Machine Learning
Exercise 3: Usage of different regression algorithms in python

Prof. Dr. Thomas Kopinski

July 10, 2023

Abstract

The goal of this exercise is to use python and scikit-learn to perform logistic and polynomial regression on different datasets.

Task 1: Getting familiar with the regression algorithms

- Further information and examples about the implementation of logistic regression can be found in this jupyter notebook.
- Additional information about polynomial regression in python can again be found here.
- Make sure that you understand the concepts you are trying to implement before starting to write any code.

Task 2: Logistic Regression

- Download the MNIST Dataset with the scikit-learn function:
  sklearn.datasets.fetch_openml("mnist_784")
- Look into the data and analyze the data structure (eg. data and label shape)
- Why is logistic regression a good approach for this dataset? What is the difference when compared to linear regression?
- Plot some samples of the data with matplotlib
- Split the data into training and test dataset
- Classify the MNIST dataset with a logistic regression model (scikit-learn)
- Evaluate the model performance using the accuracy
- Utilize and interpret the output of the two scikit-learn methods *proba()* and *predict_proba()*. What is the difference here?
- Display a sample of the wrong predictions to understand the models shortfalls

Task 3: Polynomial Regression

- What is the use-case for Polynomial Regression? Why and when would you use it instead of Linear Regression?
- Read the data
  x = np.arange(0, 30)
  y = [3, 4, 5, 7, 10, 8, 9, 10, 10, 23, 27, 44, 50, 63, 67, 60, 62, 70, 75, 88, 81, 87, 95, 100, 108, 135, 151, 160, 169, 179]

- Use matplotlib to display the data
- Create polynomial features of degree 2, 5 and 10 from the input data using the *sklearn.preprocessing.PolynomialFeatures()* function
- Use Linear Regression models to fit to the transformed and untransformed data.
- Plot the models together with the data and evaluate the model performances. What do you notice? What might be the problem here when using higher-degree polynomials?
- Repeat the steps above, but this time use the *numpy.polyfit()* method to fit a model to the data and compare it to the previous models