

Machine Learning with Python

Brief introduction with a
classification example

Prof. Dr. Andreas Theissler

Machine Learning with Python

scikit-learn



- ▶ <https://scikit-learn.org>
- ▶ In the python library **scikit-learn (sklearn)** a variety of machine learning methods is available
- ▶ **please refer to the online documentation of the library, with many working examples**

<https://scikit-learn.org>

For example the reference for the k-NN classifier can be found under:

<https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html>



The general principle for classification:

1. imports
2. load data set
3. split the data into training and test set: `train_test_split(...)`
4. preprocessing, typically scaling
 - ▶ note that the scaling parameters should be obtained from the training set and then be applied on the training and test set
5. creation of classification model, for example `KNeighborsClassifier` (with parameters)
6. model training using: `model.fit(...)`
7. use the model to classify data: `model.predict(...)`
8. analyse the results, e.g. with a confusion matrix and accuracy scores

This is a basic setting. In addition to that data might be plotted, or the entire process can be repeated to find the best model

Machine Learning with Python

First example classification



```
# sklearn imports
from sklearn.datasets import make_moons
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import confusion_matrix, accuracy_score
from sklearn.preprocessing import MinMaxScaler

# create data set
data, labels = make_moons(n_samples=500, noise=0.1, random_state=123)

# split into training and test set
train_data, test_data, train_labels, test_labels = train_test_split(
    data, labels, test_size = 0.5, random_state=123)
```

Machine Learning with Python

First example classification



```
# min-max scaling: determine scaling parameters  
scaler = MinMaxScaler().fit(train_data)
```

```
# scale train set and test set  
train_data = scaler.transform(train_data)  
test_data = scaler.transform(test_data)
```

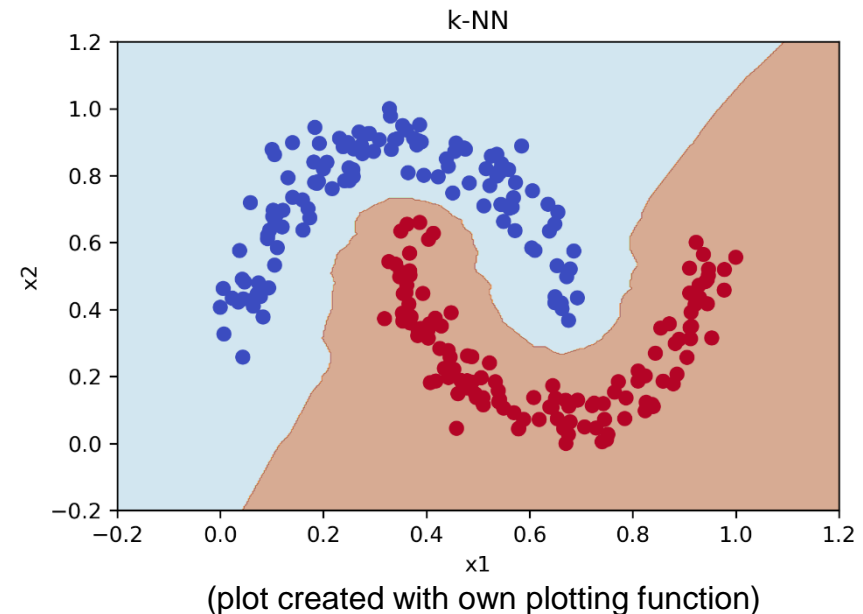
```
# create k-nearest neighbours with k = 3  
model = KNeighborsClassifier(n_neighbors = 3)
```

```
# train model on training set  
model.fit(train_data, train_labels)
```

```
# classification of test set  
predictions = model.predict(test_data)
```

```
acc = accuracy_score(test_labels, predictions)  
print("Overall accuracy: ", acc)
```

```
print("Confusion matrix")  
cm = confusion_matrix(test_labels, predictions)  
print(cm)
```



Machine Learning with Python

First example classification



- ▶ for other classification models, the source code requires only minor changes, e.g.:
 - ▶ the use of a different classes like SVC
 - ▶ or different preprocessing steps
 - ▶ or loading different data sets