CH1

CRISP-DM consists of six phases:

The main objective of the first **project understanding** step is to identify the potential benefit as well as the risks and efforts of a successful project, such that a deliberate decision on conducting the full project can be made.
What is the problem and benefit?
What should a solution look like?

Next we need to make sure that we will have sufficient data at hand to tackle the problem. To achieve this, we proceed in the **data understanding** phase with a review of the available databases and the information contained in the database fields. A visual inspection of the relationships between the attributes and an inspection of outliers. For instance, outliers appear to be abnormal in some sense and are often caused by faulty insertion, but sometimes they give surprising insights on closer inspection. Missing fields too.
What data do we have available?
Is the data relevant to the problem?
Is the data quality and quantity sufficient?

In the **data preparation** phase the data is selected, corrected, modified, dimension is reduced, treat missing values, even new attributes are generated, such that the prepared data set best suits the problem and the envisaged modeling technique. Basically all deficiencies that have been identified in the data understanding phase require special actions.
How the data is best transformed for modeling?
Which data should we concentrate on?

Once the data is prepared, we select and apply **modeling** tools to extract *knowledge* out of the data in the form of a *model.* Depending on what we want to do with the model, we may choose techniques that are easily interpretable (to gain insights) or less demonstrative black-box models, which may perform better. Background knowledge may provide hints on useful *transformations* that simplify the representation of the solution.
Compared to the modeling itself, which is typically supported by efficient tools and algorithms, the data understanding and preparation phases take considerable part of the overall project time as they require a close manual inspection of the data, investigations into the relationships between different data sources, often even the analysis of the process that generated the data. New insights promote new ideas for feature generation or alter the subset of selected data, in which case the data preparation and modeling phases are carried out multiple times.
What kind of model architecture suits the problem the best?
What is the best method to get the model?

The obtained results are analyzed in the **evaluation** phase from the perspective of the problem owner. At this point, the project may stop due to unsatisfactory results, the objectives may be revised in order to succeed under a slightly different setting, or the found and optimized model may be deployed.
How is the model in terms of project requirement?
What have we learned from the project?

After **deployment**, which ranges from writing a report to the creation of a software system that applies the model automatically to aid or make decisions, the project is not necessarily finished. We better verify from time to time that these assumption still hold to prevent decision-making on outdated information.
How is the model best deployed?
How do we know that the model is still valid?

Problem Categories

classification
Predict the outcome of an experiment with a finite number of possible results
(like *yes*/*no* or *unacceptable*/*acceptable*/*good*/*very good*).
Typical questions: *Is this customer credit-worthy?*

regression
Regression is, just like classification, also a prediction task, but this time the value
of interest is numerical in nature.
Typical questions: *How will the EUR/USD exchange rate develop?*

clustering, **segmentation**
Summarize the data to get a better overview by forming groups of similar cases
(called clusters or segments). We may also obtain some insight into the structure of the whole data set. Cases that do not belong to any group may be considered as abnormal or outliers.
Typical questions: *Do my customers divide into different groups?*

association analysis
Find any correlations or associations to better understand or describe the interdependencies of all the attributes. The focus is on *relationships* between all attributes rather than focusing on a single target variable or the cases (full record).
Typical questions: *How do the various qualities influence each other?*

deviation analysis
Knowing already the major trends or structures, find any exceptional subgroup that behaves differently with respect to some target attribute.
Typical questions: *Which properties do those customers share who do not follow the crowd?*

The most frequent categories are *classification* and *regression*, because decision making always becomes much easier if reliable predictions of the near future are available. When a completely new area or domain is explored, cluster analysis and association analysis may help to identify relationships among attributes or records. Once the major relationships are understood (e.g., by a domain expert), a deviation analysis can help to focus on *exceptional situations* that deviate from regularity.

CH2

This chapter demonstrates some typical pitfalls one encounters when analyzing real-world data.

CH 3

In this initial phase of the data analysis project, we have to map a problem onto one or many data analysis tasks. In a nutshell, we conjecture that the nature of the problem at hand can be adequately captured by some data sets (that still have to be identified or constructed), that appropriate modeling techniques can successfully be applied to learn the relationships in the data, and finally that the gained insights or models can be transferred back to the real case and applied successfully.

For instance, if a regression problem has to be solved, the analyst may decide that a certain method seems to be a promising choice for the modeling phase. From the characteristics of this technique he knows that all input data have to be transformed into numerical data, which has to be carried out beforehand (data preparation phase).

To prevent us from carrying out an expensive project having almost no prospect of success, we have to carefully track all assumptions and verify them as soon as possible. Typical requirements and assumptions include:

• requirements and constraints
– *model requirements*,
e.g., model has to be explanatory (because decisions must be justified clearly)
– *ethical, political, legal issues*,
e.g., variables such as gender, age, race must not be used
– *technical constraints*,
e.g., applying the technical solution must not take more than n seconds

• assumptions
– *representativeness*:
If conclusions about a specific target group are to be derived, a sufficiently large number of cases from this group must be contained in the database, and the sample in the database must be representative for the whole population.
– *informativeness*:
To cover all aspects by the model, most of the influencing factors (identified in the cognitive map) should be represented by attributes in the database.
– *good data quality*:
The relevant data must be of good quality (correct, complete, up-to-date) and unambiguous thanks to the available documentation.

– *presence of external factors*:
We may assume that the external world does not change constantly—for instance, in a marketing project we may assume that the competitors do not change their current strategy or product portfolio at all.

CH4

The main goal of data understanding is to gain general insights about the data that will potentially be helpful for the further steps in the data analysis process

Drastic round-off errors or truncations can lead to problems in later steps of the analysis. Suppose, for instance, that a cluster analysis is to be carried out later on and that there is one numerical attribute, say X, that is truncated to only one digit right after the decimal point, while all other numerical attributes were measured and stored with a higher precision. When comparing different records, such truncation for the attribute X influences their perceived similarity and might be a dominating factor for the further analysis only for this reason.

Sturges' rule is still very often used as a default in various statistics software packages; it is tailored to data from normal distributions and data sets of moderate size.

A scatter plot with no outliers does not mean that there are no outliers in the data set. It only means that there are no outliers with respect to the combination of the attributes displayed in the scatter plot.

MDS differs from PCA in various aspects. MDS is based on the idea of preserving the distances among the original data objects, whereas PCA focuses on the variance. All these dimension-reduction methods generate scatter plots with abstract coordinate axes that do not correspond to attributes of the original data. Both can show you clusters too.

Pearson's correlation coefficient measures linear correlation. Even if there is a functional dependency between two attributes, but the function is nonlinear but monotone, Pearson's correlation coefficient will not be -1 or 1. It can even be far away from these values, depending on how much the function describing the functional relationship deviates from a line.

Kendall's tau rank correlation coefficient or simply **Kendall's tau** is not, like Spearman's rho, based on ranks, but rather on the comparison of the orders of pairs of values.

Rank correlation coefficients like Spearman's rho and Kendall's tau depend only on the order (ranks) of the values and are therefore more robust against extreme outliers than Pearson's correlation coefficient.

For categorical attributes, these correlation coefficients are not applicable. Instead, one can carry out independence tests like the χ^2 test for independence

Outliers
Outliers can be a hint to data quality problems as they will be discussed in the next section. Outliers can correspond to erroneous data coming from wrong measurements or typing mistakes when data are entered manually.

However, outliers can also be correct data that differ from the rest of the data just by chance or for other reasons like special exceptional situations. Even if outliers are correct data, it might be worthwhile to exclude them from the data set for further analysis and to consider them separately.

However, removing these "outliers" from the data set would actually make it impossible to achieve our original goal to derive a classifier from the data set that can identify the parts with failures.

For numerical attributes, outlier detection is more difficult. We have already classified certain data points in a boxplot as outliers. However, the definition of outliers in a boxplot does not take the number of data into account, so that for larger data sets, boxplots will usually contain points marked as outliers. As mentioned before, for a normal distribution, we can expect roughly 0.7% points to be marked as outliers in a boxplot. For asymmetric distributions, boxplots tend to contain more outliers.

Statistical tests for outliers are usually based on assumptions about the underlying distribution, although we might not know from which distribution the data are sampled. The standard assumption for outlier tests for continuous attributes is that the underlying distribution is a normal distribution. Try Grubb's test.

Outlier detection in multidimensional data is usually not based on specific assumptions on the distribution of the data and is not carried out in the sense of statistical tests. Visualization techniques provide a simple method for outlier detection in multidimensional data. Scatter plots can be used when only two attributes are considered.

Instead of using projections to two attributes, one can also use dimension-reduction methods like PCA or multidimensional scaling in order to identify outliers in the corresponding plots. Finding outliers in multidimensional data based on clustering the data and defining those data objects as outliers that cannot be assigned reasonably to any cluster.

Another way to approach the problem of distinguishing between missing completely at random and missing at random is the following procedure:
1. Turn the considered attribute X into a binary attribute, replacing all measured values by the values *yes* and all missing values by the value *no*.
2. Build a classifier with now binary attribute X as the target attribute and use all other attributes for the prediction of the class values *yes* and *no*.
3. Determine the misclassification rate. The misclassification rate is the proportion of data objects that are not assigned to the correct class by the classifier.

In the case of missing values of the type observed at random, the other attributes should not provide any information, whether X has a missing value or not. Therefore, the misclassification rate of the classifier should not differ significantly from pure guessing, i.e., if there are 10% missing values for the attribute X, the misclassification rate of the classifier should not be much smaller than 10%. If, however, the misclassification rate of the classifier is significantly better than pure guessing, this is an indicator that there is a correlation between missing values for X and the values of the other attributes. Therefore, the missing values for X might not be of the type observed at random but of the type missing at random or, even worse, nonignorable. Note that it is in general not possible to distinguish the case nonignorable from the other two cases based on the data only.

SUMMARY:
One important part of data understanding is to get an idea of the data quality.
Outliers are another problem, and there are various methods to support the identification of outliers. There are especially visualization techniques like boxplots, histograms, scatter plots, projections based on PCA and MDS that can help to find outliers but are also useful for other purposes. First, a distance matrix is needed for MDS. Identical objects leading to zero distances are not admitted. Therefore, if there are identical objects in a data set, all copies of the same object except one must be removed. It is necessary to exclude the categorical attribute from PCA.
Missing values are another concern of data quality. When there are explicit missing values, i.e., entries that are directly marked as missing, then one should still try to find out of which type—OAR, MAR, or nonignorable—they are.
Apart from these data quality issues, data understanding should also help to discover new or confirm expected dependencies or correlations between attributes. Scatterplots.
Specific application dependent assumptions—for instance, the assumption that a specific attribute follows a normal distribution—should also be checked during data understanding.
Representativeness of the data cannot always be checked just based on the data, but we have to compare the statistics with our expectations. If we suspect that there is a change in a numerical attribute over time, we can compare histograms or boxplots for different time periods. We can do the same with bar charts for categorical attributes.

Check the distributions for each attribute whether there are unusual or unexpected properties like outliers. Are the domains or ranges correct? Do the medians of numerical attributes look correct?
Check correlations or dependencies between pairs of attributes with scatter plots which should be density-based for larger data sets. For small numbers of attributes, inspect scatter plots for all pairs of attributes. For higher numbers of attributes, do not generate scatter plots for all pairs, but only for those ones where independence or a specific dependency is expected. Generate in addition scatter plots for some randomly chosen pairs.

Teacher: Leave categorical and binary variables in for all visualizations. KNN is a good baseline method.

CH5
When there are more than two classes, it is not possible to draw a ROC curve as described above. One can only draw ROC curves with respect to one class against all others. The **confusion matrix** is another way to describe the classification errors. A confusion matrix is a table where the rows represent the true classes and the columns the predicted classes. Each entry specifies how many objects from a given class are classified into the class of the

corresponding column. An ideal classifier with no misclassifications would have only entries different from zero in the diagonal.

A finite sample, especially when its size is quite small, will seldom exactly reflect the true distribution of the probability distribution generating the data. According to the laws of large numbers, the sample distribution converges with probability one to the true distribution when the sample size approaches infinity. However, a finite sample can deviate significantly from the true distribution, although the probability for such a deviation might be small.

When the set of considered models is too simple for the structure inherent in the data, no model will yield a small error. Such an error is also called **model error**.

Usually, the training set is chosen larger than the test data set, for instance, $2/3$ of the data are used for training, and $1/3$ for testing.

One way to split the data into a training and a test set is a random assignment of the data objects to these two sets. However, by chance it can happen that the distributions may differ significantly. When a classification problem is considered, it is usually recommended to draw stratified samples for the training and the test set. **Stratification** means that the random assignments of the data to the test and the training set are carried out per class and not simply for the whole data set. In this way, it is ensured that the relative frequency in the original data set, the training, and the test set are the same.

When predictions are made for future data for which given data is not representative, extrapolation is carried out with a higher risk of wrong predictions. In the case of high-dimensional data, it cannot be avoided to have scarce or no data in certain regions of the space of possible values

Cross-validation does not rely on only one estimation of the model error, but rather on a number of estimations. For k-fold cross-validation, the data set is partitioned into k subsets of approximately equal size. Then the first of the k subsets is used as a test set, and the other $(k-1)$ sets are used as training data for the model. In this way, we get the first estimation for the model error. Then this procedure is repeated by using each of the other k subsets as test data and the remaining $(k-1)$ subsets as training data. Altogether, we obtain k estimations for the model error. The average of these values is taken as the estimation for the model error. Typically, $k = 10$ is chosen.
Small data sets might not contain enough examples for training when 10% are left out for testing. In this case, the **leave-one-out method**, also known as the **jackknife method**, can be applied which is simply n-fold cross-validation for a data set with n data objects, so that each time only one data object is used for evaluating the model error.

Bootstrapping is a resampling technique from statistics that does not directly evaluate the model error but aims at estimating the variance of the estimated model parameters. Therefore, bootstrapping is suitable for models with real-valued parameters. Like in cross-validation, the model is computed not only once but multiple times. For this purpose, k bootstrap samples, each of size n, are drawn randomly *with replacement* from the original data set with n records. The model is fitted to each of these bootstrap samples, so that we obtain k estimates for the model parameters. Based on these k estimates, the empirical standard deviation can be computed for each parameter to provide information how reliable the estimation of the parameter is.

In the example: The standard deviation for the slope is much lower than for the intercept, so that the estimation for the slope is more reliable.

A penalty term for more complex models can be incorporated into to the pure measure for model fit as a regularization technique for the avoidance of overfitting.

CH6 Data Preparation

Before we start modeling, we have to prepare our data set appropriately, that is, we are going to modify our dataset so that the modeling techniques are best supported but least biased.

The data preparation phase can be subdivided into at least four steps. The first step is data selection. If multiple datasets are available, based on the results of the data understanding phase, we may select a subset of them as a compromise between accessibility and data quality. Within a selected dataset, we may concentrate on a subset of records (data rows) and attributes (data columns). We support the subsequent modeling steps best if we remove all useless information, such as irrelevant or redundant data. The second step involves the correction of individual fields, which are conjectured to be noisy, apparently wrong or missing. If something is known, new attributes may be constructed as hints for the

modeling techniques, which then do not have to *rediscover* the usefulness of such transformations themselves. For some modeling techniques, it may even be necessary to construct new features from existing data to get them running. Finally, most available implementations assume that the data is given in a single table, so if data from multiple tables have to be analyzed jointly, some integration work has to be done.

The goal of **feature selection** is to select an *optimal* subset of the full set of available attributes A of size n. The more attributes there are, the wider is the range of possible subsets: the number of subsets increases exponentially in the size of A. Feature selection typically implies two tasks: (1) the selection of some evaluation function that enables us to compare two subsets to decide which will perform better and (2) a strategy (often heuristic in nature) to select (some of) the possible feature subsets that will be compared against each other via this measure.

PCA was used for generating scatter plots from higher-dimensional data and to get an idea of how many intrinsic dimensions there are in the data by looking at how much of the variance can be preserved by a projection to a lower dimension. Therefore, PCA can also be used as a dimension reduction method for data preparation.

PCA belongs to a more general class of techniques called **factor analysis**. Factor analysis aims at explaining observed attributes as a linear combination of unobserved attributes.
Independent component analysis (**ICA**) is another method to identify such unobserved variables that can explain the observed attributes. In contrast to factor analysis, ICA drops the restriction to linear combinations of the unobserved variables.

The reason why we nevertheless may want to change the missing fields is that the implementations of some methods simply cannot deal with empty fields. Then the imputation of estimated values is a simple mean to get the program run and deliver some result, which is often, although affected by estimation errors, better than having no result at all.
Another option would be to remove records with missing values completely, so that only complete records remain.
By replacing all missing values of one variable by the mean of the measured values, the mean of the modified set will remain unchanged—however, its variance will decrease. This replacement procedure affects the variance of the attributes to different degrees (depending on the portion of missing values).
Replacing all missing values with the very same constant number always reduces the variability of the dataset, which often influences derived values such as correlation coefficients or fitting errors. This is less pronounced with substitutions that come from a predictor, but the problem remains.

A very simple approach is to replace the missing values by some new value, say `MISSING`. This is only possible for nominal attributes, as any kind of distance to or computation with this value is undefined and meaningless. If the fact that the value is missing carries important information about the value itself (nonignorable missing values), the explicit introduction of a new value may actually be advisable, because it may express an intention that is not recoverable from the other attributes. If we suppose that the absence of the value and the intention correlate, we may luckily capture this situation by a new constant, but the problem is that we cannot assure this from the data itself. If the values are missing completely at random, there is no need to introduce a new constant.

A better approach is to introduce a new (binary) variable that simply indicates that the field was missing in the original dataset (and then impute a value). In those cases where neither the measured nor the estimated values really help, but the fact that the data was missing represents the most important bit of information, this attribute preserves all chances of discovering this relationship: the newly introduced attribute will turn out to be informative and useful during modeling.

Another typical assumption is that some variables obey a certain probability distribution, which is not necessarily the case in practice. If the assumption of a Gaussian distribution is not met, we may transform the variable to better suit this requirement. This should be a last resort, because the interpretability of the transformed variable suffers. Typical transformations include the application of the square root, logarithm, or inverse when there is moderate, substantial, or extreme skewness.

New attributes are often constructed by applying arithmetic or logic operators to existing attributes and aggregations.

CH7

Association Rules Rather than grouping or organizing the cars, we may be interested in interdependencies among the individual variables. Several automobile manufacturers, for instance, offer certain packages that contain a number of additional features for a special price. If the car equipment is listed completely, it is possible to recover those features that frequently occur together. The existence of some features increases the probability of others, either because both features are offered in a package or simply because people frequently select a certain set of features in combination.

One technique to find associations of this kind are **association rules.** For every feature that can be predicted confidently from the occurrence of some other features, we obtain a rule that describes this relationship.

Based on the type of data to be mined, frequent pattern mining is divided into (1) frequent item set mining and association rule induction, (2) frequent sequence mining, and (3) frequent (sub)graph mining with the special subarea of frequent (sub)tree mining.

In deviation analysis a pattern describes a divergence of some target measure in a subgroup of the full dataset.

CH8

Decision trees Decision trees aim to find a hierarchical structure to explain how different areas in the input space correspond to different outcomes. The hierarchical way to partition this space is particularly useful for applications where we drown in a series of attributes of unknown importance. Decision trees are often thrown first at classification problems because they tend to be insensitive to normalization issues and tolerant toward many correlated or noisy attributes. In addition, the structure of the tree also allows for data clean-up. Quite often, the first attempt at generating a decision tree reveals unexpected dependencies in the data which would otherwise be hidden in a more complex model.

Bayes classifiers Bayes classifiers form a solid baseline for achievable classification accuracy—any other model should at least perform as well as the naive Bayes classifier. In addition they allow quick inspection of possible correlations between any given input attribute and the target value. Before trying to apply more complex models, a quick look at a Bayes classifier can be helpful to get a feeling for realistic accuracy expectations and simple dependencies in the data.

Regression models Regression models are the counterpart for numerical approximation problems. Instead of finding a classifier and minimizing the classification error, regression models try to minimize the approximation error, that is, some measure for the average deviation between the expected and predicted numerical output value. Again, many more complex models exist, but the regression coefficients allow easy access to the internals of the model. Each coefficient shows the dependency between any input attribute and the target value.

Rule models Rule models are the most intuitive, interpretable representation. Generally, one would only apply rule extraction algorithms only to data set with a reasonably well-understood structure. Many of the algorithms tend to be rather sensitive toward useless or highly correlated attributes and excessive noise in the data.

We start this chapter with likely the most prominent and heavily used methods for interpretable model extraction from large data sets: Decision Trees. The most prominent training algorithm ID3 which, with various extensions, ends up forming c4.5.

Decision Trees come in two flavors, classification and regression trees. Decision trees are one of the most well-known and prominently used examples of more sophisticated data analysis methods. However, one often forgets that they are notoriously unstable. **Stability** means that small changes to the training data, such as removing just one example, can result in drastic changes in the resulting tree.

Despite this pejorative name, naive Bayes classifiers perform very well in practice and are highly valued in domains in which large numbers of descriptive attributes have to be taken into account (for example, chemical compound and text document classification).

More robust results can usually be obtained by minimizing the **sum of absolute deviations** (least absolute deviations, LAD). However, this approach has the disadvantage of not being analytically solvable (like least squares) and thus has to be addressed with iterative methods right from the start.

CH9

Especially artificial neural networks and support vector machines, are known to outperform other methods w.r.t. accuracy in many tasks. However, due to the abstract mathematical structure of the prediction procedure, which is usually difficult to map to the application domain, the models they yield are basically “black boxes” and almost impossible to interpret in terms of the application domain. Hence they should be considered only if a comprehensible model that can easily be checked for plausibility is not required, and high accuracy is the main concern.

Nearest-Neighbor Predictors A very natural approach to predict a class or a numeric target value is to look for historic cases that are similar to the new case at hand and to simply transfer the class or target value of these historic cases. The nearest-neighbor algorithm is one of the simplest and most natural classification and numeric prediction methods: it derives the class labels or the (numeric) target values of new input objects from the most similar training examples, where similarity is measured by distance in the feature space. The prediction is computed by a majority vote of the nearest neighbors or by averaging their (numeric) target values. The number k of neighbors to be taken into account is a parameter of the algorithm, the best choice of which depends on the data and the prediction task. A common method to automatically determine an appropriate value for the number k of neighbors is **cross-validation.** Knn prediction can take a long time so better approaches rely on data structures like a kd-tree.

Artificial Neural Networks Among methods that try to endow machines with learning ability; artificial neural networks are among the oldest and most intensely studied approaches. They take their inspiration from biological neural networks and try to mimic the processes that make animals and human beings able to learn and adapt to new situations. However, the used model of biological processes is very coarse, and several improvements to the basic approach have even abandoned the biological analogy. The most common form of artificial neural networks, *multilayer perceptrons*, can be described as a staged or hierarchical logistic regression, which is trained with a gradient descent scheme, because an analytical solution is no longer possible due to the staged/hierarchical structure.

In principle, a multilayer perceptron may have any number of hidden layers, but it is most common to use only a single hidden layer, based on certain theoretical results about the capabilities of such neural networks. The number of input and output neurons is fixed by the data analysis task. The only choices left for a multilayer perceptron are the number of hidden layers and the number of neurons in these layers.

A simple rule of thumb, which often leads to acceptable training results, is to use 1/2(#inputs+#outputs) hidden neurons, where #inputs and #outputs are the numbers of input and output attributes, respectively. Or other methods (cross-validation probably).

However, even though a wrong number of hidden neurons, especially if it is chosen too small, can lead to bad results, one has to concede that other factors, especially the choice and scaling of the input attributes, are much more important for the success of neural network model building.

Support Vector Machines Though closely related to specific types of artificial neural networks, support vector machines approach the problem of finding a predictor in a way that is more strongly based on statistical considerations, especially on risk minimization approaches. The basic principle underlying support vector machines can be described as follows: when it comes to classification (with two classes), linear separation is enough; when it comes to numeric prediction, linear regression is enough—one only has to properly map the given data to a different feature space. However, this mapping is never explicitly computed, but only a certain property of it (to be precise, the scalar product in the image space) is described implicitly by a so-called *kernel function*, which intuitively can be seen as a similarity measure for the data points. As a consequence, the choice of the kernel function is crucial for constructing a support vector machine, and therefore a large number of specialized kernels for different application domains have been developed.

Ensemble Methods If we are sick and seek help from a physician but have some doubts about the diagnosis we are given, it is standard practice to seek a second or even a third opinion. Ensemble methods follow the same principle: for a difficult prediction task, do not rely on a single classifier or numeric predictor but generate several different predictors and aggregate their individual predictions. Provided that the predictors are sufficiently different, this can often lead to considerably improved prediction accuracy.

CH10

SLIDES

1

Unsupervised learning: Tools for exploratory data analysis. Can we discover structure in the data? Pre-processing information and visualizations. Finding patterns.

Cluster Analysis: Customer segmentation. Summarize data to get a better overview by forming groups of similar cases.

Association Analysis: Find any associations or correlations to better understand and describe the variables.

Deviation Analysis: Outlier detection. Knowing already the major trends and structures, find any exceptional groups that behave differently.

Visualizations

3Data Understanding

Gain general insights, check assumptions made during project understanding, suitability of data.

Ordinal attribute: bs.c ms, phd linear ordering of domain

Biasedness and non-representativeness of data are problems. Is data too old?

Sturges' rule is suitable for data from normal distributions and from data sets with moderate size $n>30$

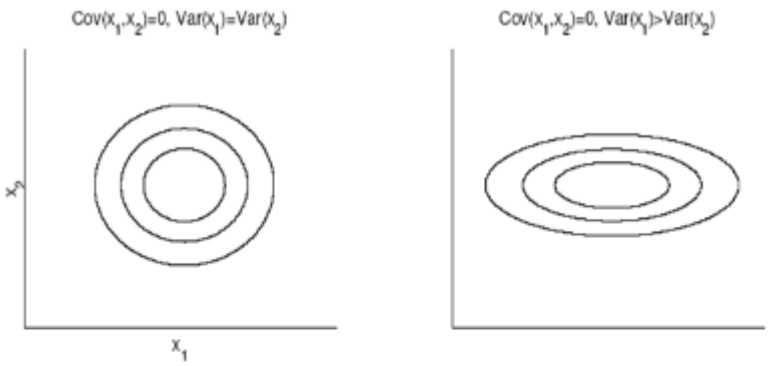
FD uses interquartile range.

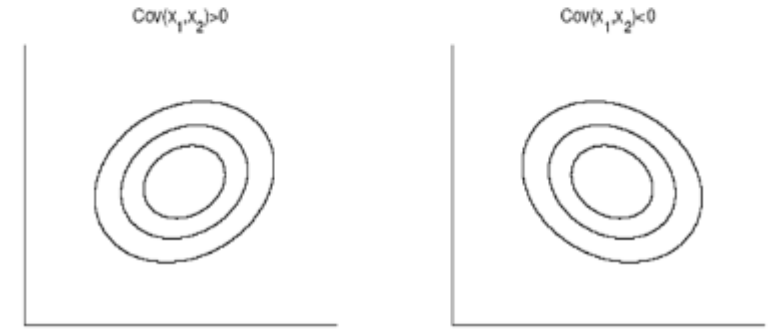
Boxplot is a visual inspection of the data. It uses interquartile range in it's construction, specifically the 75% quantile and 25% quantile. The median is plotted in the box. The whiskers are 1.5 interquartile range. If a data point does not fit into this area then it is an outlier. Not good for asymmetric distributions.

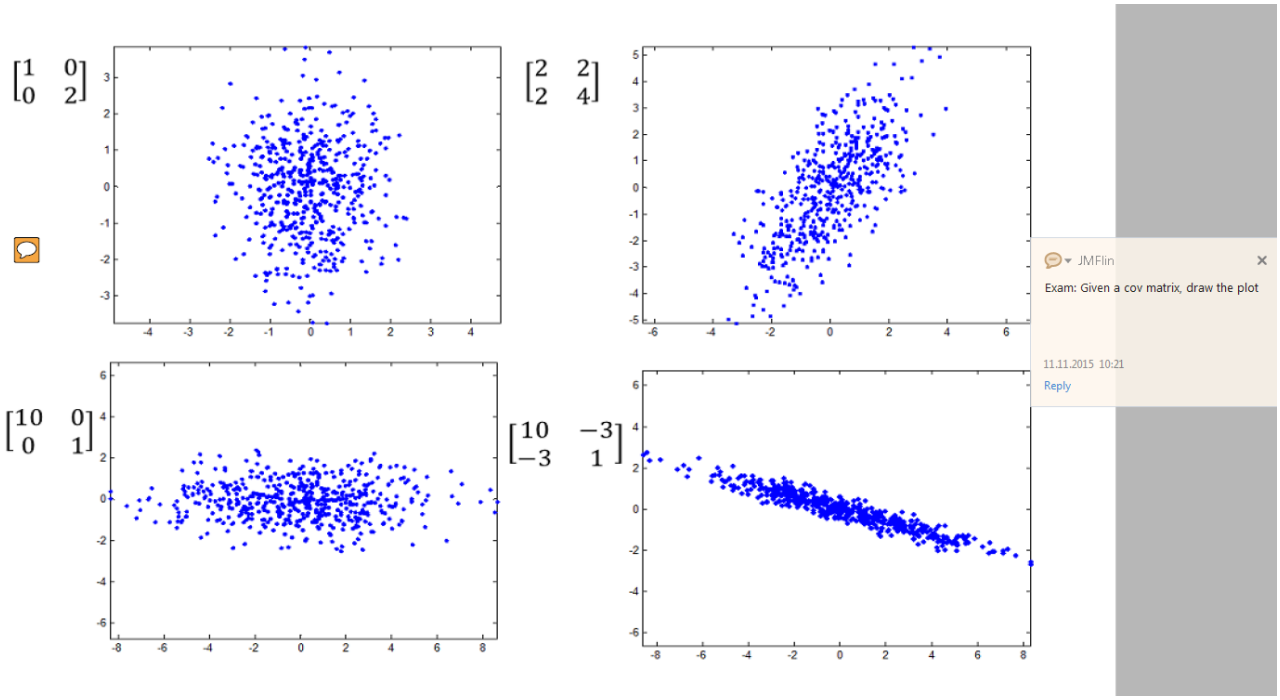
Large data scatterplots: use density plots with semitransparent points.

Variance: How much a random variable varies around the expected value.

Covariance: measure of strength of the linear relationship between two variables. It becomes more positive for each pair of values which differ from their mean in the same direction. If they are independent then covariance is zero.







PCA: uses variance to preserve structure. From high dimensional space to low dimensional space. The aim is to find a projection in the form of a linear mapping. It is a linear method for dimensionality reduction. The principal components are the normalized eigenvectors of the covariance matrix.
Standardization: all variables contribute equally to variance.
MDS: Does not construct a mapping from high dimension to low dimension. Aims to preserve the distance between data points when going from high to low dim. Usually Euclidean. Non-linear optimization problem -> gradient descent
Explicit representation not mapping like PCA.

Pearson: Linear corr
Rank: considers the ordering of values. Measures monotonous correlations (does not have to be linear).

Outliers in categorical data are those that have extremely less frequency.

Outlier detection: scatterplots, pca, mds, clustering.

4 Data preparation
Feature extraction: refers to construct new features from the given attributes. Ex: we are interested in finding the best workers in the company. Tasks, number of hours, etc can be used as a new feature called efficiency by diving them or multiplying etc.

Dimensionality reduction: PCA and MDS. (standardize and center)
Feature selection: choose a subset of the features. Remove irrelevant and redundant features. Performance measure is needed like BIC or AIC.
Forward/backward
Rare events: stock market crash etc. Artificially increase these rare events in the data set by adding copies of them or only choose a subset of the other data.
Missings: Missing at randoms can be deleted, Replaced by mean or median, can be replaced by MISSING.
Always turn categoricals into bunary if you need a numericl format. Don't turn them into 1..6 if they are not hierarchical.

5 Finding Patterns
Finding patterns in an exploratory data analysis task which summarizes the data.

Cluster analysis: finding groups of similar observations. Normalization should be done.
Association rules: Finding associations between attributes or typical combinations of values.
Deviation analysis: Finding groups that deviate from the rest of the observations.

Hierarchical Clustering: Step by step bottom up strategy, where each data object is considered it's own cluster and then joining clusters together that are close. Uses a dissimilarity measure in order to decide which data objects should belong to the same cluster -> centroid on mean, single link on lähin ja complete link on kaukaisin.
The cluster merging process causes a binary tree form and then draw a connection between clusters to form a dendogram.

Choosing the clusters: Specify a minimum distance between clusters and stop merging clusters if the closest two clusters are farther apart than this distance.

K-Means Clustering: Choose a number of K clusters. Initialize the cluster centers randomly. Assign each data point to the nearest cluster center. Compute the new center as a mean of the data points. Repeat.
This is sensitive to the initial position and the number of K.

Density based clustering: high density of data = cluster.

Gaussian Mixture Models: Data is generated by some normal distributions, find the parameters and how much each distribution contributes to the data.

Association rule mining: Market basket analysis. Aims at finding patterns in shopping behavior aka find sets of products that are frequently bought together.
Assessing quality of association rules: Support of an item set -> Proportion of transactions that contain the item set.
Confidence: probability of items being bought together.

Item set tree pruning: Support based -> No superset of an infrequent item set can be frequent. Prune infrequent item sets.
Free item set: any frequent item set that has a support greater than minimum.
Closed item set: No superset has the same support
Maximal item set: no superset is frequent.

SUMMARY:
Find the frequent item sets (support)
Form the relevant association rules (confidence)
Form all possible association rules from the frequent item sets. Filter "interesting" association rules.
Association rules with confidence of close to 100% could be business rules!
This process can be seen as a discovery of subgroups of the dataset that share common properties.

Deviation analysis: The goal is to find subgroups that are statistically most interesting. They deviate from the whole population with respect to the property of interest.
The ingredients are a target measure and a verification test for irrelevant patterns, a quality measure to rank subgroups, a method that enumerates candidate subgroups. Significance of deviation is done by statistical tests.

Mahalnobis distance: Nonisotropic distance measure (distance tries to capture variations in variable values), which is scale-invariant.

6 Modeling

Supervised learning: data, features, labels

Modeling steps:
- Model class: linear, quadratic, tree, NNs etc.
- Score function: evaluates quality of different models. Squared error etc. How complex is the model?
- Algo: find good model based on score function.
- Validation: cross-validation.

Fitting criterion: How well does the model fit the data.
- Model complexity: Usually, simpler models are preferred because they are easier to understand, easier to fit and avoid overfitting.

Once you have done your best with simple methods, you can try more advanced ones.

OLS: Ordinary least squares: Method for estimating parameters in a linear regression model. Minimizes least squared error.

Maximum Likelihood: Model fitting approach. It determines the optimal value for a parameter given the data. The aim is to maximize the likelihood function. Ex: Logistic regression

Bayes MAP estimator: Maximum a posterior. It is a mode of the posterior distribution. It is closely related to ML and can be seen as a regularized version of ML.

Hillclimbing: a greedy strategy. Start with a random soloution and generate a new solution in the neighborhood. If the new solution is better then generate another one etc.

7 K-NN

Z-score standardization. Instance-based learning.
Similar instances should have the same labels. This similarity is based on distance.
So memorize your training data and then for new instances predict the majority class of its k-nearest neighbors.
The data is the model.
In classification it is majority vote and and in regression the average of the nearest neighbors.
Disadvantages: all neighbors have an equal effect, maybe the closer ones should have more of an effect. Can be slow for large data sets, poor results on high dimensional data.
Advantages: easy to implement. Can model very complex non-linear data. Can easily be adapted to different types of data by chosing distance measure.
Distance metric-> usually Euclidean and k-neighbors need to be decided (CV). And regression vs classification. Black-box.

Remember feature selection and maybe dimensionality reduction with normalization

8 Performance measures
Misclassification rate, squared error.
AUC (area under roc) for binary classification.

Misclassification rate doesn't necessarily tell anything about the accuracy. With unbalanced classes a low misclassification rate can be easily achieved.

Cost matrix. When an intact cup is classified as broken, the cup has to be remade. The error made costs money.
Confusion matrix -> accuracy, recall, precision, f-score
Squared error: sensitive to outliers
ROC: receiver operating characteristic curve. Y is true positive rate, x is false positive rate. Diagonal is randomly guessed. 0-1 scale so 1 is perfect model

The aim of learning is to approximate the target function as closely as possible.

Two things need to be balanced in learning: fit to data and model complexity. Overfitting is modeling noise while underfitting is not modeling the relevant aspects.
Controlling for complexity: polynomial degrees, number of k, regularization parameter etc.

CV: split data randomly into training and testing sets, construct model on training set, compute performance on test set, repeat 10 times, and take average. Small pessimistic bias, the model can be slightly less accurate than the model trained on the entire data.

Minimum description length: Model performance criterian.

9 Model Validation
Models should be tested on independent data sets called test sets.

Validation means measuring the predictor's behavior on data points other than those in the training set. Needed for parameter selection and comparing different algorithms and how the final model will generalize.

For classification, stratifying is often used to so that the training and test data have the distribution of classes as the original data.
Train, test then train on whole data set.
LOOCV: train model on n-1 instances and test it on the left over. Iterate over this. This is almost unbiased. Take the model parameters with the lowest error! Not good for large data sets.

K-fold CV: Instead of leaving out one instance of data at a time, leave out several. Data is randomly divided into K different parts, on each round the model is trained on K-1 folds and tested on the one left out. You can repeat this with different fold divisions and take the average too.

We can also split 50% 25% 25% where we train models, test all models, and then best model is tested again.

Nested CV: CV within CV. Outer loop is for evaluation and inner loop for model selection.

10 Regression
Lin reg: a line that fits the data the best. Determined by minimizing mean squared error
OLS is sensitive to outliers and prone to overfitting in high dimensional data.
Ridge penalized model complexity, more robust for higher dimensional data. Compact linear model.

Logistic regression: predictions scaled between 0-1 and can be interpreted as probabilities.

11 Bayes Classification
Classification method. A probabilistic model with very simple independence assumptions.
Pros: very simple, scales well to large data sets, fast, can handle categorical variables naturally, nothing to tune.
Cons: might not work well on complex problems, not that great for handling numerical features.
Typical application: text classification.
Given the features I observed, what are the probabilities that this instance belongs to different classes.
Email probability of junk vs non-junk.

Bayes classifier: The computations are carried out by the assumption that the attributes are independent given the classes. Then we assign the attribute to the class with the highest likelihood.

Example
Remember that a full bayes classifier would not be able to classify it!

Laplace error is for correcting the fact that if a single likelihood is zero then the total likelihood is also zero. Laplace error adds 1 to each occurrence. Kind of like add one smoothing in NLP!
Add the amount of features to the denominator but not for sex.

Missing values in bayes: not counted in the frequencies and only probabilities of those attributes are multiplied.

Full bayes: each class has a bell-shaped probability density.
Naïve Bayes: Covariance matrices are diagonal matrices.

In a picture full bayes is more diagonal!

TENTTI

Entropy: expected value of information contained in the data

mahalnobis distance is a normalized something covariance matrix something
Igenvectors in PCA
OLS and Maximum Likelihood MAP
Association rule mining – deviation analysis etc