COMP534 First Assessment - Supervised learning methods for solving a classification problem

Dr. Blaine Keetch
23rd of February 2024

Project goal:

This project aims to compare the performance of various supervised learning methods on a binary classification problem, which will help to understand the advantages and disadvantages of each classification algorithm.

The dataset is provided in a ".csv" file, its basic format is as follows:

feature1	feature2	feature3	feature4	feature5	feature6	feature7	feature8	featue9	class
5	1	1	1	2	1	3	1	1	0
5	4	4	5	7	10	3	2	1	0
3	1	1	1	2	2	3	1	1	0
6	8	8	1	3	4	3	7	1	0
4	1	1	3	2	1	3	1	1	0
8	10	10	8	7	10	9	7	1	1
1	1	1	1	2	10	3	1	1	0

Project Introduction:

Write python code to compare the performance of three different classification methods, you can choose any three classification methods based on the following list:

- Decision Tree
- Random Forest
- SVM/kernel SVM
- KNN
- Naïve Bayes
- Logistic Regression

You can refer to any python libraries (pandas, numpy, matplotlib, seaborn, scikit-learn, ...) to use the classification methods. However, your code must include the following steps:

1. Indicate the imported packages/libraries

2. Load the dataset and print the data information

3. Understand the dataset
   - Print-out the number of samples for each class in the dataset
   - Plot some figures to visualize the dataset (e.g., histogram, box plots, scatterplots etc.)
   - For each class, print-out the statistical description of features (e.g., the input
   - variable x), such as mean, std, max and min values, etc.

4. Randomly split data into a training dataset and a testing dataset (i.e., 80% v.s. 20%)

5. **For each classification algorithm** you chose, please complete the below steps in Python:
   - Train the model using the training dataset.
   - - If there are hyperparameters in the algorithm, please use K-Fold Cross Validation (e.g., you could choose k = 5 for K-Fold Cross Validation) to tune the hyperparameters of the algorithm (e.g., explore the best value for hyperparameter “k” for KNN, or the best kernel for kernel SVM, etc.).
   - - Please use different evaluation metrics, including precision, recall and accuracy, F1-Score, to pick up a model that gives you the best result on the validation dataset (e.g., via the Cross Validation, for kNN model, which k value gives the best precision, recall, accuracy, and F1-Score respectively)
   - Test the model (the best one you obtained from the above stage) on the testing dataset
   - - Plot the confusion matrix
   - - Please use different evaluation metrics, including precision, recall and accuracy, F1-Score, to report the performance of the algorithm, you can use tables or plot figures to summarize the results

Submission:

The deadline is the **13th of March at 5PM.** You will need to submit your completed assessment on Canvas.

Submit a single zip file which includes a report and the python code, please write the name and student ID in the report, please name your zip file in the format of “lastname_firstname_studentID”.

Assessment:
The project assessment contains two parts:

1) Python code **(65%)**: You need to provide your Python code in .py or .ipynb script, which includes all the steps in Section Project Introduction, and add comments to describe/explain each line of your python code.

- Step 1 - Step 4 **(5%)**
- For each classification method, Step-5 accounts **20%** (since there are three different classification methods, overall it is **20%*3 = 60%**)

2) Report **(35%)**: A short report of a maximum of three pages (not considering the cover page and references, in case you want to include them). This report should contain:

   An introduction Section explaining the development process **(10%)**, including (but not limited to):

   - The libraries used.
   - The classification methods used, and discuss how you set up or select the hyperparameters for each classification method
   - The training and testing process (e.g., data-split ratio, cross-validation, how to set up or use API/libraries for training and testing, etc.)

   An evaluation section describing and explaining your results **(20%)**, including (but not limited to):

   - Describe and explain the confusion matrix of the best version of each classification method
   - Describe and explain the tables comparing the Precision, Recall, F1-Score, and accuracy of all methods

   Final conclusions **(5%)**, including (but not limited to):

- Your own analysis on the performance of different classification methods based on your own AI knowledge (you could discuss the pros and cons of each classification method, and your thoughts on why particular classification methods perform better than others in this dataset)

- Your own reflection on this project, for example, what you have learned via this project, which part you could do better next time, etc.