| | |
|---|---|
| **Business Analytics** | Business analytics makes extensive use of statistical analysis, including explanatory and predictive modeling, and fact-based management to drive decision making. Analytics may be used as input for human decisions or may drive fully automated decisions.<br>• *Descriptive analytics* (What has occurred?)<br>• *Predictive analytics* (What is likely to occur?)<br>• *Prescriptive analytics* |
| **From data to information** | 1. *Data consolidation*<br>2. *Selection and Preprocessing*<br>3. *Predictive analytics*<br>4. *Interpretation and evaluation* |
| **Numeric Prediction** | Given a collection of data with known numeric outputs, create a function that outputs a predicted value from a new set of outputs |
| **Classification** | From data with known labels, create a classifier that determines which label to apply to a new observation |
| **Clustering** | Identify "natural" groupings of data (no predefined groups) |
| **Association Rule Analysis** | Identify relationship s in data from co-occurring terms or items |
| **Machine Learning** | • *Supervised learning*: attempt the discovery of the relationships between input attributes and a target attribute. (→ form a description that can be used to predict unseen examples)<br>  ○ Training: estimate the prediction function f of the training data set<br>  ○ Testing: apply f t a never seen test example<br>• *Unsupervised learning*: there is no supervisor and only input data is available. (→ find regularities, irregularities, relationships, similarities and associations in the input) |
| **Model** | Representation of a system that allows for investigation of the properties of the system and, in some cases, prediction of future outcomes.<br>• *Decision Trees*<br>• *Neural Networks* |
| **Causal Inference** | Causal inference is the process of deriving cause-and-effect conclusions by reasoning from knowledge and factual evidence.<br>Randomized control trials are the gold standard for causal inference in the social sciences, but often only observational data is available. |
| **Econometrics** | Data analysis focusing on causal relationships in economics (Correlation ≠ Causation) |
| **Descriptive statistics** | Descriptive statistics can be used to summarize the data, either numerically or graphically, to describe the sample (e.g. mean, standard deviation,…) |
| **Inferential statistics** | Inferential statistics is used to model patterns in the data, accounting for randomness and drawing inferences about the larger population (estimation, hypothesis testing, forecasting, correlation, regression) |
| **Random Variables** | X is a random variable if it represents a random draw from some population and is associated with a probability distribution<br>• *Discrete*: random variable can take only selected values<br>• *Continuous*: random variable can take any value in the real interval<br>**Standard Normal**: any random variable can be "standardized" by subtracting the mean and dividing by the standard deviation.<br>**Expected Value**: the expected value of a probability weighted average of X is the mean or expected value of the distribution of X. |
| **Random Sample** | A random sample is a set of independent, identically distributed random variables (every combination of n sample points has an equal chance of being selected) |
| **Statistical Estimation** | Every member of the population has the same chance of being selected in the sample.<br>An estimator is a statistic that is used to estimate an unknown population parameter.<br>• *Point estimate*: sample mean, sample proportion<br>• *Interval estimate*: confidence interval for mean, confidence interval for proportion<br>Point estimate is always within the interval estimate |
| **Standard Deviation** | The standard deviation of the sample means is equal to the standard deviation of the population divided by the square root of the sample size |
| **Central Limit theorem** | The central limit theorem states that the standardized average of any population of random variables is asymptotically ~N(0,1) |
| **Confidence Interval (CI)** | Provide us with a range of values that we believe, with a given level of confidence, contains a population parameter CI for the population means<br>• Larger sample = smaller interval<br>• Lower confidence interval = smaller interval |

| | |
|---|---|
| | • More variation = larger interval |
| **Statistical test** | 1. *Formulate hypothesis*<br>2. *Collect data to test hypothesis* (systematic error can be controlled by statistical significance or by confidence interval)<br>3. *Accept or reject hypothesis* |
| **Hypothesis testing** | 1. *State null and alternative hypothesis* ($H_0$ usually a statement of no effect)<br>2. *Choose α level* (related to confidence interval) - probability of falsely rejecting the $H_0$<br>3. *Calculate test statistic and find p-value* (how far data are from null hypothesis)<br>4. *State conclusion*<br>a. $P \leq \alpha$, reject null hypothesis<br>b. $P > \alpha$, insufficient evidence to reject null hypothesis<br>**Type I error**: reject null hypothesis when the null hypothesis is actually true<br>**Type II error**: fail to reject the null hypothesis when the alternative hypothesis is true |
| **Student t-Distribution** | When the population is normally distributed, the statistic t is Student t distributed (bell-shaped and symmetric around zero) |
| **T-Test** | • *Single sample*: tests whether a sample mean is significantly different from a pre-existing value<br>• *Paired samples*: tests the relationship between 2 linked samples<br>• *Independent samples*: tests the relationship between 2 independent populations |
| **Selected Statistical Tests** | • *Parametric tests*<br>○ T-Tests: compares two sample means or tests a single sample mean<br>○ F-Test: compares the equivalence of variances of two samples<br>• *Non-parametric tests*<br>○ Wicoxon signed-rank test: independence of two means for 2 paired samples when normality is not assumed<br>○ Mann-Whitney-U test: used for 2 independent samples<br>○ Kruskal-Wallis-Test: equivalence of multiple means in case of several non-normally distributed samples<br>• *Tests of the Probability Distribution*<br>○ Kolmogorov-Smirnov and Chi-square test: used to determine whether two underlying probability distributions differ, or whether an underlying probability distribution differs from a hypothesized distribution |
| **Linear Regression** | Regressions identify relationships between dependent and independent variables. The linear regression is a statistical tool for numerical predictions<br>$Y = \beta_0 + \beta_1 X + \varepsilon$ (first order linear model)<br>$\beta_0$ and $\beta_1$ are unknown, therefore, are estimated from the data |
| **Ordinary Least Squares (OLS)** | Minimize the sum of squared residuals.<br>OLS requires choosing values of the estimated coefficients, such that Residual Sum of Squares (RSS) is as small as possible for the sample. |
| **Residual Sum of Squares (RSS)** | Sum of squared differences between the points and the regression line (how well line fits the data) |
| **Total Deviation** | The Total Sum of Squares (TSS) is the sum of the Explained Sum of Squares (ESS) and the RSS |
| **Coefficient of Determination** | $R^2$ measure the proportion of the variation in y that is explained by the variation in x ($R^2 = 1$ → perfect match between the line and data points, $R^2 = 0$ → no linear relationship between x and y) |
| **Multiple Linear Regression** | A p-variable regression model can be expressed as a series of equations. Beta coefficients are known as partial regression coefficients<br>$\hat{Y} = \hat{\beta}_0 + \hat{\beta}_1 X_1 + \hat{\beta}_2 X_2 = \mathbf{X}\hat{\beta}$<br>**Adjusted $R^2$**: It represents the proportion of variability of y explained by X. $R^2$ is adjusted so that models with a different number of variables can be compared<br>F-Test: significant F indicates a linear relationship between y and at least one of the Xs<br>T-Test: significant t indicates that the variable in question influences the response variable while controlling for other explanatory variables |
| **Model Specification** | In regression analysis the specification is the process of developing a regression model (selecting an appropriate functional form and choosing which variables to include). The model might include irrelevant variables or omit relevant variables.<br>Non-linear models are challenging, but some nonlinear regression problems can be linearized. |
| **Subset Selection** | Fit a parsimonious model that explains variation in Y with a small set of predictors<br>• *Best subset* (computationally expensive)<br>• *Backward elimination* (top-down approach)<br>• *Forward selection* (bottom-up approach)<br>• *Stepwise regression* (combines forward and backward) |

| | |
|---|---|
| **Gauss-Markov Theorem** | The Gauss-Markov theorem states that in a linear regression model in which errors<br>• Have expectation zero<br>• Are uncorrelated<br>• Have equal variances<br>The best (lowest variance) linear unbiased estimator (BLUE) of the coefficients is given by the ordinary least squares (OLS) estimator. |
| **Unbiased** | Expected value for estimator "is true" ($E(\beta) = \beta$) |
| **Consistent** | Var($\beta$) decreases with increasing sample size n |
| **Efficient** | Estimator $\beta$ has lower variance than any other estimator |
| **OLS assumptions** | • *Linearity*: linear relationship in parameters $\beta$ (when linearity does not hold, try to reformulate)<br>• *No multicollinearity*: no linear dependency between predictors<br>• *Homoscedasticity*: residuals exhibit constant variance (the spread of the data points does not change much)<br>• *No autocorrelation*: there is no correlation between the I and j residual terms<br>• *Exogeneity*: expected value of the residual vector, given X, is 0 |
| **Outlier** | An outlier is an observation that is unusually large or small. **Possibilities**:<br>• There was an error in recording the value<br>• The point does not belong in the sample<br>• The observation is valid |
| **Multicollinearity check** | 1. *Calculate the correlation coefficient for each pair of predictor variables*<br>→ Large correlations (greater than the correlations between predictor and response) indicate problems.<br>2. *Variance Inflation Factor (VIF)*: $\text{VIF} = \frac{1}{1-R_k^2}$ where the $R_k^2$ is the value when the predictor in question (k) is set as the dependent variable (how much of the variance can be explained)<br>→ remove variables with VIF scores greater than 10<br>If the variable has a non-significant t-value, then either<br>• The variable is not related to the response<br>• The variable is not related to the response, but it is not required in the regression because it is strongly related to a third variable that is in the regression<br>→ The usual remedy is to drop one or more variables from the model |
| **Heteroscedasticity** | When the requirement of a constant variance is violated.<br>Breusch-Pagan test or White test are used to check for heteroscedasticity. If there is heteroscedasticity, the estimated Var(β) is biased and OLS might not be efficient anymore |
| **Autocorrelation** | By examining the residuals over time, no pattern should be observed. Reasons of autocorrelation:<br>• Omission of an important variable<br>• Functional misfit<br>• Measurement error in the independent variable<br>Durbin-Watson (DW) statistic to test for first order autocorrelation. |
| **Modeling seasonality** | A regression can estimate both the trend and additive seasonal indexes:<br>• Create dummy variables which indicate the season<br>• Regress on time and the seasonal variables<br>• Use the multiple regression model to forecast<br>$y = \beta_0 + \beta_1 t + \beta_2 Q_1 + \beta_3 Q_2 + \beta_4 Q_3$ (if all 4 Qs are modelled → multicollinearity!) |
| **Exogeneity** | Other factors, which are not explicitly accounted for in the model but are contained in the error term, are not correlated with X |
| **Endogeneity** | Endogeneity is given when an independent variable is correlated with the error term and the covariance is not null (e.g. omitted variable) |
| **Cross-section data** | Refers to data observing many subjects at the same point in time, or without regard to differences in time (there might be omitted variables describing important characteristics of individuals) |
| **Panel Data** | A panel data set is one where there are repeated observations on the same units<br>• *Balanced panel*: every unit is surveyed in every time period<br>• *Unbalanced panel*: some individuals have not been recorded in some time period<br>Individual effects:<br>• *Fixed effects*: individual-specific effects are correlated to other covariates (endogeneity)<br>• *Random effects*: individual-specific effects are uncorrelated to other covariates<br>The Hausman test can help decide on one or the other (the test takes into account the covariance matrix of the FE and RE estimators as well as the estimates and follows a chi-square distribution) |
| **Fixed effect model** | Treat $\lambda$ (the individual-specific heterogeneity) as a constant for each individual<br>$y_{it} = (\beta_0 + \lambda_i) + \beta_1 x_{1it} + \beta_2 x_{2it} + \ldots + \beta_p x_{pit} + \varepsilon_{it}$ |

| | |
|---|---|
| | **Estimators**: first differences, within, between, least squares dummy variable estimator |
| **Random effect model** | The larger the variance of individual effects $\lambda$, the more random effects accounts for it.<br>$Y_{it} = \beta_0 + \beta_1 X_{1it} + \beta_2 X_{2it} + \beta_3 X_{3i} + \beta_4 X_{4i} + \varepsilon_{it}$ |
| **Logistic Regression** | The logistic function Pr[Y\|X] constraints the estimated probabilities to lie between 0 and 1. Pr[X\|Y] is the estimated probability that the i$^{th}$ case is in a category and $\beta 0 + \beta 1X$ is the regular linear regression equation → Probability of success (Y=1) given the predictor variable (X) is a logistic function (non-linear)<br>• $\beta 0$ is the regression constant<br>• $\beta 1X$ is the regression slope (if $\beta 1 < 0$ and X increases → the odds go down, opposite by $\beta 1>0$)<br>**Odds**: by algebraic manipulation, the logistic regression equation can be written in terms of an odd of success (range from 0 to INF).<br>**Logit**: taking the natural log of both sides, we can write the equation in terms of logits (log-odds).<br>The presence of multicollinearity will not lead to biased coefficients, but it will have an effect on the standard errors.<br>The inclusion f irrelevant variables can result in a poor model fit (consult Wald statistics).<br>Multiple Logistic Regression: more than one independent variable |
| **Multinomial Logit model** | The dependent variable, Y, is a discrete variable that represents a choice, or category, from a set of mutually exclusive choices or categories (more than 2). → **conditional logit**<br>Ordered logit models have ordinal dependent variables. |
| **Generalized Linear Models (GLM)** | GLMs are a general class of linear models that are made up for three components:<br>• *Random component*: identifies dependent variable $\mu$ and its probability distribution<br>• *Systematic component*: identifies the set of explanatory variables ($X1$ , … , $Xk$ )<br>• *Link function*: identifies a function of $\mu$ that is a linear function of the explanatory variables<br>  ○ Identity link: form used in normal linear regression models<br>  ○ Log link: used when $\mu$ cannot be negative as when data are Poisson counts<br>  ○ Logit link: used when $\mu$ is bounded between 0 and 1 as when data are binary |
| **Maximum Likelihood Estimation (MLE)** | Maximum Likelihood estimation is a statistical method for estimating the coefficients of a model. (MLE involves finding the coefficients that make the log of the likelihood function (LL<0) as large as possible.<br>Likelihood Function (L): measures the probability of observing the particular set of dependent variable values that occur in the sample |
| **Goodness of Fit** | • *Null model*: assumes one parameter (the intercept) for all of the data points, which means you only estimate 1 parameter<br>• *Fitted model*: assumes you can explain your data points with p parameters and an intercept term, so you have p+1 parameters<br>• *Null deviance* (-2ln(L(null))): how much is explained by a model with only the intercept<br>• *Residual deviance* (-2ln(L(fitted))): small values mean that the fitted model explains the data well<br>• *Likelihood ratio test*: Non-significant χ 2 values indicate that a significant amount of the variance is unexplained<br>• *Wald test*: used to test the statistical significance of each coefficient in the model hypothesis that $\beta i$ = 0 |
| **Count Variables** | Count variables are non-negative integers. (OLS bad regression model as count variables cannot be negative and are often highly skewed. |
| **Count models** | • *Poisson regression model*<br>• *Negative binomial regression model* |
| **Poisson Regression** | In Poisson regression, y is typically conceptualized as a rate (positive coefficients indicate higher rate).<br>Poisson models are non-linear (coefficients don´t have a simple linear interpretation<br>Poisson Model has a log form; exponentiation aids interpretation<br>**Assumptions:**<br>• The mean and variance are the same (often not met in real data --> variance is often greater than $\mu$: Overdispersion)<br>• **Overdispersion** → standard errors will be underestimated; potential overconfidence in results<br>→ Negative binomial regression as alternative to Poisson regression |
| **Zero-Inflation** | If outcome variable has many zero values it tends to be highly skewed.<br>**Zero-inflated model**: assume two types of groups in your sample:<br>1. *Type A:* Always zero (no probability of non-zero values)<br>2. *Type ~A*: non-zero chance of positive count value<br>→ Use logit to model group membership (A or ~A)<br>→ Use Poisson or NB regression to model counts for those in group ~A<br>→ Compute probabilities based on those results |

| | |
|---|---|
| **Classification** | Given a database D of tuples and a set of classes C, the classification problem is to define a mapping D-→ C where each x is assigned to one class. A class contains precisely those tuples mapped to it. Prediction is similar, but usually implies a mapping to numeric values instead of a class C.<br>Algorithms:<br>• *Logistic Regression*<br>• *Statistical Modeling* (e.g. Naïve Bayes)<br>• *Decision Trees*: divide and conquer<br>• *Classification Rules* (e.g. PRISM)<br>• *Instance-Based Learning* (e.g. kNN)<br>• *Support Vector Machines*<br>• ... |
| **Naïve Bayes Classifier** | Naïve Bayes classifier takes all attributes into account. **Assumptions**:<br>• All attributes are equally important<br>• All attributes independent (value of one value tells nothing about value of another attribute)<br>Adding to many redundant attributes will cause problems |
| **Bayesian (Belief) Networks** | Bayesian Belief Network describe conditional independence among subsets of attributes: combing prior knowledge about dependencies among variables with observed training data (Graphical representation: directed acyclic graph)<br>Ex. P(A,B,C,D,E) = P(A) P(B\|A) P(C\|A,B) P(D\|A,B,C) P(E\|A,B,C,D)<br>**Inference**: other than the joint probability of specific events, we may want to infer the probability of an event, given observations about a subset of other variables |
| **Conditional probability** | Probability of events change when we know something of the world.<br>P(A\|B) → Probability of A given that we know B is true = P(A∩B)/P(B)<br>If events A and B do not influence each other → P(A\|B) = P(A)P(B)<br>$\Pr[h\|e] = \frac{\Pr[e\|h]\Pr[h]}{\Pr[e]}$<br>**Normalization**:<br>$\Pr[h\|e] = \frac{\Pr[e_1\|h]\Pr[e_2\|h]\ldots\Pr[e_n\|h]\Pr[h]}{\Pr[e]}$<br>Dealing with numeric attributes → **Probability density function**<br>$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$ (normal or gaussian probability)<br>For unknown distributions → **Kernel Density Estimate** |
| **Zero Frequency Problem** | Attribute value does not occur with every class value → add 1 to the numerator for every attribute-class combination, and the probability can never be 0<br>**Modified Probability Estimates**: in some cases adding a constant different from 1 might be more appropriate |
| **Probability Laws** | • *Chain rule*: the joint distribution is independent of the ordering<br>• *Conditional independence*: event A causes something in event B (rain → umbrella) |
| **Decision Trees** | • *Internal node*: test on an attribute<br>• *Branch*: outcome of the test<br>• *Leaf node*: class label or class label distribution |
| **Building Decision Tree** | • *Top-down tree construction*: all training examples are at the root. Partition the examples recursively by choosing one attribute each time<br>• *Bottom-up tree pruning*: remove subtrees or branches, in a bottom-up manner, to improve the estimated accuracy on new cases (on the basis of an evaluation)<br>The output of decision trees can be used for descriptive as well as predictive purposes |
| **Attribute selection** | • Choose the attribute which will result in the smallest tree (not good)<br>• Choose the attribute that produces the "purest" nodes (heuristics)<br>• Choose attribute that results in the greatest information gain |
| **Entropy** | Gives the information required in bits. The distribution´s entropy represents the info required to predict an event<br>$\text{entropy}(p_1, p_2, \ldots, p_n) = -p_1\log_2 p_1 - p_2\log_2 p_2 \ldots - p_n\log_2 p_n$<br>(entropy = 0 → perfectly ordered system) |
| **Expected Information Gain** | Gain(S,a) is the information gained adding a sub-tree. Problems:<br>• Attributes with a large number of values (e.g. ID codes)<br>• Subsets are more likely to be pure if there is a large number of values → Overfitting<br>**Gain ratio**: modification of the information gain that reduces its bias on high-branch attributes. Gain ratio takes number and size of branches into account when choosing an attribute<br>(Problem: gain ratio may overcompensate → choose an attribute just because intrinsic information is very low) |

| | |
|---|---|
| | Intrinsic Information: how much info do we need to tell which branch an instance belongs to |
| **Splitting Criterion** | • *Gini Index*: frequency of positive and negative classes (select the split that decreases the Gini index most)<br>• *C4.5 and CART* |
| **Industrial-Strength Algorithm** | For an algorithm to be useful in a wide range of real-world applications it must: (e.g. C4.5)<br>• *Permit numeric attributes*: every attribute has many possible split points (evaluate info gain) Numeric attributes may be tested several times along a path in the tree<br>• *Allow missing values*<br>• *Be robust in the presence of noise* |
| **Handling Missing Values (Tree)** | • *Ignore instances with missing values*<br>• *Ignore attributes with missing values*<br>• *Treat missing value as another nominal value*<br>• *Estimate missing value*<br>• *Follow the leader*: an instance with a missing value is sent down the branch with the most instances<br>• *Partition the instance*: send down part of the instance proportional to the number of training instances |
| **Overfitting** | Chasing every abnormality (noise, outliers) causes overfitting:<br>• Decision tree gets too large and complex<br>• Good accuracy on training set, poor accuracy on test set<br>• Does not generalize the data any more<br>→ **Prune the tree** |
| **Propositional rules** | Rules comparing attributes to constants are called propositional rules |
| **Pruning** | • *Prepruning*: tries to decide a priori when to stop creating subtrees (halt tree construction)<br>• *Postpruning*: simplifies an existing decision tree (construct complete tree)<br>  ○ Subtree replacement: replace a subtree with a single leaf node<br>To determine if a node should be replaced, compare the error rate estimate for the node with the combined error rates of its children → replace the node if combined error rate is higher |
| **Data understanding** | • *Quantity*<br>  ○ Number of instances (>5000)<br>  ○ Number of attributes (<50)<br>  ○ Number of targets (>100 for each class)<br>• *Visualization*<br>• *Data summaries* |
| **Data preparation** | • *Data cleaning*:<br>  ○ Missing values<br>  ○ Discretization: reduces the number of values for a continuous attribute (binning)<br>  ▪ Equal Frequency: bins with equal number of instances<br>  ▪ Class Dependent (supervised discretization)<br>• *Conversion*: ordered attributes to numeric attributes<br>• *Building balanced train sets* → **Subset Selection**<br>  ○ Best subset (computationally expensive)<br>  ○ Backward elimination (top-down approach)<br>  ○ Froward selection (bottom-up approach)<br>  ○ Stepwise regression (combines forward/backward) |
| **Validity** | • *External validity*: a statistical study has external validity if its inferences and conclusions can be generalized from the population and setting studied to other populations and settings<br>• *Internal validity*: a statistical analysis has internal validity if the statistical inferences about causal effects are valid for the population being studied. Problems:<br>  ○ Misspecification of the functional form of the model<br>  ○ Measurements errors in the independent variable<br>  ○ Simultaneous causality<br>  ○ Omitted variable bias<br>  ○ Sample selection bias (if data is not collected via a randomized controlled trial) |
| **Data Collection** | • *Randomized controlled trials (RCTs)*: randomized experiments, where each subject is randomly assigned to a treated group or a control group in order to control for extraneous factors<br>• *Quasi-experiments*: compare natural groups and measure effects without randomization of the subjects. The independent variable is controlled, but the assignment of subjects is not random |

<table>
<tr><td></td><td>

- *Observational studies*: draw inferences from a sample to a population where the independent variable is not under control of the researcher
  - Cross-sectional studies: data collection at one specific point in time
  - Longitudinal study: repeated observations of the same variables over long periods of time
  - Panel study: group of subjects is closely monitored over a span of time
  - Case-control study: two existing groups differing in outcome are identified and compared on the basis of some supposed causal attribute

</td></tr>
<tr><td>

**Confounding variable**

</td><td>

Is an extraneous variable in a statistical model that correlates (directly or inversely) with both the dependent variable and the independent variable, in a way that “explains away” some or all of the correlation between these two variables.

**Identification strategies**:

- *Randomized controlled trials*
- *Fixed effects models for panel data*: eliminates alternative explanations that are “fixed” across units
- *Propensity score matching*: only difference in similar subjects is treatment
  a. Estimate propensity score: individual being selected in the treatment
  b. Match subjects with similar propensity score (balance the pretreatment covariates)
  c. Evaluate quality of matching: check if the treatment and comparison group are similar
  d. Evaluate outcomes
- *Instrument variables*: determines observed versus unobserved explanations for taking treatment, and only uses observed portion
- *Regression discontinuity analysis*
- *Difference-in-differences for quasi-experimental data*: if the treatment in quasi-experiment is as if subjects were randomly assigned, we can use the differences regression
- ...

</td></tr>
<tr><td>

**Common trend assumption**

</td><td>

Treatment and control group have the same overall trend

</td></tr>
<tr><td>

**Experiments**

</td><td>

- *Lab experiment*: create a situation with desired conditions, manipulate some variables while controlling others, examine the dependent variable
- *Field experiment*: research study in a natural setting, manipulate some variables, examine the dependent variable

<table>
<tr><th></th><th>Randomized experiment</th><th>Quasi-experiment</th></tr>
<tr><td>Field</td><td>High internal validity/<br>High external validity</td><td>Low internal validity /<br>High external validity</td></tr>
<tr><td>Lab</td><td>High internal validity/<br>Low external validity</td><td>Low internal validity/<br>Low external validity</td></tr>
</table>

</td></tr>
<tr><td>

**CRISP Data Mining Process**

</td><td>

1. *Business understanding*
2. *Data evaluation*
3. *Data preparation*
4. *Modeling*
5. *Evaluation*
6. *Deployment*

</td></tr>
<tr><td>

**Bias-Variance Tradeoff**

</td><td>

The bias-variance tradeoff provides a conceptual framework for determining a good model

- Models with too few parameters are inaccurate because of a large bias (not enough flexibility)
  → **Underfitting**: model is too simple to represent all relevant characteristics (high bias, low variance)
- models with too many parameters are inaccurate because of a large variance (too much sensitivity to the sample)
  → **Overfitting**: model is too complex and fits irrelevant characteristics/noise (low bias, high variance)

</td></tr>
<tr><td>

**Model selection**

</td><td>

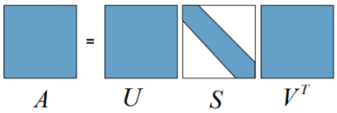
Estimating performances of different models to choose the best one (minimum of the test error)

- *Akaike Information Criterion (AIC)*
- *Minimum description length*
- *Resampling methods* (cross validation, jackknife, bootstrap, etc...
  - Holdout procedure: reserve some data for testing
  - Stratified holdout: guarantee that class are proportionally represented in the test and training set
  - Repeated holdout: randomly select holdout set several times and average the error rate estimates

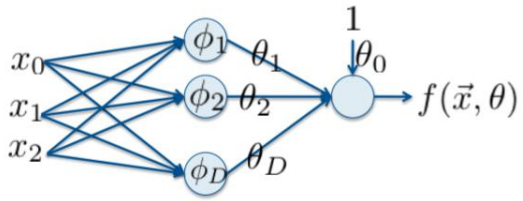
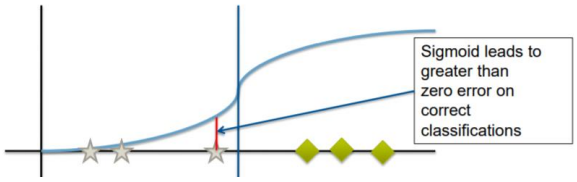
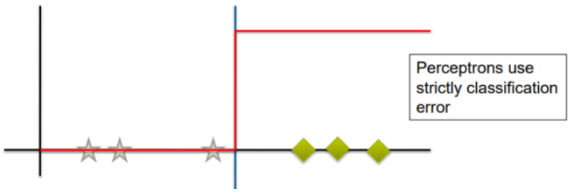
</td></tr>
</table>

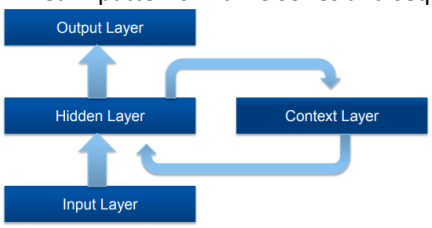
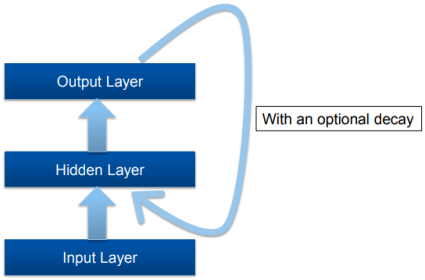
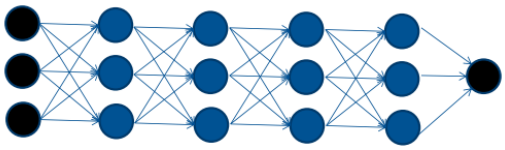
| | |
|---|---|
| **Resampling methods** | • *Cross validation*: Train most of the data and use remaining for testing (use all parts of the data once for testing) → k-fold cross validation: divide data in k partition and use 1 for testing (estimate error rates)<br>• *Jackknife* (Leave-One Out Holdout): use all but one instance for training. Each iteration is evaluated by predicting the omitted instance<br>• *Bootstrap*: sampling several times with replacement from training set to form a "bootstrap" data set. Some observations are considered more than once and others not at all. Prediction are made for original training set (process repeated many times) |
| **Model assessment** | Having chosen a model, estimating the prediction error on new data |
| **Measuring errors** | • *Error rate* = (False negative (FN) + false positive (FP)) / N (Instances)<br>• *Recall* (hit rate) = TP / (TP+FN)<br>• *Precision* = TP / (TP+FP)<br>• *Specificity* = TN / (TN+FP)<br>• *False alarm rate* = FP / (FP+TN) |
| **Cost-sensitive learning** | Weighting of instances according to costs (ex. Increase the "no" instances in training --> when testing on the original test data set, there will be fewer false positives) |
| **Gain Curve** | Instances are sorted according to their predicted probability (x axis = sample size, y axis = number of positives)<br>• *Random List* (ex. 5% of random list have 5% of targets)<br>• *Model-Ranked List* (ex. 5% of model ranked list have 20% of targets Gain(5%) = 20%) |
| **Lift Curve** | How much times better the model-ranked is in comparison to the random list |
| **ROC Curves** | (Receiver Operating Characteristics) Recall vs. False alarm rate: y axis shows percentage of true positives in sample, x axis shows percentage of false positives in sample<br>• *Jagged curve*: one set of test data<br>• *Smooth curve*: use cross-validation and average |
| **Real-world comparison studies** | • *Logistic regression* (discriminant analysis) - widely used<br>• *Decision trees* – widely used<br>• *K-nearest neighbor*<br>• *Non-parametric statistical methods*<br>• *Neural networks* |
| **Algorithmic Information Theory** | • *Kolmogorov complexity*<br>• *Minimum description length principle*<br>A good model is a simple model that achieves high accuracy on the given data<br>**Theory 1**: very simple, elegant theory that explains the data almost perfectly (preferable)<br>**Theory 2**: significantly more complex theory that produces the data without mistakes |
| **Kolmogorov complexity** | The Kolmogorov complexity (K) of a binary object is the length of the shortest program that generates this object on a universal Turing machine (random strings are not compressible)<br><u>Kolmogorov complexity is not computable</u>! (it needs approximation) |
| **Minimum description length principle** | MDL restricts the set of allowed codes in such a way that it becomes possible (computable) to find the shortest codelength of the data, relative to the allowed codes (model selection criterion)<br>**DL** = space required to describe a theory + space required to describe the theory's mistakes |
| **Computational learning theory** | • *Probably approximately correct (PAC) learning*: only reasonable expectation of a learner is that with high probability it learns a close approximation to the target concept<br>• *Vapnik-Chervonenkis (VC) theory*: provides a measure of the expressiveness of infinite hypothesis spaces<br>• *Bayesian inference*<br>**Concepts**:<br>• *Sample complexity*: how many training examples are needed for a learner to converge to a successful hypothesis<br>• *Computational complexity*: how much computational effort is needed for a learner to converge to a successful hypothesis<br>• *Mistake bound*: how many trainings examples will the learner misclassify before converging to a successful hypothesis |
| **Ensembles** | Combining multiple models<br>• **(+)** often improves predictive performance<br>• **(-)** usually produces output that is very hard to analyze<br>Methods:<br>• *Bagging*<br>• *Random Forests* |

| | |
|---|---|
| | • *Boosting*<br>• *Stacking* |
| **Bagging** | Combining predictions by voting/averaging<br>1. Sample several training sets of size n from the population<br>2. Build a classifier for each training set<br>3. Combine the classifiers´ predictions<br>- If the learning scheme is unstable (small change in the data causes big change in the model) bagging almost always improves the performance<br>• (+) Can be applied to numeric prediction and classification<br>• (+) Can help a lot if the data is noisy<br>• (+) Can easily be parallelized because ensemble members are created independently |
| **Bias-Variance decomposition** | The bias-variance decomposition is used to analyze how much restriction to a single training set affects performance.<br>• *Bias*: expected error of the ensemble classifier on new data<br>• *Variance*: component of the expected error due to the particular training set being used to build our classifier<br>• *Total expected error* = bias + variance<br>→ Combining multiple classifiers generally decreases the expected error by reducing variance |
| **Random Forests** | Random forests randomize data and features (more generally applicable than bagging).<br>1. Draw a bootstrap sample from the data<br>2. Grow a "random" tree, where at each node, the best split is chosen among mtry randomly selected variables. The tree is grown to maximum size and not pruned back<br>3. Store the resulting decision tree<br>4. For each of the decision tree predict class of instance |
| **Boosting** | Boosting tries to minimize the bias in terms of training performance of simple learners (ex. AdaBoost)<br>Boosting needs weight, but you can apply boosting without weights (resample data with probability determined by weights)<br>- Boosting implements forward stagewise additive modelling (well-known statistical technique) |
| **Forward Stagewise additive model** | 1. Build simple regression model<br>2. Gather residuals, learn model predicting residuals, and repeat<br>3. Sum up individual predictions from all regression models |
| **Additive Regression** | Additive regression greedily minimizes squared error of ensemble if base learner minimizes squared error |
| **Stacking** | In stacking, the predictions from heterogeneous classifiers are used as input into a meta-learner, which attempts to combine the predictions to create a final best predicted classification<br>• *Level-0 Models*:<br>  ○ Decision tree<br>  ○ Naïve Bayes<br>  ○ Instance-based<br>• *Level-1 Model*<br>  ○ Meta learner |
| **Meta Learning** | 1. Holdout part of the training set<br>2. Use remaining part for training level-0 methods<br>3. Use holdout data to train level-1 learning<br>4. Retrain level-0 algorithms with all the data<br>If the base learner can output class probabilities, use those as input to meta learner instead of plain classifications |
| **Clustering** | Find a natural partitioning of the dta set into a number of clusters such that:<br>- *Intra-cluster similarity is maximized* (items in same cluster are similar)<br>- *Inter-cluster similarity is minimized* (items in different clusters are different)<br>Clusters are not known a priori.<br>Many different methods and algorithms:<br>- *Numeric and/or nominal data*<br>- *Deterministic vs. probabilistic*<br>- *Partitional vs. overlapping*<br>- *Hierarchical vs. flat* |
| **Hierarchical Clustering** | **Bottom up**:<br>1- Start with single instance clusters<br>2- At each step, join the two closest clusters<br>**Top down**:<br>1- Start with one universal cluster |

| | |
|---|---|
| | 2- Find two clusters<br>3- Proceed recursively on each subset<br>(**MST Algorithm**: compute the minimal spanning tree of the graph) |
| **K-Means clustering** | K-Means is an example of a partial clustering algorithm<br>1- Pick a number (k) of cluster centers (at random) → number o clusters determined a priori<br>2- Assign every item to its nearest cluster center (e.g. using Euclidean distance)<br>3- Move each cluster center to the mean of its assigned items<br>4- Repeat steps 2 and 3 until convergence<br>Results can vary significantly depending on the initial choice of seeds. To increase chance of finding global optimum restart with different random seeds |
| **K-Means Pros and Cons** | **(+)** Simple and understandable<br>**(+)** Items automatically assigned to clusters<br>**(-)** Must pick number of clusters before hand<br>**(-)** All items forced into clusters (sensitive to outliers) |
| **Probability-based clustering** | Model each cluster with a probability distribution (mixture). Each probability distribution gives the probability of an instance being in a given cluster.<br>1- Start with initial guesses for the parameters<br>2- Calculate cluster probabilities for each instance (expectation)<br>3- Re-estimate the distribution parameters form probabilities (maximization)<br>4- Repeat<br>The maximum found by EM could be a local optimum, so repeat several times with different initial values.<br>How do we know the parameters for the mixture? Use an iterative approach similar in spirit to the k-means algorithm.<br>**Extending the model**<br>- *Multiple clusters*: use k normal distributions<br>- *Multiple attributes*: multiply the probabilities of all attributes to get the probability of an instance (in case of correlation among attributes use multivariate normal distribution)<br>- *For nominal attributes*: crate probability distributions for the values |
| **Dimensionality problem** | Dimensionality reduction is used when the number of variables exceeds the number of observations |
| **Principal Component Analysis (PCA)** | Principal component analysis (PCA) converts a set of possibly correlated variables into a (possibly smaller) set of values of linearly uncorrelated variables called principal components. The principal components are orthogonal (they are the Eigenvectors of the symmetric covariance matrix) → Rotation of the axes<br>→ Eigenvalues explain the proportion of variance explained by PC<br>→ The PCA score for any of the x is just its coefficient in each of the y<br>PCA Stages<br>1. *Calculate Zero mean data* (calculate the men of each column and subtract the mean from each value)<br>2. *Calculate the variance matrix*, which summarizes the relationship between variables (if variables are measured in different units use the correlation matrix<br>3. *Calculate the Eigenvectors and Eigenvalues of the covariance matrix*<br>4. *Order the Eigenvectors by Eigenvalues* (from highest to lowest)<br>5. *Rotate to get the Eigenvectors as axes*<br>You can decide to ignore the components of lesser significance (lose some information)<br>→ How many dimensions should remain? Take enough Eigenvalues to cover 80-90% of the total variance (the first k components display as much as possible of the variation in the data)<br>- The first PC is direction of maximum variance from the origin, subsequent PCs are orthogonal to first PC and describe maximum residual variance<br>PCA can be used for example for image compression |
| **PCA Assumptions** | - PCA assumes relationships among variables are linear (if the structure of the data is nonlinear, the principal axes will not be an efficient and informative summary of the data)<br>- PCA uses the Euclidian distance among points assuming continuous variables. With discrete variables special techniques are in order) |
| **Singular Value Decomposition** | SVD of the data matrix can be seen as an alternative technique to compute the same Eigenvectors.<br>$A = USV^T$<br>  <br>$A$ $U$ $S$ $V^T$<br>- The diagonal values of S are called the *singular values* (it is accustomed to sort them by size)<br>- The Columns of U are called the *left singular vectors* |

| | |
|---|---|
| | - The columns of V are called the *right singular vectors* (principal axes)<br>- The columns of US are the *principal components* (scores)<br>- Singular values of the SVD decomposition of the matrix A is the square root of the Eigenvalues of the matrix ($AA^t$) or ($A^tA$) |
| **Principal Component Regression** | PCR will tend to do well in cases when the first few principal components are sufficient to capture most of the variation in the predictors as well as the relationship with the response.<br>1. PCA to compress the data<br>2. PC regression<br>In PCR, the number of principal components is typically chosen |
| **Partial Least Squares** | Partial Least Squares (PLS) is just like PC Regression except in how the components are computed. PLS makes use of the response y in order to identify new features that do not only approximate the old features well, but also that are related to the response.<br>→ Weights are calculated from the covariance matrix of the predictors<br>→ Weights reflect the covariance structure between predictors and response y |
| **Regularization** | By regularizing the estimator in some way, its variance will be reduced. If the corresponding increase in bias is small, this will be worthwhile.<br>• *Subset selection* (forward, backward, all subsets)<br>• *Ridge regression*<br>• *The lasso* |
| **Ridge regression** | Ridge coefficient minimize a penalized RSS. This is a biased estimator that for some value of ?>0 may have smaller mean squared error than the least squares estimator.<br>Ridge regression estimates will be more biased than the OLS ones but have lower variance |
| **The Lasso** | The lasso coefficients minimize the quantity. The lasso has a major advantage over the ridge regression, in that it produces simpler and more interpretable models that involved only a subset of predictors.<br>The lasso can generate more accurate predictions compared to ridge regression.<br>→ Cross-validation can be used in order to determine which approach is better on a particular data set |
| **Ridge-PCA-PLS-Lasso** | • Ridge regression and PCR outperform PLS in prediction<br>• Lasso outperforms ridge when there are a moderate number of sizable effects, rather than many small effects. It also produces more interpretable models. |
| **Association Rule Discovery** | Aims to discover interesting correlations or other relationships in large databases.<br>- *Market Basket Analysis*: analyze customer buying habits by finding associations and correlations between the different items that customers place in their "shopping basket"<br>**Steps**:<br>1- Find all (frequent) itemsets that meet minimum support<br>2- Find all rules that meet minimum confidence<br>3- Prune<br>**Subset Property**: every subset of a frequent set is frequent<br>**Apriori Algorithm**: use one-item sets to generate two-item sets, two-item sets to generate three-item sets, … |
| **Support of itemset** | Supp(I) is the proportion of transactions that support (contain) I |
| **Frequent Itemset** | A frequent itemset I is one with at least the minimum support (supp(i) > minsupp). Association rules with maximum support and confidence are sometimes called "strong" rules.<br>Frequent itemsets represent sets of items which are positively correlated. |
| **Recommender Systems** | Systems for recommending items to users based on examples of their preferences<br>- *Collaborative filtering*: based on similarities among users tastes. For a given user, find other users whose ratings strongly correlate with the current user → recommend items rated highly by these similar users<br>  ○ Product associations (ex. 90% of users who like A and B also like C)<br>  ○ User associations (ex. 90% of products liked by A and B are also liked by C)<br>  ○ Combination of product and user associations<br>  **(-)** Cold start: needs to be enough other users already in the system to find a match<br>  **(-)** Sparsity: hard to find users that have rated the same items<br>  **(-)** First rater: cannot recommend an item that has not been previously rated<br>  **(-)** Popularity bias: tends to recommend popular items<br>- *Content-Based filtering*: recommendations are based on information on the content of items rather than on other users' opinions<br>  **(+)** Able to recommend to users with unique tastes<br>  **(+)** Able to recommend new and unpopular items<br>  ➔ **SVD**: doesn't work on sparse matrices, but one can estimate the vectors by minimizing the Euclidian distance between $r_{ui}$ and the dot product for the relevant vectors of U and V |

| | |
|---|---|
| **Similarity Weigjting** | Typically use Pearson correlation coefficient between ratings for active user, a, and another user, u |
| **Significance Weighting** | Include significance weights, $s_{a,u}$, based on number of co-rated items, n<br>➔ Important not to trust correlations based on very few co-rated items |
| **Rating prediction** | Predict a rating for each item (for active user A) by using the k selected neighbor users |
| **Neural Network** |  <br>- Edges multiply the signal by a weight<br>- Nodes apply a function, $\phi_d$<br>- Each neuron has its own bias and weights<br>Combining (Activation) Function – **Sigmoid function classification error**<br>  <br>**Perceptron Error**<br>  <br>**Problems with Neural networks:**<br>• *Interpretation of hidden layers* (what are the hidden layers doing?)<br>○ Feature extraction (the non-linearities in the feature extraction can make interpretation of hidden layers very difficult --> NN treated as black boxes)<br>• *Overfitting* |
| **Multi-Layer Feed-Forward Networks** | Multi-layer networks can represent arbitrary functions, but an effective an effective learning algorithm for such networks was thought to be difficult.<br>A typical multi-layer network consists of an input, hidden and output layer, each fully connected to the next, with activation feeding forward. The weights determine the overall function computed<br>**Feed-Forward**: predictions are fed forward through the network to classify<br>--> The output from one layer is the input to the next (each layer has its own sets of weights) |
| **Cost Function (NN)** | Use a cost function to compute the average (misclassification) cost for all training data (for a particular set of weights and biases it computes a single number) --> **Minimize the cost function!**<br>**(-)** There is no "closed-form" solution<br>**(+)** Convex (for any pair of points within a region, the line is in the region) |
| **Backpropagation** | Backpropagation is a method used in artificial neural networks to calculate a gradient that is needed in the calculation of the weights to be used in the network. It describes how a single training example, starting from the output neurons, determines the goal for the neurons on the next layer and steps backwards recursively<br>• *Epoch*: one pass through the training set, with an adjustment to the network weights for each training example. (perform as many epochs as needed to reduce the classification error)<br>• *Loss function*: function that maps values of one or more variables onto a real number intuitively representing the cost associated with the event<br>• *Risk function R*: expectation of the loss function |
| **Gradient descent** | 1. Gradient points in the direction of fastest increase<br>2. To minimize R, move in the opposite direction<br>3. Nearly guaranteed to converge to the minimum<br>→ Can only find a local minimum unless the function is convex<br>→ Can Oscillate if the steps are too large (stall if derivate is ever 0 not at the minimum) |
| **Linear Regression Neural Networks** | The product of two linear transformations is itself a linear transformation (nothing special)<br>→ Non-linearities to identify complex regions in space |
| **Linearly separability** | Two classes of points are linearly separable, if there exists a line such that all the points of the class fall on one side of the line, and all the points of the other class fall on the other side of the line |

| | |
|---|---|
| | If f(x) is linear, the NN can only draw straight decision boundaries<br>→ Use the non-linear, differentiable sigmoidal "logistic" function f(x), which can draw complex boundaries |
| **Hidden nodes** | The number of nodes in the hidden layer affects generality and convergence<br>• Too few hidden nodes → convergence may fail<br>• Few but not too few nodes → possibly slow convergence but good generalization<br>• Too many nodes → rapid convergence, but "overfitting" happens |
| **Error Backpropagation** | Error backpropagation unravels the multivariate chain rule and solves the gradient for each partial component separately. The target values for each layer come from the next layer<br>→ Feeds the errors back along the network |
| **Recurrent Neural Networks** | Output or hidden layer information is stored in a context or memory layer<br>→ Learn patterns in time series and sequential learning tasks<br>  <br><br>**Time Delayed Recurrent Neural Networks (TDRNN)**: Output layer from time t are used as inputs to the hidden layer at time t+1<br>  <br> |
| **Deep Neural Networks** | Many-layer neural network architectures capable of learning the true underlying features and "feature logic", and therefore generalize very well. Regularization is used to combat overfitting.<br>  <br>There is no universally agreed threshold of depth dividing shallow learning from deep learning (most researchers agree that deep learning has multiple nonlinear layers) |
| **Problems with Neural Networks** | • *Lack of transparency* (where is the knowledge?)<br>• *Difficulty in predicting convergence*<br>• *Difficulty in scaling up* (NNs are often useful subsystems, but highly complex systems must be carefully structured into separately trainable subsystems) |

| | |
|---|---|
| **Point Estimate** | $\bar{X} = \frac{\sum X}{n}$ |
| **Estimate of variability in population** | $s = \sqrt{\frac{1}{n-1}\sum_i (X_i - \bar{X})^2}$ |
| **True Standard deviation in sample mean** | $SD = \sigma/\sqrt{n}$ |
| **Standard error of sample mean** | $SE = s/\sqrt{n}$ |
| **95% Confidence interval** | Depends on degrees of freedom |
| **Welch Test**<br><br>**(2 Samples, independent)** | $t_0 = \frac{\bar{x}_2 - \bar{x}_1 - \mu_0}{s_{\bar{x}-\bar{w}}}$ |
| **Paired t-test**<br><br>**(2 Samples, dependent)** | $t_0 = \frac{\bar{d} - \mu_0}{s_d}\sqrt{n}$ |
| **Z-test / Gauss test**<br><br>**(1 Sample, σ known)** | $z = \frac{\bar{X} - \mu_0}{\sigma/\sqrt{n}}$<br>Rejection region:<br>$\mu \neq \mu_0$ $\quad$ $\lvert z \rvert \geq z_{\alpha/2}$<br>$\mu > \mu_0$ $\quad$ $z \geq z_\alpha$<br>$\mu < \mu_0$ $\quad$ $z \leq -z_\alpha$ |
| **T-test**<br><br>**(1 Sample, σ unknown)** | $t = \frac{\bar{d} - \Delta_0}{s/\sqrt{n}}$<br>Rejection region:<br>$\mu_d \neq \Delta_0$ $\quad$ $\lvert t \rvert \geq t_{\alpha/2,\, n-1}$<br>$\mu_d > \Delta_0$ $\quad$ $t \geq t_{\alpha,\, n-1}$<br>$\mu_d < \Delta_0$ $\quad$ $t \leq -t_{\alpha,\, n-1}$ |
| **Formulas for coefficients** | $\hat{\beta}_1 = \frac{\sum_i^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_i^n (x_i - \bar{x})^2} = \frac{\frac{1}{n}\sum_i^n x_i y_i - \bar{x}\bar{y}}{\frac{1}{n}\sum_i^n x_i^2 - \bar{x}^2}$ |
| **Residual Sum of Squares (RSS)** | $RSS = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$ |
| **Total Deviation** | $\sum (y - \bar{y})^2 = \sum (\hat{y} - \bar{y})^2 + \sum (y - \hat{y})^2$<br>Total deviation (TSS) = explained deviation (ESS) + unexplained deviation (RSS) |
| **$R^2$ (Coefficient of Determination)** | $R^2 = \frac{TSS - RSS}{TSS} = 1 - \frac{RSS}{TSS} = \frac{ESS}{TSS}$<br>$R^2$ = 1 --> Perfect match between the line and data points<br>$R^2$ = 0 --> There is no linear relationship between x and y |
| **Multiple linear regression model** | $\hat{Y} = \hat{\beta}_0 + \hat{\beta}_1 X_1 + \hat{\beta}_2 X_2 = \mathbf{X}\hat{\beta}$ |
| **Variance Inflation Factor (VIF)** | $\text{VIF} = \frac{1}{1 - R_k^2}$ |
| **Durbin-Watson statistic (DW)** | $DW = \frac{\sum_{i=2}^n (e_i - e_{i-1})^2}{\sum_{i=1}^n e_i^2}$<br>• $DW = 2$ – no autocorrelation<br>• $DW = 0$ – perfect positive autocorrelation<br>• $DW = 4$ – perfect negative autocorrelation |
| **Logit (log odds)** | $\ln\left(\frac{p(x)}{1 - p(x)}\right)$<br>$p$ = 0.50, then logit = 0<br>$p$ = 0.70, then logit = 0.84<br>$p$ = 0.30, then logit = -0.84 |

| | |
|---|---|
| **Odds of success** | $\frac{p(X)}{1-p(X)} = e^{\beta_0+\beta_1 X}$ |
| **Likelihood function for the logit model** | $\Pr(Y_i=1)=F(\beta_0+\beta_1 X_{1i})=\frac{e^{(\beta_0+\beta_1 X_{1i})}}{1+e^{(\beta_0+\beta_1 X_{1i})}}=\frac{1}{1+e^{-(\beta_0+\beta_1 X_{1i})}}$ |
| **Likelihood Ratio test** | $\mathrm{D} = -2\ln\left(\frac{\mathrm{L(null)}}{\mathrm{L(fitted)}}\right) = -2\left(\mathrm{LL(null)} - \mathrm{LL(fitted)}\right)$ |
| **McFadden $R^2$** | $R^2_{McFadden} = 1 - \frac{LL(fitted)}{LL(null)}$ |
| **Conditional Probability (Bayes Rule)** | $\Pr[h\|e] = \frac{\Pr[e\|h]\Pr[h]}{\Pr[e]}$ |
| **Normalization** | $\Pr[h\|e] = \frac{\Pr[e_1\|h]\Pr[e_2\|h]\ldots\Pr[e_n\|h]\Pr[h]}{\Pr[e]}$ |
| **Density function f(x)** | $f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$ |
| **Entropy($p_1$, $p_2$, … , $p_n$)** | $\mathrm{entropy}(p_1, p_2, \ldots, p_n) = -p_1\log_2 p_1 - p_2\log_2 p_2 \ldots - p_n\log_2 p_n$ |
| **Gini Index** | $Gini(S) = 1 - P^2 - N^2 \quad \in [0,0.5]$<br>with<br>$P = p/(p+n)$<br>$N = n/(p+n)$ |
| **C4.5 Method** | $e = p = \left(f + \frac{z^2}{2n} + z * \sqrt{\frac{f}{n} - \frac{f^2}{n} + \frac{z^2}{4n^2}}\right) \Big/ \left(1 + \frac{z^2}{n}\right)$<br>$f$ is the error on the training data<br>$n$ is the number of instances covered by the node |
| **Euclidian distance** | $d_2(x,y) = \left(\sum_{j=1}^{p} \|x_j - y_j\|^2\right)^{1/2}$ |
| **Manhattan distance** | $d(x,y) = \left(\sum_{j=1}^{p} \|x_j - y_j\|\right)$ |
| **Correlation Matrix** | $\rho = corr(X,Y) = \frac{cov(X,Y)}{\sigma_X \sigma_Y}$ |
| **Reconstruction of Original data with one Eigenvector** | **X** ≈ PCA Scores * Eigenvectors + original mean |
| **Similarity Weighting** | $c_{a,u} = \frac{\mathrm{cov}(r_a, r_u)}{\sigma_{r_a}\sigma_{r_u}}$ |
| **Significance Weighting** | $w_{a,u} = s_{a,u} c_{a,u}$<br>$s_{a,u} = \left\{\begin{matrix} 1 \; if \; n > 50 \\ \frac{n}{50} if \; n \leq 50 \end{matrix}\right\}$ |
| **Rating prediction** | $p_{a,i} = \bar{r}_a + \sum_{u=1}^{k} \frac{w_{a,u}(r_{u,i} - \bar{r}_u)}{\sum_{u=1}^{k} \|w_{a,u}\|}$ |
| **Neural network** | $f(\vec{x}, \theta) = \sum_{n=0}^{N-1} \sum_{d=1}^{D} \theta_d \phi_d(x_n) + \theta_0$ |

| | |
|---|---|
| **Logistic Neuron Optimization (gradient descent)** | $$R(\theta) = \frac{1}{2N}\sum_{i=0}^{N-1}(t_i - g(\theta^T x_i))^2$$ derivative of $f(z)^2 \Rightarrow 2f(z)f'(z)$ (chain rule) $$\nabla_\theta R = \frac{1}{2N}\sum_{i=0}^{N-1} 2(t_i - g(\theta^T x_i))(-1)g'(\theta^T x_i)x_i = 0$$ |