09-Exercises

January 8, 2025

1 09 - Exercises: Machine Learning

This week we saw: - Introduction to machine learning: supervised and unsupervised learning methods - The Scikit-Learn package

Here are some exercises to help you get comfortable with these concepts :)

2 Basic ML - 8 points

Use Google, Wikipedia, ChatGPT, or any Machine Learning book you can find online to inform yourself about these new topics, and then reply to the following questions! Every answer should be no longer than 200 words, and can include long sentences, bullet-point lists, formulas, or whatever you feel like is necessary. Please, write the answers yourself! Remember that it's very easy to spot a LLM-generated answer ;)

1. What is the difference between supervised learning, unsupervised learning, and semi-supervised learning? (2P)
2. What is Lasso regression? Briefly describe its key properties. (2P)
3. How can you perform decision tree classification with Scikit-Learn? Briefly describe the classification algorithms that Scikit-Learn provides. (2P)
4. What is the perceptron algorithm? Why is it so important? (2P)

[ ]:

3 Intermediate ML - 4 points

Follow the tutorial at https://scikit-learn.org/stable/getting_started.html and summarize its main points with a bullet-point list.

[ ]:

4 Advanced ML - 18 points

4.1 The diabetes dataset - 8 points

Scikit-Learn provides the following diabetes data set that has been published by:

Bradley Efron, Trevor Hastie, Iain Johnstone and Robert Tibshirani (2004) "Least Angle Regres- sion," Annals of Statistics (with discussion), 407-499 (https://www.researchgate.net/publication/264955448_Least_Angle_Regression_with_discussions).

The authors describe the data set as having ten baseline variables: - Age, sex, body mass index, and average blood pressure, - and six blood serum measurements, obtained for each of the 442 diabetes patients, - as well as the **response of interest** (our label), a quantitative measure of disease progression one year after baseline.

Explore the dataset, take a look at the original publication, and when you feel ready, perform the following analyses: 1. Perform a **regression** analysis of the diabetes data set with Lasso regression. (2P) 2. Describe and visualize your results. (3P) 3. Compare your results with the response data (`diabetes.target`). (3P)

[11]:

```python
from sklearn.datasets import load_diabetes
import pandas as pd

diabetes = load_diabetes()
data = pd.DataFrame(diabetes.data, columns=diabetes.feature_names)

data.head()
```

[11]:

```
        age       sex       bmi        bp        s1        s2        s3  \
0  0.038076  0.050680  0.061696  0.021872 -0.044223 -0.034821 -0.043401
1 -0.001882 -0.044642 -0.051474 -0.026328 -0.008449 -0.019163  0.074412
2  0.085299  0.050680  0.044451 -0.005670 -0.045599 -0.034194 -0.032356
3 -0.089063 -0.044642 -0.011595 -0.036656  0.012191  0.024991 -0.036038
4  0.005383 -0.044642 -0.036385  0.021872  0.003935  0.015596  0.008142

         s4        s5        s6
0 -0.002592  0.019907 -0.017646
1 -0.039493 -0.068332 -0.092204
2 -0.002592  0.002861 -0.025930
3  0.034309  0.022688 -0.009362
4 -0.002592 -0.031988 -0.046641
```

[ ]:

```

```

4.2 The breast cancer Wisconsin diagnostic data set - 10 points

Another data set that Scikit-Learn provides is the breast cancer Wisconsin diagnostic data set that was first published by:

W.N. Street, W.H. Wolberg and O.L. Mangasarian. Nuclear feature extraction for breast tumor diagnosis. IS&T/SPIE 1993 International Symposium on Electronic Imaging: Science and Tech- nology, volume 1905, pages 861-870, San Jose, CA, 1993 (https://www.spiedigitallibrary.org/conference-proceedings-of-spie/1905/1/Nuclear-feature-extraction-for-breast-tumor-diagnosis/10.1117/12.148698.short).

The data set includes data of 569 patients and consists of features that are computed from a digitized image of a fine needle aspirate (FNA) of a breast mass. They describe characteristics of the cell nuclei present in the image.

Explore the dataset, take a look at the original publication, and when you feel ready, perform the following **classification** analyses: 1. Use cross validation in your analysis; justify your choice(s) of the number of partitions. (2P) 2. Run the analysis for all decision tree algorithms that Scikit-Learn provides. (3P) 3. Evaluate the classification quality of the algorithms with your (justified) choice of metric. (3P) 4. Visualize your results. (2P)

[4]:
```
from sklearn.datasets import load_breast_cancer
import pandas as pd

breast_cancer = load_breast_cancer()
data = pd.DataFrame(breast_cancer.data, columns=breast_cancer.feature_names)

data.head()
```

[4]:
```
   mean radius  mean texture  mean perimeter  mean area  mean smoothness  \
0        17.99         10.38          122.80     1001.0          0.11840
1        20.57         17.77          132.90     1326.0          0.08474
2        19.69         21.25          130.00     1203.0          0.10960
3        11.42         20.38           77.58      386.1          0.14250
4        20.29         14.34          135.10     1297.0          0.10030

   mean compactness  mean concavity  mean concave points  mean symmetry  \
0           0.27760          0.3001              0.14710         0.2419
1           0.07864          0.0869              0.07017         0.1812
2           0.15990          0.1974              0.12790         0.2069
3           0.28390          0.2414              0.10520         0.2597
4           0.13280          0.1980              0.10430         0.1809

   mean fractal dimension  …  worst radius  worst texture  worst perimeter  \
0                 0.07871  …         25.38          17.33           184.60
1                 0.05667  …         24.99          23.41           158.80
2                 0.05999  …         23.57          25.53           152.50
3                 0.09744  …         14.91          26.50            98.87
4                 0.05883  …         22.54          16.67           152.20

   worst area  worst smoothness  worst compactness  worst concavity  \
0      2019.0            0.1622             0.6656           0.7119
1      1956.0            0.1238             0.1866           0.2416
2      1709.0            0.1444             0.4245           0.4504
3       567.7            0.2098             0.8663           0.6869
4      1575.0            0.1374             0.2050           0.4000

   worst concave points  worst symmetry  worst fractal dimension
0                0.2654          0.4601                  0.11890
1                0.1860          0.2750                  0.08902
2                0.2430          0.3613                  0.08758
3                0.2575          0.6638                  0.17300
```

```
4                0.1625          0.2364                  0.07678

[5 rows x 30 columns]
```

```
[ ]:
```