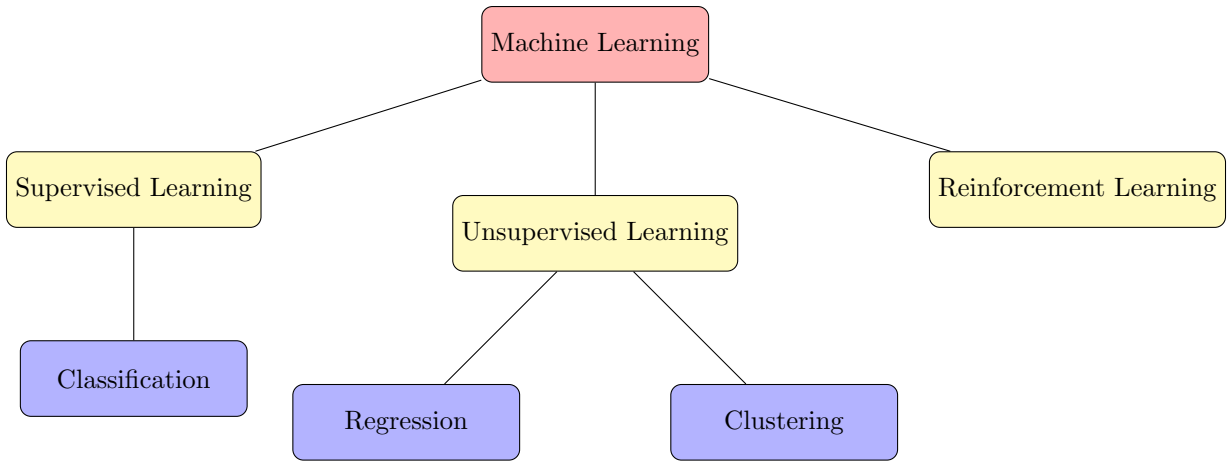
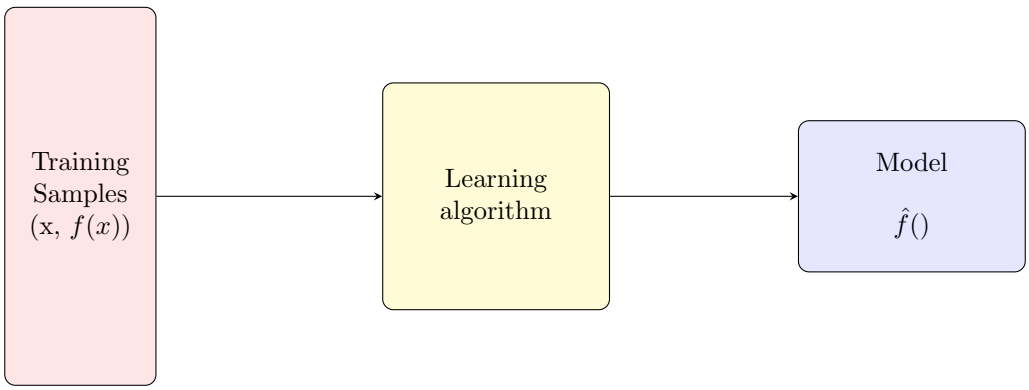
Professorship of Business Analytics & Intelligent Systems
TUM School of Management
Technical University of Munich

Course Contents

1. Naive Bayes & Bayesian Networks
2. Decision Trees
3. Clustering
4. Regression
5. Neural Networks
6. Data Preparation, Generalization & Evaluation





Focus of this course: **Predictive Analysis**

using statistical models to forecast future outcomes based on past data

Datasets: Features and Target Variables

Dataset

Let $D = (x_i, y_i)_{i=1}^{N}$ be a dataset
- x_i is K-dimensional feature vector (independent variables)
- y_i is the respective target variable (dependent variable)

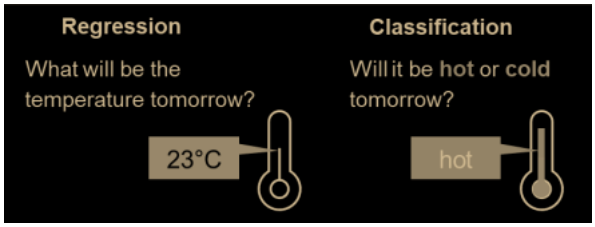
Features

1. Categorical Features (mutually exclusive, no natural ordering, equality is the only meaningful comparisonėx: purpose of loan)
2. Ordinal Features (encoded as numbers to preserve their ordering, meaningful to compare values apart with $>$ and $<$, difference shouldn't be measured or used to perform any mathematical operationsėx: savings and employment)
3. Numerical Features (meaningful to perform mathematical operations, normalizationėx: loan amount)

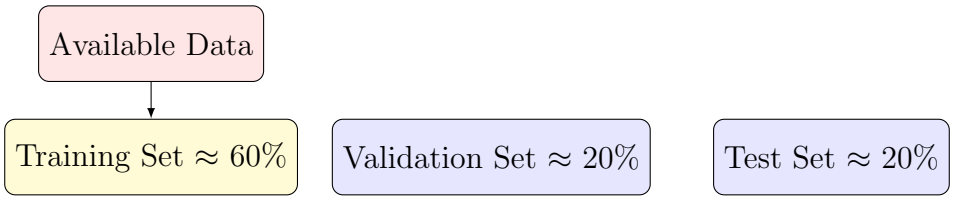
Normalization

rescaling numerical data to ensure that all features have similar scales and prevent one feature from dominating during training. min-max normalization on $[0, 1]$ based on $x' = \dfrac{x - x_{\min}}{x_{\max} - x_{\min}}$

Regression vs Classification



(Continuous) (Coarse Discrete)



Cross Validation

K-fold cross validation, multiple training and test sets
data is divided into K equal subsets, model is trained and tested K times (each time with a different subset used as test set)

1st iteration: (K-1) training fields, the last subset is the test field [result R1]
Kth iteration: 1st subset is the test field, (K-1) training fields [result R2]
$R = \frac{1}{k} \sum_{i=1}^{k} R_i$ is is the robust estimate of model performance

Accuracy and un-balanced classes

$$\text{Precision} = \frac{TP}{TP + FP} \tag{1}$$

actually positive

$$\text{Recall} = \frac{TP}{TP + FN} \tag{2}$$

positive correctly predicted

$$\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3}$$

Probabilistic Models

using probability distributions to predict outcomes and quantify uncertainty which helps us in predicting and estimation all using probability theory

1. Spam Detection
2. Sentiment Analysis
3. Credit Risk Assessment

Bayes Theorem

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)} \tag{4}$$

using prior beliefs to find out new beliefs

1 Naive Bayes Classifier and Bayesian Networks

discrete-valued features $x \in \{1, \dots K\}^D$, all features are conditionally independent

$$P(x|y=c) = \prod_{j=1}^{D} p(x_j|y=c) \tag{5}$$

classes = $\{\dots\}$
features = $\dots, \dots \dots$
training examples = classes + features
model = Gaussian

Zero-frequency problems

don't allow zero probabilities → **Laplace Smoothing**

$$\text{Laplace's estimate: } P_{LAP}(x) = \frac{num(x)+1}{\sum_X [num(x)+1]} = \frac{num(x)+1}{N+|X|} \tag{6}$$

where N = number of observations and $|X|$ = amount of possible values of n

$$P_{LAP,\varepsilon}(x) = \frac{num(x)+\varepsilon}{N+\varepsilon|X|} \tag{7}$$

extended Laplace's estimate

$$P_{LAP,\varepsilon}(x|y) = \frac{num(x \cap y)+\varepsilon}{y+\varepsilon|X|} \tag{8}$$

estended Laplace's estimate for conditionals

Assumed Indepedence → Feature Modeling

feature vector for filtering out words and ignoring them for the classification, ex: disregarding common words like "the" in spam filters

Missing Values: for some feature x_j

conditional independence: ignore the missing feature and instead compute the likelihood for the observation based on known features.

$$P(x_1, \dots, x_j, \dots, x_D|y) = \sum_{x_j} P(x_1|y) \dots P(x_j|y) \dots P(x_D|y)$$

$$P(x_1, \dots, x_j, \dots, x_D|y) = \sum_{x_j} P(x_j|y) \prod_{i \neq j}^{D} P(x_i|y) = 1 \prod_{i \neq}^{D} P(x_i|j) \tag{9}$$

Drawback of Naive Bayes

time complexity of calculating both conditional proabbilities and the class with $\mathbb{O}(n)$ and $\mathbb{O}(cp)$ where n is the number of instances, c classes and p is the number of features.
independence may not hold for some features and therefore we use **Bayesian Belief Networks**

Bayesian Belief Networks (BBNs)

Bayesian Belief Networks (BBNs, also called Bayesian Nets) model the conditional dependency between some features. Thus, BBNs are less constraining and more powerful than Naïve Bayes models that always assume feature independence. BBNs allow us to combine prior knowledge about variable dependencies with patterns learned from observed training data.

Key Characteristics of Bayesian (Belief) Networks

- **Graphical Representation:** BBNs are represented as *directed acyclic graphs (DAGs)* where:
  - Each **node** represents a feature or a random variable.
  - **Edges** between nodes represent the local conditional dependencies between features.
- **No Directed Cycles:** The structure of BBNs ensures there are no directed cycles in the graph.

Definition of Bayesian Networks

Let $X = \{x_1, x_2, \ldots, x_n\}$ be random features. A Bayesian Network is a directed acyclic graph (DAG) that specifies a joint distribution over X as a product of local conditional distributions. For each node x_i, a local joint conditional distribution exists. Given that we know the set of parents $\text{Pa}(x_i)$ for each feature x_i, the resulting joint probability distribution can be written as:

$$P(x_1, x_2, \ldots, x_n) = \prod_{j=1}^{n} P(x_j \mid \text{Pa}(x_j)).$$

The chain rule of probability allows us to represent a joint distribution as follows:

$$P(x_1, x_2, \ldots, x_V) = P(x_1) \cdot P(x_2 \mid x_1) \cdot \cdots \cdot P(x_V \mid x_1, x_2, \ldots, x_{V-1}),$$

where V is the total number of variables and $\text{Pa}(x_i)$ denotes the set of parents of node x_i. Note that the order of the variables can be interchanged in any way.

Key Components of BBNs

- **Nodes:** Represent random variables or features.
- **Edges:** Represent conditional dependencies between variables.
- **Conditional Probability Tables (CPTs):** Quantify the relationships between connected nodes.

Advantages of Bayesian Belief Networks

- Ability to handle dependencies among features.
- Less constraining than the assumption of conditional independence in Naïve Bayes.
- Combines prior knowledge about variable dependencies with observed training data.

Disadvantages and Challenges

- Learning Bayesian Belief Networks is computationally complex.
- This remains a subject of ongoing research.

Example Graphical Representation:

$$\begin{array}{ccc} 1 \to 2 & 3 & \dots \\ \downarrow & \nearrow & \dots \\ 4 \to 5 & \cdots & \cdots \end{array}$$

2 Decision Trees

1. Gini Impurity

The goal in building a decision tree is to create the smallest possible tree in which each leaf node contains training data from only one class. To evaluate splits, we use a measure of node purity called Gini impurity:

$$\phi(p) = \sum_i p_i(1 - p_i)$$

where $p = (p_1, \dots, p_n)$ and each p_i is the fraction of elements from class i.

This represents the fraction of incorrect predictions if each element's class is predicted by randomly selecting a label according to the class distribution. The Gini impurity is:

- $\phi(p) = 0$ when all elements belong to the same class.
- $\phi(p)$ increases as the class mix becomes more uniform.

Example:

Calculate the Gini impurity for the following dataset:

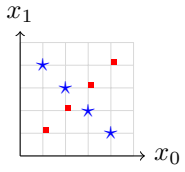


Figure 1: Dataset visualization for Gini impurity calculation.

Solution:

- Squares (class 1): 4, Stars (class 2): 16.
- $p_1 = \frac{4}{20} = 0.2, \quad p_2 = \frac{16}{20} = 0.8.$
- $\phi(p) = 0.2 \times 0.8 + 0.8 \times 0.2 = 0.32.$

2. Tree Construction

The decision tree construction algorithm recursively splits the training data into smaller subsets. Splits are selected based on the greatest decrease in impurity, measured by:

$$\Theta(s,t) = \phi(p) - P_L\phi(p_L) - P_R\phi(p_R)$$

where:

- s: Possible split.
- t: Node.
- P_L, P_R: Fraction of elements in the left and right child nodes, respectively.

Example: Recursive Tree Construction

Assume the best split places similar elements on the same side.

Split 1:

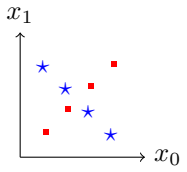


Figure 2: Visualization of Split 1.

$$\text{Left Impurity: } \phi(p_L) = 0 \times 1 + 1 \times 0 = 0$$
$$\text{Right Impurity: } \phi(p_R) = \frac{4}{13} \times \frac{9}{13} + \frac{9}{13} \times \frac{4}{13} = 0.426$$
$$\Theta(s,t) = 0.32 - (0.35 \times 0 + 0.65 \times 0.426) = 0.0431$$

Split 2:

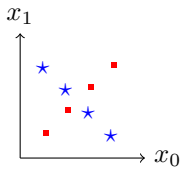


Figure 3: Visualization of Split 2.

$$\text{Left Impurity: } \phi(p_L) = \frac{4}{8} \times \frac{4}{8} + \frac{4}{8} \times \frac{4}{8} = 0.5$$
$$\text{Right Impurity: } \phi(p_R) = 0 \times 1 + 1 \times 0 = 0$$
$$\Theta(s,t) = 0.426 - (0.615 \times 0.5 + 0.385 \times 0) = 0.118$$

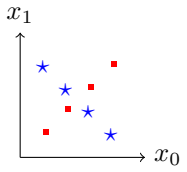


Figure 4: Visualization of Split 3.

Split 3:

$$\text{Left Impurity: } \phi(p_L) = 0$$
$$\text{Right Impurity: } \phi(p_R) = 0$$
$$\Theta(s,t) = 0.5 - (0.5 \times 0 + 0.5 \times 0) = 0.5$$

3. Classification Problem

Using the constructed decision tree, classify the following points:

1. $(0.4, 1.0)$ belongs to **Class 2 (star)**.
2. $(0.6, 1.0)$ belongs to **Class 1 (circle)**.
3. $(0.6, 0.0)$ belongs to **Class 2 (star)**.

The classifications were determined by traversing the tree to the appropriate leaf nodes based on the splitting criteria and assigning the class label of the respective leaf node.

3 Clustering

Overview

Clustering is an unsupervised learning technique to group data points:

- **Intra-cluster similarity**: Maximized within a cluster.
- **Inter-cluster similarity**: Minimized between clusters.

Types of Clustering

Partitional Clustering (e.g., k-means)

- Divides data into k disjoint clusters.
- Algorithm:
  1. Randomly initialize k centroids.
  2. Assign points to the nearest centroid.
  3. Recalculate centroids until convergence.

Hierarchical Clustering

- **Agglomerative**: Starts with individual data points, merges clusters iteratively.
- **Divisive**: Starts with one cluster, splits into smaller clusters iteratively.
- Can use Minimal Spanning Trees (MST) for visualization.

Probabilistic Clustering (e.g., Expectation Maximization)

- Data points belong to clusters with probabilities.
- Fits data to probabilistic models like Gaussian Mixture Models (GMM).

Metrics and Challenges

- **Distance measures**: Euclidean, Manhattan, Jaccard, etc.
- Sensitivity to initial conditions (e.g., centroids in k-means).
- Handling outliers and cluster validation are critical challenges.

Applications

- Customer segmentation.
- Plant species identification.
- Recommender systems.
- Image segmentation.

Kruskal's Algorithm

Kruskal's algorithm is used to find the Minimal Spanning Tree (MST) for a given graph. The MST connects all vertices without forming any cycles while minimizing the total edge weight.

Algorithm Steps

1. Sort all edges of the graph in non-decreasing order of their weights.
2. Initialize an empty set to store the edges of the MST.
3. For each edge in the sorted edge list:
   (a) Add the edge to the MST if it does not form a cycle.
   (b) Discard the edge if it forms a cycle.
4. Repeat until the MST contains $V - 1$ edges, where V is the number of vertices.

Example

Consider the following graph with vertices and weighted edges:

Steps to Compute the MST

1. Sort edges by weight: $(0, 1) : 0.16$, $(1, 2) : 0.19$, $(2, 3) : 0.26$, $(1, 3) : 0.29$, $(3, 0) : 0.34$.
2. Add $(0, 1)$ to the MST.
3. Add $(1, 2)$ to the MST.
4. Add $(2, 3)$ to the MST.

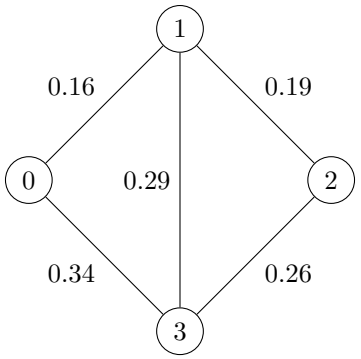


Figure 5: Graph for Kruskal's algorithm example.

5. $(1, 3)$ forms a cycle, so discard it.
6. Add $(3, 0)$ to the MST.

Resulting MST:

- Edges: $(0, 1), (1, 2), (2, 3), (3, 0)$.
- Total weight: $0.16 + 0.19 + 0.26 + 0.34 = 0.95$.

4 Regression

Simple Linear Regression

For a quantitative response Y on the basis of a single predictor variable X, we assume that there is approximately a linear relationship between X and Y. We can mathematically write this relationship as:

$$Y \approx \beta_0 + \beta_1 X$$

$\beta_0 \rightarrow$ **intercept** of the linear model
$\beta_1 \rightarrow$ **slope** of the linear model

They are known as the **model coefficients** or just **paraneters**. We will then use our training data and determine the *estimates* of $\hat{\beta}_0$ and $\hat{\beta}_1$ for the model coefficients.

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x \qquad \text{(estimates)}$$

where $\hat{y}$ indicates a prediction of Y on the basis of $X = x$

Estimating the Coefficients

β_0 and β_1 are unknown, we'll use our data to estimate the coefficients. Our data is:

$$(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$$

Our goal is to find the estimates $\hat{\beta}_0$ and $\hat{\beta}_1$ such that the linear model fits the available data well that is to
say $y_i \approx \hat{\beta}_0 + \hat{\beta}_1 x_i$ for i = 1,2,. . . ,n
Our task is to measure the *closeness*. The most common approach involves **minimizing the least**

squares criterion
We define e_i as the residual

$$e_i = y_i - \hat{y}_1$$

As evident, the residual is basically the difference between the observed i-th value and the i-th response value that is predicter by our linear model. We define the **residual sum of squares RSS** as

$$\text{RSS} = e_1^2 + e_2^2 + \cdots + e_n^2$$

R^2 Statistic

R^2 statistic provides an alternative measure of fit. It is the proportion of variance and takes a value between 0 and 1 and is independent of the scale of Y. It is a measureof the linear relationship between X and Y.

$$R^2 = \frac{\text{TSS} - \text{RSS}}{\text{TSS}} = 1 - \frac{\text{RSS}}{\text{TSS}}$$

1. $\text{TSS} = \sum (y_i - \overline{y})^2$ is the total sum of squares and measures the total variance in the response Y. It can be considered as the amount of variability inherent in the response before the regression is performed.
2. RSS measures the amount of variability that is left unexplained after performing the regression.
3. TSS − RSS measures the amount of variability in the response that is explained by performing the regression.
4. R^2 measures the proportion of the variability in the response (Y) that is explained using (X)
   (a) R^2 statistic that is close to 1 indicates that a large proportion of the variability in the response that is explained by the regression
   (b) R^2 statistic that is close to 0 indicates that the regression does not explain much of the variability in the response that is explained by the regression

Multiple Linear Regressions

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p + \varepsilon$$

where X_j represents the jth predictor and β_j quantifies the association between that variable and the response. β_j is interpreted as the *average effect* on Y of a one unit increase in X_j, holding all other predictors fixed.

Estimating the Regression Coefficients

estimates:

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \cdots + \hat{\beta}_p x_p$$

We choose $\beta_0, \beta_1, \ldots, \beta_p$

$$\text{RSS} = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

The values that **minimize the RSS** are the multiple least squares regression coefficent estimates

Gauss-Markov Theorem

The theorem states that, under specific assumptions, the OLS estimator minimizes the variance among all linear unbiased estimators. The assumptions are:

1. Linearity in parameters.
2. No multicollinearity among features.
3. No autocorrelation in residuals.
4. Homoscedasticity (constant variance of residuals).
5. Zero mean of residuals.

1. Linearity

1. a linear map or linear function is a function that satisfies:
2. additivity: $f(x+y) = f(x) + f(y)$
   homogenity of degree 1: $f(\alpha x) = \alpha f(x)\ \forall \alpha$
3. if linearity doesn't hold $\rightarrow$ non-linear regression function (piecewise linear regression)

$$y = \beta_0 + \beta_1 x[x > x_K] + \varepsilon \tag{10}$$

where $[x > x_k] = 0$ if $x \leq x_K$, $[x > x_k] = 1$ if $x \geq x_K$

2. Multicollinearity

A model is considered to be multicollinear if several of its features are correlated. Multicollinearity among features will result in less reliable statistical inference.

1. two features x and y are perfectly collinear if their correlation coefficent $\rho_{x,y}$ is ± 1.0

$$\rho_{x,y} = \frac{Cov[x,y]}{\sqrt{Var(X)}\sqrt{Var(Y)}} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{11}$$

2. we can check multicollinearity with the **variance inflation factor (VIF)**

$$\text{VIF}_K = \frac{1}{1 - (R_k)^2} \tag{12}$$

   VIF = 10 $\rightarrow {R^2}_k$ would be 90%, meaning 90% of the variance in the feature can be explained by the other features.

3. a feature that is correlated to other features can be determined by the tstatistic

$$t_{\hat{\beta}} = \frac{\hat{\beta} - \beta H_0}{\sigma_{(\hat{\beta})}} \tag{13}$$

   If the t-statistic is not significant:

   (a) feature is not related to the target variable (small correlation with target variable)
   (b) feature is related to the target variable (large correlation with target variable), but not required in regression due to strong relation with another feature $\rightarrow$ drop either of the related features

3. No Autocorrelation

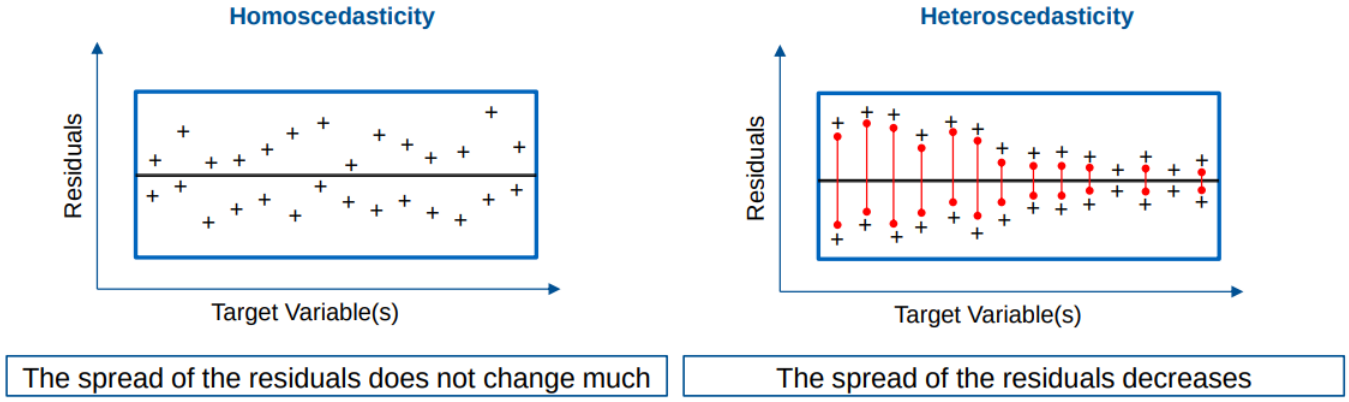
1. correlation of any time series with its own past and future values. In OLS regression there should be no pattern in the residuals over time if the errors are independent.
2. Durbin-Watson (DW) statistic to test for autocorrelation for residuals $\varepsilon_1, \ldots, \varepsilon_n$

$$\mathrm{DW} = \frac{\sum_{i=2}^{n}(\varepsilon_i - \varepsilon_{i-1})^2}{\sum_{i=1}^{n} {\varepsilon_i}^2} \tag{14}$$

DW = 0 (perfect positive correlation)
DW = 2 (no autocorrelation)
DW = 4 (perfect negative autocorrelation)

4. Homoscedasticity

Homoscedasticity is preserved if the variance of residuals is constant. In contrast, if the requirement of a constant variance of residuals is violated, we observe heteroscedasticity and $\mathrm{Var}(\varepsilon|x)$is not constant.



White Test (Example)

Model:
$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \epsilon$

Question: Is $\mathrm{Var}(\epsilon|\mathbf{x})$ constant?

Null-hypothesis:
$H_0 : \sigma_i^2 = \sigma^2$ for all $i = 1, \ldots, n$

Auxiliary OLS regression:
$\epsilon^2 = \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1^2 + \beta_4 x_2^2 + \beta_5 x_1 x_2 + \nu$

Test statistic (n = sample size, I = # coefficients in aux. model):
$nR^2_{\epsilon^2} \sim \chi^2_{I-1}$

Reject H_0 if test statistic is greater than critical value of chi-square distribution:

	Probability less than the critical value				
v	0.90	0.95	0.975	0.99	0.999
1	2.706	3.841	5.024	6.635	10.828
2	4.605	5.991	7.378	9.210	13.816
3	6.251	7.815	9.348	11.345	16.266
4	7.779	9.488	11.143	13.277	18.467

Heteroscedasticity does not necessarily invalidate a regression model, but may affect the efficiency of the estimates and the robustness of hypothesis tests

5. Exogeneity and Endogeneity

Exogeneity of a feature = feature is not correlated with the residuals
Endogeneity of a feature = feature correlated with the residuals and the assumption $\mathbb{E}[\epsilon|\mathbf{x}] = 0$ is violated because $\text{corr}(\epsilon, \text{x}) \neq 0 \Rightarrow \mathbb{E}[\epsilon|\text{x}] \neq 0$

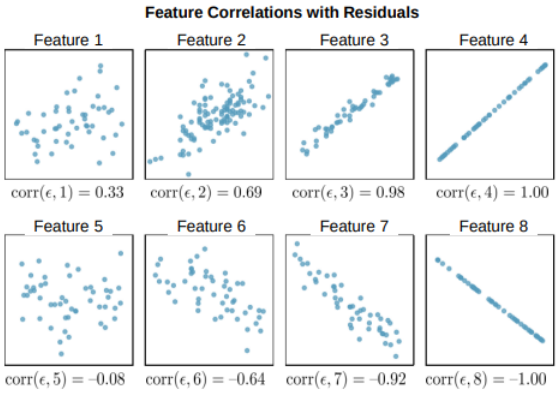
Reason for endogeneity:

- Measurement errors
- Features affecting each other
- Omitted / confounding features

Formula:

$$\text{corr}(\epsilon, \text{x}) = \frac{\text{cov}(\epsilon, \text{x})}{\sqrt{\text{var}(\epsilon) \cdot \text{var}(\text{x})}} \quad \text{with}$$

$$\text{cov}(\epsilon, \text{x}) = \frac{\sum_{\text{i}=1}^{\text{n}} \left(\text{x}_\text{i} - \overline{\text{x}}\right) \cdot \left(\epsilon_\text{i} - \overline{\epsilon}\right)}{\text{N} - 1}$$



[6]