Assignment 1

Due on 16.03.2025 (23:00)
Programming Language: Python 3.9

Introduction

The goal of this project is to make you understand and familiarize yourself with basic machine learning concepts with help of 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. This project consists of two parts. The first part involves classification according to given textual features of the rice images where the second part focuses on rice images itself. There are 75,000 samples in the dataset. Data must be split data into train (80%) and test (20%) sets randomly. Moreover, K-Fold Cross Validation must be applied.

Technical Background

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)** 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** 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** 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)** 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. You can also use libraries like Matplotlib and Seaborn (and so on) for visualization purposes.

PART I: Textual Data Analysis

In this part of the project you are given 106 features with corresponding classes for each sample. You are supposed to check the dataset and see if it needs preprocessing. Apply the necessary preprocessing to classify.

PART II: Image Data Analysis

In this part of the project you are given the image forms of the samples given at the first part. You are supposed to extract features from the images in order to do classification process. As a way to do this, you can use RGB features of the images and apply some threshold to handle pixels of the images as binary values. It is expected to both handling the RGB values as-is and the thresholded version, you can also apply any other feature extraction methodologies in addition to these two.

Steps to Follow

1. Import and visualize the data in any aspects that you think it is beneficial for the reader's better understanding of the data.

2. Split data into train and test set randomly (you can use 80% of the data for training and 20% of it for the test purposes).

3. For the test set that you separated at the previous step try to determine classes for the rice species.

4. Also apply K-Fold Cross-Validation method to check the model performance in a better aspect.

5. Finally compute performance of the model to measure the success of the classification methodologies for each setting you have used:

$$\textbf{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}$$
$$\textbf{Precision} = \frac{TP}{TP+FP}$$
$$\textbf{Recall} = \frac{TP}{TP+FN}$$

   You will report Accuracy, Precision, and Recall measures.

6. The most important part of this project is doing as much experiment as you can to show strengths and weaknesses of the algorithms. In short, you are supposed to experiment with different scenarios and comment about them, note that commenting is as much important as the experimenting, so, please explain your reasoning and inference for every experiment that you did. Some examples that you may try are (Note that the following ones are only examples, you can add anything that you think it is beneficial to try for better understanding about these algorithms.):

   - You can use different hyper-parameters and try to determine optimal hyper-parameters for this dataset according to desired algorithms. You may benefit from validation error for this purpose.
   - You can compare the performance of the models with raw data and eliminated data.

- You can compare the performance of the models with raw data and scaled data.
- You can compare the best performance of the models with each other.
- **Please include only necessary studies and comparisons in your report. Including all studies and comparisons in your report (whether they are necessary or not) may reduce the readability of your report and may negatively affect your score.**

What to Hand In

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 and your notebook must be ran and have outputs for each cell in the order of the cells before submission.** Your report should be self-contained and include a concise overview of the problem and the details of your implemented solution. Note that your report also has to contain necessary libraries to be installed with the versions that are used (!pip install commands are preffered). Feel free to include pseudocode or figures to highlight or clarify specific aspects of your solution. Submission hierarchy must be as follows:

```
- <GroupID>.zip
    - assignment1.ipynb
    - *.(jpg|jpeg|png|gif|tif|tiff|bmp|svg|webp) (optional)
```

Do not send the dataset.
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 (or any content you believe that it is beneficial to show your all work) to report your results.
Note that submission format is crucial and submit system is set to give you score as one if you follow the submission hierarchy, which is really easy (there might be some issues for the MacOS users but it can be overcomed via the mini guide that is shared at the Piazza). If you do not score one from the submit system **you will penalized by 20% even if your submission hierarchy is correct**.

Academic Integrity

All work on assignments must be done on your own group 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] Murat Koklu, Ilkay Cinar, and Yavuz Selim Taspinar. Classification of rice varieties with deep learning methods. *Computers and Electronics in Agriculture*, 187:106285, 2021.

[2] İlkay Çınar and Murat Köklü. Determination of effective and specific physical features of rice varieties by computer vision in exterior quality inspection. *Selcuk Journal of Agriculture and Food Sciences*, 35(3):229–243, 2021.

[3] İlkay Çınar and Murat Koklu. Identification of rice varieties using machine learning algorithms. *Journal of Agricultural Sciences*, 28(2):307–325, 2022.

[4] Ilkay Cinar and Murat Koklu. Classification of rice varieties using artificial intelligence methods. *International Journal of Intelligent Systems and Applications in Engineering*, 7:188–194, 09 2019.