UNIVERSITÀ
DI PARMA
DIPARTIMENTO DI INGEGNERIA E ARCHITETTURA

Lab Machine Learning

Gianfranco Lombardo, Ph.D
gianfranco.lombardo@unipr.it

What we need today?

- Scikit-learn: pip install sklearn
- Matplotlib: pip install matplotlib
- Pandas: pip install pandas
- Numpy: pip install numpy

Ex. 1: Boston house price

- Regression task with 13 features

- from sklearn.datasets import load_boston
- X, y = load_boston(return_X_y=True)

- Divide the dataset in 70/30 train and test

- Train a linear regression model

- Repeat data splitting and training for 10 times and report the average root mean square error

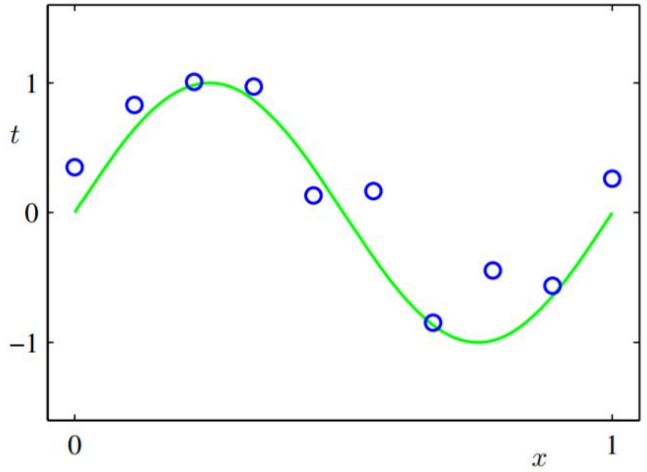
Ex. 3: Polynomial fitting and Regularization

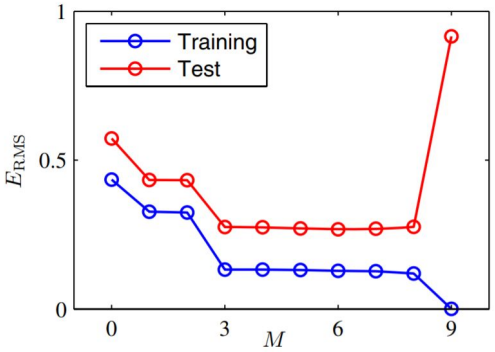
1. Try to replicate the experiment seen before while introducing Regularization methods

2. Try to fit sin(2πx)function with magnitude M from 1 to 9
   a. For each M make 5 different runs and take the average

3. Plot the RMSE for training and test as in Figure

4. Try to use L_2 Regularization to reduce overfitting





Ex. 3: Polynomial fitting and Regularization

```
import matplotlib.pyplot as plt
import numpy as np
from sklearn.preprocessing import PolynomialFeatures
def f(x):
    return np.sin(2*np.pi*x) + np.random.normal(scale=0.1, size=len(x))

NUM_SAMPLES=15
x = np.random.uniform(0,1,NUM_SAMPLES)
y = f(x)
plt.plot(x,y,'bo')
plt.show()
poly = PolynomialFeatures(degree=2)
X= poly.fit_transform(np.array(x).reshape(-1, 1))
```

Ex. 4: Breast cancer with Logistic

- X,y=datasets.load_breast_cancer(return_X_y=True)
- 0 is malignant and 1 is benign
- Split the dataset in 70% training and 30% test
- Train a logistic classifier with the default parameters in case of error use (solver=”liblinear”)
- If is it possible compute the accuracy and explain why
- Compute the confusion matrix and compute precision, recall and F1-score for each model which class is better classified?

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)
- **class:**
  - WDBC-Malignant
  - WDBC-Benign

Ex. 5: Iris plant dataset

Using the iris_dataset.csv:

- Read the dataset with pandas or with traditional python functions
- Divide columns in a X matrix and y vector
- Split the dataset in train and test to have both balanced
- Train a decision tree and a logistic and compare the two models in terms of accuracy
- For the best one compute also the confusion matrix
- If you try to compute the precision what happens?