Building a Classification Model - Iris Data set - Random Forest Classification

We will explore here a simple data set included in a standard Python package and a simple model for a prediciting certain values.

First we load the data set from the library.

In [11]:

```
from sklearn.datasets import load_iris
```

Loading iris.

In [17]:

```
iris = load_iris()
```

Inserting panda library in order to create the data set from the iris. Looking at the data labels.

In [18]:

```
import pandas as pd
```

In [19]:

```
dir(iris)
```

Out[19]:

```
['DESCR',
 'data',
 'feature_names',
 'filename',
 'frame',
 'target',
 'target_names']
```

Creating data set "data" using pandas and looking at the first 20 rows:

In [20]:

```
data = pd.DataFrame(iris.data, columns = iris.feature_names)
```

In [27]:

```
data.head(20)
```

Out[27]:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)
0	5.1	3.5	1.4	0.2

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)
1	4.9	3.0	1.4	0.2
2	4.7	3.2	1.3	0.2
3	4.6	3.1	1.5	0.2
4	5.0	3.6	1.4	0.2
5	5.4	3.9	1.7	0.4
6	4.6	3.4	1.4	0.3
7	5.0	3.4	1.5	0.2
8	4.4	2.9	1.4	0.2
9	4.9	3.1	1.5	0.1
10	5.4	3.7	1.5	0.2
11	4.8	3.4	1.6	0.2
12	4.8	3.0	1.4	0.1
13	4.3	3.0	1.1	0.1
14	5.8	4.0	1.2	0.2
15	5.7	4.4	1.5	0.4
16	5.4	3.9	1.3	0.4
17	5.1	3.5	1.4	0.3
18	5.7	3.8	1.7	0.3
19	5.1	3.8	1.5	0.3

Looking at the Featuer names, targets and target names.

In [26]:

```
print(iris.feature_names)
```

```
['sepal length (cm)', 'sepal width (cm)', 'petal length (cm)', 'petal width (cm)']
```

In [24]:

```
print(iris.target)
```

```
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

```
 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2
 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 2 2]
```

In [25]:

```
print(iris.target_names)
```

```
['setosa' 'versicolor' 'virginica']
```

We want to investigate the relationship between iris data and iris target. To give to the algorithm the data input and as an output receive a

In [56]:

```
X = iris.data
Y = iris.target
```

Importing Random Forest Classifier from ScikitLear, defining it as "classif" and fitting X and Y:

In [57]:

```
from sklearn.ensemble import RandomForestClassifier
```

In [58]:

```
classif = RandomForestClassifier()
```

In [59]:

```
classif.fit(X, Y)
```

Out[59]:

```
RandomForestClassifier()
```

Import classificaition and test in on two arrays from iris.data:

In [60]:

```
from sklearn.datasets import make_classification
```

In [61]:

```
X[(1)]
```

Out[61]:

```
array([4.9, 3. , 1.4, 0.2])
```

In [62]:

```
X[(23)]
```

Out[62]:

```
array([5.1, 3.3, 1.7, 0.5])
```

In [65]:

```
print(classif.predict(X[[23]]))
```

```
[0]
```

In [66]:

```
print(classif.predict_proba(X[[23]]))
```

```
[[1. 0. 0.]]
```

Import model selection and spliting features from ScikitLearn, classifying:

In [69]:

```
from sklearn.model_selection import train_test_split
```

In [72]:

```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.3)
```

In [71]:

```
classif.fit(X_train, Y_train)
```

Out[71]:

```
RandomForestClassifier()
```

Now let the RFC do the classification and prediction and print both values (predicted and actual):

In [74]:

```
print(classif.predict(X_test))
```

```
[0 0 1 2 2 2 0 1 2 2 2 0 2 1 2 1 0 1 0 2 1 1 1 0 1 1 0 1 1 2 2 2 2 0 1 1 1
 0 2 0 0 1 2 2 2]
```

In [75]:

```
print(Y_test)
```

```
[0 0 1 2 2 2 0 1 2 2 2 0 2 1 2 1 0 1 0 2 1 1 1 0 1 1 0 1 1 2 2 2 2 0 1 1 1
 0 1 0 0 1 2 2 2]
```

In [76]:

```
print(classif.score(X_test, Y_test))
```

```
0.9777777777777777
```

97% accuracy of the prediction.