Exercises 6: High dimensional data

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

If you have not done so already 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`

1 Vector operations

- Create two vectors, represented by numpy arrays. Print both vectors.
- Compute the length of the first vector. Print the length.
- Use the length to normalize the first vector to a unit vector.
- Project the second vector on the unit vector and print the projected vector.

2 Distance matrix

Import the sklearn submodules for generating *blob* data and computing pairwise distances.

You can generate any blob data you want but make sure that `shuffle=False` is given as argument.

```
import matplotlib.pyplot as plt
from sklearn.datasets import make_blobs
from sklearn.metrics import pairwise_distances
X, y = make_blobs(n_samples=50,
                  n_features=10,
                  centers=5,
                  cluster_std=1.0,
                  random_state=0,
                  shuffle=False)
```

- Compute the distance matrix `D` from the blob data matrix `X`.
- Create a figure and plot the distance matrix using `imshow`.
- What can you say from the structure of the matrix by looking at the plot?

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

3 Diabetes multilinear regression

Load the diabetes dataset for regression models that you used in the exercises for lecture 5. https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_diabetes.html

Use the following code to load the data into a data matrix `X` and target vector `y`.

```
from sklearn.datasets import load_diabetes
X, y = load_diabetes(return_X_y=True)
```

Instead of using one hand picked feature use the whole data matrix to create a multilinear regression model.

- Split the data into a test and a validation set using sklearns `train_test_split`.
- Use sklearn to create a `LinearRegression` model and fit it to the data.
- Apply the estimator model by calling the `predict(X)` function on both the training and the test set data.
- Use the R^2 metric to validate the performance of your model.
- **(Optional)** Repeat you model creation from the exercises from lecture 5, using only one of the features. Compare the performance of this model to your multivariate model.

See the documentation and previous examples/exercises for data splitting and validation metrics:
https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html
https://scikit-learn.org/stable/api/sklearn.metrics.html

4 Breast cancer multivariate logistic regression

Load the breast cancer dataset for classification models that you used in the exercises for lecture 5. https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_breast_cancer.html

Use the following code to load the data into a data matrix `X` and target vector `y`.

```
from sklearn.datasets import load_breast_cancer
X, y = load_breast_cancer(return_X_y=True)
```

Instead of using one hand picked feature use the whole data matrix to create a multivariate logistic regression model.

- Split the data into a test and a validation set using sklearns `train_test_split`.
- Use sklearn to create a `LogisticRegression` model and fit it to the data.
- Apply the estimator model by calling the `predict(X)` function on both the training and the test set data.
- Use *e.g.* precision, recall and F1 score metrics to validate the performance of your model.
- **(Optional)** Repeat you model creation from the exercises from lecture 5, using only one of the features. Compare the performance of this model to your multivariate model.

See the documentation and previous examples/exercises for data splitting and validation metrics:
https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html
https://scikit-learn.org/stable/api/sklearn.metrics.html