**Summary of BI Lectures HS21/22**

**Table of content**

[Business Performance Management, Balanced scorecards](https://github.com/nico-fhnw/msc-medical-informatics/blob/main/Semester%201/Lecture%20Summaries/Business%20Intelligence.md#Business-Performance-Management,-Balanced-scorecards)

[Reporting, dashboarding](https://github.com/nico-fhnw/msc-medical-informatics/blob/main/Semester%201/Lecture%20Summaries/Business%20Intelligence.md#Reporting,-dashboarding)

[Multidimensional modelling](https://github.com/nico-fhnw/msc-medical-informatics/blob/main/Semester%201/Lecture%20Summaries/Business%20Intelligence.md#Multidimensional-modelling)

[Data mining: classifications](https://github.com/nico-fhnw/msc-medical-informatics/blob/main/Semester%201/Lecture%20Summaries/Business%20Intelligence.md#Data-mining-classification)

[Big Data](https://github.com/nico-fhnw/msc-medical-informatics/blob/main/Semester%201/Lecture%20Summaries/Business%20Intelligence.md#Big-Data)

[Information Extraction and Text Mining](https://github.com/nico-fhnw/msc-medical-informatics/blob/main/Semester%201/Lecture%20Summaries/Business%20Intelligence.md#Information-Extraction-and-Text-Mining)

**Business Performance Management, Balanced scorecards**

*BPM Primer - all chapters*

***Learning goals:***

* *Understand how Performance Management works in general and know the basic principles of how Balanced Scorecard is built*
* *Be able to formulate strategic goals and Key Performance Indicators and place them into the appropriate perspective of a Balanced Scorecard*

**Business Performance Management** consists of four different activities:

* Strategize: selection of objectives the organisation wants to achieve
* Plan: top-level objectives broken down into more manageable strategic goals, metrics are derived
* Monitor & analyse: technology is being used to measure the metrics and the achievement of each goal
* Take corrective action: when targets are missed and causes understood, to make sure that goals are achieved in the future

**Balanced Scorecards** describe the different perspectives of a company. The original perspectives are financial and customer (external) and internal business process and Learning & Growth (internal).

A **KPI** is a measurable number that indicates whether a strategic goal has been reached. Following ingredients are usually necessary:

* a formula that defines how the measure results from business data
* the unit of measurement
* whether the goal is to maximize, minimize the measure or keep it in a specific range
* the periodicity of measurement (e.g. daily, monthly, quarterly)
* a target value that defines when the strategic goal is reached (or two e.g. in case of a traffic light system)

**Reporting, dashboarding**

*Statistic Primer - all chapters*

***Learning goals:***

* *Be able to distinguish analytical questions from non-analytical questions*
* *Be able to identify relevant instances and variables descriptions of information needs and categorise these variables into categorical and numerical ones*
* *Be able to suggest suitable representations (statistical summaries, visualisations) based on description of information needs and available data*
* *Be capable to discuss pros and cons of representations; this includes knowing pitfalls that lead to misleading and / or incomplete representations of data*
* *Be able to suggest improvements for visualizations, especially that help stakeholders to quickly spot the important information in charts / tables*

**Individuals, variables and distributions**

Company X has collected data about their customers. Each customer in the data set is a *individual*. Customers are described by several *variables*, such as their city and the number of products they bought.

* categorical variables (qualitative or nominal variables)
* numerical variables (quantitative variables)

A *measurement* is the value of a variable for an individual, such as the customer John Doe lives in Zurich and has 3 kids. The set of all these values is called the *distribution*.

**Summarising and visualising distributions**

**1 categorical variable**

To summarise the distribution of a single categorical variable, the frequency (or relative frequency) of the values of the variable can be used. The data can be visualised by either a *bar chart* or a *pie chart*

*Beware: the pie chart must contain all categories and the categories may not be overlapping.*

**1 numerical variable**

The simplest (and very informative) way of summarising a numerical values visually is to draw a *histogram*. To obtain a histogram, one divides the value range of the measurements into equally sized bins and count how many measure fall into each bin.

*Beware: Bin width must be equal for all bins*

A histogram shows the different shapes of a distribution. Many distributions are *symmetric* (normal distribution), others can be *right-skewed* (bins are small on the right) or *left-skewed* (bins are small on the left).

One special distribution is the *power law distribution*, it is a heavily right-skewed distribution. It appears when very small values are extremely frequent whereas the frequency of the many higher values decrease very quickly. An example of the power law distribution would be the distribution of wealth among individuals.

**Central tendency** Central tendency is usually measured by the arithmetic mean or median. The weakness of the arithmetic mean is that it is heavily influenced by outliers, while the median doesn't suffer from this problem. Mean and median also differ when distributions are skewed. In a right-skewed distribution for instance, the mean is pulled to the right by the large values that exist on the right side, the median on the other hand is smaller since there are many measurements with low values.

**2 or more numerical variable**

Analysing the distribution of more than one numerical variable has the goal to compare the two variables has the goal to compare the two variables - most of the time, to find out if they are correlated. **Correlation is not equal to causation** - but s frequent interest in analysing correlations is to identify causes.

The simplest way of analysing correlation is to create a *scatter plot*. Usually the explanatory variable (or independent variable) is put on the horizontal axis - i.e. the variable that is assumed to cause a change in the value of the other variable, which is the response or dependent variable.

**1 or more categorical and 1 numerical variable**

Often it's interesting to compare the distribution of a numeric variable between two or more groups of individuals. Usually the value of a categorical variable is used to form groups. An option to visualise the comparison are *side-by-side box plots*, where each value of the categorical variable represents a box plot.

For instance, if within a company the salaries of different genders should be compared. One can divide all employees into N amount of gender and then plot the distribution of the numerical variable "salary" for all N groups.

Often, one does not want to compare the distribution as such, but rather compare the values of a function of the numerical variable (sum, average, count, etc.) between different categories. For instance, the revenue (numerical) by year (categorical) or the cost (numerical) by department (categorical). In this case, popular options for visualisations are *bar or line charts*. Line charts in particular when the categorical variable is time.

**Avoid misleading representations**

The following mistakes and pitfalls should be avoided in descriptive statistics as they will lead to misleading summaries or visualisations:

* **Cutting axes:** in a bar chart, when the vertical axis does not start at 0, differences between bars might seem larger than in reality
* **Using absolute numbers** when the underlying number of observations differs
* **Using differently sized bins** in histograms
* **Using pie charts** when the frequencies of the categories do not add up to the number of individuals (e.g. individuals fall into more than one category or only sample of individuals top10 are selected)
* **Using a line chart** for very few data points: line charts always evoke the impression of a trend and people tend to extrapolate the trend to the future

**Multidimensional modelling**

*Multidimensional modelling Primer - all chapters*

***Learning goals:***

* *Be able to identify a suitable grain, as well as facts, dimensions and dimension hierarchies from business information needs and a description of source data. Be able to judge which fact measures and dimension are true to a given grain*
* *Be able to represent multidimensional models using ME/R notation*
* *Know how multidimensional models are represented in BI tools for self-service BI*

**Key concepts**

**Measures, facts and dimensions**

Usually there is either an event triggering a measurement or it's a periodical snapshot, e.g. a customer orders a product or at the end of the day the inventory levels are stored. Each measurement (or order in the prev. example) is a *fact*. With each fact, several *measures* can be associated i.e. things that get measured simultaneously. A measure could be the number of items bought or the total amount paid by the costumer.

The definition of when/how often a measurement should be made is called the *grain* of a multidimensional model.

For instance in a supermarket, we have at least the following three possible grains:

* closing of cash register in one store on one day
* the shopping cart of a customer
* a single line item from a customers shopping cart

When storing measures associated with a fact, more information that describes the event/situation need to be included. This is done via a categorical attribute, called *dimension*. Usually the definition of a grain implies a certain set of dimensions. Additional dimensions can be introduced to satisfy analytical needs - but they must be **"true to the grain"**, i.e. it must be possible to associate one single value of each dimension to a given fact.

**Notations**

**ME/R**

ME/R is used to create very simple models - where attributes of dimensions can't be modelled. There are three different modelling elements:

* A cube symbol: The name of the fact written at the top part of the cube, with the name of the different measures written below
* Hierarchy levels: These are denoted by boxes with a hierarchy symbol inside (dimension).
* An arrow for hierarchical relations: used to connect different hierarchy levels (e.g. city -> state -> country)

**Data mining: classification**

*Predictive Primer - chapters 1-5 & 7-9*

***Learning goals:***

* *Understand the special challenges of ETL operations in data mining, e.g. handling of missing values*
* be able to cast business-relevant questions formally as a classification or regression problem and explain how training data will be prepared according to the formal definition\*
* *be able to design and discuss attributes (feature engineering) and be able to discuss the influence of attribute types on predictive models.*
* *understand the basic operating mode of decision tree and k-nearest neighbour classifiers and be able to assess them with regard to certain quality criteria (and to justify/explain such assessment!); be able to interpret decision trees (including statistical summaries of their nodes)*
* *understand and be able to discuss the limitations of parametric and non-parametric regression methods*
* *Be able to design evaluation procedures, e.g. by deciding how to divide classified samples into training and test set*
* *Be able to set up a cost matrix for a given classification problem*
* *Be able to apply evaluation measures (in particular accuracy and cost), interpret their meaning and use them to inform decisions about classifier selection*
* *Understand typical problems in predictive analytics, such as overfitting and class imbalance and know how to solve them using e.g. pruning and undersampling*
* *Understand the meaning of bias and variance and how variance can be assessed*

**Data preparation**

**Feature selection**

Selecting only a subset of attributes can be a good idea because:

* it decreases the complexity and computing time of algorithms
* some algorithms are not good at dealing with irrelevant and/or redundant attributes and will deliver lower quality if attribute selection is not performed

Most data mining tools offer a range of feature selection algorithms. Typically, these algorithms select attributes which:

1. correlate well with the class attribute
2. are independent, i.e. do not correlate well among each other

**Type conversion**

Some algorithms cannot work with certain types of attributes. Some can only handle numerical features, others only categorical ones.

**Discretisation** means converting numerical to categorical attributes. The values of the variable are mapped into bins such as high, middle, low. There are different discretisation methods used, usually one of those three:

* equal-width binning: bins of the same size of range (0,...,1000; 1001,...2000)
* equal- frequency binning: bins of the same number of sample objects
* Supervised discretisation: tries to determine intervals such thath objects falling into the same bin tend to have the same class value

**Binarization** means converting from categorical to a series of binary attributes. For instance, instead of "country" = "Switzerland" it would be "is\_switzerland" = 1.

**Aggregation** means to arrange categorical attributes in hierarchies. For instance, instead of having a variable with the town for each costumer, it could be aggregated to being either "city", "agglomeration" or "rural".

**Handling missing values**

There are different approaches to handling missing values:

* eliminate data objects with missing attribute values (i.e. delete a row)
* eliminate attribute with missing values from all data objects (i.e. delete a column)
* insert a special value such as "n/a" or "unknown", some algorithms can handle such values, but not all of them
* insert a statistical summary of all other values e.g. mean or median
* copy the value from the data object that is most similar e.g. with kNN

The last option require the most effort, but often manages to come the closest to the real values.

**Modelling**

There are plenty of algorithms offered by data mining tools that can be seen as black boxes. However, to select the right classifier for a given task, one requires some understanding.

**OneRule**

One of the simplest classifiers is the OneRule classifier, which makes a prediction by looking at only one attribute value and only with categorical attributes. This algorithm can be useful as a baseline classifier, i.e. before creating a "real" classifier, OneRule can create a baseline to then evaluate the performance of the chosen classifier

**Decision trees**

A decision tree is a set of cascading questions forming to a shape of a tree. The questions test whether a feature of the data satisfies a certain condition. In order to predict, it starts from the top of the tree and traverse deeper and deeper using the attributes to make decisions along the tree, until a leaf node is reached.

**k-nearest neighbour**

The kNN algorithm is a so called "lazy learner". It does not really learn a model, but simply stores all training examples. It defers the whole prediction effort to the moment when a new data object has to be classified. This requires a measure of similarity or distance between data objects. One has to define k = the nearest neighbours, say k = 3, the algorithm will then decide the prediction of the new data object based on the 3 most similar/nearest other data objects and predicts it's class with a simple majority vote (hence it's best to use an odd number for k).

**Evaluation**

see [Lecture 5 Data Science 2](https://github.com/nico-fhnw/msc-medical-informatics/blob/main/Semester%201/Lecture%20Summaries/Biomedical%20Data%20Science.md#Lecture-5-Data-Science-2) of the Bio Medical Data Science module

**Typical problems**

**Bias - variance problem**

When evaluating a learned model, two problems often need to be balanced:

* **Bias:** your model is said to have a high bias when it is too simple to capture the relevant patters, such as the relationships between features and class attributes. That means you have an **underfitting**. This can be recognized by observing the performance of the model on the training data: if the model performs poorly even on the training data, there is an underfit / high bias
* **Variance:** your model is said to have a high varience if it is sensitive to certain particularities in the (training) data. In that case you have an **overfitting**. This can be recognized by observing a good performance on the training data, but a significantly worse performance on the test data.

**Class imbalance**

Another common problem is the class imbalance, i.e. the situation where one value of the class is much more frequent than other values. A typical example would be fraud detection, where the number of non-fraudulent cases is much higher than the number of fraud cases. In that case the fraud cases would be the "minority class" and the non-fraudulent cases the "majority class".

One possible way to address that would be to expose the classifier to more examples of the minority class while training - via what is called *oversampling* or *undersampling*. Most classifiers have a natural tendency to become more sensitive to something that they have seen more often.

**Tuning the model**

Once you have evaluated your model, deciding the next action to improve the model becomes rather straightforward:

* Does the model have a high bias? it's underfitting, so increase complexity and tune its parameters
* Does the model have a high variance? it's overfitting, use more training data, reduce complexity
* Does the model have any frequent errors? often, such frequent errors can be tackled by changing the sampling strategy for the training set, e.g. adding more data of a specific value/variable can help the model to better pick up corresponding patterns.

**Big Data**

*Video primer*

***Learning goals:***

* *Velocity: be able to recognize - given the description of some raw data stream - threats or opportunities that can be detected as high-level events from the event stream and what is the special business value of early detection; be able to specify simple and complex events for event processing*
* *Orchestration: be able to propose a big data architecture / orchestration of big data technologies for a given scenario of source data characteristics and analytical needs; be able to describe the components of a lambda architecture and the kind of processing happening inside these components*

**Velocity**

Regarding Velocity, we speak about data streams, so data that gets produced at fast rates. Single data points are called events.

**Simple event processing** Goal: detect simple events of certain type as early as possible in order to react immediately, e.g. filter out delayed flights from all incoming flight on an airport

**Complex event processing** Same starting point: a stream of events coming in

Complex events can be formed out of simple events. For instance, when we have a camera taking pictures of the traffic, each picture would be a simple event. Predicting a traffic jam based on this data would be a complex event.

Typical characteristics:

* high-level event (complex event) unfolds over time
* consists of low-level events (simple event) which are tracked electronically
* Recognising the event early (before it is over) has a special business value

**Orchestration**

**Data Lakes** idea: build a giant repository of all data that is produced in a company, store it for later (potential) use

* data lakes require no pre-processing, cleansing or integration -> will be done later when building a data mart out of the data
* storage of data lakes is usually based on HDFS/Hadoop

Advantages: long-term flexibility for answering new questions Criticism: data graveyards, companies often lose track

**Lambda architectures** In a Lambda architecture, all data entering the system is dispatched to both a batch layer and a speed layer:

1. The batch layer has to functions:
   1. manage an append-only set of raw data (master dataset), e.g. a data lake
   2. pre-compute the batch views (data that is refreshed once a day, week)
2. The serving layer indexes the batch views to that they can be queried in low-latency, ad-hoc way
3. The speed layer deals only with recent data (shallow analyses), provides real-time views, performs stream processing
4. Any incoming query can be answered by merging results from batch views and real-time views

**Information Extraction and Text Mining**

*Text mining primer - all chapters*

***Learning goals:***

* *be able to suggest an extraction template (database scheme) for a given data source and given scenario and/or information needs, i.e. be able to recognize which entities and relations should be extracted from input data*
* *understand which (linguistic) obstacles can have a negative impact on IE results and which types of errors (false positives/negatives) they may result in; be able to identify false negatives/positives in extraction results*
* *be able to annotate training data for machine learning-based entity extraction using B-I-O notation; be able to discuss how / under which circumstances manual annotation can be partially avoided*
* *be able to suggest required text mining operations for given information needs and to derive suitable means of representing and/or visualizing data to satisfy these needs.*
* *be able to describe how certain text mining / visualization paradigms (such as spring graphs and/or topic modelling) will be used and interpreted to reach analytical goals and what their pros and cons are*
* *understand the meaning of the output of topic modelling*

**Text mining process**

**Pre-processing:**

1. Segmentation
2. Part-of-speech (POS) tagging
3. Stemming or lemmatization
4. Recognition of synonyms
5. Information extraction

**Mining:**

1. Term extraction
2. Sentiment analysis
3. Association analysis

**Linguistic problems**

Roughly, we can distinguish two types of mistakes:

* **False positives**: these are cases where text portions are annotated as entity mentioned, which are in fact not entities (of the given type), but either ordinary words or entities of a different type
* **False negatives**: these are missing annotations, i.e. cases where entity mentions are not annotated although they are in fact entity mentions.