Statistical Methods in AI (CSE/ECE 471)
Spring-2020
Assignment-1 (200 points)
Posted on: 21/01/20
Due on: 11:59 P.M. 31/01/20

Instructions

1. The assignment contains three questions. All the questions are compulsory
2. Assignment must be implemented in Python 3 only.
3. You are not allowed to use libraries (like scikit-learn etc.) for building the model or for hyper-parameter optimization (hyperopt, dragonfly etc.).
4. You are allowed to use libraries for data preprocessing (numpy, pandas etc.) and for evaluation metrics, data visualization (matplotlib etc.).
5. You will be evaluated not just on the overall performance of the model on the test set but also on the experimentation with hyper parameters, data prepossessing techniques etc.
6. Datasets for all the questions are provided at http://bit.ly/$smai_a1_data$
7. For each question, you are required to make two files - one for the code and the second for the report.
8. The report file must be a well documented jupyter notebook, explaining the experiments you have performed, evaluation metrics and corresponding code. The code must run and be able to reproduce the accuracies, figures/graphs etc.
9. For all the questions, you must create a train-validation data split and test the hyperparameter tuning on the validation set. Your jupyter notebook must reflect the same.
10. The code file must be a python(.py) file. You are expected to define a class for each question which is compatible with the test.py file provided here. This file should not contain any libraries for building the model (like scikit-learn etc.). Make sure your code can be run by "python test.py". Double check this.
11. Your assignment will be evaluated with undisclosed test files.
12. Your final submission folder should be named "RollNo.zip". This should contain a single folder "RollNo" that has 7 files - q1.py, q2.py, q3.py, q1.ipynb, q2.ipynb, q3.ipynb, test.py. Do not include any other files. Strictly adhere to the naming convention.
13. Any attempts at plagiarism will be penalized heavily.

Questions

1. (60 points) k-Nearest Neighbors - Task 1

    1. Implement a KNN based classifier to predict digits from images of handwritten digits in the dataset.
    2. Featurize the images as vectors that can be used for classification.
    3. Experiment with different values of K(number of neighbors).
    4. Experiment with different distance measures - Euclidean distance, Manhattan distance,
    5. Report accuracy score, F1-score, Confusion matrix and any other metrics you feel useful.
    6. Implement baselines such as random guessing/majority voting and compare performance. Also, report the performance of scikit-learn's kNN classifier. Report your findings.

2. (40 points) k-Nearest Neighbors - Task 2

    1. Implement a KNN based classifier to classify given set of features in Mushroom Database. Missing data must be handled appropriately.(Denoted by "?").
    2. Choose an appropriate distance measure for categorical features.
    3. Experiment with different values of K(number of neighbors).
    4. Report accuracy score, F1-score, Confusion matrix and any other metrics you feel useful.
    5. Implement baselines such as random guessing/majority voting and compare performance. Also, report the performance of scikit-learn's kNN classifier. Report your findings.

3. (100 points) Decision Tree

    1. Implement a decision tree to predict housing prices for the given dataset using the available features.
    2. The various attributes of the data are explained in the file `data_description.txt`. Note that some attributes are categorical while others are continuos.
    3. Feel Free to use Python libraries such as binarytree or any other library in Python to implement the binary tree. However, you cannot use libraries like scikit-learn which automatically create the decision tree for you.
    4. Experiment with different measures for choosing how to split a node(Gini impurity, information gain, variance reduction) . You could also try different approaches to decide when to terminate the tree.
    5. Report metrics such as Mean Squared Error(MSE) and Mean Absolute Error(MAE) along with any other metrics that you feel may be useful.

6. For feature engineering, you may consider normalizing/standardizing the data.
7. Implement simple baselines such as always predicting the mean/median of the training data. Also, compare the performance against scikit-learn's decision tree. Report your findings.