Exercises 10: Support vector machines, random forests and over fitting

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 Classification

Load the breast cancer dataset 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 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 and fit at least three regression models:
  1. `LogisticRegression`
  2. `RandomForestClassifier`
  3. `SVC` (Support vector classifier)
- Apply the estimator models by calling the `predict(X)` function on both the training and the test set data.
- Report their performances by printing the `confusion_matrix`.
- Use *e.g.* precision, recall and F1 score metrics to validate the performance of your models for both the training and the test set.

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

2 Regression

Load the diabetes dataset 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 data matrix `X` and target vector `y`.

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

Use the whole data matrix for your regression models.

- Split the data into a test and a validation set using sklearns `train_test_split`.
- Use sklearn to create and fit at least three regression models:
  1. `LinearRegression`
  2. `RandomForestRegressor`
  3. `SVR` (Support vector regressor)
- Apply the estimator models 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 models. Calculate the score for both the training and the test set and compare the resulting metrics.
- The behavior of the `SVR` estimator is primarily determined by its kernel type and the regularization parameter `C`, try the linear, poly and rbf kernels and use a small (10) and a high (1 000 000) for `C`. What happens with your R^2 scores?

See the documentation and previous examples/exercises for data splitting and validation metrics:
https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html
https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.

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