Laboratorium: SVM

1 Cel/Zakres

- Klasyfikacja SVM.
- Skalowanie.
- Budowanie potoków.
- Regresja SVM.
- Poszukiwanie wartości hiperparametrów.

2 Przygotowanie danych dla klasyfikacji

Załaduj zbiory danych, które będą używane w klasyfikacji.

```
from sklearn import datasets
```

Pierwszy zbiór zawiera dane obrazów przypadków nowotworów piersi:

```
data_breast_cancer = datasets.load_breast_cancer(as_frame=False)
print(data_breast_cancer['DESCR'])
```

```
.. _breast_cancer_dataset:

Breast cancer wisconsin (diagnostic) dataset
--------------------------------------------

**Data Set Characteristics:**

:Number of Instances: 569

:Number of Attributes: 30 numeric, predictive attributes and the class

:Attribute Information:
    - radius (mean of distances from center to points on the perimeter)
    - texture (standard deviation of gray-scale values)
    - perimeter
    - area
    - smoothness (local variation in radius lengths)
    - compactness (perimeter^2 / area - 1.0)
    - concavity (severity of concave portions of the contour)
```

- concave points (number of concave portions of the contour)
- symmetry
- fractal dimension ("coastline approximation" - 1)

The mean, standard error, and "worst" or largest (mean of the three worst/largest values) of these features were computed for each image, resulting in 30 features. For instance, field 0 is Mean Radius, field 10 is Radius SE, field 20 is Worst Radius.

- class:
  - WDBC-Malignant
  - WDBC-Benign

:Summary Statistics:

	Min	Max
radius (mean):	6.981	28.11
texture (mean):	9.71	39.28
perimeter (mean):	43.79	188.5
area (mean):	143.5	2501.0
smoothness (mean):	0.053	0.163
compactness (mean):	0.019	0.345
concavity (mean):	0.0	0.427
concave points (mean):	0.0	0.201
symmetry (mean):	0.106	0.304
fractal dimension (mean):	0.05	0.097
radius (standard error):	0.112	2.873
texture (standard error):	0.36	4.885
perimeter (standard error):	0.757	21.98
area (standard error):	6.802	542.2
smoothness (standard error):	0.002	0.031
compactness (standard error):	0.002	0.135
concavity (standard error):	0.0	0.396
concave points (standard error):	0.0	0.053
symmetry (standard error):	0.008	0.079
fractal dimension (standard error):	0.001	0.03
radius (worst):	7.93	36.04
texture (worst):	12.02	49.54
perimeter (worst):	50.41	251.2
area (worst):	185.2	4254.0
smoothness (worst):	0.071	0.223
compactness (worst):	0.027	1.058
concavity (worst):	0.0	1.252
concave points (worst):	0.0	0.291
symmetry (worst):	0.156	0.664
fractal dimension (worst):	0.055	0.208

=== ====== ======

:Missing Attribute Values: None

:Class Distribution: 212 - Malignant, 357 - Benign

:Creator: Dr. William H. Wolberg, W. Nick Street, Olvi L. Mangasarian

:Donor: Nick Street

:Date: November, 1995

This is a copy of UCI ML Breast Cancer Wisconsin (Diagnostic) datasets.
https://goo.gl/U2Uwz2

Features 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.

Separating plane described above was obtained using
Multisurface Method-Tree (MSM-T) [K. P. Bennett, "Decision Tree
Construction Via Linear Programming." Proceedings of the 4th
Midwest Artificial Intelligence and Cognitive Science Society,
pp. 97-101, 1992], a classification method which uses linear
programming to construct a decision tree. Relevant features
were selected using an exhaustive search in the space of 1-4
features and 1-3 separating planes.

The actual linear program used to obtain the separating plane
in the 3-dimensional space is that described in:
[K. P. Bennett and O. L. Mangasarian: "Robust Linear
Programming Discrimination of Two Linearly Inseparable Sets",
Optimization Methods and Software 1, 1992, 23-34].

This database is also available through the UW CS ftp server:

ftp ftp.cs.wisc.edu
cd math-prog/cpo-dataset/machine-learn/WDBC/

|details-start|
References
|details-split|

- 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 Technology, volume 1905, pages 861-870,
  San Jose, CA, 1993.
- O.L. Mangasarian, W.N. Street and W.H. Wolberg. Breast cancer diagnosis and

```
  prognosis via linear programming. Operations Research, 43(4), pages 570-577,
  July-August 1995.
- W.H. Wolberg, W.N. Street, and O.L. Mangasarian. Machine learning techniques
  to diagnose breast cancer from fine-needle aspirates. Cancer Letters 77 (1994)
  163-171.

|details-end|
```

Drugi zawiera „klasyczny” zbiór parametrów irysów:

```
data_iris = datasets.load_iris()
print(data_iris['DESCR'])
```

```
.. _iris_dataset:

Iris plants dataset
--------------------

**Data Set Characteristics:**

:Number of Instances: 150 (50 in each of three classes)
:Number of Attributes: 4 numeric, predictive attributes and the class
:Attribute Information:
    - sepal length in cm
    - sepal width in cm
    - petal length in cm
    - petal width in cm
    - class:
            - Iris-Setosa
            - Iris-Versicolour
            - Iris-Virginica

:Summary Statistics:

============== ==== ==== ======= ===== ====================
                Min  Max   Mean    SD   Class Correlation
============== ==== ==== ======= ===== ====================
sepal length:   4.3  7.9   5.84   0.83    0.7826
sepal width:    2.0  4.4   3.05   0.43   -0.4194
petal length:   1.0  6.9   3.76   1.76    0.9490  (high!)
petal width:    0.1  2.5   1.20   0.76    0.9565  (high!)
============== ==== ==== ======= ===== ====================

:Missing Attribute Values: None
:Class Distribution: 33.3% for each of 3 classes.
:Creator: R.A. Fisher
:Donor: Michael Marshall (MARSHALL%PLU@io.arc.nasa.gov)
```

```
:Date: July, 1988

The famous Iris database, first used by Sir R.A. Fisher. The dataset is taken
from Fisher's paper. Note that it's the same as in R, but not as in the UCI
Machine Learning Repository, which has two wrong data points.

This is perhaps the best known database to be found in the
pattern recognition literature.  Fisher's paper is a classic in the field and
is referenced frequently to this day.  (See Duda & Hart, for example.)  The
data set contains 3 classes of 50 instances each, where each class refers to a
type of iris plant.  One class is linearly separable from the other 2; the
latter are NOT linearly separable from each other.

|details-start|
**References**
|details-split|

- Fisher, R.A. "The use of multiple measurements in taxonomic problems"
  Annual Eugenics, 7, Part II, 179-188 (1936); also in "Contributions to
  Mathematical Statistics" (John Wiley, NY, 1950).
- Duda, R.O., & Hart, P.E. (1973) Pattern Classification and Scene Analysis.
  (Q327.D83) John Wiley & Sons.  ISBN 0-471-22361-1.  See page 218.
- Dasarathy, B.V. (1980) "Nosing Around the Neighborhood: A New System
  Structure and Classification Rule for Recognition in Partially Exposed
  Environments".  IEEE Transactions on Pattern Analysis and Machine
  Intelligence, Vol. PAMI-2, No. 1, 67-71.
- Gates, G.W. (1972) "The Reduced Nearest Neighbor Rule".  IEEE Transactions
  on Information Theory, May 1972, 431-433.
- See also: 1988 MLC Proceedings, 54-64.  Cheeseman et al"s AUTOCLASS II
  conceptual clustering system finds 3 classes in the data.
- Many, many more …

|details-end|
```

Podpowiedź: funkcje `load_...` domyślnie zwracają obiekty numpy, ale jeżeli przekażemy im argument `as_frame=True`, elementy `data` oraz `target` będą strukturami pandas, a dodatkowo dostępny będzie element `frame`, ktory zawiera `data` połączone z `target`.

3 Klasyfikacja

1. Podziel zbiór danych na uczący i testujący w proporcjach 80/20.
2. Zbuduj modele klasyfikacji SVM dla średnich (*mean*) wartości cech `area` oraz `smoothness`; stwórz dwa modele:
   1. LinearSVC, z funkcją straty “hinge”,
   2. LinearSVC, z funkcją straty “hinge”, po uprzednim automatycznym skalowaniu wartości cech.
3. Policz dokładność (*accuracy*) dla ww. klasyfikacji osobno na zbiorze uczącym i testują-

cym, zapisz wartości na liście w kolejności: zbiór uczący bez skalowania, zbiór testujący bez skalowania, zbiór uczący ze m, zbiór testujący ze skalowaniem. Listę zapisz w pliku Pickle `bc_acc.pkl`.

4 pkt.

4. Czy skalowanie coś dało?
5. Ekperyment powtórz dla zbioru irysów; zbuduj model wykrywający, czy dany przypadek jest gatunku *Virginica* na podstawie cech: długość i szerokość płatka.
6. Policz dokładność (accuracy) dla w/w klasyfikacji osobno na zbiorze uczącym i testującym, zapisz wartości na liście w kolejności: zbiór uczący bez skalowania, zbiór testujący bez skalowania, zbiór uczący ze skalowanie, zbiór testujący ze skalowaniem. W.w. listę zapisz w pliku Pickle `iris_acc.pkl`.

   4 pkt.

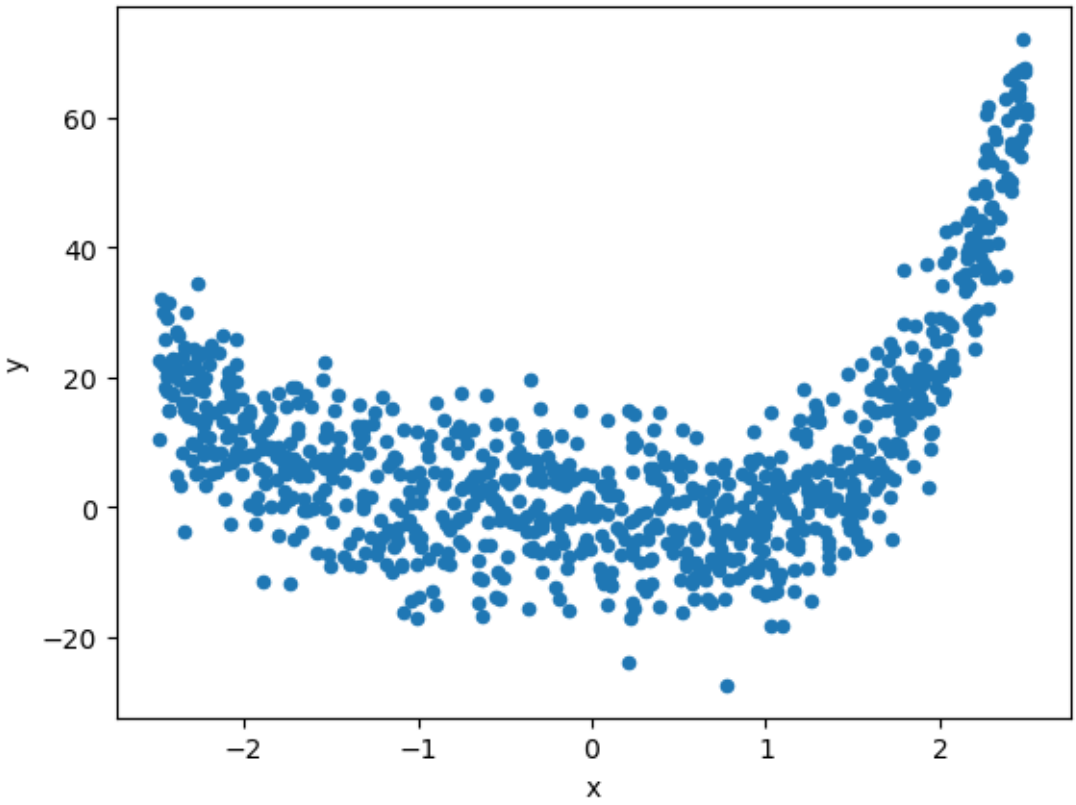
7. Czy skalowanie coś dało?

4 Przygotowanie danych dla regresji

1. Użyj tej samej funkcji co z laboratorium o regresji.

```
import numpy as np
import pandas as pd
size = 900
X = np.random.rand(size)*5-2.5
w4, w3, w2, w1, w0 = 1, 2, 1, -4, 2
y = w4*(X**4) + w3*(X**3) + w2*(X**2) + w1*X + w0 + np.random.randn(size)*8-4
df = pd.DataFrame({'x': X, 'y': y})
df.plot.scatter(x='x',y='y')
```

```
<Axes: xlabel='x', ylabel='y'>
```



2. Podziel zbiór uczący i testowy w proporcji 80:20.

5 Regresja

1. Zbuduj potok rozszerzający cechy do 4 wymiarów, za pomocą wielomianu 4 stopnia oraz regresora *LinearSVR* z domyslnymi parametrami.
2. Oblicz MSE dla zbioru uczącego i zbioru testowego. Wyniki powinny być podobne do najlepszych rezultatów z ćwiczenia o regresji, lub nawet lepsze.
3. Powtórz uczenie dla regresora *SVR* z kernelem *poly* 4 stopnia i pozostałymi parametrami z wartościami domyslnymi. Wyniki MSE powinny być … rozczarowujące.
4. Jakie hiperparametry użyć żeby *SVR* miał podobną jakość co *LinearSVR*? Użyj *GridSearchCV* na całym zbiorze danych (nie tylko uczącym!). Do znalezienia optymalnej pary parametrów *coef0* oraz *C*. Jak funkcje oceny zastosuj *neg_mean_squared_error*. Poszukaj optymalnych wartości spośród: `"C" : [0.1, 1, 10], "coef0" : [0.1, 1, 10]`.
5. Dla wyliczonych optymalnych wartości hiperparametrów przeprowadź proces uczenia *SVR* raz jeszcze. Oblicz wyniki MSE dla zbioru uczącego i testowego.
6. Zapisz wyniki MSE z punktu 2 i 5 na liście (4 elementy), którą następnie zapisz w pliku Pickle o nazwie: `reg_mse.pkl`.

   4 pkt.

6 Prześlij raport

Prześlij plik o nazwie `lab04/lab04.py` realizujący ww. ćwiczenia.

Sprawdzane będzie, czy skrypt Pythona tworzy wszystkie wymagane pliki oraz czy ich zawartość jest poprawna.