University of Waterloo
ECE 657A: Data and Knowledge Modeling and Analysis
Winter 2023
Assignment 1 - Data Preprocessing, Experimental Setup and KNN Classification
Submission Due: February 10, 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 and Crowdmark. If you are working alone, you should still join a group in LEARN with one member. 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. Each main question has an entry in Crowdmark. For each question

- submit a pdf or image (from scanner or a screencap, or good a clear photo) for the entire question and all it's subparts. Be sure to label the subparts clearly.
- For coding questions, you should also submit a python jupyter notebook that summarizes your results and is easy to read. You can generate the pdf part for coding questions from the jupyter notebook and upload that if you want to.
- For the theory questions, you can write the answer out by hand on paper or a tablet and take a picture or save to pdf/image. You could also choose to type it out, Microsoft Word and Google Docs now have quite good math notation entry, or you could use the best (but not the easiest) option LaTeX using the Overleaf online editor (`https://www.overleaf.com/edu/uwaterloo`). UWaterloo students get free access.

If anything goes wrong with submission on Crowdmark, you can always use the LEARN Dropbox for your group to submit the files before the submission deadline or in the worst case, just email it to one of the course staff. 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: As described in the course outline, assignments this year will be *lightly graded*, meaning we are not going to dig super-deep into every plot and piece of code, we want to see that the general questions have been answered, and the results are presented in a professional manner. So don't try to have long justifications, or included all your code in your report, or every single iteration

of the plots. Instead, focus on the core insights you find, show the final plot or table and explain it succinctly and fully. Name and number every section and question clearly; make sure plots and tables are *fully labeled* in a professional way, look at academic papers for examples. For theory questions, state your assumptions if needed, and follow a step-by-step process. If you had to correct something, or change it, don't add more layers and notes, rewrite it neatly so the answer can be seen clearly and quickly.

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, run a simple classification algorithm and generate some exploratory plots and tables. Ensure all plots and tables have labels for axis, titles and short captions explaining them.
- Practice how to apply the methods discussed in class.
- Demonstrate understanding by *explaining* the final results you obtain, in straightforward, and short, pieces of text about what the results mean and how you came to them.

Dataset

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

- An `abalone.csv` file will be available in the asg1 directory on learn.
- Original 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

First you need to look at the dataset as a whole and analyse if we need to carry out any preprocessing. Load the dataset and explore the features and their ranges and distribution. Show just a few highlights of the approaches you use. Be sure to answer the following questions along the way:

1. Is there any missing data?

2. Compute the moments or summarization statistics on the data features (mean, median, variance, skew, kurtosis). Do these highlight anything interesting about the different features?

3. Use a pairsplot (the seaborn library has a nice one, for example) to look at the whole of the datset. Choose a subset, just some features, and show it in your report to highlight some features that seem important.

4. Is this a balanced dataset? If not, what kind of correction could we apply?

5. Normalize the data using `z-score` normalization as a preprocessing step.

2 Classification with KNN

Classify the data using a KNN classifier. You will explore some parameters of the KNN classifier, 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. Start by training the model with the classifier's default parameters using the training set and test the model with the test set.

2. Note that different values of k will lead to different results. Note that this is a hard dataset to get high accuracy on, with kNN accuracy the accuracy could be below 50%.

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.

4. First divide the data into Training and Testing sets in the ratio 80%/20%.

5. You then need to test a range of values for the KNN parameter k using **5-fold cross validation** on the Training set. So each round of cross-validation will use 4 of the folds for training, and 1 of the folds for computing accuracy to observe how well that k does.

6. All of these 5-folds can be plotted, analysed, and averaged to determine the best value of k to use for KNN. You can decide which information to show, but at least produce one plot of the mean validation accuracy vs. k across all folds.

7. Once a promising value of k is chosen, you can retrain KNN using that k on all the training data and calculate accuracy on the held out Test set and report this result.

8. **Improving on KNN:** You can also 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.

    - Minimum Requirement: using your chosen value of k above compute the values on your Test set using the other weighting methods.
    - (optional ungraded, extra task) 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. (consider where is the right place to do this, given the hyper-parameter tuning step we did earlier)

3 MLE Derivation

In this problem we will find the *maximum likelihood estimator (MLE)* estimator for given posterior and prior distributions.

We assume we have N samples, $x_1, \ldots, x_i, \ldots, x_N$, each *independently* drawn from a *Kumaraswamy distribution*, defined as :

$$x_i \sim \text{Kumaraswamy}(a, b) : f(x|a, b) = abx^{a-1}(1 - x^a)^{(b-1)} \tag{1}$$

where $x_i \in (0, 1)$ The parameters a, b are positive real numbers and act as *shape parameters* in a similar as the Beta distribution. **For the purposes of this assignment, we assume that a is known, and we need to derive a formula for estimating b from data.**

Derive the MLE estimator for a. Make sure to show all of your work.
Steps for MLE:

1. Derive the form of the likelihood term from the given posterior distribution. **NOTE:** For defining the log-likelihood, assume that $\log(x) = \log_e(x) = \ln(x)$.
2. Take the log of the likelihood
3. Take the derivative of the log-likelihood and set to equal to zero
4. Solve for the target parameter, if possible. If not, then simplify it as much as much as you can.

4 MAP Estimator

In this problem we will find the *maximum a-posteriori (MAP)* estimator for the same distribution. Derive the MAP estimator for a. We assume that as a

prior distribution that a is itself sampled from a normal distribution with known mean μ and known variance σ^2.

$$a \sim \mathcal{N}(\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(a-\mu)^2}{2\sigma^2}} \tag{2}$$

Steps for MAP:

1. Derive the form of the log-posterior term from the given posterior distribution and the given prior distribution of it's parameter(s):
2. Take the derivative of the log-likelihood and set to equal to zero. Derive the formula for MAP_a in general terms using the known distribution's and reduce for a as much as you can.

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`