the moodle forum for this module.
- elicit answers or help from paid or unpaid third parties, including online discussion groups.

If you think any of the description of the task below is ambiguous or unclear, please post to the moodle forum, explaining what your concerns are, or raise it in person with your module leader, Luke Dickens, or a TA. If you are unsure of any of the above points, please post your concern to the moodle forum.

Finally, if there is any reason you think you cannot contribute fully to the group work in the alloted time, you should discuss your reasons with the module coordinator, Luke Dickens at `l.dickens@ucl.ac.uk` at the earliest opportunity.

1 Task Description

For this assessment, your task is to carry out an experiment with multiclass classification or using clustering to partition a binary dataset prior to performing binary classification on clusters. There are three different pathways, called here research templates, and these are described in Section 1.1. All research templates require you to choose **one** classification dataset. This may be a multiclass dataset (research templates 1 & 2) or a binary classification dataset (research template 3). Your first job is to decide on the research template and associated dataset. These two choices are interdependent and should be decided within **one week** of the release of this document. To facilitate this, **you are strongly encouraged to decide as a group on a shortlist** of three such choices before or during the first lab session following the release of this document, and discuss these choices with your teaching team

during that lab session. In exceptional cases, you may be asked to finalise your decision after the lab session. Further advice on the choice of research template can be found in Section 1.1 and on the choice of dataset can be found in Section 1.2. **You do not have time to do everything, or read up on everything, so be selective.**

This project consists of three assessed components: a group report (group submission), an accompanying code archive (group submission) and an individual reflective report (individual submission). The guidance on these three parts is provided in Sections 2, 3 and Section 4 respectively. The group report counts 35% towards your final grade of the module. The code counts 15% to the final grade. The individual report counts 30% to your final grade and is due one week after the deadline for your group report and code submission.

On completion, each group should submit files with the following titles:

- `fomlads_group_report_<GROUP_ID>.pdf` containing a group report
- `fomlads_code_archive_<GROUP_ID>.zip` containing a zip archive of the associated code

Individuals should also submit a file titled

- `fomlads_individual_report_<GROUP_ID>_<STUDENT_ID>.pdf` containing their individual report

In the titles above <GROUP_ID> and <STUDENT_ID> should be replaced with your group id and student id respectively. For group submissions, only one person out of the group needs to submit. Every student needs to individually submit the individual report.

DO NOT include your names anywhere in the group report, code or individual report.

1.1 Research Templates

You must choose one of the following three research templates to guide your investigation. The research template will help you to decide what your research questions or objectives are and whether you have addressed them.

RT1: Model/algorithm comparison

Machine Learning practitioners often have many algorithms at their disposal to solve a particular problem. However, it is not always clear a priori, which model/algorithm works the best for the problem at hand. This research template involves choosing **one multiclass classification dataset** with **5 or more classes**, **one data representation** for that dataset and **one evaluation metric**. Your objective is to evaluate a number of models to see which performs best and understand a little about why. For this research template you should use between 3 and 4 models (not more), including **at least one lab model**, and **at least one external model**.

You will need to briefly explain the principles of any external classification model you use in your report. This is not a tutorial explanation, but a high level description that gives the intuition behind what is happening and the characteristic outcomes of the approach, e.g. under what conditions it works well and under what conditions it fails.

RT2: Dimensionality Reduction

One of the most vital ingredients to a machine learning algorithm is the choice of representation. In this research template, you are going to explore how dimensionality reduction techniques can be used

to provide information rich representations of your data. This research template involves choosing **one multiclass classification dataset** with **5 or more classes**, **one data representation** for that dataset, **one evaluation metric** and **one** dimensionality reduction approach from the following list:

- Principle Component Analysis (PCA) – see [Mur12] for a fairly clear description. You may use the SciKit learn model implementation `sklearn.decomposition.PCA` but only the `fit` and `fit_transform` functions.
- T-distributed Stochastic Neighbor Embedding (t-SNE) – see [MH08]. You may use the SciKit learn model implementation `sklearn.manifold.TSNE` but only the `fit` and `fit_transform` functions.

Your task will be to explore how the number of dimensions of your given reduced dimensionality representation affects the performance of the final model. It makes sense to choose a dataset with a relatively high dimension, e.g. more than 20, but avoid data with very high dimension, e.g. more than 200-300 features. This number will be reduced by your dimensionality reduction step, but you may need more data to effectively reduce dimensionality the more features you have. You can, if you choose, start with a lower dimensionality dataset and upscale it with a feature mapping, e.g. a large number of RBFs, then apply dimensionality reduction to that feature vector, but you should justify such a choice. You must choose exactly **one** classification model. You may use a lab model, a kNN model, Lasso, ElasticNet or a SVM with a linear kernel. **You must not use Random Forest, SVM with a non-linear kernel or MLP**.

You will need to briefly explain the principles of any dimensionality reduction method you use in your report. This is not a tutorial explanation, but a high level description that gives the intuition behind what is happening and the characteristic outcomes of the approach, e.g. under what conditions it works well and under what conditions it fails.

RT3: Multicluster linear classification

In this research template, you should explore the use of clustering to partition your data, so that separate linear models can be trained on each cluster. The argument here is that some classification datasets may best be modelled with a non-linear classifier, but this can be approximated to a collection of linear classifiers applied to different regions (and that these regions can be discovered with a clustering algorithm). This research template involves choosing **one dataset** with a binary target, **one data representation**, **one evaluation metric** and **one clustering approach** from the following list:

- K-means, see the lecture notes. You may use the SciKit implementation of `sklearn.cluster.KMeans`
- Agglomerative Clustering, see the lecture notes. You may use the SciKit implementation of `sklearn.cluster.AgglomerativeClustering`, but you will need to cut the hierarchy to give a flat clustering.
- DBSCAN, see [EKXM96] or [SSE+17]. You may use the SciKit implementation of `sklearn.cluster.DBSCAN`

You can choose to explore different numbers of clusters, but you need to decide whether you will choose best number of clusters based on some clustering metric, or on your final classification performance or to investigate whether good clustering performance corresponds to good classification performance. If you wish to use a clustering metric, you may use the SciKit implementation of **one** from the following list:

- sklearn.metrics.calinski_harabasz_score
- sklearn.metrics.silhouette_score
- sklearn.metrics.davies_bouldin_score

Descriptions of these metrics and appropriate references can be found at the linked page.

You will need to briefly explain the principles of any clustering method and any clustering metric you use in your report. This is not a tutorial explanation but a high level description that gives the intuition behind what is happening and the characteristic outcomes of the approach, e.g. under what conditions it works well and under what conditions it fails.

1.2 Dataset

In conjunction with your research template, you should also choose a dataset. The constraints on your dataset will depend partly on your research template. This choice should be made in consultation with the teaching staff. Make sure they are happy with your choice before you proceed.

1.2.1 Choosing a dataset

You must choose a **one dataset** and this requires you to search out an appropriate dataset online. Good sources for datasets are from Kaggle (`https://www.kaggle.com/datasets`) or UCI (`https://archive.ics.uci.edu/ml/index.php`). The dataset should ideally have no more than 100000 datapoints, but you can always take a larger dataset and then take a subset for your experiments. However, you will need to make this subset available to the markers, so take the subset first save it down to file, work with that file, and submit that file as part of your code submission. Your final code should run on a desk computer (CPU only) in less than 15 minutes to obtain all the necessary results presented in your report from scratch.

1.2.2 Representations and basis functions

You must decide on how you are going to represent your datapoint inputs. This partly depends on what your raw data is like, let's call the ith datapoint $\mathbf{x}_i$, and partly on whether you are using a feature mapping $\boldsymbol{\phi}(.)$. As stated, $\boldsymbol{\phi}$ is a feature mapping takes raw inputs $\mathbf{x}_i \in \mathbb{R}^D$ and produces a feature vector $\boldsymbol{\phi}(\mathbf{x}_i) \in \mathbb{R}^M$ (M may be larger than D).

Your raw data inputs can themselves involve some work to define and might involve processing of some kind. In particular, your data may have an existing data representation which needs to be transformed in some way before it will work well with your model. For instance, if you have a movie data-set, then each movie may have columns to indicate: the genre, the director, the production company, the nationality of the production company, the box-office turnover, the year, a column for cast members and so on. Some columns may be real (box-office turnover), some may be integer (e.g. year), some may be categorical with a small number of categories (e.g. language) or with a large number of categories, and some may be sets of entities (e.g. genre, cast members). Some of these will not work well as features of your model. You may also want to do a little additional work to improve an already usable representation further. Here are just some ideas:

- For numerical features, it is often desirable to have the feature normally distributed or at least not strongly skewed. To see if this is the case for feature j then you can histogram the jth column

of the original (untransformed) data-matrix. If the histogram is significantly skewed then simple mathematical transformations can help. Histogramming the jth column of the transformed data-matrix will then help you to see if your processing has improved matters. For instance, box-office turnover for films, say column j, may contain a small number of very large values, with most values much smaller, and be better transformed by the log function, e.g. a new feature $\ln x_{ij}$ for each i.

- Non-binary categorical features may need to be represented as one-hot-vectors. This allows different classes to be associated with different weights in your model. For instance, for a film rating dataset if the jth column of the data-matrix refers to nationality of the production country and can take only one of a small number of values, say `"US"`, `"UK"`, or `"China"` then you could replace this single feature with a three column subvectors $(1,0,0)^T$, $(0,1,0)^T$ and $(0,0,1)^T$ respectively.

- "Set-of" features are features that can take one or more of a small number of discrete items. One good representation for this is a binary vector with ones indicating the presence of a particular item. For instance, in a film rating dataset the jth feature could correspond to genre and may contain one or more of the strings: `"Thriller"`, `"Action"`, `"Comedy"` or `"Romance"`. Thus the jth feature could be replaced by a binary subvector of length 4 indicating which genres are identified, and a film indicated as `"Thriller, Comedy"` would be represented by sub-vector $(1,0,1,0)^T$. Alternatively, you may choose to have a single category for every unique set of features, and then represent this as a one hot vector over unique sets.

- Some features may be categorical or set-of but will contain many different categories. You may choose to group these based on your understanding of the feature. For instance, if you have a film dataset where the jth feature is production company, then you may want to group smaller production companies all under the same category, e.g. `Other`. To do this, you can count the frequency of each category in the jth column, and put any production company with fewer than K entries into the `Other` category.

- You may want to use text descriptions of each item, i, to get it's raw input, $\tilde{\boldsymbol{x}}_i$. For instance, the raw data matrix $\tilde{\boldsymbol{X}} = (\tilde{\boldsymbol{x}}_1, \ldots, \tilde{\boldsymbol{x}}_N)^T$, may be a unigram matrix based on these descriptions, e.g. if items are books and each book comes with a text synopsis, then $\tilde{\boldsymbol{x}}_i$ could be the unigram representation of a synopsis of the ith book.

- You may want to combine different representations, e.g. categorical features with a text derived unigram representation. This can be achieved by concatenating the two representations. More formally, if you have two data matrices $\tilde{\boldsymbol{X}}_1 \in \mathbb{R}^{N \times D_1}$ and $\tilde{\boldsymbol{X}}_2 \in \mathbb{R}^{N \times D_1}$ then concatenating these along each row would result in new matrix $\tilde{\boldsymbol{X}} = (\tilde{\boldsymbol{X}}_1 \; \tilde{\boldsymbol{X}}_2) \in \mathbb{R}^{N \times (D_1 + D_2)}$.

- Finally, you may have a data-matrix which is now very wide (dimension D is large), and/or with a large number of zeros. Such data-matrices mean that each datapoint is represented by a long vector and/or each weight parameter plays only a very smaller part in the prediction. This in turn means your model has very high variance (in a bias-variance sense) and this can lead to poor fitting of your models. Regularisation can help this, but another common preprocessing step is to run dimensionality reduction on your data. If you wish to do this, then you should commit to research template 2.

1.3 Model/Algorithm

This is something we have spent some time on in the lectures and labs. How to model a classifier on your data and how to fit the best parameters for that model. The **lab models** – those described in the lectures are:

- Fisher's linear discriminant
- Logistic Regression
- Shared Covariance Model

Although, we have implementations for these in the `fomlads` library, you may instead use the SciKit implementations of these models, but you may only use the fit, predict and predict_prob methods for these models.

Under some circumstances, you may also use one (or more) of the following **external models** (again this depends on your choice of research template – see Section 1.1). For these models, you are allowed to directly use the Sklearn API but consult Section 1.5 for the allowed functionality. The appropriate library function is indicated for each.

- Lasso regression – see here
- Elastic net – see here
- Random Forest Classifier – see here.
- Support Vector Machine (SVM) – see here.

  Treat SVMs with different kernel types as different models. In particular, a SVM with a linear kernel is a linear model (it fits a linear decision boundary). A SVM with a gaussian kernel is non-linear. Polynomial kernels can be slow and perform less well than gaussian kernels.
- K-Nearest Neighbours (kNN) – see here.
- Multilayer Perceptron (MLP) – see here.

  You can use at most one hidden layer.

When using either **lab models** or **external models** some effort to select best performing hyperparameters is expected. This can include **model hyperparameters**, which influence the family of functions fit (such as RBF location and width, random forest tree depth or branching factor, MLP activation function or number of hidden units) as well as **training hyperparameters** (such as gradient stepsize or regularisation parameter). Note that some models have a high number of model hyperparameters and you may not be able to evaluate everything. A general rule of thumb is that at most 3 hyperparameters per model should be investigated.

If using **external models** then some effort should be made in the report to demonstrate an understanding of the model, and the influence of each model hyperparameter on the model (e.g. in terms of flexibility of function and/or variance/bias). For each of the models above, there are suggested references on the associated SciKit learn documentation page which you can read. You should only read up on the models you use.

1.4 Evaluation Metrics and other investigations

One of the most important decisions, you will make as part of this project is the choice of classification metric for evaluation. If you are performing multiclass classification (research templates 1 and 2) or you have some class bias, then you should take extra care. A good overview of metrics for classification can be found in [BBC+00]. If you want more detail, then I give some additional references for individual choices below, and you can always find references on the SciKit learn documentation pages (but be selective about what you read). You must choose **exactly one** of the following classifcation metrics to evaluate headline performance, e.g. which model gives better classification performance (research template 1), which dimensionality leads to better classification performance (research template 2) or which number of clusters leads to better classification performance (research template 3):

- Accuracy – you may use SciKit implementation `sklearn.metrics.accuracy_score`
- The F1 score for binary labels, or macro average F1 for multiclass classification – you may use SciKit implementation `sklearn.metrics.f1_score`
- Area under the roc (sensitivity-specificity) curve or area under the precision-recall curve – you may use SciKit implementation `sklearn.metrics.auc`
- Pearson correlation coefficient, also known as Matthews correlation coefficient (MCC), see [Gor04] for more detail – you may use SciKit implementation `sklearn.metrics.matthews_corrcoef`
- Cohen's Kappa, although some advise against this [DT19] – you may use SciKit implementation `sklearn.metrics.cohen_kappa_score`
- Cross-Entropy loss, also known as log loss or logistic loss – you may use SciKit implementation `sklearn.metrics.log_loss`

Note that you can only choose one of the above methods to select among your different models. You will need to justify which of these methods you use.

1.4.1 Other notes on evaluation and hyperparameter/model selection

Your evaluation will be judged on how well you apply the concepts and methods in the module too. This includes appropriate partition of data into training, validation and test sets. You will need to be careful to apply dimensionality reduction (research template 2) and clustering (research template 3) consistently to training, validation and test sets (you should not refit these algorithms separately for each partition). The best projects will evaluate models and/or hyperparameters with multiple trials and report confidence intervals for their validation results. Note that one way to achieve this is to use cross-validation. However, **you may not use the SciKit functionality for cross-validation or hyperparameter selection**. See Section 1.5 for a full list of allowed functionality.

If you wish to inspect your class bias qualitatively with a confusion matrix, ROC curve or precision-recall curve, then you may do so. However, this should be done **after you have performed model selection** with your chosen evaluation metric. You can use at most one of these three approaches and you should justify it in the report. To implement it, you may use the SciKit functionality for this listed in Section 1.5.

1.5 Allowed functionality from SciKit Learn

Although, you have been working with the `fomlads` library for most of this module, for this project you can explore the use of some SciKit learn functionality too. However, you are only allowed to use certain functionality, not all of it. A list of allowed functionality from Scikit learn appears below. If it is not included on that list, then it is not allowed and you will lose marks on the assessment as a result. You may not need to use all this functionality though, so be selective. Allowed functionality:

- The class constructor for prediction models listed in Section 1.3, as well as their methods `fit`, `predict` and `score` methods.
- If you are following research template 2, then you may use the class constructor for dimensionality reduction models listed in Section 1.1, as well as methods `fit`, `fit_transform`, `transform` and `score` methods.
- If you are following research template 3, then you may use the class constructor for clustering models listed in Section 1.1, as well as methods `fit`, `fit_predict`, `fit_transform`, `predict`, `transform` and `score` methods.
- One of these classification metrics:
  - `sklearn.metrics.accuracy_score`
  - `sklearn.metrics.f1_score`
  - `sklearn.metrics.auc`
  - `sklearn.metrics.matthews_corrcoef`
  - `sklearn.metrics.cohen_kappa_score`
  - `sklearn.metrics.log_loss`
- If you are following research template 3, one of these clustering metrics:
  - `sklearn.metrics.calinski_harabasz_score`
  - `sklearn.metrics.silhouette_score`
  - `sklearn.metrics.davies_bouldin_score`
- For qualitative assessment of a model, you may use `sklearn.metrics.confusion_matrix`, `sklearn.metrics.ConfusionMatrixDisplay`, `sklearn.metrics.roc_curve` or `sklearn.metrics.plot_roc` `sklearn.metrics.precision_recall_curve` or `sklearn.metrics.plot_precision_recall_curve`

Any methods or classes that are not on the above list, are forbidden and marks will be deducted for their use. In particular, you may not use any functionality from the module `sklearn.model_selection` including `train_test_split`, `cross_validate`, `learning_curve`, `GridSearchCV` and so on.

2 Group Report

The assessment's main output is a technical report motivating and describing the experiments you have performed in a format similar to what you would see in a machine learning or data science workshop or conference proceedings. The report should be **no longer than 6 A4 pages**. References count

in this page limit. No Appendix is allowed. You should use the IEEE conference standard **double column** paper format with font size 10pt (you can find word or LaTeXtemplates at: `https://www.ieee.org/conferences/publishing/templates.html`). We strongly encourage you to use LaTeXto do the formatting of your report and more specifically the Overleaf platform* which has been made free for UCL students. It has extensive collaborative features which enable you to work efficiently together as a group even remotely. You should use the IEEE referencing style provided to you in the template and it is very easy to manage your bibliography if you are using Overleaf (guidence here: `https://www.overleaf.com/learn/latex/bibtex_bibliography_styles`). You may lose some marks if you do not format your document with LaTeX. Marks will be based partly on how well you meet the discription as well as whether you have demonstrated a degree of creativity or novelty in doing so.

The report should contain an abstract and the following sections: Introduction, Background, Dataset and comparison-framework, Method, Results/Experiments/Evaluations and Discussion. Potential marks for each section as well as an overall marking criterion are as follows:

1. **Abstract and Introduction [10 marks]** The abstract should give a broad concise overview about what the domain and investigation is (without too much motivation). It should mention the domain, the general aims of the investigation, and something about the findings.

   The introduction should more fully motivate the project (say why it is interesting). It should explain again what has been done (method) at a high level and why (referring to a particular research template is good, but not essential). If relevant it should outline what the general measure of success is in terms of the domain. It should outline the achievements and findings: this may be interleaved with the method, but is better kept separate. Ideally the section should end with a brief outline of the document.

2. **Background [10 marks]** A brief description of the models/algorithms or data representations used in your experiments. For the methods described in lectures, you can write a few sentences of textual description and reference. You can add an equation of the objective function (e.g. the loss) that is being optimised or the type of basis functions you use, if you feel this is appropriate. For other classification models, performance metrics, validation approaches (e.g. cross-validation), model selection and other choices, you must describe the concept, and cite source materials appropriately. For classification models or data representations not explained in the lectures, this should include a description of the objective function or an equation. You do not need to explain in detail how models are fit, or give a tutorial explanation of how/why it works.

3. **Dataset and comparison framework [10 marks]** A brief description of the dataset you use and your choice of comparison framework. This should help to contextualise and justify the classification task and your approach to it. For this you should discuss such things as: who would be interested in the findings, how might the classifier be used in practice.

4. **Method [20 marks]** This is a key section. It should be made clear (ideally here) what the general research aims/questions are and why they are interesting. The description of your research template should help you to formulate this. Imagine that this document will be read by people who do not know it is an assessment, and have not read the assessment brief. Note you should not write: *we did research template X*. Instead, describe how you have interpreted your chosen research template as a plan of research. You must assume that your reader does not know what is described in this document.

*`https://www.overleaf.com/`

This section should describe and justify the following:

- What data representation(s) is (are) used. If some of your data-columns are discrete, you need to describe how you will represent these in your data-matrix.
- What feature mapping(s) is (are) used.
- What model(s) is (are) used. If this requires (these require) a model selection step to choose hyperparameters, you should say that here as well explaining as your general procedure for doing so.
- What your performance criterion is (namely what classification metric you use, e.g. accuracy, F1-score or AUROC). Choose one.
- Whether you will investigate your best performing model qualitatively.

Your research aims/objectives/questions will influence how the model is constructed, what experiments are included in the evaluation, and how things are measured. It is not enough to just describe what you have done (although this is important) you must also justify why this is interesting and how it relates to your research objective (as set out by your research template).

For instance, if using research template 2, then you should explain: what data representations you are comparing and why you have chosen these, how this relates to your chosen dataset and ideally which choices you anticipate will work most effectively and why. You should also describe your single choice of model and the **single** measure you are using to compare performance between representations (you can use other measures to interrogate performance of a single data representation).

5. **Results/Experiments/Evaluation [20 marks]** This section should describe the experimental evaluations conducted, and should include a clear description of each experimental set up and conditions. Moreover, you should report on the results/findings, plotting necessary graphs and interpreting and highlighting key results. If model selection is required then the report should illustrate this process, and ideally report (in a plot) the outcome of **at most one** parameter selection step. For reasons of space, you do not need to report model selection plots for all selected parameters. Describe the process first then say that other parameters were selected following the same procedure.

   Results should be presented such that they are readable, and should be interpreted in the text. Plots/tables which are poorly labelled or described, or not properly referred to in the body text, may lead to lower marks in this section. Ideally, confidence intervals should be given when metrics contain an element of randomness. Finally, you should remember to guide the reader towards the experimental outcomes and plot features that are most relevant, surprising or informative, then interpret them, e.g.

   > "Figure X shows the accuracy of the evaluated models (y-axis) for different sizes of training data (x-axis). Note that method Y (blue line) achieves the best accuracy (highest) overall, but method Z (red line) performs within 95% of this accuracy with half as many training points."

6. **Discussion [10 marks]** This should include a brief summary of what the report contains, with an emphasis on the overarching picture that emerges from the experiments. It should then include a

description of assumptions made, limitations associated with the chosen methods/model/evaluation and/or whether you can conclude anything more generally about the application of these methods more widely. Creativity can be demonstrated by pulling out a *general picture* from disparate strands of the evaluation. Future work should be included. Creativity can also be demonstrated by understanding non-core literature and relating it appropriately to the investigation.

7. **Overall [20 marks]** This assesses your report as a whole, including elements that may be present in more than one section.

   Presentation The report should be professionally presented. This includes consistent overall appearance, font type and font size, use of appropriate heading styles, consistent use of tenses, grammar, spelling and so on. LaTeXcould be a great help in this. Do not try to change the margins or the formatting of the template - we will see this.

   Narrative The report should tell a coherent story from beginning to end. If the report deviates from the recommended structure then this should be well judged.

   Structure Although the sections are given to you, you should use subsections and bullet-pointed lists appropriately. Sub-section names should be appropriate in length and relate well to the content, as well as sit well within the parent section. Lists should be used to improve readability, and should not negatively impact clarity or presentation.

   Equations These should be used when appropriate, properly typeset, and appropriately described. Ideally, key equations that are later referred to should be given a reference number and referred to by reference.

   Figures/tables A key element, figures should be clear and readable, with appropriate fontsize and your own work. They should be used proportionately. **Do not** include images created by others. **Do not** include screenshots of code or output. Tables should be clearly presented with appropriate fontsize. Captions should clearly and concisely describe the figure/table. Some main body text should refer to each figure or table and explain, describe or interpret more fully than the caption.

   References The report should cite other work appropriately and include a bibliography. Direct quotes should not be used without proper contextualising text. All words and ideas from others should be appropriately cited. Citation style should be consistent. Citations should be proportionate. Bibliography should be well typeset and contain essential information for citation type.

Quality not quantity is the key here. General principles for writing a report are expected to be known and adhered to. Similarly for practices in conducting experiments.

The group report makes up 35% of your total marks of this module.

3 Code

Groups should provide a zip archive of the code developed during the assessment. This may use all functionality from the libraries `numpy`, `scipy`, `pandas`, `argparse` and the `fomlads` code provided in the tutorial solutions.

You may use the `sklearn` library as described in Section 1.5. No other functionality from `sklearn` is allowed, including code for model selection and cross-validation. If you are unsure whether a specific functionality of `sklearn` is allowed, please contact module tutors or TAs. Marks will be deducted for inappropriate use of this functionality.

Code archives when unzipped, **must allow each experiment described in the report to be run independently**, but **all experiments must also be aggregated into a single `main` runner file** taking the location of a data-file as input. The easiest way to achieve this is to have a command line interface as shown in the solutions to tutorials, or via an `argparse`[†] library, which would allow us to choose a particular experiment to run. You must provide us with instructions how to use that interface and what commands to use for the different experiments. Therefore we expect the run to be performed with:

```
python main.py <DATA FILE> <COMMAND LINE OPTIONS/FLAGS>
```

and this should load data from `<DATA FILE>` and recreate all or particular results/figures for that experiment, then save these down to file. Any results not plotted in a figure, should be printed clearly to screen. Code archives should contain a succinct `README` file describing how each experiment should be run.

Here is a general breakdown of the marks for project code:

1. **Reproducability and Faithfulness to Criteria [10 marks]** We should be able to easily reproduce the results from your report including figures, plots and numerical results. Each experiment should ideally have independent function calls that can be easily executed. It should take us **no more than 15 minutes** to run through all your experiments on a **standard consumer desktop machine** (CPU only). **Experiments taking significantly more than 15 minutes will be penalised.** Code should be error free. If there is randomness in your experiment (e.g., sampling from a data distribution), you should provide the random seed you are using so we can recreate the exact same results as seen in the document. A good starting example is `https://github.com/pinga-lab/paper-template/blob/master/README.md`. If you use any external libraries other than `numpy`, `scipy`, `matplotlib`, `pandas` and `sklearn`, you must first approve this with the module leader and if approved you must include a dependency file either in text or yaml format which contains all the external libraries you use for running the experiments. Remember, you can only use the specified functionality of `sklearn`.

2. **Readability: Documentation and Code [10 marks]** You **must** provide a detailed `README` file explaining the structure of your code base and different functionalities of modules. The `README` file **must also** give clear instructions of how each experiment should be run and what are the expected behaviours (e.g., how long each experiment will take, which figures are output). Your code should be appropriately documented and commented so that the functionalities of the core modules are clear. You are not expected to strictly follow python programming style (i.e., PEP8), but we do expect you to write clean code with good variable and function naming.

3. **Reusability, Reuse and Attribution [10 marks]** You should consider if your code can be easily reused or even extended by another user. The overarching principle for having good reusablity involve modularising your code such that different functionalities are as independent as possible to one another. For example, if a new user plans to add new evaluation methods, they should not have to rewrite a lot of the model fitting code from scratch. Moreover, your code shouldn't contain many/any duplications of the same procedures (e.g. via copy-pasting). Instead you should define and reuse lower level functions. Moreover, if some user wish to explore different parameters of the model or basis functions, ideally they should be able to do so on the top level of your code

[†] `https://docs.python.org/3/library/argparse.html`

base. We also strongly recommend you to think about how to reuse the code that you wrote as part of the solutions to the exercises which has been grouped into the `fomlads` library.

Any use of code or ideas outside of this should be properly attributed. However, reuse of others code is unlikely to be appropriate, so please check with your module leader if you are unsure.

We encourage you to use collaborative version control platforms such as GitHub `https://github.com/` which help you to manage your work, collaborate and iterate across different versions of the codebase as you are adding content. You can create a free account and as students you can create public or private code repositories to store your code. A succinct tutorial on how to use version control based around GitHub can be found here `https://guides.github.com/activities/hello-world/`.

The code submission makes up 15% of your total marks for this module.

4 Individual Report

The Individual report deadline is one week after the submission of your group project and the code. Each individual student should submit a short report (min 1,000 words, max 1,200 words) which reflects on their experience of the group project. You will be penalised for exceeding the word limit.

You might want to use the same template format as the main report. You should reflect on what you did, how the group worked together to address the task and the challenges you faced. You can also discuss general concepts that you have understood more clearly or practical skills you have developed through this exercise. Be aware that you should keep things anonymous throughout, e.g. refer to team members by IDs rather than names.

Throughout all sections emphasis should be given on how you (and your group) tried to engage with all aspects of the project. So when talking about skills or challenges try to describe a diversity of aspects, e.g. not just implementation. Other aspects that you may want to talk about include: management, background reading, researching datasets or methodologies, planning, defining, designing evaluation frameworks, implementing code, presentation of results, analysis/interpretation of results and the different aspects of writing up (motivating, describing, explaining, formulating, interpreting, reflecting/discussing). You can't talk about everything, so choose two or three aspects for each part. Repeatedly referring to the same aspects through all sections will lead to fewer marks than if new things are brought to light in each section.

Here is a breakdown of the marks:

1. **Team working and task allocation [10 marks]** Describes who in the project was given what task, how you worked together as a group, and what tools or practices were employed. These choices should be justified, e.g. group breakdown in terms of appropriate skills/knowledge and fairness. It is reasonable to expect some assessment of how fairly/evenly the load was distributed in terms of effort. Students may describe how work was reallocated at a later date if a member met difficulties, did not attend group meetings or failed to produce expected work.

   Students should not just describe one aspect of the work but they should try to explain a broad coverage of the effort while working in a group. It is fine to define sub-groups that worked together on particular aspects, but you should also distinguish at some level between what each member did. For activities where you worked together as a whole group, or in teams, you could describe how you structured that activity, to ensure it was productive and that all members had an opportunity to participate. You can describe ways that you work together as a team, and technologies you used to facilitate that. Some marks will be reserved for explaining whether and how each student

was engaged with all stages of the project: management, planning, implementation, evaluation and writing up.

2. **Challenges [10 marks]** Describe 2 or 3 challenges faced when carrying out the project. In particular, students may refer to conceptual difficulties (understanding), implementational difficulties (making things work) and group/interpersonal difficulties (getting people to do what they should). Student should ideally describe the potential impact of a given issue, as well as what efforts were made to address these challenges, e.g. background reading, debugging sessions, changes in group work distribution or additional meetings.

   It is best to explain what is tried to address difficult challenges before explaining that an easier course of action was taken. For each challenge, try to include:

   - a description of the problem
   - how the problem was identified
   - what practices you first put in place to address things
   - whether these practices were effective straight away or whether you needed to evolve further
   - what lessons were learned for the future about how to address similar problems

   Here marks are given with respect to the above aspects as well as to the diversity of challenges described.

3. **Skills [10 marks]** Describe 2 or 3 new skills that have been developed during the project. This may include coding skills, statistical analysis skills, independent learning, ability to read the literature, group management, interpersonal skills and so on. Students should be specific about what they have learned, e.g. not just "I improved my coding" but "I better understood how to run python scripts from the command line and how to plot images with the matplotlib library".

   You could explain how you applied a particular skill within the project to help demonstrate your new mastery of that skill. It helps for you to describe in sufficient detail that you demonstrate your new competence, e.g. examples of what you now do that you did not before. You should also reflect briefly on what you might use these skills for in the future (why is this a useful skill?).

4. **Concepts [30 marks]** This section should describe concepts that you have understood more clearly, that are *evidenced in your group report or code* and in such a way that understanding is demonstrated. Therefore you **must describe** the concepts you have understood and demonstrate that understanding. This may include concepts taught directly in the course, or related concepts from the textbooks and wider literature. Note that this section carries a higher weighting than the other sections and should make up roughly half of your individual report. For example, you may want to discuss classification models, basis functions and feature vectors, optimisation, model selection, approximating samples with expectations, proper evaluation criteria, limitations of various algorithms or approaches or anything else you feel is relevant. However, particular emphasis should be placed on how these related to your specific project.

   Concepts should be **evidenced in group report or code**. It should be clear that the concepts being described played a role in: the formulation of the research objective, the design of the model, the evaluation procedure, the interpretation of the results, the analysis of the limitations or the proposed future work. You should describe this relationship to your group work and then perhaps

describe how these concepts have a more general applicability too. However, you should avoid long narrative descriptions of your personal experience of developing an understanding.

You must **demonstrate understanding**. Saying the names of concepts, ideas or methods is not enough. You should either describe a concept in your own words and its relevance to your project, compare different aspects identifying similarities and differences, hypothesise appropriately about consequences or in other ways communicate what has been learned in a way that goes beyond simply recalling. In other words, you should do one or more of the following: explain, analyse, compare, reason about, hypothesise, deconstruct, evaluate (qualitatively) or relate (to your project). Feel free to include images or equations to help make your point here.

The individual report makes up 30% of your total marks for this module.

References

[BBC+00] Pierre Baldi, Søren Brunak, Yves Chauvin, Claus A F Andersen, and Henrik Nielsen, *Assessing the accuracy of prediction algorithms for classification: an overview*, Bioinforma. Rev. **16** (2000), no. 5, 412–424.

[DT19] Rosario Delgado and Xavier-Andoni Tibau, *Why Cohen ' s Kappa should be avoided as performance measure in classification*, PLoS One (2019), 1–26.

[EKXM96] Martin Ester, Hans-peter Kriegel, Xiaowei Xu, and D Miinchen, *A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise*, KDD, 1996, pp. 226–231.

[Gor04] J Gorodkin, *Comparing two K -category assignments by a K -category correlation coefficient*, Comput. Biol. Chem. **28** (2004), 367–374.

[MH08] Laurens Van Der Maaten and Geoffrey Hinton, *Visualizing Data using t-SNE*, JMLR **9** (2008), 2579–2605.

[Mur12] Kevin P. Murphy, *Machine Learning: A Probabilistic Perspective*, MIT Press, 2012.

[SSE+17] Erich Schubert, Jorg Sander, Martin Ester, Hans-Peter Kriegel, and Xiaowei Xu, *DBSCAN Revisited , Revisited : Why and How You Should (Still) Use DBSCAN*, ACM Trans. Database Syst. **42** (2017), no. 3, 21.