Research and Innovation Center of Excellence

MSc on Intelligent Critical Infrastructure Systems

Machine Learning

Lecture 7: Feature engineering, Evaluation

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

FUNDED BY:

University of Cyprus

Imperial College London

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**
  - Monitoring and Control

Recap

- **Linear / Logistic Regression**
  - Fast, scalable, easy to understand and implement.
  - It often achieves a descent performance.

- **Support Vector Machines (SVMs)**
  - **Idea**: Transform the original space to a higher dimensional so that data become linearly separable.
  - Superior performance (structured data, e.g., tabular).
  - It doesn’t scale well with big data.

- **Neural Networks**
  - **Idea**: Representation learning
  - Superior performance (unstructured data, e.g., images).
  - It scales well with big data.

Steps for Machine Learning

Data gathering → Data cleaning → Feature engineering → Model selection → Model training and testing

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

Categorical features

- Some algorithms are heavily affected by categorical data.
- **Integer** vs. **one-hot** encoding
- If the order of the feature's values is not important then using integer encoding may affect the learning algorithm, e.g., countries are not sequential!
- **Drawbacks** of one-hot encoding
  - 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

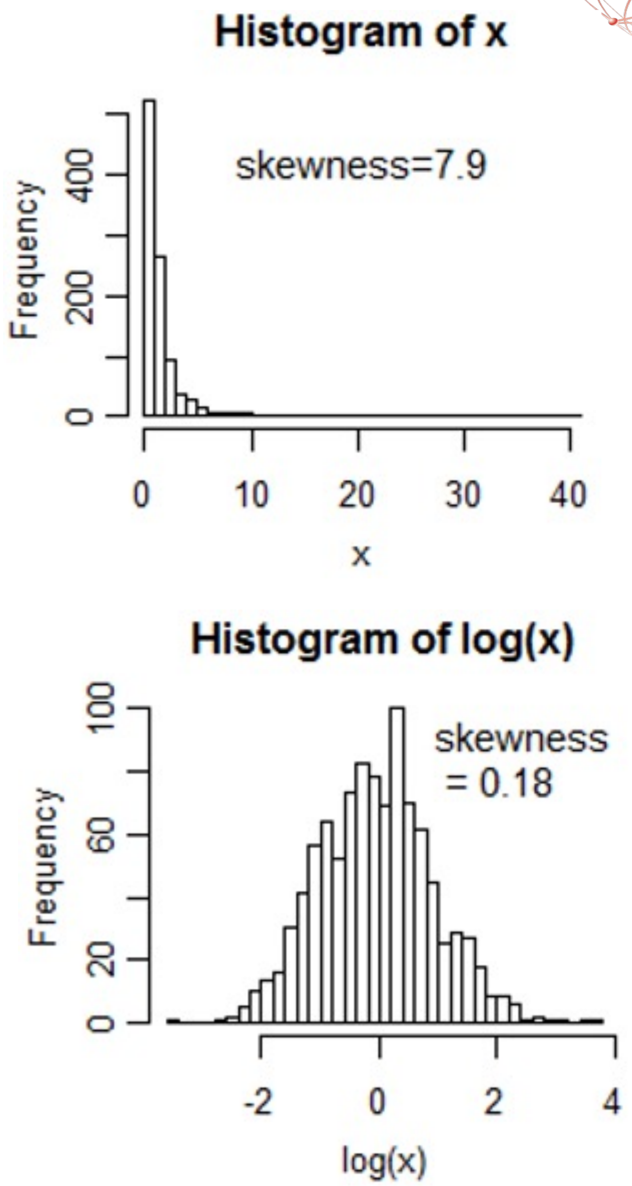
Skewed (high-cardinality) categorical features

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

- **Knowledge-based**
  - E.g., group “post codes” in to “areas”
- **Prediction power-based**
  - Group as “Other” those which have less predictive power

Skewed numerical features

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



- **Log** transformations $x \leftarrow \log(x + c)$

- **Power** transformations $x \leftarrow (x)^p$

Skewed class

- Class imbalance occurs when at least one class is under-represented ➔ minority class
  - $p(y = y_0) \gg p(y = y_1)$

- **Algorithm-level** approach
  - **Cost-sensitive learning**
  - Anomaly detection
  - One-class classification
  - …

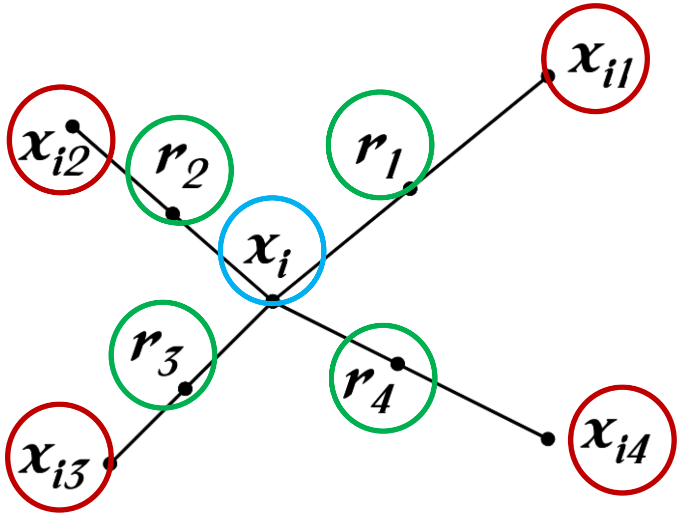
$$J(\theta) = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{w}_{\boldsymbol{y}^i} L(\hat{f}(x^i), y^i)$$

- **Data-level** approach
  - Undersampling the majority class (e.g., random)
  - **Oversampling** the minority class (e.g., SMOTE, data augmentation)

Skewed class: SMOTE

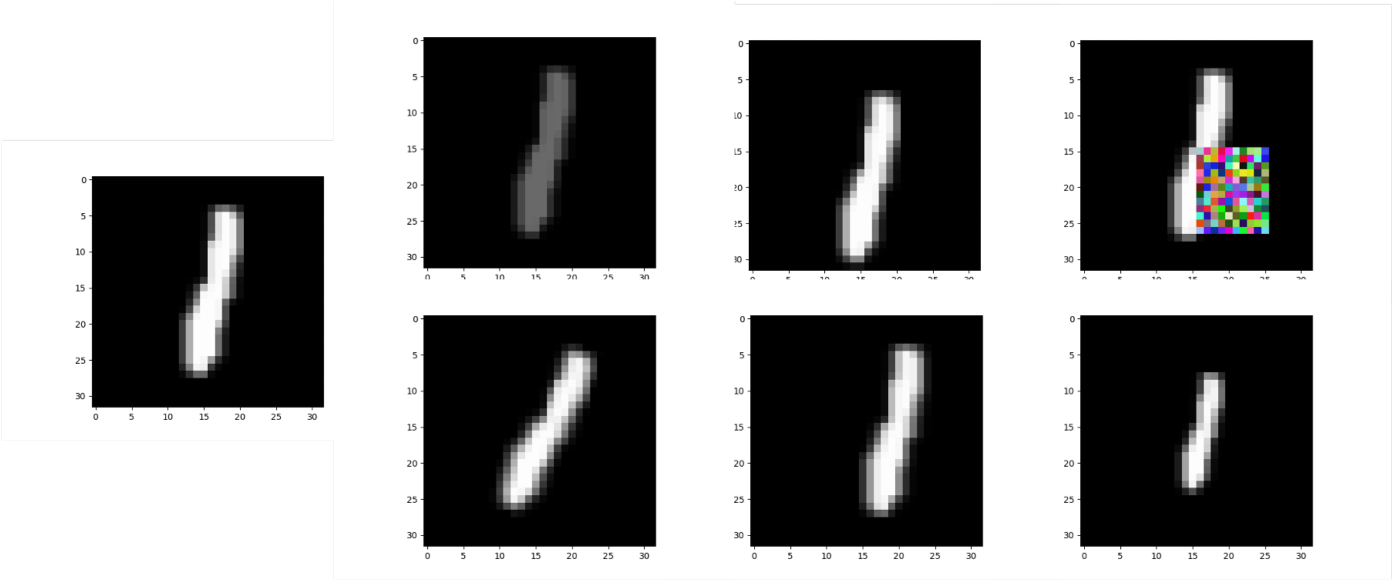
Repeat:

- A minority class instance is selected at random
- Select K nearest neighbours
- Select N of K to create synthetic points using interpolation
  - $r_i = x_i + rand(0,1) \times (x_i - n_i)$



Skewed class: Data augmentation

- Data augmentation is applied to a dataset to expand its size by artificially creating variations of the data.
- It enhances the diversity of the dataset which could improve the learning performance, and improve generalisation.



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.

Missing values

- Remove the examples with missing features.
- Use a **data imputation** method.
  - Numerical features
    - Replace the missing value of a feature by its average value, or the middle of the range values, or use interpolation (for time-series data).
    - Use regression to estimate the missing feature value.
  - Categorical data
    - 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.

Outliers

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

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

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

- 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.

Feature scaling

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

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

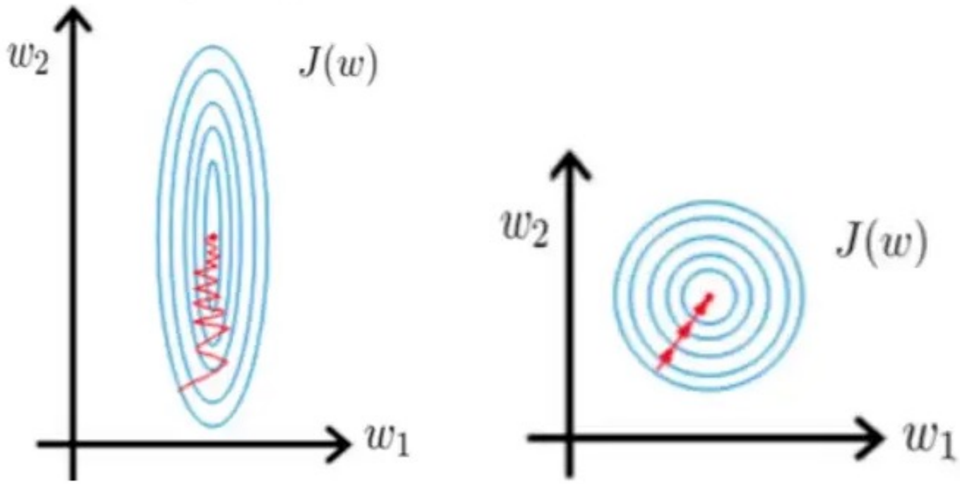
- **Standardization** rescales the numerical values of a feature so that it has a standard normal distribution (mean = 0; standard deviation = 1).

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

- Both normalization and standardization may improve the learning speed.

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)

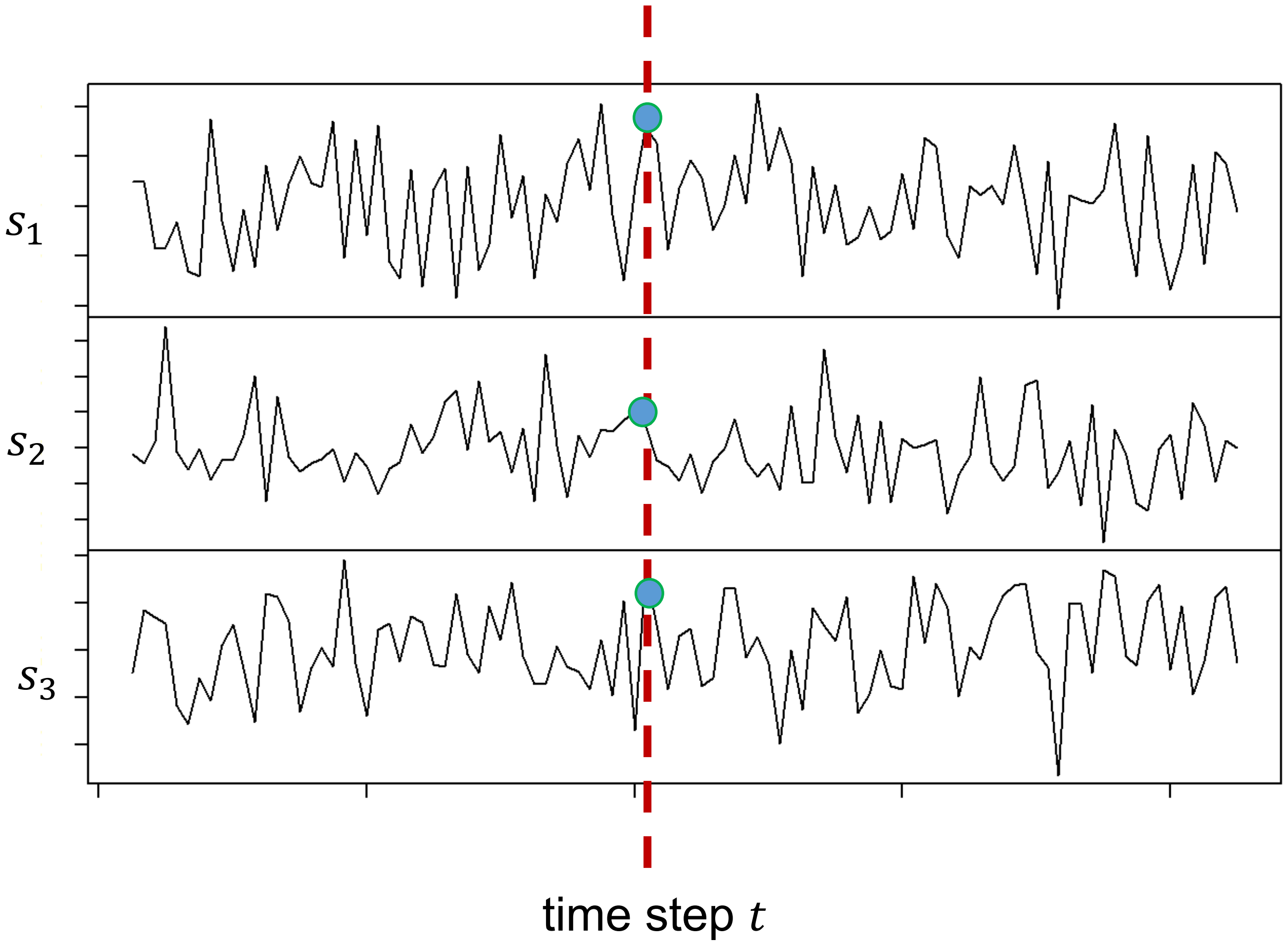


Non-iid data

- In statistical learning, it is assumed that the training data (X, y) consists of examples that are **independently** drawn from the **same** joint distribution $p_{X,y}$.

- **Temporal correlations** (e.g., time-series data) affect learning algorithms
  - Time-series forecasting
  - Time-series classification

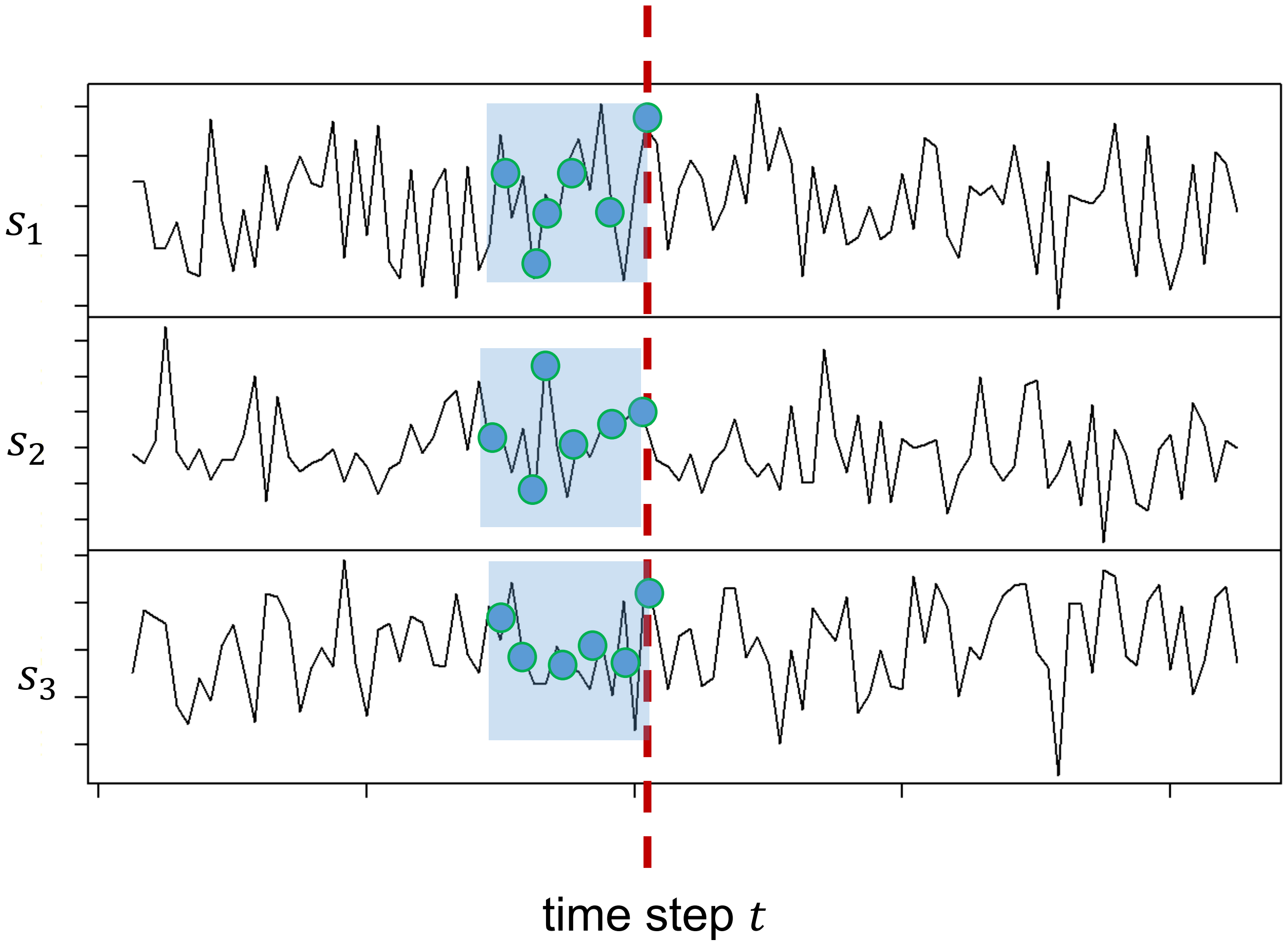
Non-iid data: Traditional approach



Traditional approach

- Each example is of the form:
  $$\boldsymbol{s}^{\mathrm{t}} =< \boldsymbol{s}_1^t, \boldsymbol{s}_2^{\mathrm{t}}, \boldsymbol{s}_3^{\mathrm{t}} >$$
- Typically, it yields a poor performance.

Non-iid data: Sliding window approach



Sliding window approach

- Consider a window of size W:
  - $\mathbf{s}_1^{t} =< s_1^t, s_1^{t-1}, \ldots, s_1^{t-W} >$
  - $\mathbf{s}_2^{t} =< s_2^t, s_2^{t-1}, \ldots, s_2^{t-W} >$
  - $\mathbf{s}_3^{t} =< s_3^t, s_3^{t-1}, \ldots, s_3^{t-W} >$
- Each example is now of the form:

  $\mathbf{x}^{t} =< \mathbf{s}_1^t, \mathbf{s}_2^t, \mathbf{s}_3^t >$
- Use $\mathbf{x}^{t}$ (instead of s^t) as input to the learning algorithm.
- Additionally:
  - LSTM
  - Further feature extraction

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.

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)

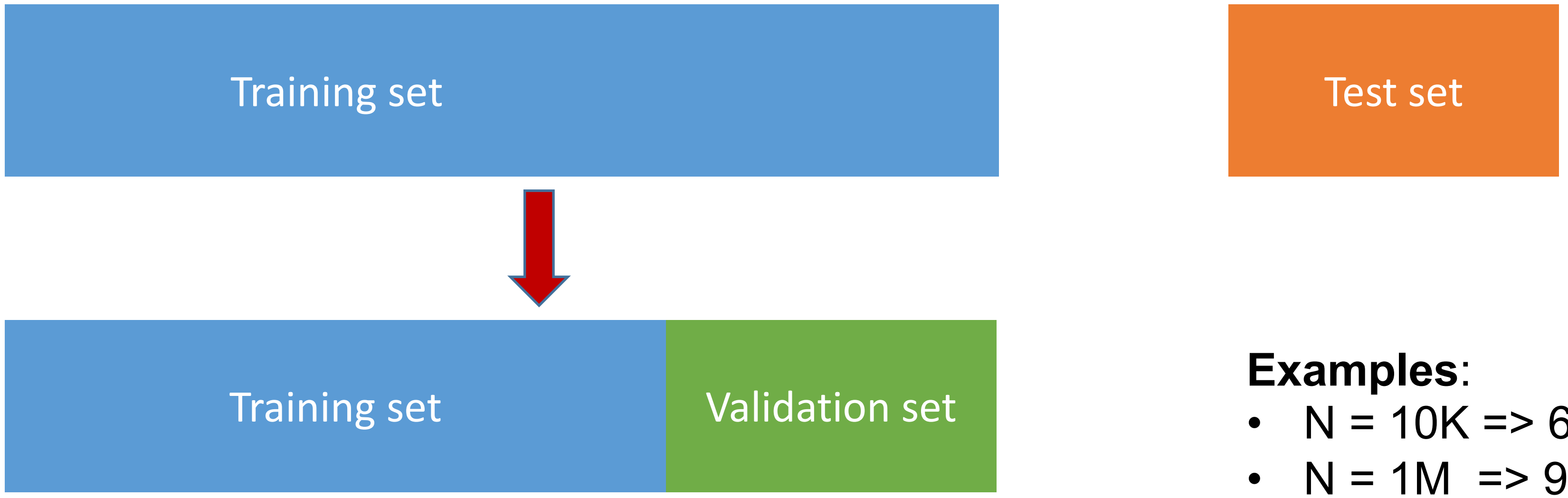
→ Possible to try various learning algorithms and select one by testing on validation test.

- **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.

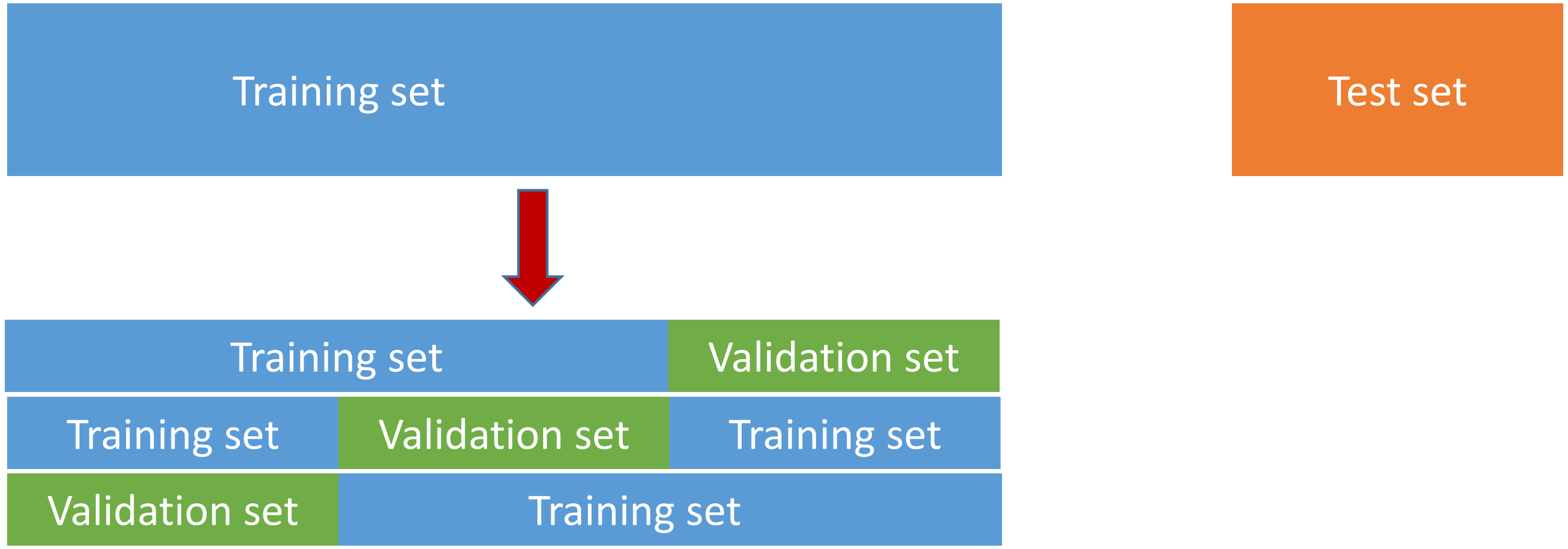
Holdout cross-validation



Examples:
- N = 10K => 60-20-20%
- N = 1M => 95-2.5-2.5%

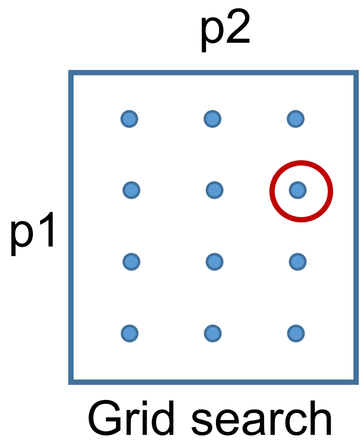
- **Training set**: is used to fit the models.
- **Validation set**: is used to estimate prediction error for model selection.
- **Test set**: is used for assessment of the generalization error of the final chosen model.
  - “Ideally, the test set should be kept in a “vault,” and be brought out only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

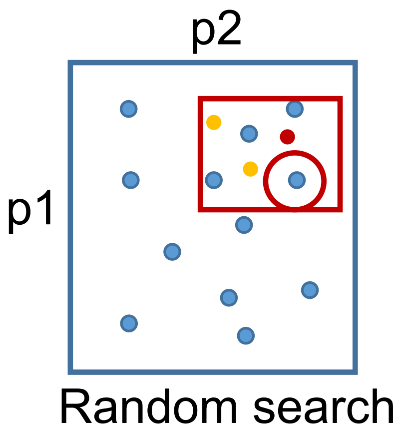
k-Fold cross-validation



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

Hyper-parameter search





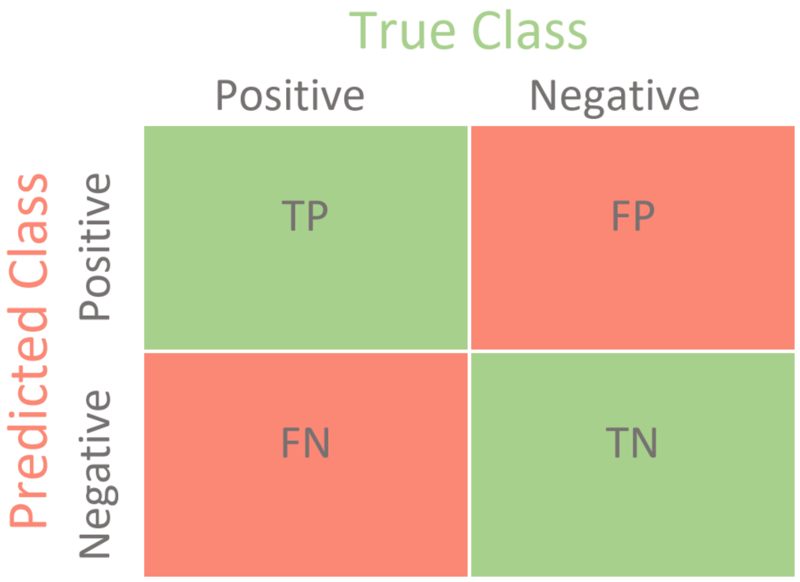
- Rule of thumb:
  - For algorithms with **a few** hyper-parameters, use grid search.
  - For algorithms with **many** hyper-parameters, and of different **importance**, use random search.

Classification metrics

- Accuracy

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

- Example:
  - 10 Positive class, 990 Negative class
  - Prediction: 1000 Negative class
  - **Accuracy 99%**



Classification metrics (cont'd)

- Precision $$P = \frac{TP}{TP + FP}$$ $$\frac{true\ positive}{predicted\ positive}$$
- Recall (Sensitivity) $$R = \frac{TP}{TP + FN}$$ $$\frac{true\ positive}{actual\ positive}$$
- F-score $$F_\beta = \frac{(\beta^2 + 1)PR}{\beta^2 P + R}$$
- F1-score $$F_1 = \frac{2PR}{P + R}$$
  - $\beta = 1 \rightarrow$ harmonic mean

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