2024 COMP309/AIML421 ML Tools and Techniques: Assignment 1

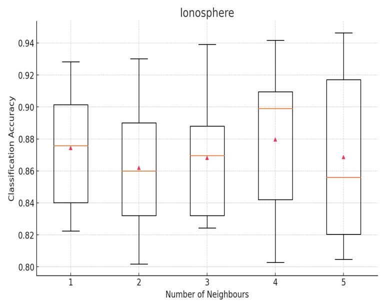
This assignment has 100 marks (120 marks for AIML 421 students) and is due on ***23:59, Friday, 2nd August 2024***. Please submit your report as a single *.pdf* file **including figures and tables as required** (This is crucial. Failing to do this will result in a loss of marks), PLUS your code as a separate file(a Jupyter notebook as a *.ipynb* file or *.py* file). Make sure you read the *Assessment* and *Submission* sections at the end of this file. This assignment contributes 20% to your overall course grade.

Objectives

This assignment involves trying out a variety of classification and clustering algorithms. It requires use of *python*, *numpy*, *matplotlib*, and *scikit-learn*, and serves as an introduction to all those tools. You can run python in *a jupyter notebook* but there are others option to do the development.

1 Classification [70 marks]

The part of assignment is to explore several classifiers in scikit-learn and investigate the hyperparameter for complexity control for each of these classifiers on three datasets by setting the hyperparameter to a range of plausible values and seeing how well it does on "held out" data. To do this you will need *train_test_split* from scikit-learn. To get better estimates, simply repeat *50 times* with *different random splits* (set the seed to get reproducible results). For simplicity, use a *50:50 train:test split* in all cases. For each setting of the hyperparameter, you then have a distribution over 50 different classification accuracies on the test set. A nice way to visualise these scores is to produce a box plot where the x-axis gives options for a parameter of the model, while the y-axis indicates the spread for classification accuracies of the classifier. *Titles and Axis Labels* are needed for clarity.



1.1 Machine Learning Models:

You will be trying out the following classifiers in scikit-learn:

(a) KNeighborsClassifier (K nearest neighbours)

(b) GaussianNB (the Gaussian form of Naive Bayes)

(c) DecisionTreeClassifier (A decision tree (DT))

(d) LogisticRegression (essentially, a perceptron)

(e) GradientBoostingClassifier (Gradient Boosted DTs)

(f) RandomForestClassifier (Random Forest)

(g) MLPClassifier (Neural Network)

1.2 Datasets:

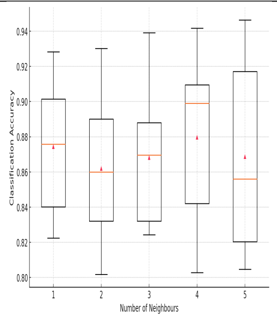
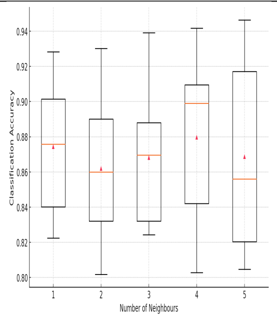
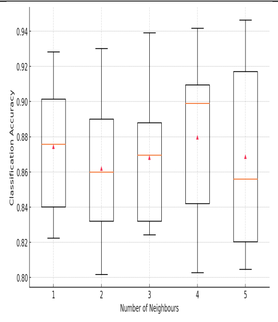
You will need to test the models on the following three datasets (you can easily download them from OpenML).

1. steel-plates-fault
2. ionosphere
3. banknotes

1.3 Tasks:

You should:

(i) Build a 7-by-3 (7 classifiers and 3 datasets) table to present the boxplots on the classifier accuracy versus parameter values (as the example table below). It probably won't fit into one summary figure. You should structure it as separate main plots, one per classifier, each consisting of several subplots for different datasets.

	steel-plates-fault	ionosphere	banknotes
KNN			
GNB	...	...	...
...	...	...	...

You need to consider **the parameter and the corresponding values** for each classifier as shown in Table i. Other hyperparameters should be left as default.

	Classifier	HyperParameter	Values
1	KNeighborsClassifier	”n_neighbors”	$[1, 2, 3, 4, 5]$
2	GaussianNB	”var_smoothing”	$[1e-9, 1e-5, 1e-1])$
3	LogisticRegression	”C”	$[0.1, 0.5, 1.0, 2.0, 5.0])$
4	DecisionTreeClassifier	”max_depth”	$[1, 3, 5, 8, 10]$
5	GradientBoostingClassifier	”max_depth”	$[1, 3, 5, 8, 10]$
6	RandomForestClassifier	”max_depth”	$[1, 3, 5, 8, 10]$
7	MLPClassifier	”alpha”	$[1e-5, 1e-3, 0.1, 10.0])$

Table 1: Parameters for 7 classifiers

(ii) Present *two* summary tables, with rows being classifiers, and columns being datasets. Table (1) is to contain the lowest mean value of the test errors obtained from each classifier with various hyperparameter settings.
Table (2) is to contain the corresponding hyperparameter values for obtaining the best test errors.

(iii) Write a paragraph to compare and analyse the overall results as captured in these two tables including which model has the best performance and why, and how sensitive these models are to the complexity control hyperparameter.

2 Clustering [30 marks]

The scikit-learn library provides a suite of different clustering algorithms to choose from, each offering a different approach to discovering the natural groups in data.

In this part of assignment, you will explore the characteristics of different clustering algorithms by testing them on three toy datasets. You will use the make_blobs() function, make_classification (with ”n_clusters_per_class=1”) and make_circles (with ”noise=0.3”) to create three toy clustering datasets. Each of them will have $1,000$ examples/instances, with *two* input features (for ”make_classification”, ”n_infomative=2”). Keep Other hyperparameters as default, and set the seed in the dataset generators to get reproducible results.

2.1 Machine Learning Models:

You will be trying out the following clustering algorithms in scikit-learn:

(a) K-Means

(b) Affinity Propagation

(c) DBSCAN

(d) Gaussian Mixture Model

(e) BIRCH

(f) Agglomerative Clustering

(g) Mean Shift

2.2 Tasks:

You should:

(i) Fit the clustering models on the datasets and predict a cluster for each example in the datasets. For clustering models that require specifying the number of clusters, use the insights obtained from the three data-generating functions. Build a 7-by-3 (7 clustering algorithms and 3 datasets) table to present the scatter plots showing the clusters generated by each algorithm.

(ii) Write a paragraph to compare and analyse the results of the clustering algorithms on the three datasets, highlight the characteristics of these algorithms.

3 Classification and Clustering (for AIML421 Students only) [20 marks]

The real world often throws up situations in which we have a mountain of unlabelled data but only relatively few examples with their labels. Ultimately this is because acquiring accurate labels is expensive, since it often requires human expertise, whereas acquiring unlabelled data is often automatic and cheap.

This part asks you to investigate whether it is possible to use features derived from unsupervised learning to enhance the predictions made by a supervised learner (in this case, a Classifier). To do this, act as most of the data is missing its label: you will use the iris data and split it with *95 percent* as a test set, and only *5 percent* is available for training. Set the seed for splitting to get reproducible results. You need to:

- Use k-means with k=3 as the cluster method, run on the whole dataset, and add the cluster label as a new column in X.
- For a hundred repeats:
  - Randomly split the iris dataset (**5:95 splitting** as above);
  - Train the *GaussianNB()* Classifier on the training data [a] *without* and [b] *with* the new feature, i.e. the new column derived from clustering;
  - Obtain the classification accuracy on the test set of [a] and [b].

3.1 Tasks:

You should then,

(i) Provide a scatterplot of the accuracy of [a] on the x-axis against the the accuracy of [b] on the y-axis. The Titles and Axis labels for the plots are needed for clarity.

(ii) Write a paragraph explaining whether the new feature derived from unsupervised learning can enhance the classification predictions and why.

Assessment

Format: You can use any font to write the report, with a minimum of single spacing and 11 point size (hand writing is not permitted unless with approval from the lecturers). Reports are expected to be 2-6 pages for COMP309 students and 2-8 pages for AIML421 students. Reports exceeding the maximum page limit will be penalised.

Communication: A key skill required of a scientist is the ability to communicate effectively. No matter the scientific merit of a report, if it is illegible, grammatically incorrect, mispunctuated, ambiguous, or contains misspellings, it is less effective and marks will be deducted.

Late Penalties: You have three automatic extension days, which can be applied to any assignments throughout the course (but not the final project). Please note that these three days are for the whole course, not for each assignment. No formal application is required; instead, any remaining late hours will be automatically deducted when submitting assignments after the due date. You have the flexibility to use only a portion of your late day and retain the remainder for future use. The penalty for assignments that are handed in late without prior arrangement (or use of ”late days”) is one grade reduction per day. Assignments that are more than one week late will not be marked.

Plagiarism: Plagiarism in programming (copying someone else's code) is just as serious as written plagiarism, and is treated accordingly. Make sure you explicitly write down where you got code from (and how much of it) if you use any other resources asides from the course material. Using excessive amounts of others' code may result in the loss of marks, but plagiarism could result in zero marks!

Submission

You are required to submit a single *.pdf* report PLUS the python code file (.ipynb or .py) through the web submission system from the COMP309/AIML421 course website *by the due time*. Provide a *README.txt* file if you use any non-standard python libraries.