University of Waterloo

ECE 657A: Data and Knowledge Modeling and Analysis

Spring 2021

Assignment 1: Data Cleaning and Classification

Due: Sunday June 6, 2021 at 11:59pm ET

Overview

Collaboration/Groups: You may do your work individually or in pairs. Either way, every student must sign up for a Asg1Group in LEARN *and* in Crowdmark for grading purposes. 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, via the CROWDMARK website in PDF format. You should receive an invite to Crowdmark by email. You will need to divide the PDF up into one file for each `[CMX]` question. Some "questions" in this assignment are descriptive and have no output, so no PDF if needed. It is best to start each `[CMX]` question on a new page, split up your PDF after completion, and then drag and drop each PDF onto the relevant question in Crowdmark. Overlapping or duplicated text between questions is fine, as long as the *entire answer* for each question is fully contained within that question's PDF file.

You must also submit the code/scripts needed to reproduce your work as a Python Jupyter notebook to the LEARN dropbox associated with your group (this is why you need to join a group of 1 if working alone, to access your dropbox).

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.
- Study how to apply some of the methods discussed in class and gain experience on the use of moments, k-NN classification and evaluation methods.
- Whenever you give a result, provide an short *explanation* in straightforward text for what the results mean and how you came to them.

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

Dataset 1

Use the Iris dataset (file uploaded on learn dropbox). This dataset is a classic and fairly simple benchmark for basic machine learning algorithms. It includes different features (attributes) of three Iris flower species (setosa, versicolor, virginica).

Dataset 2

Another dataset is the heart disease dataset (file uploaded on learn dropbox). The original dataset is from `https://archive.ics.uci.edu/ml/datasets/heart+disease` but some of the values have been altered for this assignment. The dataset contains 14 features (attributes) and 303 instances. The features are multivariate with types - nominal, ordinal, binary, and interval/real ratio data. The classification target is a binary variable (named `target`) indicating the presence or absence of heart disease using 1 and 0 respectively.

Answer Question 1, 2, and 3 for both the datasets.

Question 1: Data Exploration

1. `[CM1]` **Data Cleaning:** [***TODO:*** *this should go first*] deal with any missing values in the data (use any of the methods discussed in class: dropping data, interpolating, replacing with approximations,...). You can also remove any noise from the data by applying smoothing on some features. **Report any changes you make and justify them.** You can make comparisons of any of these approaches have an impact on classification performance using your validation set. **Normalization:** Normalize the data using min-max and zscore and compare to unnormalized version of the data. Explain any differences that you see. You may want to do some of the visualization in `[CM2]` to see the impact of Data Cleaning on the distribution of the data.

2. `[CM2]` To begin understanding the dataset, generate a "pairs plot" (also called a scatter plot matrix, `seaborn.pairplot` is one method to do this) of the data. Note that the pairs plot includes the scatter plots of every dimension versus every other dimension. From the pair plot, identify the subplots corresponding to the pairs of features where you see correlation.

   - **For Iris:** Make a single pair plot of all the features and data.
   - **For Heart Disease:** Use pair plots to explore different subsets of features and their correlations, distribution, etc. You don't need to include this large pair plot in your report. Instead, choose your own subset of 3-5 features for the plot which highlight some *interesting*

pattern. Show this smaller pair plot only and provide a short justification why you chose those features.

3. `[CM3]` **Calculate and report** the correlation coefficient for all pairs of features. To what extent are the features correlated? Calculate and report the mean, variance, skew, and kurtosis for all the variables in datasets for which is makes sense. Do you find any interesting or significant relationships? **Briefly explain your observations** about the nature of data and the relationships between the features of the dataset. Do you chosen subset of features from Heart Disease have any particular pattern on any of these statistics?

Question 2: KNN

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:

Basic Model: The intent for the steps 1-4 is to confirm your numerical answer, so please follow the steps exactly.

1. Divide the data into a training set and a test set (80%, 20%) **Note:** set the random seed for splitting, use `random_state=98` in the sci-kit learn `train_test_split` function to get the same split every time you run the program.

2. `[CM4]` Train the model with the classifier's default parameters. Use the train set and test the model on the test set. Store the accuracy of the model and report it.

3. Next, you need to find the best parameters, to "tune", the classifier. In this case it is k, the number of neighbours, used in KNN.
   General Process: You need to test a range of values for the parameter k using a validation set. You will try two different ways of doing this **k-fold cross validation** and **train-validate-test**.

   - **For Iris:** Since this is a very small dataset, we need to leverage as much of the data as we can for parameter tuning. So you will use **5-fold cross validation** on the *training set* to train the classifier and produce accuracy scores.
   - **For Heart Disease:** Instead of 5-fold cross validation, here you will further divide the training set into an (90%,10%) split into train and validate sets (you also, of course, still have your test set held out). Fit the KNN classifier for a range of k values on the new training set only, producing accuracy scores with the validation set.

For each dataset, using the corresponding validation method above, do the following for a range of k values from *at least* $k \in \{1, 5, 10, 15, 20, 25, 30, 35\}$:

(a) Fit the model using the training set or fold.

(b) Score the trained model with the validation set or fold. Store the accuracy or average across accuracies for all folds.

(c) `[CM5]` Using these scores, **plot a figure** of *accuracy* vs k. Report the best k in terms of classification accuracy. For k-fold cross validation, report the variance as well. Do you find any advantage to one form of validation over the other?

(d) `[CM6]` Now, using the best found parameter, fit the model on the entire training set and predict the target on the test set. **Report the accuracy, AUC, f-score of your kNN classifier.** Explain your results briefly and the following points: What was the effect of changing k? Was the accuracy always affected the same way with an increase of k? Why do you think this happened?

Your Improved Model: Try to improve your classification results using any of the performance metrics we have discussed by exploring different ways to improve using your validation set.

5. **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. Try using different weighting schemes (default, manhatten, euclidean) to see the effect. You can also define your own distance metric to try to improve performance further (using a validation-testing approach of course).

6. **Different NN Algorithms:** Consider changing other parameters of KNN, such as the algorithm selection, which effects accuracy but also the speed of computation. Whatever improvement you explore, report relevant statistics to show any differences which your initial experiments above.

7. `[CM7]` After making these improvements, compute your new classification results on the test set and **report the accuracy, AUC and f-score**.

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