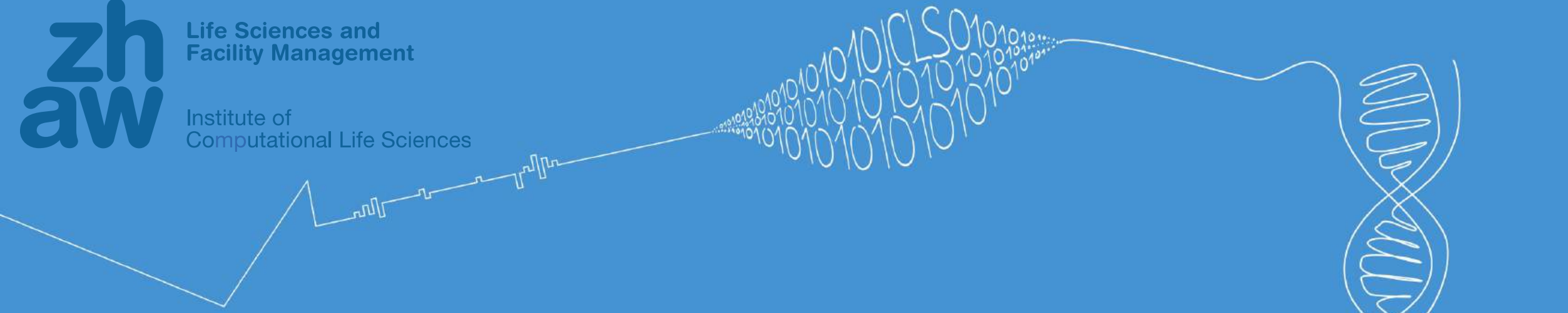
Life Sciences and
Facility Management

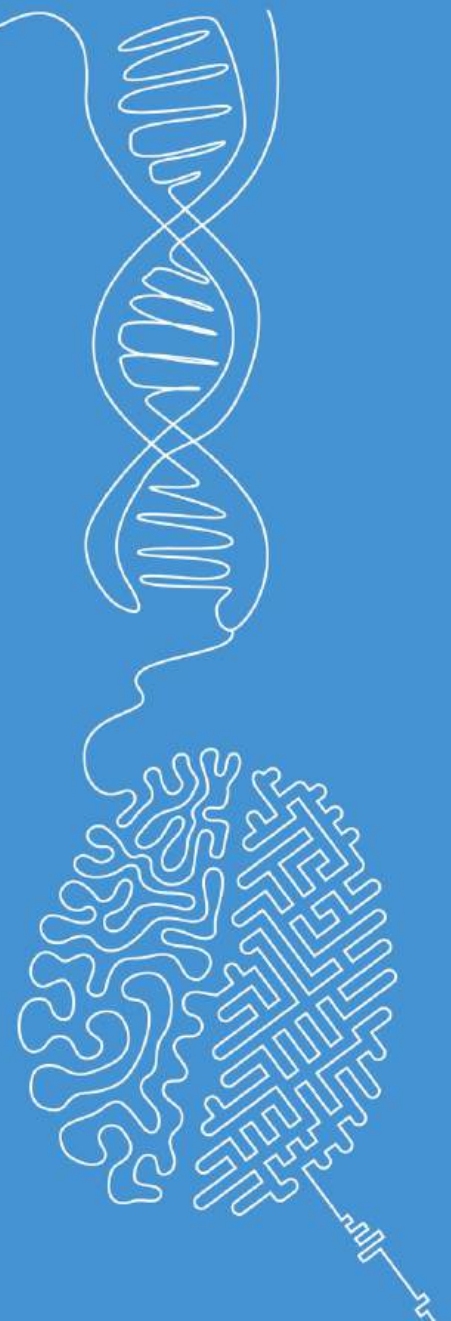
Institute of
Computational Life Sciences



A practical example

Machine Learning

Norman Juchler



Slide stack based on materials by Klaus Horn

Learning objectives

- Appreciate that building an ML model can be done in a few simple steps. (Don’t worry if you don’t understand every detail yet.)
- What the main steps of the machine learning modeling process are.
- How to perform them with scikti-learn in Python.

The problem: Iris flower classification

Iris Setosa

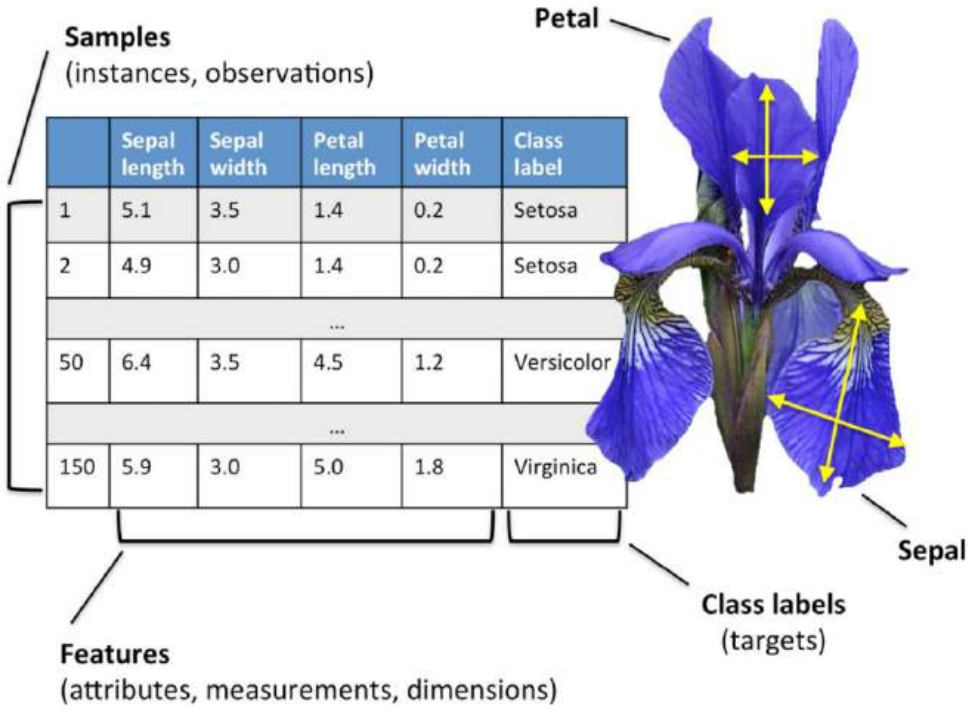
Iris Versicolor

Iris Virginica

Our aim: Develop a model that can tell the type of Iris flower

The iris flower dataset

- To build an ML model we need data!
- Feature extraction has already been done here:

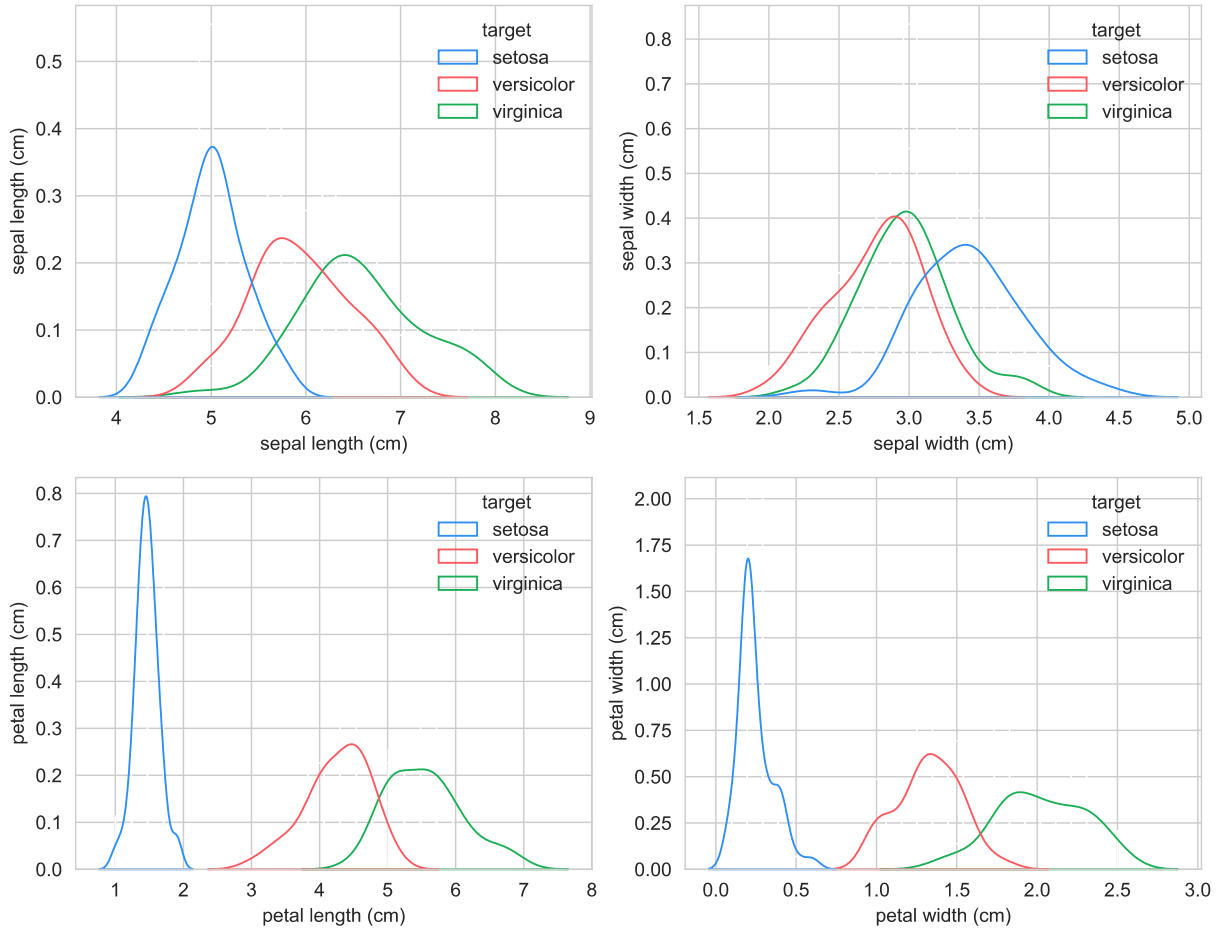


	Sepal length	Sepal width	Petal length	Petal width	Class label
1	5.1	3.5	1.4	0.2	Setosa
2	4.9	3.0	1.4	0.2	Setosa
...					
50	6.4	3.5	4.5	1.2	Versicolor
...					
150	5.9	3.0	5.0	1.8	Virginica

	A	B	C	D	E
1	Sepal Length	Sepal Width	Petal Length	Petal Width	Class
2	5.1	3.5	1.4	0.2	Iris-setosa
3	4.9	3	1.4	0.2	Iris-setosa
4	4.7	3.2	1.3	0.2	Iris-setosa
5	4.6	3.1	1.5	0.2	Iris-setosa
6	5	3.6	1.4	0.2	Iris-setosa
7	5.4	3.9	1.7	0.4	Iris-setosa
8	4.6	3.4	1.4	0.3	Iris-setosa
9	5	3.4	1.5	0.2	Iris-setosa
10	4.4	2.9	1.4	0.2	Iris-setosa
11	4.9	3.1	1.5	0.1	Iris-setosa
12	5.4	3.7	1.5	0.2	Iris-setosa
13	4.8	3.4	1.6	0.2	Iris-setosa
14	4.8	3	1.4	0.1	Iris-setosa
15	4.3	3	1.1	0.1	Iris-setosa
16	5.8	4	1.2	0.2	Iris-setosa
17	5.7	4.4	1.5	0.4	Iris-setosa
18	5.4	3.9	1.3	0.4	Iris-setosa
19	5.1	3.5	1.4	0.3	Iris-setosa
20	5.7	3.8	1.7	0.3	Iris-setosa
21	5.1	3.8	1.5	0.3	Iris-setosa
22	5.4	3.4	1.7	0.2	Iris-setosa
23	5.1	3.7	1.5	0.4	Iris-setosa
24	4.6	3.6	1	0.2	Iris-setosa
25	5.1	3.3	1.7	0.5	Iris-setosa

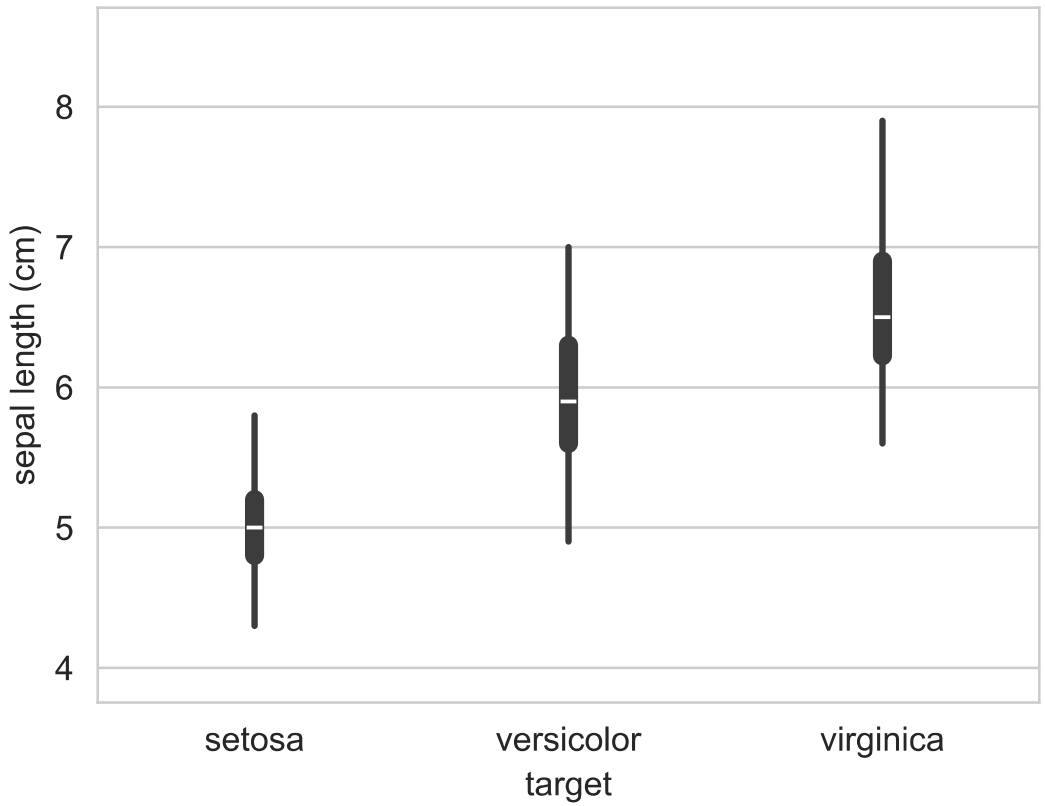
Step 1: Data exploration

- Understanding the dataset: Overlaps of univariate distributions



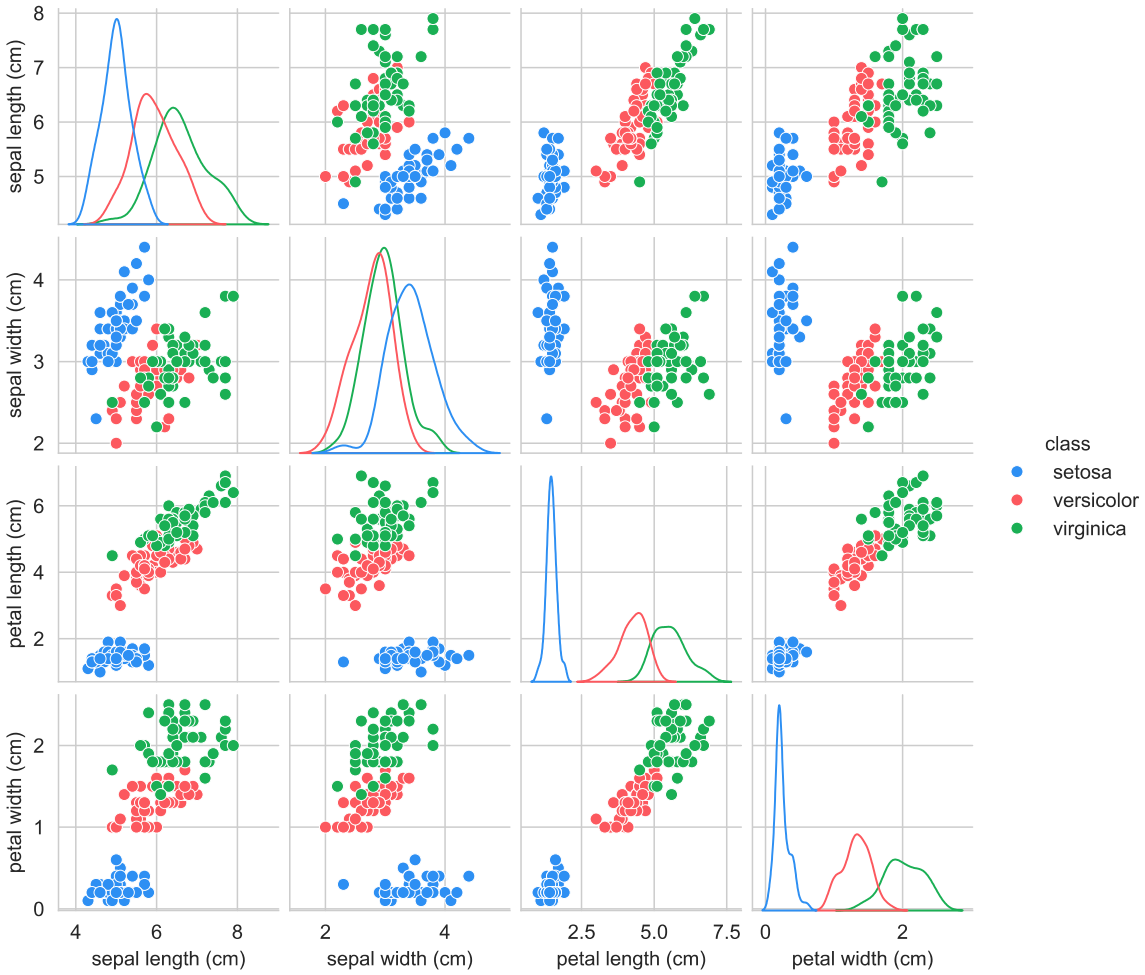
Step 1: Data exploration

- Understanding the dataset: Violine plots



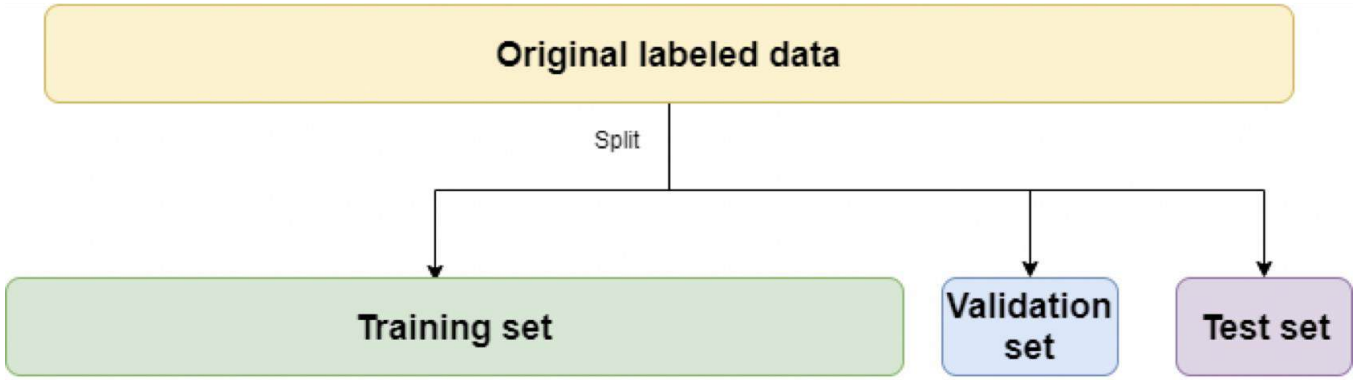
Step 1: Data exploration

- Understanding the data set: Correlations between features



Step 2: Data preparation

- Splitting the data into a train and a test set




```
from sklearn.model_selection import train_test_split

# Split the dataset into training set and test set
X_train, X_test, y_train, y_test = train_test_split(X, y,
                                                    test_size=0.3) # 70% for training
```

- Data cleaning, Replacement of missing values
- Encoding of features

Step 3: Building the model

- Many options:
  - Linear regression
  - Logistic regression
  - Naïve Bayes
  - Fisher discriminant
  - SVM
  - Random forest
  - Neural networks
  - etc.

```
# LogisticRegression
from sklearn.linear_model import LogisticRegression
classifier = LogisticRegression()
```

Hyperparameters go here

Step 4: Training the model

- There’s not much to it...

```
classifier.fit(X_train, y_train)
```

Step 5: Evaluating the model

- Make predictions on new samples (not seen during training).
  In ML-lingo, this is called **inference:**

```
y_pred = classifier.predict(X_test)
```

- Evaluate the predictive performance of the model based on one or more evaluation metrics:

```
from sklearn.metrics import accuracy_score
print('accuracy is',accuracy_score(y_pred,y_test))
```

Takeaways

- The main modeling steps are:
  - Data exploration
  - Data preparation
  - Model building,
  - Model training,
  - Evaluation

- The most important Python functions we used were:
  `train_test_split()`, `clf.fit()`, `clf.predict()`.

- Judging by the code, the machine learning part looks easier than the visualization part. ☺