University of Waterloo
ECE 657A: Data and Knowledge Modeling and Analysis
Winter 2022
Assignment 1 - Data Preprocessing, Experimental Setup and KNN Classification

Submission Due: February 4, 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 genearted on the provided data in a readable way.
- A pdf version of your report, you can print the jupyter notebook to generate this.

If anything goes wrong with submission on Kritik, you can use the LEARN Dropbox for your group to submit the files before the submission deadline. If you do this, be sure to contact course staff in a private message on piazza to explain the difficulty and you will not be counted as submitting late.

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 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:

- Establish your software stack to carry out data analysis assignments for the rest of the course.
- Load datasets and perform some exploratory plots. Ensure all plots have labelled axis, titles and short captions explaining them.
- Practice how to apply the methods discussed in class .
- Demonstrate understanding by *explaining* any result you obtain, in straightforward, and short, pieces of text about what the results mean and how you came to them.

Dataset 1

We will use the Wine Quality Data Set that we saw in Asg0:

- Wine Quality Dataset:
  `https://archive.ics.uci.edu/ml/datasets/wine+quality`
- Original Data Directory
  `http://archive.ics.uci.edu/ml/machine-learning-databases/wine-quality/`
- Python Jupyter Template : there is a ipynb file on LEARN for assignment0 that can be used as a starting point.
- The dataset original consists of one datafile for red wine and one datafile for white win. The provided jupyter notebook loads the data and adds an additional column for wine `color` where `color` = 0 indicates white wine and `color` = 1 indicates red wine.
- **Classification task:** predict ‘quality‘ from the other features

Dataset 2

The second dataset is the Abalone dataset about the abalone fish and it’s physical characteristics.

- Abalone Dataset:
  `https://archive-beta.ics.uci.edu/ml/datasets/abalone`
- **Classification task:** predict ‘Rings‘ from the other features, this feature is essentially the age of the fish. Note that one of the features is categorical.

1 Assessment of Data and Applying Normalization (on Abalone Dataset only)

NOTE: Since you have a solution for this on the Wine Dataset already from Asg0, you should repeat that analysis now on your own with the Abalone Dataset only.

1. Load the dataset and explore the features and their ranges and distribution. Show a couple approaches you use.
2. Is there any missing data? Present evidence or refer to figures or to earlier tables or figures clearly (eg. by a figure/table name or by the jupyter line number).
3. Compute the moments or summarization statistics on the data features. Comment on the diversity of data types and ranges of the features.
4. Do there seem to be outliers that should be watched out for? Is it possible they are errors or just naturally occurring? How would you evaluate this?
5. Is this a balanced dataset? If not, where is it unbalanced? Would the dataset still be usable?
6. **Normalization:** Assess the need for normalization and implement it. You will normalize the data using two normalization levels `min-max` and `z-score` normalization.
   (a) Is normalization necessary for this dataset given what you've seen? Explain why briefly.
   (b) Using the min-max normalized data, pick two or three numeric features of your own choosing and compare the interaction between the variables and how it differes from the unnormalized data.
   (c) Now perform the same analysis but for the data with z-score normalization.

2 Classification with KNN (on both datasets)

Classify the data using a KNN classifier. You will tune the parameter of the KNN classifier using sci-kit functions (see links at the end of the assignment), plot the different validation accuracies against the values of the parameter, select the best parameter to fit the model and report the resulting accuracy. Carry out the following activities and reporting:

1. Divide the data into a training set and a test set (80%, 20%) **Note:** set the random seed for splitting, use `random_state=27` in the sci-kit learn `train_test_split` function to get the same split every time you run the program.

2. Start by training the model with the classifier's default parameters. Use the train set and test the model on the test set. Note that different values of k will lead to different results.

3. To find the best value for k, you need to compute accuracy for a range of values of k so you can "tune" the classifier. Using these scores, **plot a figure** of *accuracy* vs k. Report the best k in terms of classification accuracy.

4. **Improving on KNN:** You can try to improve on your classification results using the method of *weighted* KNN. The `KNeighborsClassifier` class has an option for *weighted* KNN where points that are nearby to the query point are more important for the classification than others. Compare the three different weighting schemes (default, manhatten, euclidean) by plotting accuracy vs k for all three of them on the same figure to see the effect.

5. **Ablation Study on Normalization:** An ablation study is where some aspect of the model or analysis is dropped, in order to see what its effect was on the entire outcome. We can do a simple form of ablation here by removing normalization from our pipeline. Replot the three curves from the previous question on weighted KNN, but this time remove the normalization step from the preprocessing. Comment on the difference, was normalization effective or necessary in this case?

Notes

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

- `https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html`

- `https://scikit-learn.org/stable/modules/cross_validation.html#cross-validation`

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

- `https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingClassifier.html`