University of Waterloo
ECE 657A: Data and Knowledge Modeling and Analysis
Winter 2023
Assignment 2 - Representation Learning,
Parameter Tuning and Classification Comparisons
Submission Due: March 10, 2023 by 11:59pm

Overview

Collaboration/Groups: You may do your work individually or with a partner. If you are working with a partner, you must sign up for an `Assignment Group` in LEARN which will provide you a dropbox to upload your assignment content to. In **Crowdmark** you also need to join a group with your partner, there are no numbers for groups, we will use names and student id's to match students and group submission. If you are working alone, you should still join a group in LEARN so that you have a dropbox to put your code. You can also collaborate with other classmates on the right tools to use and setting up your programming environment, but your submitted worked must be only from members of your group.

Submission: Hand in one report per person, or group, to Crowdmark and LEARN. Your report should be submitted as two files:

- *To your group dropbox on LEARN:* Your code in the form of a jupyter notebook that has the code, and results already generated on the provided data in a readable way.
- *To Crowdmark:* Your report in the form of a pdf that is *no more than about 10 pages*. You can use the jupyter notebook to generate this report document, but you need to **remove all the code and preliminary processing** from it. Only keep what is actually essential to explain what you did, why you did it and what your results show. It might be easier to copy the text, headings, plots and tables out into a separate document and save that.

Your group on LEARN has an associated **dropbox**.

Tools: You can use libraries available in python. You need to mention explicitly which libraries you are using, any blogs or papers you used to figure out how to carry out your calculations.

Specific objectives:

- Practice how to apply the methods discussed in class.
- Demonstrate understanding by making well reasoned design choices, and *explaining* any result you obtain in straightforward, short, text. Remember, there isn't always one particular right answer to how to do something, but you must justify what you did and demonstrate how it worked.
- Experiment with how to use different methods of feature extraction, parameter tuning, analysis and visualization to improve the performance of your model and your understanding of its results.

Presentation of Results (Descriptive, Concise, Clear)

- **Report document:** All tasks and questions below should be carried out or answered for both datasets. The report is a pdf document with your answers for the questions, description of your process, plots and tables for results. ***It does not contain code!***
- **Essential Plots and Tables:** Along the way, minimize the number of different plots shown to the *essentials* for the reader to understand what you did.
- **Summary Table:** At the very end you will produce a summary table of all the accuracy results for the different experiments.
- **Code:** You must submit your code, in a self-contained python jupyter notebook to LEARN. Your, you should collapse most of your code before printing to improve readability. You only need to show critical code which relates to the central task of the question or a point you are highlighting in your text.

Datasets

To keep things simple, for this assignment we will use the `Abalone` dataset from assignment 1 and another dataset called `Wine`. Since you've already preprocessed those and have classification results for kNN you can reuse that code and those results.

1 Representation Learning

You will apply PCA and LDA onto the dataset, analyse the resulting new representations in terms of interpretability and classifier impact, then create new reduced dimension datasets for use in later questions.

1. Run PCA on each dataset, look at the total variance explained by the principle components. At least, show a plot of the first two principle components using easily distinguishable colours and markers to indicate the labels of each datapoint.
2. Plot and show a **scree-plot** to look at the cumulative variance represented by the PCA eigenvectors.
3. You now want to experimentally find the best reduced dimensionality for the dataset with respect to *how it impacts the accuracy of a classifier*.
    - Produce a plot that shows accuracy of the kNN classifier on the PCA features using different numbers of dimensions. The accuracy should be listed in increasing order from 2 up to D, the original dimensionality of the dataset.
    - For the kNN classifier, you should choose the best one you found from asg1, one of the weighted versions using a normalized dataset.
    - Comment briefly on the difference in accuracy from asg1.
    - How do the best number of features suggested by the scree plot and this analysis compare?
4. Try using the t-SNE method to visualize the datasets by producing a 2D plot, comment on any useful patterns that this shows.

Once you've completed the above analysis, you can create two new versions of your datasets using the best reduced dimensionality representation, as measured against kNN performance or the scree plot analysis, whichever you choose. For the rest of the assignment you will have the following datasets

Original Dataset	Feature Extraction Datasets
`wine-raw`	`wine-pca` / `wine-lda`
`abalone-raw`	`abalone-pca` / `abalone-lda`

Include summary accuracy scores for kNN on all six datasets in the table in the last question.

2 Naive Bayes Classifier

Now you will classify the two datasets using the **Naive Bayes Classifier**. There are a number of these available, for our datasets, the **Multinomial Naive Bayes** and **Complement Naive Bayes** forms seem most appropriate, so we will experiment with those.

1. Use 5-fold cross validation to compare both versions of Naive Bayes against your previous best results from kNN. Do this on all 6 of your datasets.
2. You can have some analysis here or plot to highlight any interesting issues. There are also variants of Naive Bayes you may want to explore.
3. Produce a table comparing the accuracies on the different datasets.

Include summary accuracy scores on all six datasets in the table in the last question.

3 Decision Trees Classifier

You will now do classification on your datasets using Decision Trees. Decision Trees have a number of parameters that can effect peformance. You can use the `GridSearchCV` function for this question.

1. Use 5-fold cross validation and a range of parameter values to evaluate the best settings for classification on each dataset.
    - the maximum depth of trees
2. Produce a plot showing the mean accuracy vs. relative to tree depth.
3. **Interpretability:** Use the decision tree library functions, to examine the final resulting splitting rules used for the trees. Do they indicate any interesting patterns that explain the data? Can you find support for this from any analysis you've done or see on this dataset previously? For this part, *use the original raw feature space only*, not the PCA/LDA space. (Why not?)
    - Relevant decision tree visualizers, whichever one you use, make sure it is readable in useful way, don't show information that isn't helpful:
    - `tree.plot_tree()`: the built-in tree plot function for
    - `sklearn.tree.DecisionTree` `tree.export_graphviz` : another simple visualizer
    - `sklearn.tree.export_text` : text view of the tree data

Include summary accuracy scores on all six datasets in the table in the last question.

4 Random Forest Classifier

You will now do classification on your datasets using Random Forests. Random Forests have a number of parameters that can effect performance. You can use the `GridSearchCV` function for this question.

1. Use 5-fold cross validation and a range of parameter values to evaluate the best settings for classification on each dataset.

   - the maximum depth of trees, you can try values as low as 2 or 3 and as high as needed, decision trees have an upper limit on how deep they can go determine by the size of the dataset.
   - the number of trees, try values at regular intervals, you can go as low as 3 and as high as a few hundred trees.

2. Produce a plot showing the mean accuracy vs. the above parameter settings. This can be individually or using a **heat plot** showing a grid of mean accuracies for different combinations of the two parameters.

NOTE: *do not produce a tree plot or export for each tree in the forest!*
Include summary accuracy scores on all six datasets in the table in the last question.

5 Gradient Tree Boosting

You will now do classification on your datasets using Gradient Tree Boosting, on sklearn one is `GradientBoostingClassifier`, but you can use other implementations if you prefer. Use your judgement and experience from the other methods to decide how to train this algorithm and choose it's settings. At a minimum, pick some good parameter settings, train the model and show some analysis of it's performance and runtime compared to Random Forests.

Include summary accuracy scores on all six datasets in the table in the last question.

6 Final Results

In this question, summarize your findings concisely in words and tables.

- Comment on which pipeline resulted in the best classification accuracy overall, or for each dataset.
- What was the effect of Dimensionality Reduction on the different algorithms. Did some benefit from it more than others? Explain why this might be.
- Feel free to make additional observations about the results beyond these.
- Produce results tables summarizing all the final results in the following general form:

	setting(*)	wine-raw	wine-pca	wine-lda
kNN	best setting k=?, etc.	acc	acc	acc
Naive Bayes	form and parameters	acc	acc	acc
Decision Tree	best settings	acc	acc	acc
Random Forest	best settings	acc	acc	acc
Gradient Boosted Tree	best settings	acc	acc	acc

	setting	abalone-raw	abalone-pca	abalone-lda
kNN	best setting k=?, etc.	acc	acc	acc
Naive Bayes	form and parameters	acc	acc	acc
Decision Tree	best settings	acc	acc	acc
Random Forest	best settings	acc	acc	acc
Gradient Boosted Tree	best settings	acc	acc	acc

Notes:

- (*) If settings are used for multiple datasets. Alternatively, multiple settings could be used for each algorithm and described in some other way within the table or outside the table.
- Values in the a table are accuracy.
- Settings columns just needs to clarify which setting from the questions earlier are shown. You should pick the best one (or two) models for each algorithm, you don't need to show everything.
- Additional rows with multiple settings per algorithm can be included, *if it is relevant and explained in the text.* (but don't go overboard! remember, keep it to 10 pages or less!).

Notes

You might find the following links are useful to solve this assignment:

- `https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.GaussianNB.html`
- `https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingCla html`
- `https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV html`