Assignment 1

Due on November 1, 2022 (23:59:59)

Instructions. The goal of this problem set is to make you understand and familiarize with K-Nearest Neighbor Algorithm. There are two parts in this assignment. The first part involves a KNN-classification experiment and and the second part involves KNN-regression experiment.

1 PART 1: Personality Classification

In this part of the assignment, you will implement a nearest neighbor algorithm to classify different personality types of people. You will also extend your implementation as weighted KNN algorithm.

A dataset [1] is provided for your training phase. In other words, you should split your training dataset into two set; training set which will be used to learn model, and validation set which will be used to measure the success of your model. You will use 5-fold cross-validation method which is explained in the class.

1.1 Classification Dataset: Personality Classification Dataset [1]

- You can download the dataset from given link.
- Dataset consists of 60.000 samples with discrete 16 ("Personality" attribute) ground-truth class types.
- Each sample in the dataset includes 60 attribute/feature informations (excluding "Response Id") and these attribute values represents the quantitative response with respect to the their counterpart questions.

1.2 Classification Performance Metric

You will compute "Accuracy", "Precision" and "Recall" of your model to measure the success of your classification method:

$$\textbf{Accuracy} = \frac{TP + TN}{TP + FP + FN + TN} \tag{1}$$

$$\textbf{Precision} = \frac{TP}{TP + FP} \tag{2}$$

$$\textbf{Recall} = \frac{TP}{TP + FN} \tag{3}$$

You will report accuracy, precision and recall of each test with respect to 5-fold cross validation and average accuracy, precision and recall of these tests.

1.3 Feature Normalization

You will use min-max normalization on the features of your samples to re-scale each feature (feature/attribute column on data) between (0-1) range.

$$n_i = \frac{f_i - min(f)}{max(f) - min(f)} \tag{4}$$

Where, n_i represents the i^{th} normalized element of your specific feature column (f) and f_i represents the i^{th} original element of your specific feature column (f).

1.4 Error Analysis for Classification

- Find a few misclassified samples and comment on why you think they were hard to classify.
- Compare performance of different feature normalization choices and investigate the effect of important system parameters (number of training samples used, k in k-NN, etc.). Wherever relevant, feel free to discuss computation time in addition to classification rate.

Steps to Follow for Classification

1. Read your classification data and transform it to the Numpy array collection.
2. For the test samples
    - predict their classes using k-NN. (with feature normalization and without feature normalization)
    - predict their classes using weighted k-NN. (with feature normalization and without feature normalization)

3. Compute and report your accuracy, precision and recall of your different k-NN and weighted k-NN models with different k parameters (Your must experiment with this k parameters: **(1,3,5,7,9)**) on 5-fold cross validation.
4. Report your findings in "Error Analysis for Classification" section.

2 PART 2: Energy Efficiency Estimation from Data

In this part of the assignment, you will implement a nearest neighbor algorithm to estimate two different energy efficiency values of different building shapes. You will also extend your implementation as weighted KNN algorithm.

A dataset [2] is provided for your training phase. In other words, you should split your training dataset into two sets; training set which will be used to learn model, and validation set which will be used to measure the success of your model. You will use 5-fold cross validation method which is explained in the class.

2.1 Regression Dataset: Energy Efficiency Estimation Dataset [2]

- You can download the dataset from given link.
- Dataset consists of 768 samples with two different continuous energy efficiency output rate ("Heating Load", "Cooling Load") values.
- **Attribute information for each sample in dataset:**
  1. Relative Compactness
  2. Surface Area
  3. Wall Area
  4. Roof Area
  5. Overall Height
  6. Orientation
  7. Glazing Area
  8. Glazing Area Distribution
  9. First ground-truth energy efficient rate output ("Heating Load")
  10. Second ground-truth energy efficient rate output ("Cooling Load")

2.2 Regression Performance Metric

You will compute "Mean Absolute Error" of your model to measure the success of your regression method:

Mean Absolute Error (MAE) $= \frac{1}{n}\sum_{i=1}^{n}|d_i - \hat{d}_i|$
where,
d_i is the actual (ground-truth) energy efficient rate value of sample
$\hat{d}_i$ is the predicted/estimated energy efficient rate value of sample
n is the number of samples

You will report MAE of each test with respect to 5-fold cross validation and average MAE of these tests.

2.3 Feature Normalization

You will use min-max normalization on the features of your samples to re-scale each feature (feature/attribute column on data) between (0-1) range.

$$n_i = \frac{f_i - min(f)}{max(f) - min(f)} \tag{5}$$

Where, n_i represents the i^{th} normalized element of your specific feature column (f) and f_i represents the i^{th} original element of your specific feature column (f).

2.4 Error Analysis for Regression

- Compare performance of different feature normalization choices and investigate the effect of important system parameters (number of training samples used, k in k-NN, etc.). Wherever relevant, feel free to discuss computation time in addition to regression/estimation rate.

Steps to Follow for Regression

1. Read your regression data and transform it to the Numpy array collection.
2. For the test samples
    - estimate their energy efficiency values using k-NN. (with feature normalization and without feature normalization)
    - estimate their energy efficiency values using weighted k-NN. (with feature normalization and without feature normalization)

3. Compute and report your MAE of your different k-NN and weighted k-NN models with different k parameters (Your must experiment with this k parameters: **(1,3,5,7,9)**) on 5-fold cross validation.
4. Report your findings in "Error Analysis for Regression" section.

3 Implementation Details

- **You can't use ready-made libraries for your K-fold cross-validation/Shuffle methods for your data for both of your Part 1 and Part 2 implementations. You must implement these on your own**
- **You can't use ready-made libraries for your k-NN/weighted k-NN methods for your data for both of your Part 1 and Part 2 implementations. You must implement these on your own**
- **You can't use ready-made libraries for computing "Accuracy", "Precision", "Recall" and "MAE" metrics. You must implement these on your own**
- **You can't use ready-made libraries for computing "min-max" feature normalization.**
- You may use Numpy array functions for your intermediate implementation steps for your Part 1 and Part 2 implementations.
- You may use "Pandas" library for reading and writing/creating .csv files: `https://pandas.pydata.org/docs/index.html`

Submit

You are required to submit all your code (*all your code should be written in Jupyter notebook* long with a report in ipynb format (should be prepared using Jupyter notebook). The codes you will submit should be well commented. Your report should be self-contained and should contain a brief overview of the problem and the details of your implemented solution. You can include pseudocode or figures to highlight or clarify certain aspects of your solution.

Finally, prepare a ZIP file named **name-surname-pset1.zip** containing

- report.ipynb (PDF file containing your report)
- code/ (directory containing all your codes as Python file .py)

Grading

- Code (50): k-NN: 20, Weighted k-NN: 30
- Report(50):Analysis of the results for KNN-Classification: 25 points, Analysis of the results for Knn-Regression: 25 points.

  Note: Preparing a good report is important as well as the correctness of your solutions! You should explain your choices and their effects to the results. You can create a table to report your results.

Academic Integrity

All work on assignments must be done individually unless stated otherwise. You are encouraged to discuss with your classmates about the given assignments, but these discussions should be carried out in an abstract way. That is, discussions related to a particular solution to a specific problem (either in actual code or in the pseudocode) will not be tolerated. In short, turning in someone else's work, in whole or in part, as your own will be considered as a violation of academic integrity. Please note that the former condition also holds for the material found on the web as everything on the web has been written by someone else.

References

[1] `https://www.kaggle.com/datasets/anshulmehtakaggl/60k-responses-of-16-personalities-tes`

[2] `https://www.kaggle.com/datasets/ujjwalchowdhury/energy-efficiency-data-set`