KIOS Research and Innovation Center of Excellence

MSc on Intelligent Critical Infrastructure Systems

Machine Learning

Lecture 7: Feature engineering, Evaluation

Dr Kleanthis Malialis
Research Associate
KIOS Research and Innovation Center of Excellence
University of Cyprus

University of Cyprus

Imperial College London

FUNDED BY:

Announcements

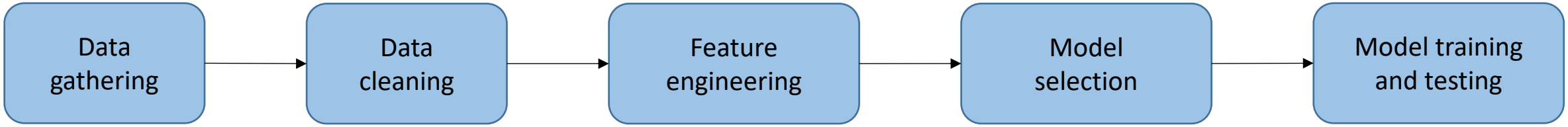
- **Homework 1**
  - **Correction**: Exercise 1a, least squares function
  - **Deadline extension**: 11 Apr. 2022, 16:00

Course outline

- **Week 1**
  - Introduction and Preliminaries
- **Week 2**
  - Linear Regression
  - Regularisation, Logistic Regression, SVMs
- **Week 3**
  - Neural Networks and Deep Learning
- **Week 4**
  - Feature Engineering and Evaluation
  - Online Learning
- **Week 5**
  - Unsupervised Learning
- **Week 6**
  - Reinforcement Learning
- **Week 7**
  - Advanced topics & Applications

- **Assignments**
  - 31/03 – 11/04
  - 08/04 – 15/04
  - 15/04 – 22/04

Steps for Machine Learning



Feature Engineering

Feature Engineering

- Feature engineering is the process of using domain knowledge of the data to create features that enhance machine learning algorithms. Basically, it transforms raw data into a dataset.
- If feature engineering is done correctly, it increases the predictive ability of machine learning algorithms.
- Feature engineering is an art!

STEPS FOR ML

- Data gathering
- Data cleaning
- ***Feature engineering***
- Model selection
- Model training and testing

Dealing with categorical data

- **Integer** vs. **one-hot** encoding
- If the order of the feature's values is not important then using integer encoding may confuse the learning algorithm.
  - E.g., colours are not sequential!
- **Drawbacks**
  - It can create many new features.
  - It introduces sparsity.

colour
red
blue
green

1-hot encoding →

red	blue	green
1	0	0
0	1	0
0	0	1

integer encoding →

new
0
1
2

Dealing with high-cardinality features

- **Frequency-based** approach:
  - Keep the values that correspond to the most frequent ones (e.g., 90%).
  - Group the rest as “Other” (e.g., 10%).

- **Prediction power-based** approach:
  - Group as “Other” those which have less predictive power

Binning (or Bucketing)

- Converting a **numerical** feature into a categorical feature
- **Example 1**: 3 bins

$$\alpha = \mu - k \times \sigma$$
$$\beta = \mu + k \times \sigma$$

$$x = \begin{cases} 0 \; if \; x \leq a \\ 1 \; if \; a < x < b \\ 2 \; if \; x \geq b \end{cases}$$

- **Example 2:** representation of age
  - Instead of age (e.g., 1 – 100) use age bins (e.g., 1-12 child, 13-17 teenager, 18 – 59 adult, 60 – 100 senior).
- **Advantages**
  - Binning can help the learning algorithm to learn using fewer examples.
  - Similar to giving “hints” to the learning algorithm.

Dealing with missing values

- Remove the examples with missing features.
- Use a **data imputation** method:
  - Replace the missing value of a feature by an average value of this feature in the dataset, or the middle of the range values.
  - Use regression to estimate the missing feature value.
  - Replace the missing value by the most frequent value of a feature.
  - Replace the missing value with a value outside the normal range of values for that feature.
- Use ML algorithms that can handle missing features.

Dealing with skewed features

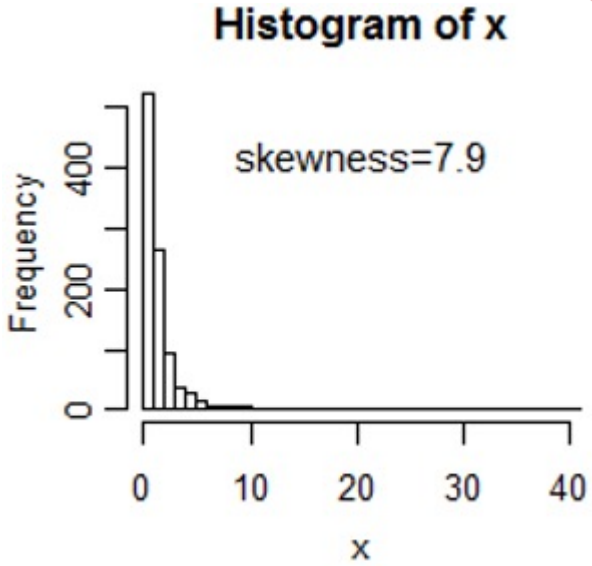
- **Identify skewness**
  - Visual inspection.
  - Calculate metrics, e.g., skewness.
- **Log** transformations

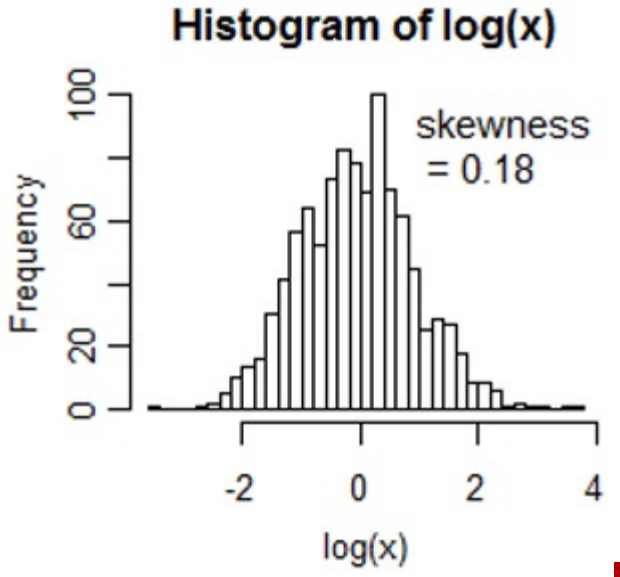
$$x \leftarrow \log(x + c)$$

- **Power** transformations

$$x \leftarrow (x)^p$$

- Use a learning algorithm which remains unaffected by skewed features.





Dealing with outliers

- The presence of outliers may “confuse” a learning algorithm.
- Identify outliers:

$$x < \mu - 2.5\,\sigma \quad x > \mu + 2.5\,\sigma$$

$$x < Q1 - 1.5\,IQR \quad x > Q3 + 1.5\,IQR$$

- Discard outliers
  - **Note**: It may not be an option in some areas, e.g., CIs or Healthcare.
- Choose a learning algorithm that is robust to outliers.

Normalization and Standardization

- **Normalization** converts the raw range of numerical feature into a standard range of values (usually, [-1, 1] or [0, 1]).

$$x \leftarrow \frac{x - min}{\max - min}$$

- **Standardization** rescales the numerical values of a feature so that it has a standard normal distribution (mean = 0; standard deviation = 1).

$$x \leftarrow \frac{x - \mu}{\sigma}$$

- Both normalization and standardization may improve the learning speed, and also help avoid numerical overflow problems.

Which one to use?

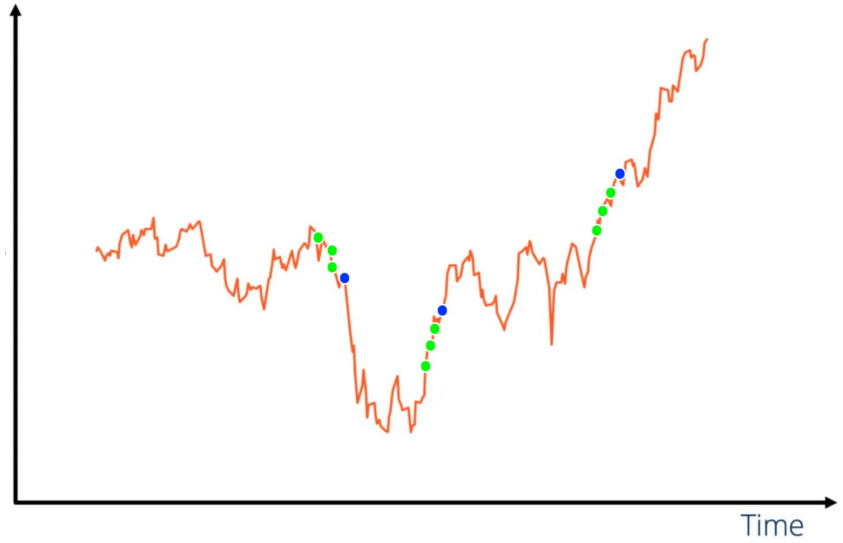
- No clear winner!
- Rule of Thumb: use normalization except in the following cases:
  - If the values of a feature are close to a bell curve
  - If the feature has extremely high or low values (outliers)
- Not always needed.

Dealing with low-variance features

- If the variance of a feature is (close to) zero, then the feature is (approximately) constant.
- These features are likely not to contain sufficient information to contribute to the prediction.
- Tuning is required to set an appropriate variance threshold.

Dealing with time-series data

- Temporal correlations affect traditional learning algorithms.
- Sliding window approach
  - Form new examples $\boldsymbol{x^t} \rightarrow < \boldsymbol{x^{t-W}}, \ldots, \boldsymbol{x^{t-1}}, \boldsymbol{x^t} >$
  - Use $< \boldsymbol{x^{t-W}}, \ldots, \boldsymbol{x^{t-1}}, \boldsymbol{x^t} >$ as the input to the learning algorithm (instead of $\boldsymbol{x^t}$).
  - Feature extraction on the window (e.g., min, max, mean, standard deviation).



- Choose an algorithm than can handle time-series data (e.g., recurrent neural networks).
- Types of applications: forecasting, time series regression and classification.

Evaluation

Learning algorithm selection

- Number of features and examples
- In-memory vs. out-of-memory
- Type of data (e.g., tabular, images, time-series)
- Type of features (categorical, numerical)
- Nonlinearity of data
- Training speed and prediction speed
- Explainability (Explainable AI)

→ Possible to try various learning algorithms and select one by testing on validation test.

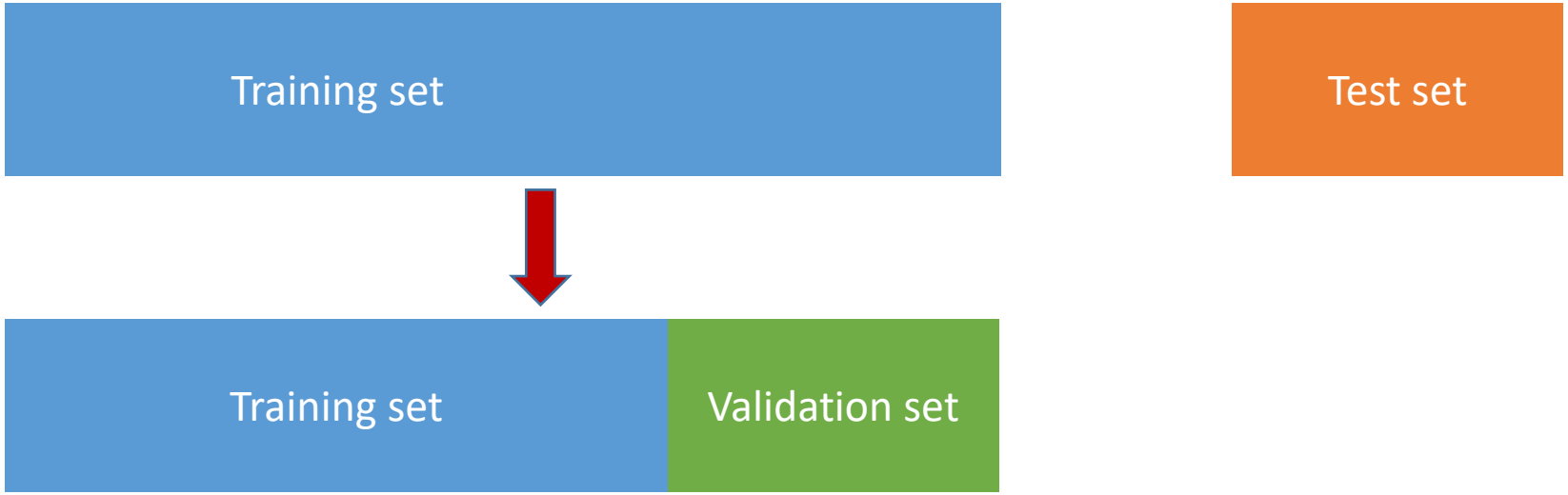
- **Training set**
- **Validation set**
- **Test set**

Model selection and assessment

- **Model selection**
  - Estimating the performance of different models in order to choose the best one.

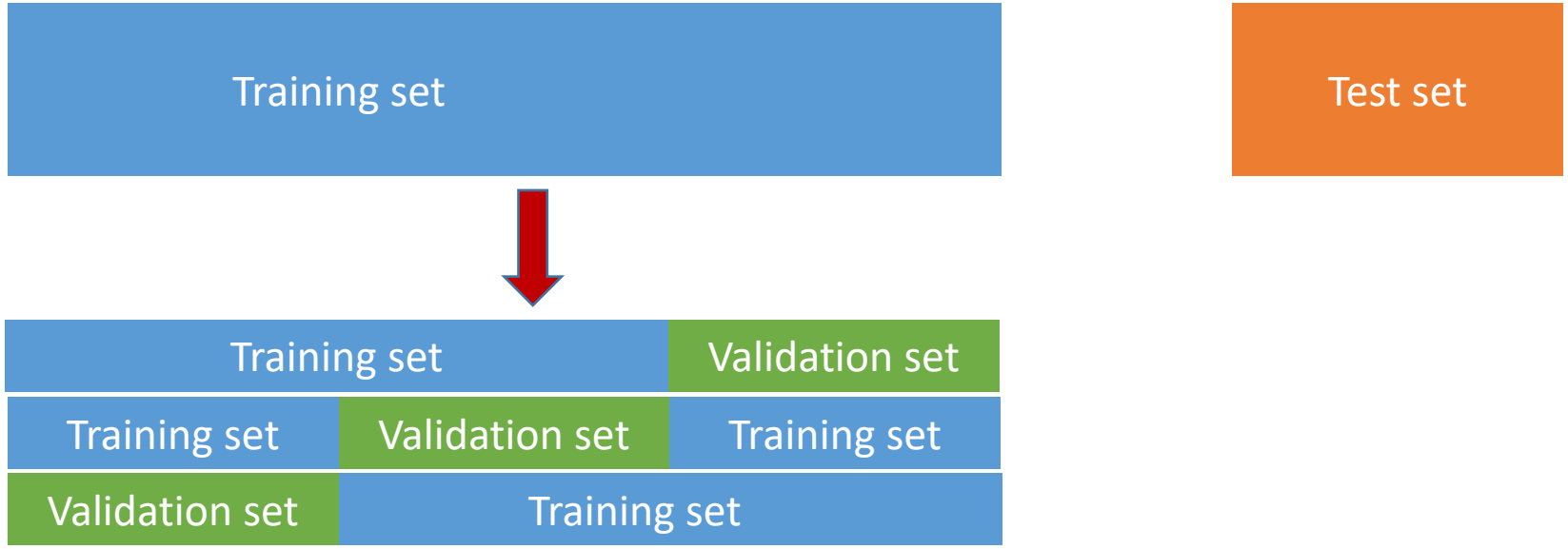
- **Model assessment**
  - Having chosen a final model, estimating its prediction (i.e., generalization) error on new data.

Holdout Cross-Validation



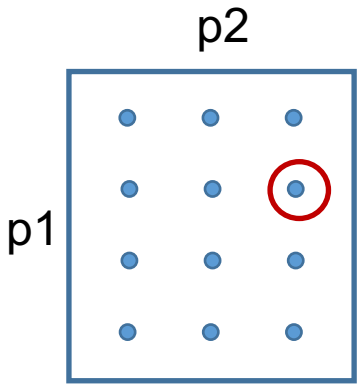
- **Training set**: is used to fit the models.
- **Validation set**: is used to estimate prediction error for model selection.
- **Test set**: is used for assessment of the generalization error of the final chosen model.
  - "Ideally, the test set should be kept in a "vault," and be brought out only at the end of the data analysis".
- **Notes**:
  - Use stratified splits.
  - The validation and test sets should come from the same distribution.

k-Fold Cross-Validation

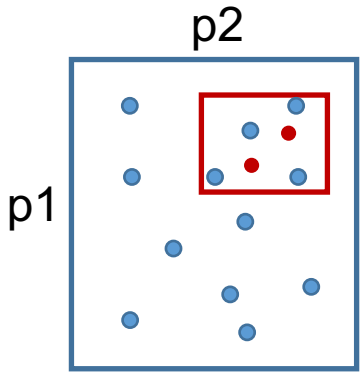


- Typical choice is **stratified 10-fold CV**.
- Leave-One-Out Cross-Validation (LOOCV)
  - Special case when # of folds = # of training examples (i.e., $\boldsymbol{k = N}$).

Hyper-parameter search



Grid search



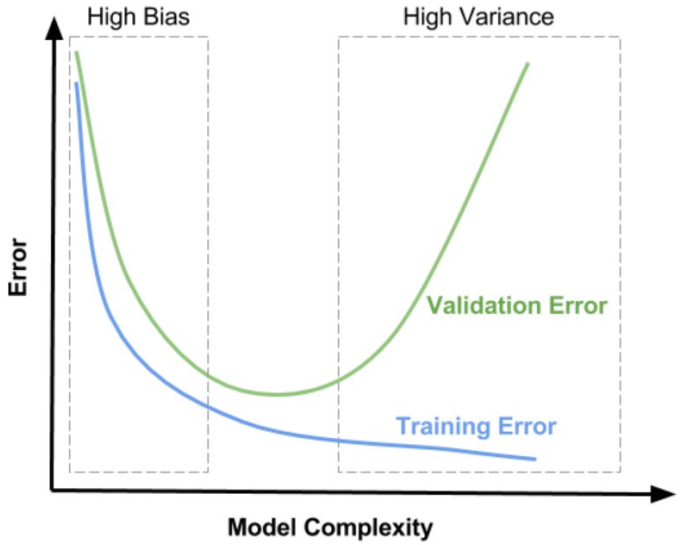
Random search

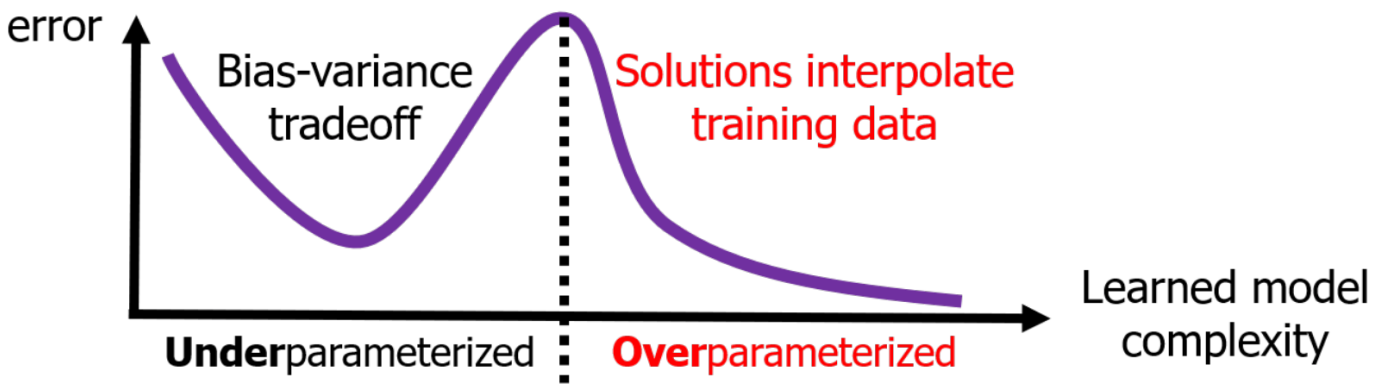
- Rule of thumb:
  - For algorithms with **a few** hyper-parameters, use grid search.
  - For algorithms with **many** hyper-parameters, and of different **importance**, use random search.

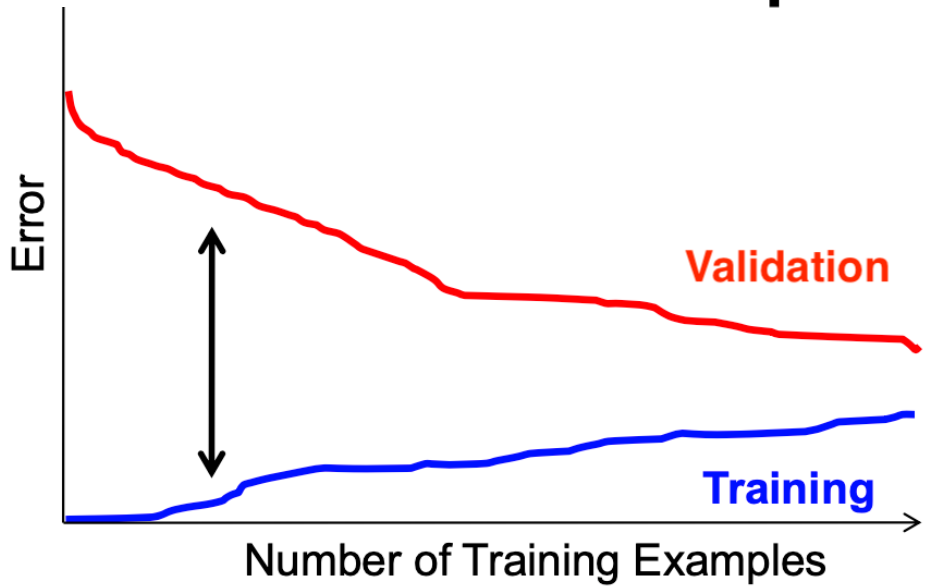
Bias and Variance

- **Main reasons for underfitting (high bias)**
  - Model is too simple for the data
  - Features not sufficiently suitable to describe the underlying correlations
- **Main reasons for overfitting (high variance)**
  - Model is too complex for the data
  - Too many features but small number of training examples
- **How to address the overfitting problem**
  - Try a simpler model
  - Reduce the dimensionality (dimensionality reduction)
  - Reduce the number of features (feature selection)
  - Add more training data
  - Regularize the learning model

Learning curves







Skewed data Classification metrics

- Accuracy

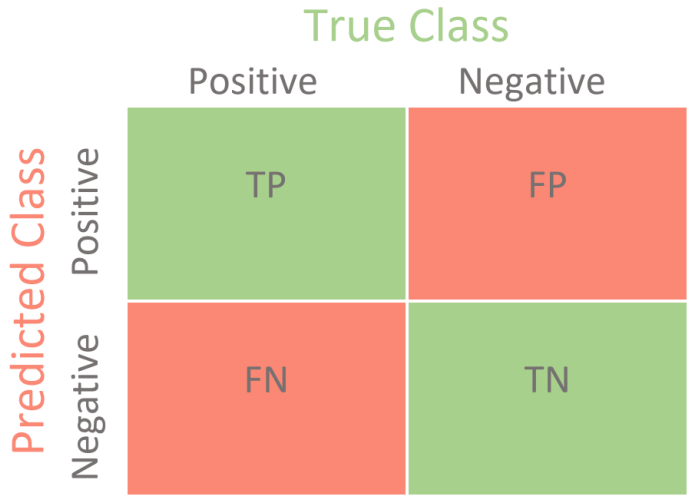
$$A = \frac{TP + TN}{TP + FP + TN + FN}$$

- Precision

$$P = \frac{TP}{TP + FP}$$

- Recall (or Sensitivity)

$$R = \frac{TP}{TP + FN}$$



- F-score
  - $\beta = 1 \rightarrow$ for F_1-score (harmonic mean)

$$F_\beta = \frac{(\beta^2 + 1)PR}{\beta^2 P + R}$$