Exercises 5: Regression

Introduction to data science and Python programming for medical research (2025)

For these exercises you need to install the sklearn package. For windows users this would require you to open your conda terminal, found in the start menu, and then typing:

```
conda install sklearn
```

You will also need the packages previously used, numpy, matplotlib, pandas and scipy.

In these exercises you will familiarize yourselves with generating and loading data examples from sklearn and perform regression modelling.

The regression model classes in sklearn expect the *data matrix* `X` to be of 2D shape. You will need to reshape your 1D data to fit this requirement. Try yourselves or look in the examples from the lecture.

1 Load data for linear regression

Download the data file and put them in your working directory or any other directory where you are able to find them from within Python.

Use Pandas to load the data.

- Extract the data from the two columns in the DataFrame into two numpy arrays.
- Use sklearn to create a `LinearRegression` model and fit it to the data.
- Use matplotlib to plot the data points using `scatter` and `plot` to plot the line from your regression model. (Tip: Look into the examples from the lecture).

See examples from the lecture or at https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html on how to use the `LinearRegression` class.

2 Generate data for linear regression

Import the sklearn submodule for generating regression data.

- Create a 1D regression dataset with some noise.
- Use sklearn to create a `LinearRegression` model and fit it to the data.

- Use matplotlib to plot the data points using `scatter` and `plot` to plot the line from your regression model. (Tip: Look into the examples from the lecture).

See the documentation for the data generation at https://scikit-learn.org/stable/modules/generated/sklearn.datasets.make_regression.html

3 Diabetes linear regression

sklearn has several dataset examples. One is a diabetes dataset suitable for regression models.

https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_diabetes.html

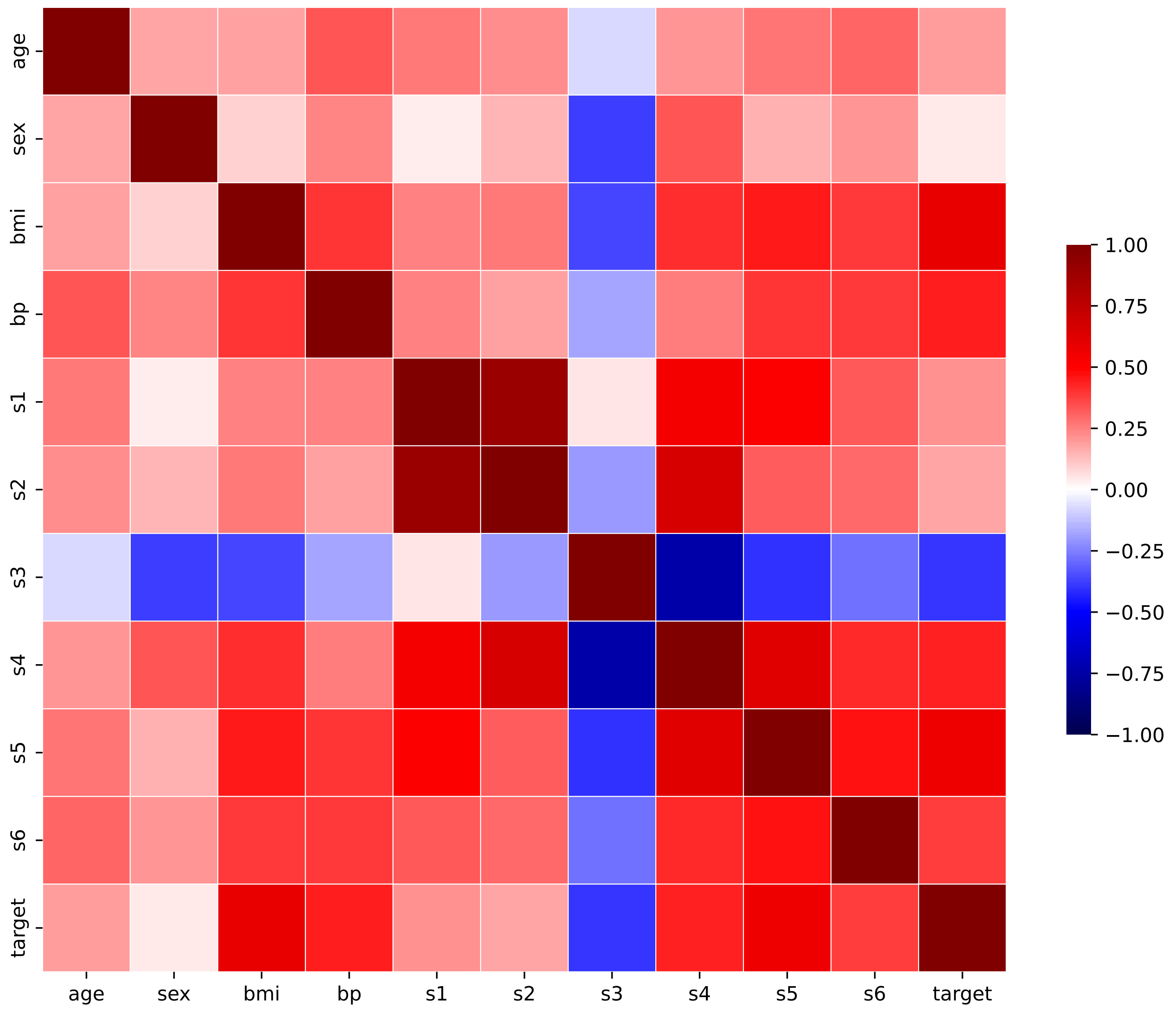
Use the following code to load the data into a pandas DataFrame.

```python
from sklearn.datasets import load_diabetes

# Load the diabetes dataset and extract the Pandas DataFrame
df = load_diabetes(as_frame=True)
df = df.frame
```

Inspect the correlation matrix below and select **one single** feature that you think should perform the best out of the ones available. Put that data in your `X` array.

The feature labeled as *target* is what you should put in your `y` array.



(Tip: You can index columns in pandas DataFrames using their names.)

- Extract the data from the two columns in the DataFrame into two numpy arrays.
- Use sklearn to create a `LinearRegression` model and fit it to the data.
- Use matplotlib to plot the data points using `scatter` and `plot` to plot the line from your regression model. (Tip: Look into the examples from the lecture).

4 Breast cancer logistic regression

sklearn also contains a breast cancer dataset suitable for classification models.
https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_breast_cancer.html

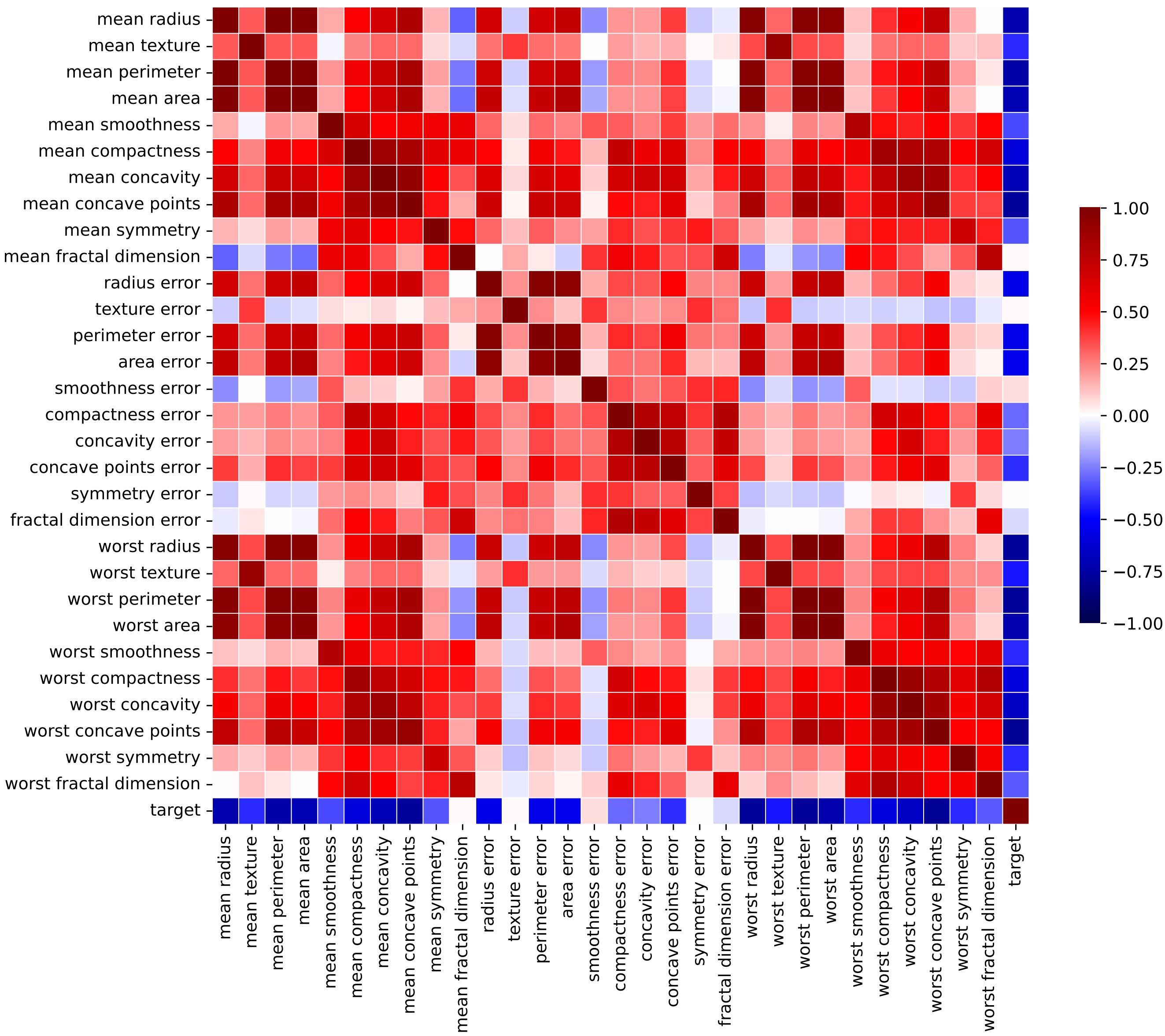
Use the following code to load the data into a pandas DataFrame.

```python
from sklearn.datasets import load_breast_cancer

# Load the diabetes dataset and extract the Pandas DataFrame
df = load_breast_cancer(as_frame=True)
df = df.frame
```

Inspect the correlation matrix below and select **one single** feature that you think should perform the best out of the ones available. Put that data in your `X` array.

The feature labeled as *target* is what you should put in your `y` array.



(Tip: You can index columns in pandas DataFrames using their names.)

- Extract the data from the two columns in the DataFrame into two numpy arrays.

- Use sklearn to create a `LogisticRegression` model and fit it to the data.
- Use the `predict(...)` function of your `LogisticRegression` instance to get predictions, labels, for the input data `X`. Use the true labels `y` and predicted labels `y_pred` to compute the performance of the model. You need to import classification validation functions from sklearn. *E.g.* precision, recall, accuracy of F1 score (https://scikit-learn.org/stable/api/sklearn.metrics.html.

5 Model validation

Chose one of the two previous exercises 3 or 4.

- Split the data into a test and a validation set using sklearns `train_test_split`.
- Perform the same regression or classification task as in the previous exercise but use only the data in the training set.
- Apply the estimator model by calling the `predict(X)` function on both the training and the test set data.
- Use a suitable metric (depending on whether you are using `LinearRegression` or `LogisticRegressi`
  to validate the performance of your model.

https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html
https://scikit-learn.org/stable/api/sklearn.metrics.html