Introduction to Machine Learning. Statistical foundations

SUMMARY

Artificial Intelligence (AI)

- Symbolic AI
- Computational Intelligence (CI)

Data mining (DM)

Machine Learning (ML)

- Deep Learning (DL)

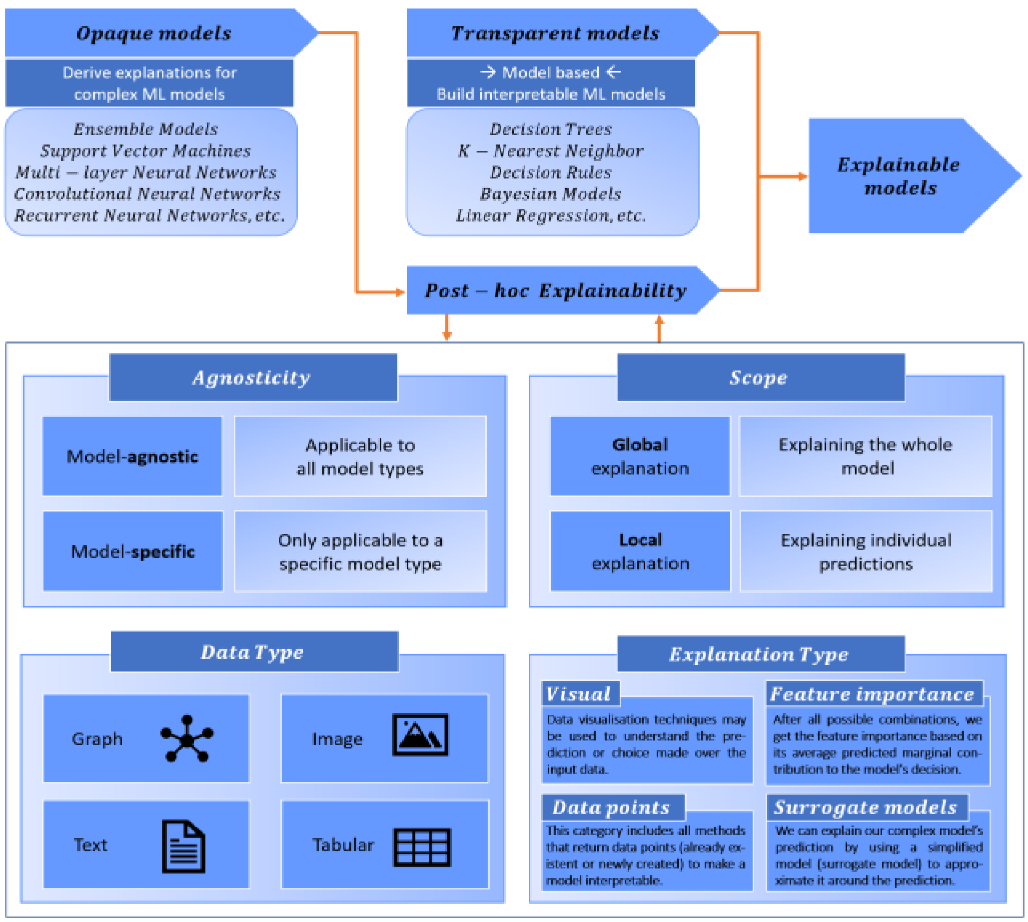
ML/DL

- rapid and significant advancements
  - natural language processing, computer vision, etc
- "hot" topics
  - **Large Language Models** (LLMs)
    - neural network-based models
      - inspired by the human brain
    - trained DL model that understands and generates text in a human-like manner
    - GPT-like models
      - **Generative Pre-trained Transformer**
      - Natural language processing (NLP), genomics, etc
      - Trustworthiness?
  - **Generative AI/ML**
  - Machine Unlearning
    - Inspiration – **human brain**
      - neuroscience-based adaptive unlearning process
    - sensitive data in training
      - the process of selectively removing specific training data points and their influence on an already trained model, making the updated model behave the same as a model that was never trained on that data
  - Representation/feature learning
    - Feature fusion
      - Computer vision tasks, bioinformatics, etc
  - Contrastive learning

            - training a model to differentiate between similar and dissimilar pairs of data instances
                - maximizing the similarity within the same class and minimizing the similarity between different classes.

AI

- interpretability of ML/DL models
    - black-box vs white-box models
- Explainable AI (XAI)
    - understand and interpret the predictions of a ML model
- interpretability methods and evaluation of ML interpretability
    - Gradients, DeepLIFT, Class Activation Maps (CAMs), etc
- methods for ML interpretability can be classified according to various criteria



General ontology of XAI (link)

- **LIME** (**L**ocal **I**nterpretable **M**odel-agnostic **E**xplanations)
    - algorithm that can explain individual predictions of any classifier or regressor
- **SHAP** **(SH**apley **A**dditive ex**P**lanations**)**
    - a game theoretic approach to explain the output of any machine learning model

Trustworthy AI (TAI)

- not only **metrics**
- principle, tools

- ethics guidelines for TAI
- **other aspects**
    - robustness, algorithmic fairness, explainability, and transparency
    - robustness
        - requirement of TAI
        - refers to the degree that a model's performance changes when using new data versus training data
            - ideally, performance should not deviate significantly
        - the ability of a model to maintain its performance when faced with uncertainties or adversarial conditions
            - noisy data, distribution shifts, and adversarial attacks
        - how to test robustness?
            - cross-validation
            - adding noisy data
            - …

1. Machine Learning

Learning problems represent an important research direction in AI, *machine learning*.

Machine Learning (ML)= the study of system models, that based on a set of data (training data) **improve** their **performance** (on a specific **task**) by **experiences** and by learning some specific domain knowledge.

Three main directions for ML

- **Data mining** – extract knowledge from data (use historical data to improve decisions, predictions, etc).
- **Software applications** we cannot program by hand
    - autonomous driving
    - speech recognition
    - handwriting recognition
    - game playing
    - etc.
- **Self customizing programs** – programs that adapt to changing conditions
    - Learn the users' interests

- The attempt of modeling the human reasoning leads to the notion of *intelligent reasoning*.
    - Most of the AI systems are *deductive* ones, capable to draw conclusions (make inferences) based on their initial (or supplied) knowledge, without having the capability to generate new knowledge
    - In situations in which a system has incomplete information (knowledge) about its environment, **LEARNING** is the only way the system could get the needed knowledge.
    - The **learning** assures the **autonomy** of a system (the ability to decide which action to perform without external intervention).

- **Learning to represent a function (target function)**

- ML algorithms are also called *function approximators*
- $f : Inputs \rightarrow Outputs$ – *target* function
- An input x $\in$ *Inputs* is characterized by a set of **features** (relevant characteristics of the input)
- Learning goal: to find $h \approx f$, *h* is called **hypothesis**
- Learning = searching for the *hypothesis* that best fits the data

Feature engineering = create features for ML

- **Manual** feature engineering
  - manually create/select relevant features characterizing the data
  - traditional approach – ***classical*** (*traditional*, *conventional*) **ML models**
- **Automated** feature engineering
  - automatically extracting useful and meaningful features from data
    - using unsupervised learning models (e.g. Principal Component Analysis – PCA, autoencoders, aso)
    - using ***deep*** **learning** (DL) models
- **Research** topics
  - unsupervised feature/representation learning
  - feature fusion
  - a.s.o

Deep learning (DL)

- is a subfield of Machine Learning
  - mainly based on ANNs
- hierarchical feature learning
- used for learning data representations
  - based on representation learning
    - automatically extract features from raw data
  - opposed to task specific algorithms
    - learn from representative examples

ML applications:

- Medicine, bioinformatics, psychology.
- Music composition, archaeology.
- Software engineering (*Search-based software engineering*)
- Computational photography.
- Computer Vision.
- Natural language processing.
- Meteorology.
- Educational data mining.
- aso

Relevant disciplines

- Artificial intelligence
  - Computational intelligence
- Computational complexity theory
- Information theory

- Philosophy
- Psychology and neurobiology
- Bayesian methods
- Mathematics
  - Probability
  - Statistics
  - Linear algebra
  - Functional analysis
  - Numerical analysis
- …

2. Types of learning

1. **Supervised learning (SL)**
   - predictive models
   - **applications**
     - intrusion detection, data rectification for process control, image (pattern) recognition, predictions
   - completely labeled training data
     - A trainer submits the input/output exemplary patterns and the learner has to adjust the parameters of the system autonomously, so that it can yield the correct output patterns when faced with a new input pattern.
     - A set of training examples (x_i, $f(x_i)$), where f is the target function to be learned, is provided to the learner and the aim is to determine (an approximation of) f by some adaptive algorithm.
   - issues in supervised learning
     - **overfitting** (learn by heart)
       - The model learns very well the training data (it has high performance on the training data set), but it does not generalize well on unseen data (low performance on a testing data set)
     - **underfitting**
       - the model is too simple, it cannot capture the structure of the data
     - Noisy training data (errors in data, outliers, irrelevant data)
   - 2 important types of supervised learning
     - **Inductive learning**
       - determine a hypothesis h such that $h(x_i) \approx f(x_i)$
       - how to compare two hypotheses approximately close to f?
         - *inductive bias*
     - **Analogical learning**
       - Identifying analogies between an experienced problem instance and a new problem
       - E.g. case-based reasoning (CBR)

2. **Unsupervised learning (UL)**
   - descriptive models
   - completely unlabeled training data

    - in absence of trainers, the desired output for a given input instance is not known, and consequently, the learner has to adapt its parameters autonomously.

3. **Reinforcement learning (RL)**
    - learning by interaction with the environment
    - an autonomous agent learns to perform an optimal sequence of actions to reach a goal
    - **applications**
        - game playing, robotics and control
4. **Semi-supervised learning**
    - falls between SL and UL
    - the learner is provided with a small amount of labeled data and a large amount of unlabeled data
    - graph-based, kernel-based, generative, pseudo-labeling methods, aso.

Inductive logic programming (ILP)

- subfield of ML which uses first-order logic to represent hypotheses and data
- specifically targets problems involving structured data and background knowledge
- tackles a wide variety of problems in machine learning, including classification, regression, clustering, and reinforcement learning.
- **applications**
    - in bio- and chemo-informatics, natural language processing, and web mining.

3. The learning problem

- **specification** of the learning task
    - *T*, *P*, *E*
        - improve over task *T*
        - with respect to performance measure *P*
        - based on experience *E*
    - e.g, learning to play a board game [1]
        - T: Play checkers
        - P: % of games won in world tournament
        - E: opportunity to play against self
    - what experience?
        - direct or indirect?
        - teacher or not?
        - is training experience representative to the performance goal?

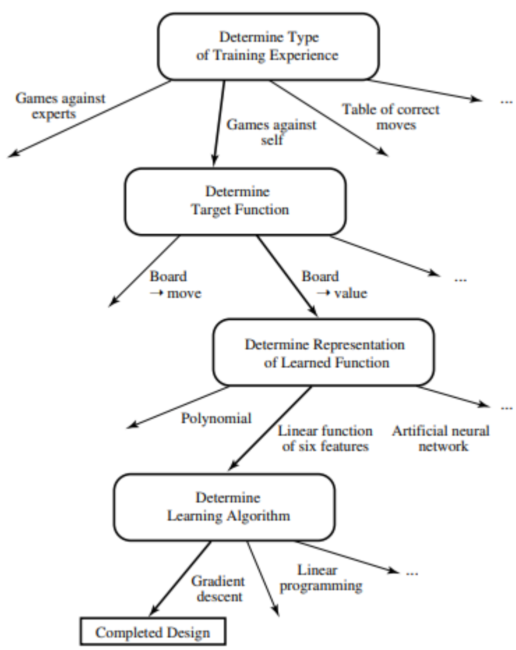
- **design** choices
    - what exactly should be learned?
        - choose the **target function**
    - how shall the learned function be represented?
    - what specific algorithm to learn the target function?

Choose the Target Function

- $ChooseMove : Board \rightarrow Move$??
- $V : Board \rightarrow \Re$??
- ...

- components of a learning system
  - the **performance system**
    - responsible with providing the output, using the learned target function
  - the **critic**
    - responsible with providing feedback to the learner (e.g., training examples in case of SL)
  - the **generalizer**
    - responsible with producing an output hypothesis that is the estimate of the target function
  - the **experiment generator**
    - mainly in RL scenarios
    - takes as input the current hypothesis (currently learned function) and outputs a new problem for the performance system to explore

Design Choices



[1]

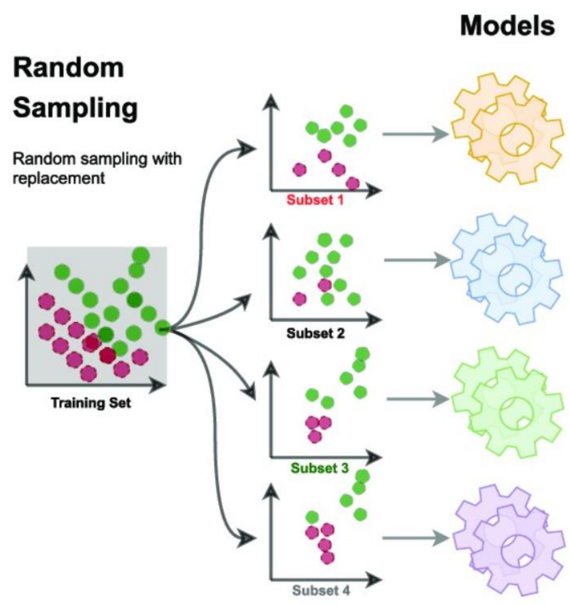
- **Learning is most reliable when the training examples follow a distribution similar to that of future test examples**

- e.g., in game playing, the learner might never encounter certain crucial board states that are very likely to be played by the human champion.
- representative training data

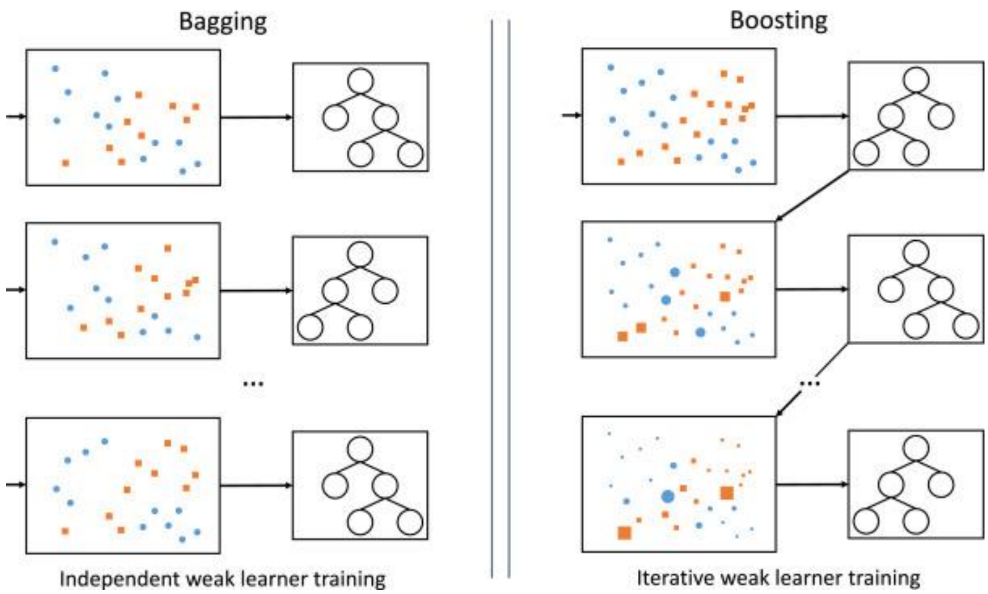
4. Ensemble learning

- meta-learning algorithms
- to combine multiple ML algorithms to obtain better predictive performance
- multiple models (often called "weak learners") are trained to solve the same problem and combined to get better results.
  - the main assumption is that we can obtain more accurate and/or robust models when weak models are correctly combined.
- An ensemble learner outputs a single hypothesis which is not necessarily contained in the hypotheses space of the small models.
- types of ensemble learning

  - **Bagging**
    - ensemble meta-learning algorithm
    - improve stability and accuracy of ML models
      - reduces overfitting
    - **B**ootstrapping **AGG**regat**ING**
      - *bootstrapping* is random sampling **with replacement** (*an observation may be selected more than once*) from the available training data.
      - **Bagging** is performing bootstrapping many times and training an estimator for each bootstrapped dataset.
      - theoretical foundation
        - sampling with replacement and then building an ensemble reduces the variance of the ensemble learner without increasing the bias.
    - homogeneous weak learners
    - E.g. Random Forests (weak learner = decision tree)

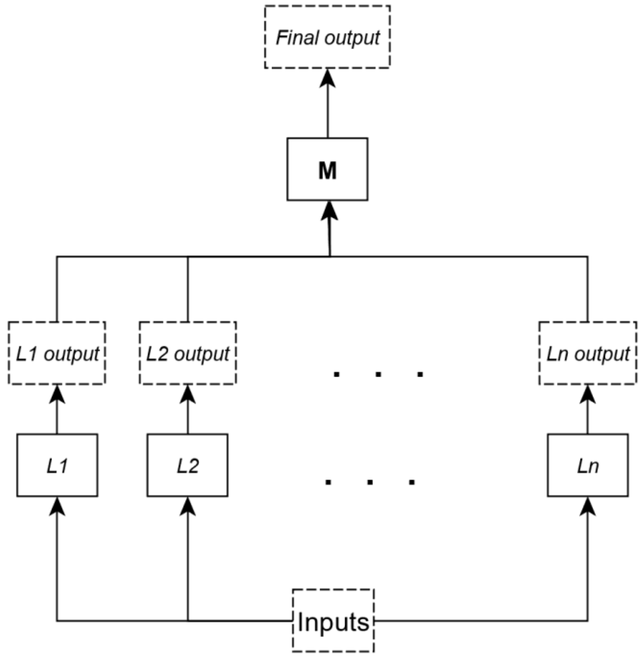


- **Boosting**
  - ensemble meta-learning algorithm
  - use multiple homogeneous weak learners to obtain a single strong learner
    - a good weak learner is one that is highly biased (*unstable*): e.g., decision tree
  - used to **reduce errors** in predictive data analysis
  - can a set of weak learners slightly correlated with the true classification to create a single strong learner?
  - seems to be better than bagging, but has the tendency to overfit
  - **algorithms**: Adaboost, EpsilonBoost, Gradient boosting (XGBoost)



Parallel **Sequential**

- **Stacking**
  - ensemble meta-learning algorithm
  - heterogeneous weak learners
  - combine the weak learners by training a metamodel



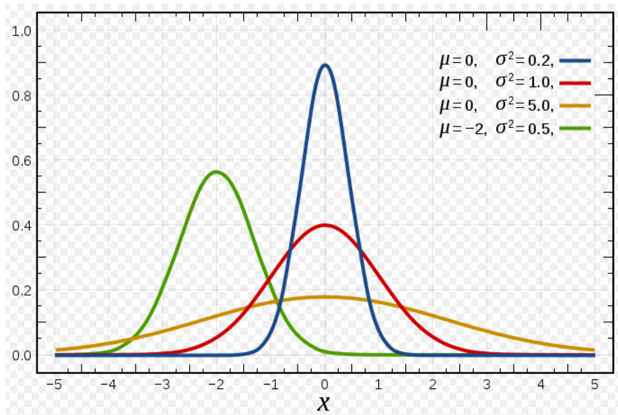
- ensemble of DL models

5. Probability, statistics and information theory

- **Probability & statistics**
    - modelling processes with uncertainty
        - **Statistics – inductive**
            - we **observe** something that has happened
              and
            - we try to figure out what **underlying process** would explain those observations
        - **Probability – deductive**
            - we consider some **underlying process** which has some randomness/uncertainty modelled by random variables
              and
            - we try to figure out **what** happens
- random variables
    - a variable whose possible values are outcomes of a random phenomenon
        - discrete (e.g. throwing a die)
        - continuous (e.g. the age of death)
    - ML
        - features
        - outcome of the learning process
- **probability** distributions [2]
    - describes probabilities of values a random variable can take
        - *discrete* (e.g. throwing a die)
        - *continuous* (e.g. the age of death)
            - continuous values (real numbers)

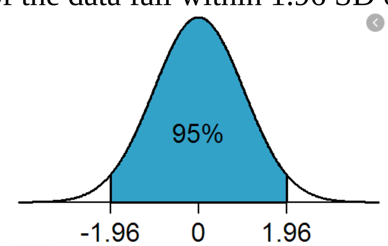
- **normal (Gaussian)**

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$



Source: https://commons.wikimedia.org/wiki/File:Normal_Distribution_PDF.svg

  - 68.26% of the data fall within 1 SD of the mean
  - 95.44% of the data fall within 2 SDs of the mean
    - 95% of the data fall within 1.96 SD of the mean



  - 99.7% of the data fall within 3 SD of the mean
    - this rule enables to
      - check for outliers
      - determine the normality of any distribution
- In **Machine Learning**, data satisfying **Normal Distribution** is beneficial for model building.
  - models like LDA, Gaussian NBC, Logistic regression, Linear Regression, etc, use the assumption of normally distributed data.
  - sigmoid functions work most naturally with normally distributed data
  - natural phenomena (e.g., financial data, forecasting data) follow a *log-normal distribution*
    - continuous probability distribution of a random variable whose *logarithm* is *normally distributed*.
  - many processes follow **normality**
    - many measurement errors in an experiment
    - the position of a particle that experiences diffusion
    - …
  - convert the data into a normal distribution - transformation techniques (statistical software packages)
    - e.g., **quantile transforms**

- will map a variable's probability distribution to another probability distribution
- **QuantileTransformer** class from *scikit-learn* Python library

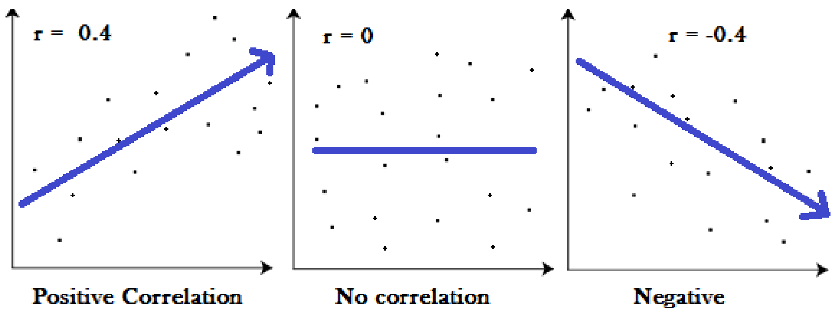
- **Normality** assumption for the ML models
  - it is not mandatory that data should always follow normality
  - ML models may work very well in the case of non-normally distributed data
    - **Decision trees**, **XGBoost**, don't assume any normality and work on raw data.
    - **Linear regression** is statistically effective if only the model errors are Gaussian, not exactly the entire dataset.
- **Statistical tests** for checking data normality
  - **Shapiro-Wilk** test, **Kolmogorov-Smirnov** (K-S) test, ..
  - statistical software

!!! Explore the data and check for the underlying distributions for each variable/feature before going to fit the ML model.

- **moments**
  - mean, variance, standard deviation
    - **mean** – expected value
      - μ, E(X)
    - **variance** – expected squared distance from mean
      - σ^2=E((X-E(X))^2
    - **standard deviation**
      - σ
      - measures spread of distribution
        - how far a set of numbers is spread out
      - low σ - close to the mean
  - **Central limit theorem**
    - the sum of a large number of random samples is normally distributed
      - the sum of N random variables with mean m and standard deviation s can be approximated by a normal distribution with
        - mean $\mu = N \cdot m$
        - standard deviation $\sigma = s \cdot \sqrt{N}$
    - if one takes sufficiently large samples from a population, the samples' means will be *normally distributed*, even if the population isn't normally distributed
    - application: **confidence intervals**
- comparing the performance of ML classifiers/regressors
    - **statistical significance tests**
      - T-test, Wilcoxon test, etc
- **correlation** – statistical relationship between two random variables
    - intuition
      - two random variables X and Y are correlated, when:
        - when X is high, Y is likely to be high
        - when X is low, Y is likely to be low
    - **correlation coefficients** – measure the degree to which the variables are monotonically related

        - **Pearson**
            - is used for measuring the degree of linear relationship between two features

        - **Spearman**
            - describes the strength and direction of the monotonic relationship between two variables (even if their relationship is not linear)
        - correlation coefficient formulas are used to find how strong a relationship is between data
        - return a value between -1 and 1, where:
            - 1 indicates a strong positive relationship.
            - -1 indicates a strong negative relationship.
            - 0 indicates no relationship at all.



Source: https://www.statisticshowto.com/probability-and-statistics/correlation-coefficient-formula/

    - in supervised learning
        - features **well correlated** with the target output
        - **independent** features

- statistics [2]: statistical independence, confidence intervals
    - two random variables are *independent* if they convey no information about each other

$$P(A|B) = P(A)$$

    - Pearson's χ^2 (Chi-squared) test for independence
    - **independence** $\Rightarrow$ uncorrelation, but uncorrelation does not imply independence
    - choosing relevant and independent features is key to ML
    - **confidence interval** (CI)
        - a statistical measure of the reliability/consistency of an estimate
        - is an interval estimate of a population parameter (e.g. the mean)
            - an interval that frequently includes the parameter
        - 95% CI of the mean μ of a population
            - $[\mu - \alpha, \mu + \alpha]$ or $\mu \pm \alpha$
            - α - *confidence value*
                - $\alpha = 1.96 * \sigma / \sqrt{N}$
                - N – sample size
                - σ - population standard deviation
        - comparing the performance of classifiers

- conditional probability (Bayes theorem)

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

    - **Bayesian learning**
- **entropy**
    - a measure from information theory
    - measures the impurity of a set

[SLIDES]

- Introduction to ML [1]

[READING]

- Introduction to ML (T. Mitchell) [1]
- Introduction to ML and DL (Zhang et al.) [2]
- ML preliminaries (N. Nillson) [4]
- ML Basics (Goodfellow et al.) [3]
- Probability and information theory (Goodfellow et al.) [3]
- Evaluating hypotheses (T. Mitchell) [1]

Bibliography

[1] Mitchell, T., *Machine Learning*, McGraw Hill, 1997 (available at www.cs.ubbcluj.ro/~gabis/ml/ml-books)

[2] Aston Zhang, Zachary C. Lipton, Mu Li, and Alexander J. Smola, *Dive into Deep Learning*, 2020 (http://d2l.ai/)

[3] Ian Goodfellow, Yoshua Bengio, Aaron Courville, *Deep Learning*, MIT Press, 2016 (online edition at http://www.deeplearningbook.org/)

[4] Nillson, N., *Introduction to Machine Learning*, Stanford University, 1996 (available at www.cs.ubbcluj.ro/~gabis/ml/ml-books)