MACHINE LEARNING HW1

HPE Machine Learning
Julian
6/09/2024

Assignment Instructions

Your assignment is to create a Jupyter notebook that demonstrates how to do the following (use methods discussed in the materials shared in this class):

1. Load the dataset in the file named `BDOShoham.csv` and produce at least one table and one graph that summarize the dataset statistics; (4 points)

2. Set up a classification problem: predicting the `FlowPattern` value based on the values of the variables named `Vsl`, `Vsg`, and `Ang`, and split the dataset into separate training and test sets in a reproducible way; (4 points)

3. Train at least two models (e.g., k-NN, logistic regression) to solve this classification problem. Use the training set you created in part 2 to cross-validate the performance of each model. Report on three different scoring methods (e.g., accuracy, weighted precision, macro recall, f1 score); (6 points)

4. Pick a model and a scoring method from part 3. Use cross-validation to evaluate the improvement/degradation of performance when you modify at least two hyperparameters (e.g., `n_neighbors`, `weights`, `metric`, `penalty`) as compared to the model's default settings; (4 points)

5. Test the performance of the best model+hyperparameters combination you found in part 4, using the test set you created in part 2. Discuss your overall results. (4 points)

There is no "perfect solution." The objective of this assignment is to provide you with hands-on practice and an opportunity to learn. The goal is to see how the different choices you make in training affect the results that you obtain, not how to obtain the best performance in the class. Good luck!

1. Dataset upload and summary

The dataset has been summarized readily using the `pandas.describe` command. The flow patterns for analysis are 1,2,3,4,5 and 7. The other variables of interest are continuous variables. These summary statistics are well described by the box and whiskers plot.

	FlowPattern	Vsg	Ang	Vsl
count	5675.000000	5675.000000	5675.000000	5675.000000
mean	4.059912	6.222612	2.727401	0.899747
std	1.379238	8.699644	46.202822	1.425159
min	1.000000	0.093720	-90.000000	0.001100
25%	3.000000	0.162255	-10.000000	0.016000
50%	5.000000	1.600000	0.500000	0.250000
75%	5.000000	10.000000	20.000000	1.500000
max	7.000000	42.956200	90.000000	25.517000

Table 1: Summary statistics of the dataset

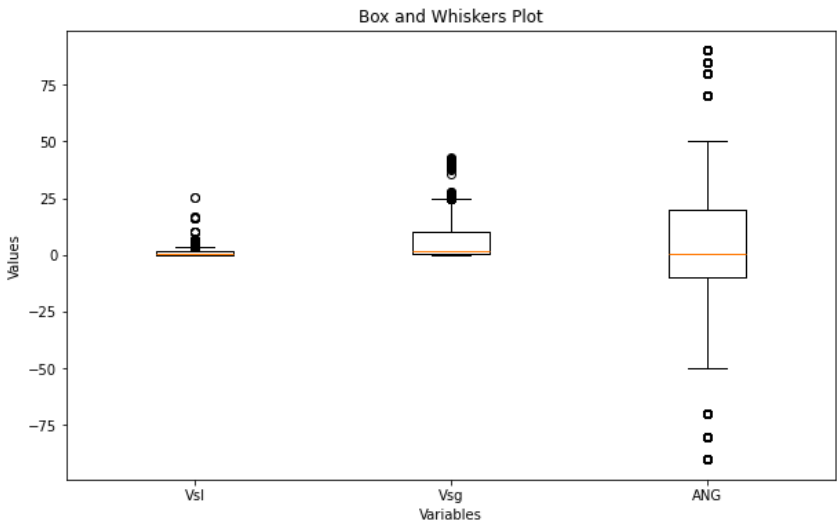

Figure 1: Box and Whiskers Plot

2. Setting up classification problem

In machine learning it is fundamental to avoid over-fitting by splitting our data into a training set and a testing set. This can be done in a reproducible way with the `model selection.train test split` command. We have reserved 20 percent of the data for testing.

3A. Logistic Regression Classification

We have trained the first classification model using logistic regression. We have cross-validated the performance, and scored the model on several metrics using the training set. We have cross validated using the f1 weighted score.

Flow Pattern	precision	recall	f1-score	support
1	0.84	0.56	0.67	478
2	0.00	0.00	0.00	113
3	0.42	0.12	0.19	694
4	0.66	0.64	0.65	816
5	0.67	0.92	0.78	2335
7	0.00	0.00	0.00	104
accuracy			0.67	4540
macro avg	0.43	0.37	0.38	4540
weighted avg	0.61	0.67	0.61	4540

Table 2: Classification report

F1 Scores for each fold: [0.62227382, 0.62696128, 0.60114643, 0.61326294, 0.60759094]
Mean score for default model: 0.6142470810065276

3B. KNN classification

We have trained the second classification model using KNN. Our initial hyperparameters are 3 neighbors, and the minowski metric. The model has been cross validated, and scored based on the training set.

	precision	recall	f1-score	support
1	0.91	0.92	0.92	478
2	0.93	0.93	0.93	113
3	0.91	0.96	0.93	694
4	0.92	0.96	0.94	816
5	0.96	0.94	0.95	2335
7	0.87	0.94	0.90	104
accuracy			0.94	4540
macro avg	0.92	0.94	0.93	4540
weighted avg	0.94	0.94	0.94	4540

Table 3: Classification report

F1 Scores for each fold: [0.86405313, 0.85577034, 0.85232023, 0.86029166, 0.84634502]
Mean score for default model: 0.8557524351331989

4. Modification of hyper parameters for KNN

We now do some basic tuning to the KNN model by changing the neighboring points to 5, and the metric to the euclidean distance. We compare the cross validation score.
Mean f1 score for changed model: 0.8486071113249408
F1 Scores for each fold: [0.85806141, 0.85770049, 0.84929183, 0.84039374, 0.83758809]
The mean score for the newer model is less than the default model.

5. Testing model

We now test to superior model (KNN with default settings) using the test set. We discuss the performance.

	precision	recall	f1-score	support
1	0.73	0.82	0.77	116
2	0.81	0.96	0.88	27
3	0.78	0.87	0.82	184
4	0.84	0.81	0.82	217
5	0.92	0.87	0.90	570
7	0.57	0.62	0.59	21
accuracy			0.85	1135
macro avg	0.78	0.82	0.80	1135
weighted avg	0.86	0.85	0.85	1135

Table 4: Classification report

In our classification report, the f1 scores appear to be plausible for all but the 7th flow pattern. Overall, the performance metrics are not as good as the ones using only the training set, but this is expected, otherwise it would probably be over fitting.