2017-01-20

Tom Rochette <tom.rochette@coreteks.org>

December 21, 2025 — 77e1b28a

0.1 Context

0.2 Learned in this study

- Using jupyter for data exploration
- Use various tools from the sklearn library
- It's useful to have a small utility library that will copy your train.py script such that you have an history of what your script was like at every run. This way if you need to backtrack to your "best" results, you can look at your tensorflow results, find the timestamp of the execution, and then find the related train.py script.

0.3 Things to explore

1 Problems faced

- I could not use sklearn right away due to the fact that my numpy library is 1.12.0b1+mkl. I had to edit sklearn/utils/fixes.py, line 406 from `if np_version < (1, 12, 0):` to `if np_version < (1, 12):`.
- Plots were very small initially.
- Pretty present matlibplot plots: when using a plot with multiple subplots, the axis labels often overlap with one another. I tried to use `tight_layout` but was unable to use it to display the data I wanted it to use.
- In the tutorial I followed, the author uses `dataset.plot(kind="box",...)` which supposedly only renders out the 4 inputs, while in my case it was also rendering the output. I was never able to only render out only my inputs without having to change the dataset variable itself.

2 Overview

Since I was able to make much progress in the last few days, I decided to take a different approach. I am far from an expert, so I decided I would follow some newbie tutorial on machine learning, but apply it to my data.

I followed the introductory tutorial available @ http://machinelearningmastery.com/machine-learning-in-python-step-by-step/. It allowed me to learn a few neat tricks that makes it possible for me to rapidly inspect the data (`head`/`tail`/`describe`/`plot(kind="box")`/`hist` on a dataset, `groupby(...).size()` to get an idea of classes distribution).

Through the tutorial, I was lead to try various models: logistic regression, linear discriminant analysis, k neighbors classifier, decision tree classifier, Gaussian naïve Bayes and support vector clustering.

Model: Cross-validation mean (std deviation)
LR: 0.466854 (0.006114)
LDA: 0.472471 (0.005050)
KNN: 0.568854 (0.005844)
CART: 0.794611 (0.005026)

NB: 0.165576 (0.005248)
SVM: No result (did not complete)

I was glad to see that the decision tree classifier was able to get 79%, compared to the ~40% I could get at best with my dense NN.

3 See also

4 References

- http://machinelearningmastery.com/machine-learning-in-python-step-by-step/
- http://stackoverflow.com/questions/40693558/typeerror-unorderable-types-str-int