Life Sciences and
Facility Management

Institute of
Computational Life Sciences

The ML workflow

Machine Learning

Norman Juchler

Update of the semester plan

CW	SW	Date	Topics
38	**1**	17.09.24	Introduction and overview
39	**2**	24.09.24	Basic concepts, types of ML problems
40	**3**	01.10.24	Data problems, exploratory analysis and preprocessing
41	**4**	08.10.24	The machine learning workflow
42	**5**	15.10.24	Supervised learning: Regression
43	**6**	22.10.24	Supervised learning: Classification
44	**7**	29.10.24	Decision trees, ensembles and boosting
45	**8**	05.11.24	Unsupervised learning: Clustering
46	**9**	12.11.24	Unsupervised learning: Dimensionality reduction
47	**10**	19.11.24	Model evaluation and selection
48	**11**	26.11.24	The machine learning workflow, revisited
49	**12**	03.12.24	Alternative learning paradigms
50	**13**	10.12.24	Common problems and challenges
51	**14**	17.12.24	Buffer / recapitulation
52	**15**	24.12.24	**Semester break**

Update of the semester plan

CW	SW	Date	Topics
38	**1**	17.09.24	Introduction and overview
39	**2**	24.09.24	Basic concepts, types of ML problems
40	**3**	01.10.24	The machine learning workflow
41	**4**	08.10.24	Data problems, exploratory analysis and preprocessing
42	**5**	15.10.24	Supervised learning: Regression
43	**6**	22.10.24	Supervised learning: Classification
44	**7**	29.10.24	Decision trees, ensembles and boosting
45	**8**	05.11.24	Unsupervised learning: Clustering
46	**9**	12.11.24	Unsupervised learning: Dimensionality reduction
47	**10**	19.11.24	Model evaluation and selection
48	**11**	26.11.24	The machine learning workflow, revisited
49	**12**	03.12.24	Alternative learning paradigms
50	**13**	10.12.24	Common problems and challenges
51	**14**	17.12.24	Buffer / recapitulation
52	**15**	24.12.24	**Semester break**

Talking of plans...

CW	SW	Date	Comments
38	**1**	16.09.24	
39	**2**	23.09.24	Input: Introduction to git / github / Jupyter
40	**3**	30.09.24	Input: Data platforms and resources
41	**4**	07.10.24	
42	**5**	14.10.24	**Deadline: Proposal of project works**
43	**6**	21.10.24	
44	**7**	28.10.24	**Start of individual project work**
45	**8**	04.11.24	
46	**9**	11.11.24	
47	**10**	18.11.24	
48	**11**	25.11.24	
49	**12**	02.12.24	
50	**13**	09.12.24	
51	**14**	16.12.24	
52	**15**	23.12.24	**Deadline: Submission of project work**

See document
“Dataset proposals”

Today’s lecture

Learning objectives

- How does a common workflow in machine learning look like?
- Which are the essential modeling steps

Recap: Machine learning models

- In ML, a **model** is a mathematical representation, function or algorithm that defines the relationship between input data and the desired output.

Model (e.g., classifier or regression model)

Model parameters

$$y = f(x|\theta)$$

$$x \in \mathbb{R}^d$$

Problem dimensionality
d = 1: univariate
d > 1: multivariate

$$\theta \in \mathbb{R}^p$$

$$y \in \bullet$$

Varies per problem type:

$y \in \mathbb{R}$: Regression
$y \in [0,1]$: Prob. assignment
$y \in \{0,1\}$: Binary classification

Response or labels
The information we want to predict. Often difficult or expensive to measure.

Predictors or features
The information we know about the subjects of interest. Usually easier to observe and access.

The data science workflow

Question: Suppose you want to solve a general data-driven problem. What do you think are the most important steps?

The data science workflow

- Problem definition
- Data acquisition
- Preprocessing
- Exploratory data analysis
- Feature engineering
- Modeling and training
- Model evaluation
- Model deployment
- Reporting

The data science workflow

- **Problem definition**
- Data acquisition
- Preprocessing
- Exploratory data analysis
- Feature engineering
- Modeling and training
- Model evaluation
- Model deployment
- Reporting

- Clearly define the problem
  - Formulate a hypothesis
  - Specify a business goal
- Identify the objectives and the questions you aim to answer using data.
- Determine the success criteria for your project.
- Identify data requirements and constraints

The data science workflow

- Problem definition
- **Data acquisition**
- Preprocessing
- Exploratory data analysis
- Feature engineering
- Modeling and training
- Model evaluation
- Model deployment
- Reporting

- Identify data sources
  - Websites (like Kaggle, zenodo, ...)
  - APIs / bots / tools
  - Internal databases
  - Surveys / interviews
- Collect the data
- Monitor data quality
  - Accuracy
  - Completeness
  - Consistency
  - Formats
  - ...

The data science workflow

- Problem definition
- Data acquisition
- **Preprocessing**
- Exploratory data analysis
- Feature engineering
- Modeling and training
- Model evaluation
- Model deployment
- Reporting

- Clean, transform, and organize the data to ensure that it's suitable for modeling and analysis.
- This may involve
  - Handle missing data, duplicates, and outliers.
  - Convert data into a usable format (e.g., handling dates, strings, and categorical variables).
  - Normalize or standardize data if necessary.
  - Remove or correct inconsistent or erroneous entries.

The data science workflow

- Problem definition
- Data acquisition
- Preprocessing
- **Exploratory data analysis**
- Feature engineering
- Modeling and training
- Model evaluation
- Model deployment
- Reporting

- Conduct preliminary investigations to discover patterns, relationships, and anomalies.
- Use descriptive statistics and visualizations to summarize the data.
  - Histograms
  - Box plots
  - Scatter plots
- Understand feature distributions, correlations, and trends.

The data science workflow

- Problem definition
- Data acquisition
- Preprocessing
- Exploratory data analysis
- **Feature engineering**
- Modeling and training
- Model evaluation
- Model deployment
- Reporting

▪ Transform raw data into a format that is more suitable for modeling.

▪ Extract features that are sensitive to the target variable(s), increase the signal to noise ratio

▪ Feature engineering is important for model accuracy, reducing overfitting, and enhancing the generalization capability of models

▪ Represents one way of introducing inductive bias into the model, and to exploit prior knowledge.

The data science workflow

- Problem definition
- Data acquisition
- Preprocessing
- Exploratory data analysis
- **Feature engineering**
- Modeling and training
- Model evaluation
- Model deployment
- Reporting

- Example: Lung cancer:
  - Most lung cancer variants form nodules that are visible on X-rays (or PET-CT).
  - If we want to predict the type of lung cancer or the treatment outcome based on medical images, it can be beneficial to focus on these nodules.
  - Creating features to enhance nodules or measure their properties (number of nodules, total volume, shape, ...) can improve prediction results.

The data science workflow

- Problem definition
- Data acquisition
- Preprocessing
- Exploratory data analysis
- Feature engineering
- **Modeling and training**
- Model evaluation
- Model deployment
- Reporting

- Choose appropriate machine learning or statistical modeling method based on the problem
  - Regression, classification, clustering, transformation
- Split data into training, validation, and testing sets.
- Train the model using the **training data**,
- Adjust hyperparameters if necessary.
- Perform cross-validation to assess model performance on unseen data and generalization.

The data science workflow

- Problem definition
- Data acquisition
- Preprocessing
- Exploratory data analysis
- Feature engineering
- Modeling and training
- **Model evaluation**
- Model deployment
- Reporting

- Evaluate the trained model's performance using the **test set**.
- Calculate relevant metrics
  - Classification: accuracy, precision, recall, F1-score, ...
  - Regression: root mean squared error, coefficient of determination, ...
  - ...
- Compare the model's performance on both the training and test sets to check for overfitting or underfitting.
- Fine-tune the model or try alternative algorithms if needed.

The data science workflow

- Problem definition
- Data acquisition
- Preprocessing
- Exploratory data analysis
- Feature engineering
- Modeling and training
- Model evaluation
- **Model deployment**
- Reporting

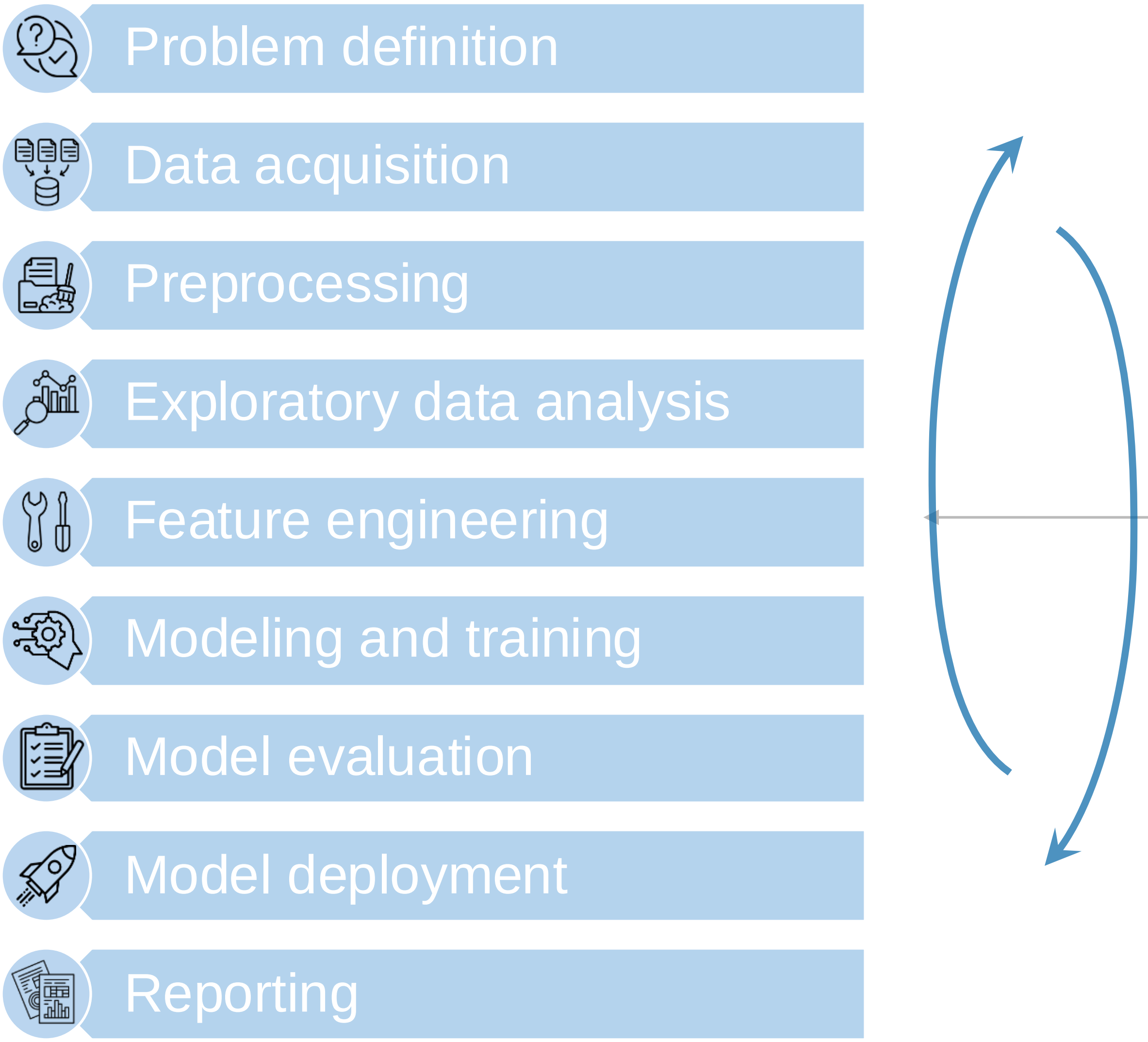
- Implement the model into production environments
- Ensure the model can scale and handle new data effectively.
- Set up monitoring to ensure the model maintains its performance over time.
- Detect concept drift!
  - The input data may change
  - Assumptions may not be valid anymore
- Update or retrain the model as necessary when new data becomes available, or the input data changes

The data science workflow

- Problem definition
- Data acquisition
- Preprocessing
- Exploratory data analysis
- Feature engineering
- Modeling and training
- Model evaluation
- Model deployment
- **Reporting**

- Communicate the insights, model results, and recommendations to stakeholders.
- Create visualizations, dashboards, or presentations to summarize findings.
- Translate technical results into actionable insights for decision-making.

The data science workflow



- Implementing the workflow is an **iterative process**
- Repeat or revisit stages of the workflow based on new findings or feedback.

Techniques and methods

Preprocessing: One-hot encoding

- **Problem**: Many machine learning methods require numerical data. However, we might encounter categorical predictors in our data.
- **Solution**: Convert categorical variables into a numerical format by mapping each each category onto a binary vector (a.k.a. dummy variables).

Feature (Color)
Red
Green
Yellow
Green
Red

One Hot Encoding

One Hot Encoded Vector	Red	Green	Yellow
[1,00]	1	0	0
[0,1,0]	0	1	0
[0,0,1]	0	0	1
[0,1,0]	0	1	0
[1,00]	1	0	0

Source / good read: https://www.scaler.com/topics/data-science/one-hot-encoding/

Preprocessing: One-hot encoding

- **Problem**: Many machine learning methods require numerical data. However, we might encounter categorical predictors in our data.
- **Solution**: Convert categorical variables into a numerical format by mapping each each category onto a binary vector (a.k.a. **dummy variables**).
- **Issues**:
  - Problem: A mapping of m categories to m features Introduces collinearity.
    Solution: Only use m-1 variables (drop one dummy variable)

Feature (Color)
Red
Green
Yellow
Green
Red

One Hot Encoding →

One Hot Encoded Vector	Red	Green	Yellow
[1,00]	1	0	0
[0,1,0]	0	1	0
[0,0,1]	0	0	1
[0,1,0]	0	1	0
[1,00]	1	0	0

Source / good read: https://www.scaler.com/topics/data-science/one-hot-encoding/

Preprocessing: One-hot encoding

- **Problem**: Many machine learning methods require numerical data. However, we might encounter categorical predictors in our data.
- **Solution**: Convert categorical variables into a numerical format by mapping each each category onto a binary vector (a.k.a. **dummy variables**).
- **Issues**:
  - Problem: A mapping of m categories to m features Introduces collinearity.
    Solution: Only use m-1 variables (drop one dummy variable)

Feature (Color)
Red
Green
Yellow
Green
Red

One Hot Encoding →

Red	Green
1	0
0	1
0	0
0	1
1	0

Yellow Column dropped to avoid the Dummy Variable Trap

Source / good read: https://www.scaler.com/topics/data-science/one-hot-encoding/

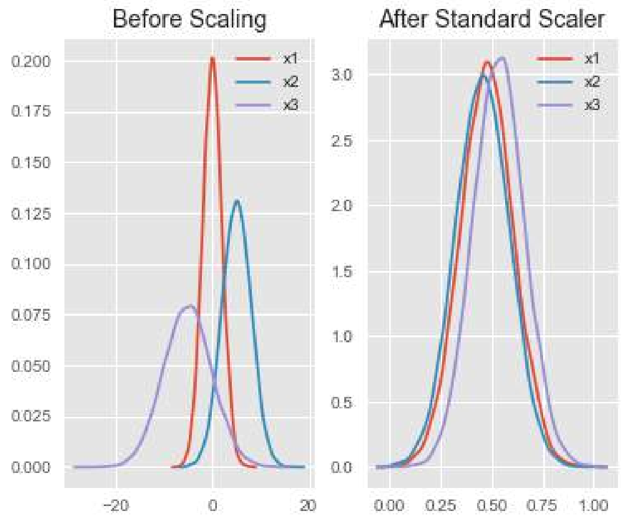
Preprocessing: One-hot encoding

- **Problem**: Many machine learning methods require numerical data. However, we might encounter categorical predictors in our data.
- **Solution**: Convert categorical variables into a numerical format by mapping each each category onto a binary vector (a.k.a. **dummy variables**).
- **Issues**:
  - Problem: A mapping of m categories to m features Introduces collinearity.
    Solution: Only use m-1 variables
  - Problem: Increases the dimensionality of the feature space (especially for many categories)
    Solution: Reduce categories, or apply dimensionality reduction techniques

Source / good read: https://www.scaler.com/topics/data-science/one-hot-encoding/

Preprocessing: Feature scaling

- **Problem**:
  - Some algorithms are sensitive to the scale of the data.
  - Features with different ranges can dominate the learning process, leading to biased models.
  - Numerical problems may arise for improperly scaled features, affecting the convergence speed for optimization algorithms (like gradient descent)



- **Solution**: Normalize or standardize features
- **Methods**:
  - Min-Max scaling (normalization):
    - Rescale the features such that all values are in the range [0, 1]

$$X_{scaled} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

  - Z-score normalization (standardization):
    - Rescale the features such that the features have a mean $\mu=0$ and a standard deviation $\sigma=1$

$$X_{scaled} = \frac{X - \mu}{\sigma}$$

Pipelines (scikit-learn)

- A sequence of data processing components is called a data pipeline.
- They are helpful if a set of operations always need to be applied in sequence.
- Example:

```
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import StandardScaler

num_pipeline = Pipeline([
                ('imputer', SimpleImputer(strategy="median")),
                ('attribs_adder', CombinedAttributesAdder()),
                ('std_scaler', StandardScaler())])

housing_num_tr = num_pipeline.fit_transform(housing_num)
```

Model development

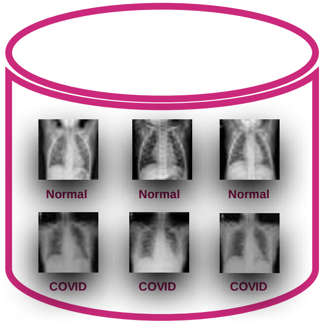
How does machine learning work?

$$f(\ X\ ) = prediction$$

How does machine learning work? Training

TRAINING PHASE

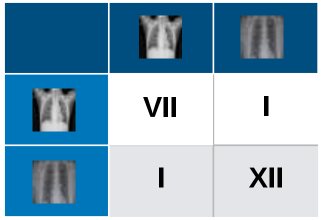
Training Dataset

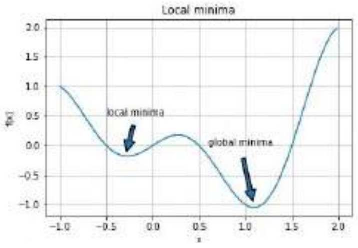


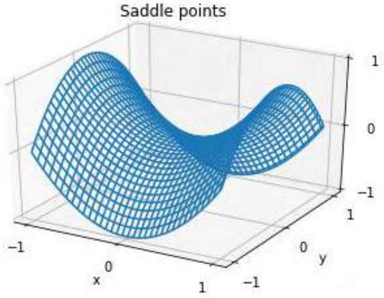
$\{x_i \in R^d, y_i \in R\}_{i=1}^{n}$

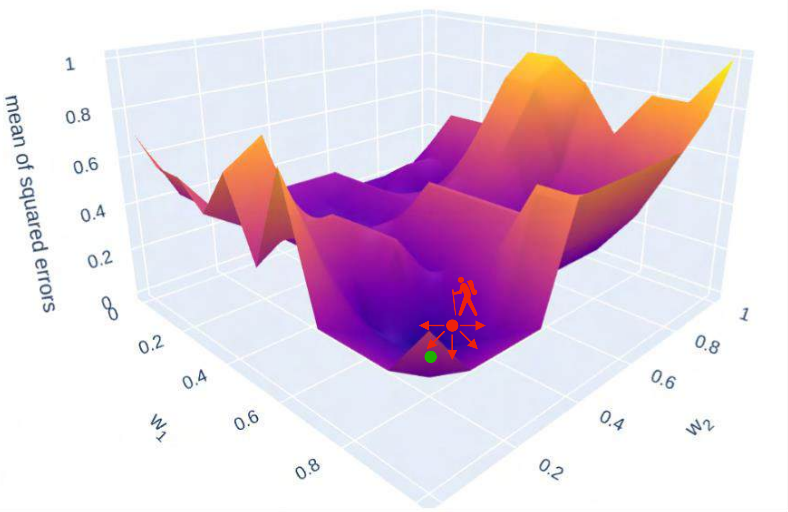
① Train → Machine Learning Algorithm → ② Predict → $f(\ )$

③ Compute Error





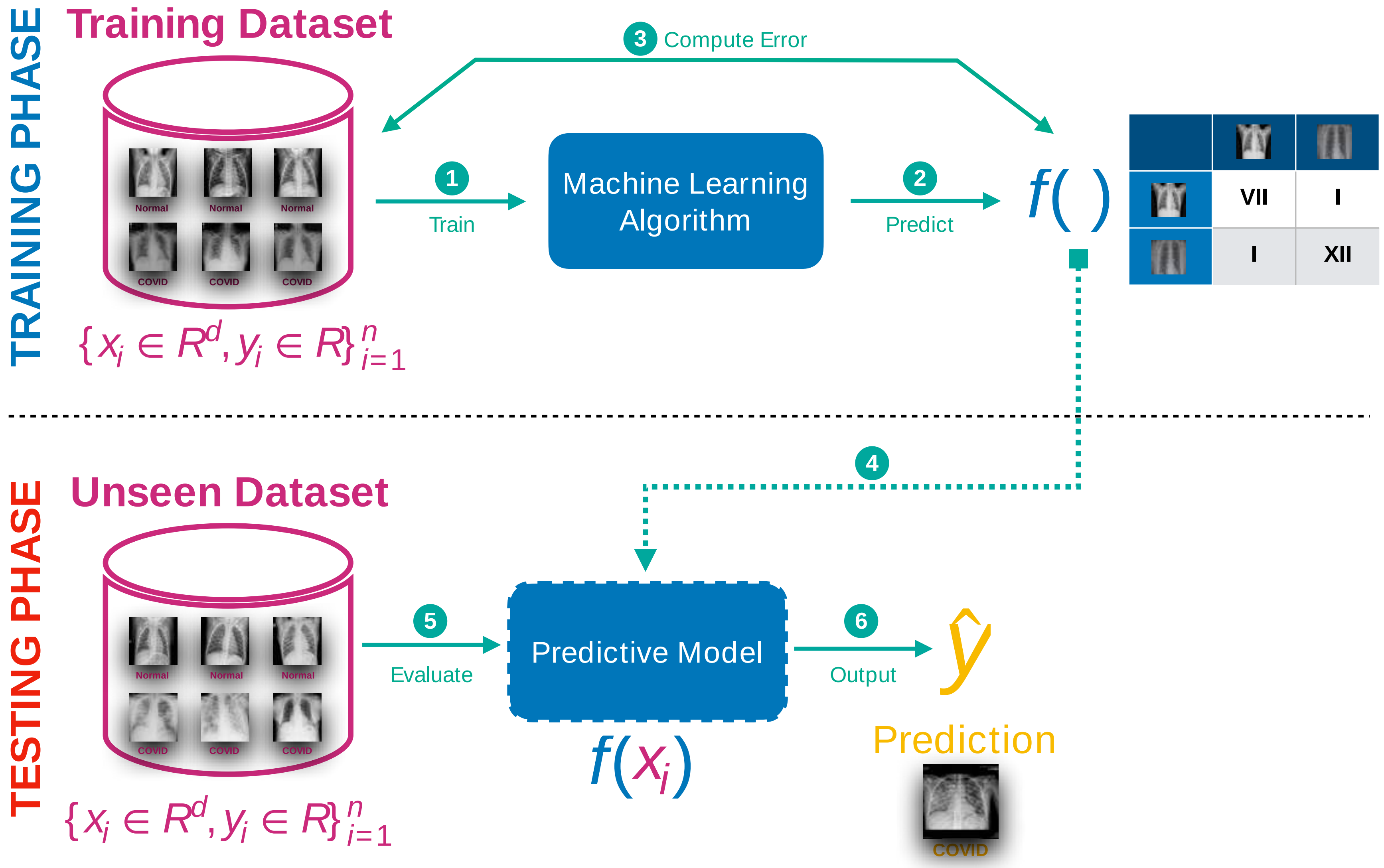




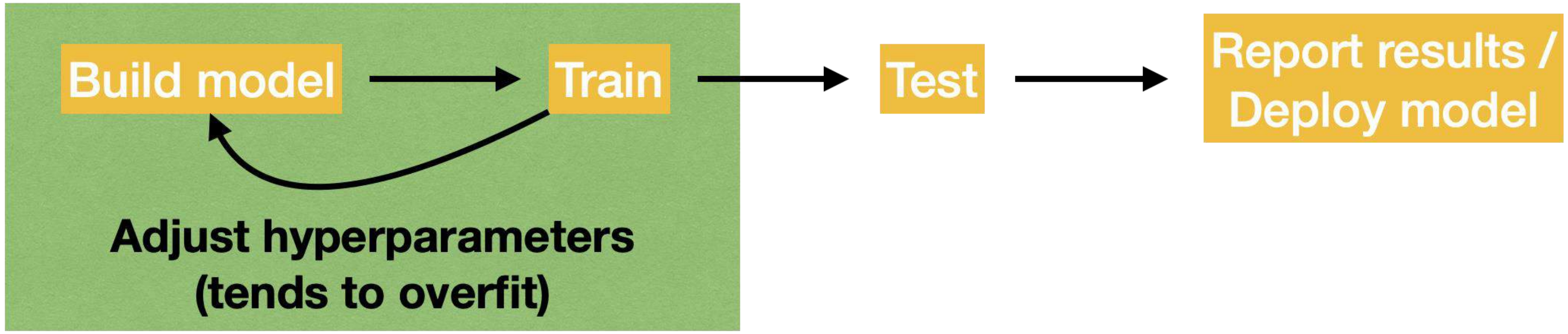
$$w_1 = w_1 - \eta \frac{\partial L}{\partial w_1}$$

$$w_2 = w_2 - \eta \frac{\partial L}{\partial w_2}$$

How does machine learning work? Testing

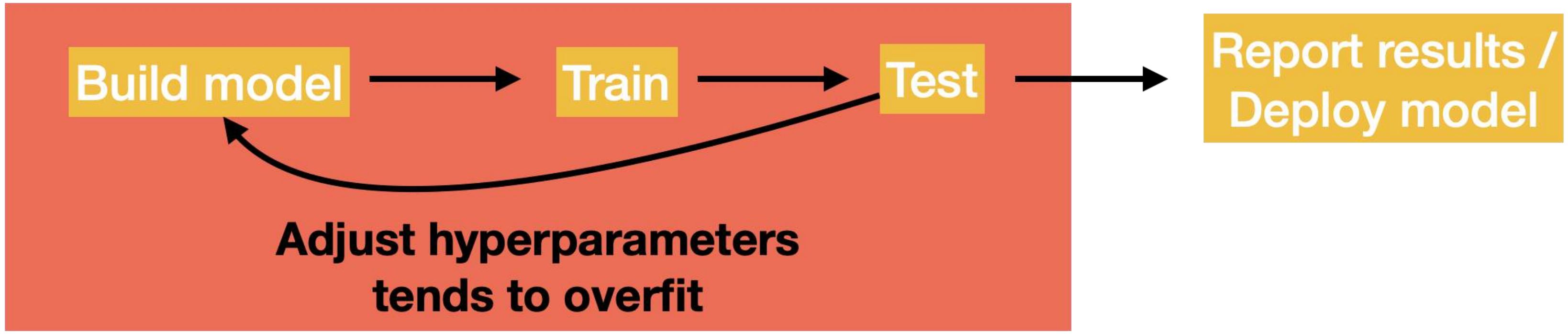


Working with training and test data



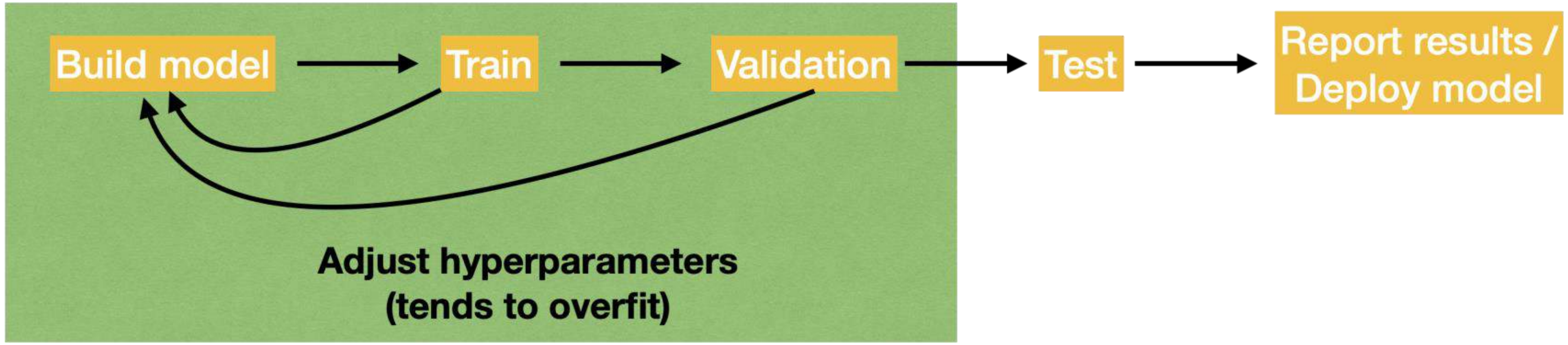
- This is **OK**!
- We might be overfitting, but it does not bias the final performance evaluation.
- **Problem**: We can not separate the amount of overfitting due to training and hyperparameter tuning.

Working with training and test data



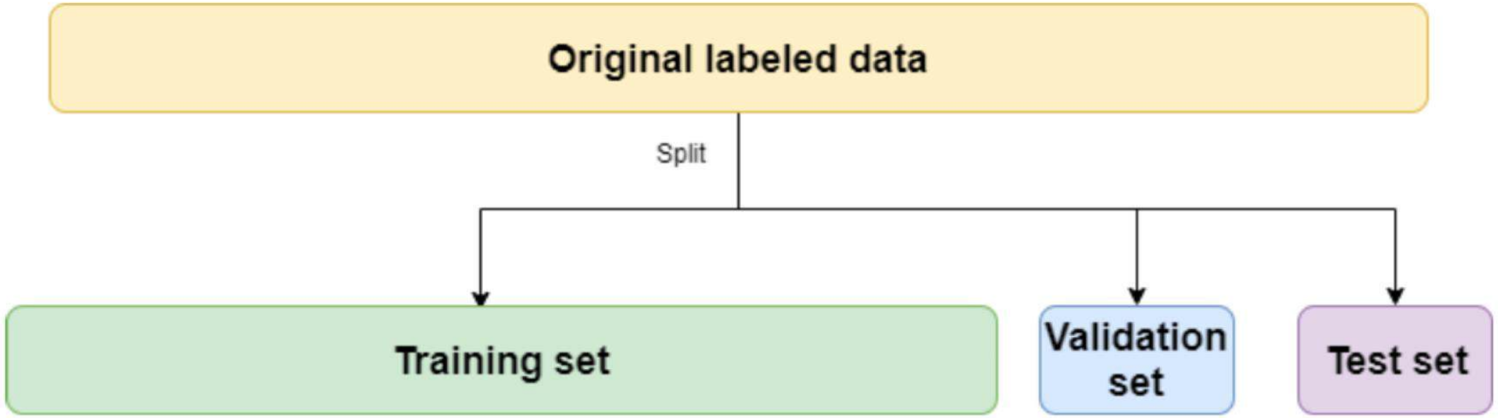
- **This is NOT OK, and needs be avoided at all cost!**
- Afterwards the measured test set performance is meaningless!
- We also should avoid any other subtle way information from the test set might slip into the model building (e.g. normalizing data or assessing outliers before train/test split).

Working with training and test data



- **The recommended approach!**
- Keep a separate validation set to tune hyperparameters.

Sub-datasets for training, validation and testing



- **Question**: How big should the training/validation/test sets be?
- **Hints**:
  - The training set should be as large as possible
  - Test and validation datasets large enough so that performance uncertainty can be estimated

Steps during modeling (detailed)

1. Creation of training, validation, and test sets
2. Model selection, feature engineering, and feature selection
3. Creation of an **in-sample model** which shows acceptable performance on the training set (some overfitting is acceptable, to determine the maximal model complexity supported by the data set)
4. Adding of regularization
5. Hyperparameter tuning