University of Waterloo
ECE 657A: Data and Knowledge Modeling and Analysis
Winter 2022
Assignment 2 - Representation Learning, Parameter Tuning and Classification Comparisons

Submission Due: March 6, 2022 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 be used to set up groups for submission in Kritik. If you are working alone, you should not need to join a group in LEARN. 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 Kritik. Your report should be submitted as two files:

- A jupyter notebook that has the output already generated on the provided data in a readable way.
- A pdf or html version of your report, you can print the jupyter notebook to generate this.

Your group on LEARN has an associated **dropbox**. This is only to be used if something goes wrong with your kritik submission or you are concerned for any reason about your submission being missed or marked as late. The kritik website is the primary way to submit your files.

Evaluation: For this course, you will grading the assignments of your classmates using a peer grading system called Kritik (Read about it on the course website:`https://compthinking.github.io/DKMA/kritik/`. Every student will be grading a small number of assignments, reading through the report and code to evaluate and give feedback based on a grading rubric. This will all be done anonymously, of course, so you won't know who you are grading or who is grading you. However, it *also* means that when you submit your assignment you know that other classmates will see your answer, and your code. So keep your output concise (ie. as short as possible, but no shorter) and make your code clean and readable, including short comments. When grading you may prefer to look at the PDF/HTML file or the jupyter notebook directly.

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.
- 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 (ie. don't make it too long!) All questions below should be performed on both datasets. At the very end you will produce a summary table of all the accuracy results for the different experiments. Along the way, try to minimize the number of different plots shown to the essentials for the reader to understand what you did. Regarding code, 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 Wine and Abalone datasets from assignment 1. Since you've already preprocessed those and have classification results for kNN you can reuse that code, or the solution guide code for assignment 1 as a starting point for this assignment. Data should be in the best formulation that you found after assignment 1 for classification, using normalization, but leaving in outliers. Also, use the same **random number seed** as you used in assignment 1, for consistency.

1 Representation Learning

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

1. Produce a plot of the data in the two lowest dimensions for PCA and LDA, using easily distinguishable colours and markers to indicate the labels of each datapoint. Also use the t-SNE method to produce a 2D plot of the datasets. You now have three 2-dimensional plots of each dataset. Comment briefly on any interesting patterns that emerge.

2. (PCA Only) Produce 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 your kNN classifier against number of reduced dimensions being used. The dimensions should 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.

4. Now run the same analysis as in Question 3 using the LDA method.

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. You can just pick the best representation for each dataset, or you can keep the best PCA and LDA reductions for each dataset. For the rest of the assignment you will have the followng datasets

Original Dataset	One or Both of These
`wine-raw`	`wine-pca` / `wine-lda`
`abalone-raw`	`abalone-pca` / `abalone-lda`

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**form **Complement Naive Bayes** forms seem most appropriate.

1. Use 5-fold cross validation to compare both versions of Naive Bayes and your previous best results from kNN. Do this on all 4 (or 6) of your datasets and produce a table comparing the accuracies.

2. Complement Naive Bayes is meant to perform better for unbalanced datasets, since our datasets our unbalanced, this seems appropriate, does it make much difference? Try to explain why either way.

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 above parameter.

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 original raw feature space only.

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 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, 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 above parameter settings. This can be done as a **heat plot** showing a grid of mean accuracies for different combinations of the two parameters.

5 Gradient Tree Boosting

You will now do classification on your datasets using Gradient Tree Boosting (on sklearn it is `GradientBoostingClassifier`): This algorithm has 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 number of estimators, try values at regular intervals, you can go as low as 3 and as high as a few hundred estimators.
     Note: the number of 'trees' grown by GBT is `n_classes` $\times$ `n_estimators` *but this is handled automatically.*

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

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.
- If you choose to just pick the best representation from PCA and LDA, then this table will have four columns of data, this is fine.
- Values in the a table are accuracy.

- Settings columns just needs to clarify which setting from the questions earlier were used, not all settings.
- Additional rows with multiple settings per algorithm can be inlcuded, if it is relevant and explained in the text. (but don't go overboard!)

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`