Assignment

Due on October 25, 2023 (23:59:59)

Instructions. The goal of this problem set is to make you understand and familiarize yourself with the Scikit-Learn library for classification methods and metrics. In this assignment, you will use k-nearest Neighbor, Naive Bayes, Random Forest, and Support Vector Machine classification algorithms and classification metrics for evaluating your results.

Stellar Classification Dataset

In this part of the assignment, you will implement a well-known machine learning algorithm to classify stellar types.

A dataset is provided for your training phase. You should use a subset of the training set to validate the performance of your model. 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. Also, you can use k-fold cross-validation method.

Stellar Classification Dataset - SDSS17 [1]

- You can download the dataset from given link.
- Dataset consists of 100,000 samples with discrete 3 class types.
- You should check the dataset and see if it needs preprocessing. Apply the necessary preprocessing to classify.
- **Attribute Information:**
  1. obj_ID = Object Identifier, the unique value that identifies the object in the image catalog used by the CAS
  2. alpha = Right Ascension angle (at J2000 epoch)
  3. delta = Declination angle (at J2000 epoch)
  4. u = Ultraviolet filter in the photometric system
  5. g = Green filter in the photometric system
  6. r = Red filter in the photometric system
  7. i = Near Infrared filter in the photometric system

8. z = Infrared filter in the photometric system
9. run_ID = Run Number used to identify the specific scan
10. rereun_ID = Rerun Number to specify how the image was processed
11. cam_col = Camera column to identify the scanline within the run
12. field_ID = Field number to identify each field
13. spec_obj_ID = Unique ID used for optical spectroscopic objects (this means that 2 different observations with the same spec_obj_ID must share the output class)
14. class = object class (galaxy, star or quasar object)
15. redshift = redshift value based on the increase in wavelength
16. plate = plate ID, identifies each plate in SDSS
17. MJD = Modified Julian Date, used to indicate when a given piece of SDSS data was taken
18. fiber_ID = fiber ID that identifies the fiber that pointed the light at the focal plane in each observation

Spliting the Dataset

- **Train-Test Split** is a straightforward method for evaluating the performance of a machine learning model. It involves dividing your dataset into two subsets: the training set and the testing set. Typically, a larger portion of the data (e.g., 80%) is used for training, while the remainder is reserved for testing. The process can be summarized as follows:
  - The training set is used to train the machine learning model.
  - The testing set is used to evaluate the model's performance by making predictions on data it hasn't seen during training.
  - Performance metrics such as accuracy, precision, recall, or F1-score are computed to assess how well the model generalizes to new, unseen data.
- **k-Fold Cross Validation** is a more robust method for estimating a model's performance, especially when the dataset is limited. It involves dividing the dataset into "k" subsets of approximately equal size (e.g., k=5). The process can be summarized as follows:
  - The dataset is divided into k subsets, or "folds."
  - The model is trained and evaluated k times. In each iteration, one of the k folds is used as the test set, while the other k-1 folds are used as the training set.

- Performance metrics are computed for each of the k iterations.
- The final performance assessment is often done by averaging the results from all iterations.

Classification Methods

- **k-Nearest Neighbor(kNN)** kNN is a simple machine learning algorithm used for classification and regression tasks. It works by finding the k training examples (data points) in the dataset that are closest to a given input point in feature space.
- **Naive Bayes** Naive Bayes is a probabilistic algorithm used for classification tasks, particularly in natural language processing and spam email detection. It is based on Bayes' theorem and assumes that features are conditionally independent.
- **Random Forest** Random Forest is an ensemble learning method that combines multiple decision trees to improve predictive accuracy and reduce overfitting. Random Forest is known for its robustness and ability to handle high-dimensional data and complex relationships.
- **Support Vector Machines(SVM)** SVM aims to find the optimal decision boundary that maximizes classification accuracy while minimizing the risk of overfitting.SVM is effective in cases where there is a clear margin of separation between classes, and it can handle both linear and non-linear data through the use of kernel functions.

Metrics

- You are expected to use different classification evaluation metrics such as "Accuracy", "Precision", "Recall" and "F1-Measure" from Scikit-Learn library.
  Evaluation Metrics from Scikit-Learn
- You are also expected to obtain a confusion matrix from Scikit-Learn:
  Confusion Matrix from Scikit-Learn

Implementation Details

- In this assignment, you will use kNN, weighted-kNN, Naive Bayes, Random Forest, and SVM classification methods.
- You can use the Scikit-Learn library for implementing classification methods, and evaluation metrics.

Submit

You are required to submit all your code in a Jupyter notebook, along with a report in ipynb format, which should also be prepared using Jupyter notebook. The code you submit should be thoroughly commented. Your report should be self-contained and include a concise overview of the problem and the details of your implemented solution. Feel free to include pseudocode or figures to highlight or clarify specific aspects of your solution. Finally, prepare a ZIP file named name-surname-a0.zip containing:

- assignment_0.ipynb (including your report and code)
- assignment_0.py (py file version of your ipynb file)
- Do not send the dataset.

The ZIP file will be submitted via Google Classroom. Click here to accept your Assignment 0.

Grading

- Code (50): k-NN: 10, Weighted k-NN: 10, Naive Bayes: 10, Random Forest: 10, SVM: 10
- Report(50): Analysis of the results for k-NN: 10, Weighted k-NN: 10, Naive Bayes: 10, Random Forest: 10, SVM: 10

Note: Preparing a good report is important as well as the correctness of your solutions! You should explain your choices and their effects on 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] Classification of Stars, Galaxies and Quasars. Sloan Digital Sky Survey DR17. `https://www.kaggle.com/datasets/fedesoriano/stellar-classification-dataset-sdss17/data`