FITPED

# Datamining

Ľubomír Benko
Michal Munk

www.fitped.eu

2021

Co-funded by the
Erasmus+ Programme
of the European Union

Work-Based Learning in Future
IT Professionals Education
(Grant. no. 2018-1-SK01-KA203-046382)

# Data Mining

**Published on**

November 2021

**Authors**

Ľubomír Benko | Constantine the Philosopher University in Nitra, Slovakia

Michal Munk | Constantine the Philosopher University in Nitra, Slovakia

**Reviewers**

Cyril Klimeš | Mendel University in Brno, Czech Republic

Anna Stolińska | Pedagogical University of Cracow, Poland

Ján Skalka | Constantine the Philosopher University in Nitra, Slovakia

Eugenia Smyrnova-Trybulska | University of Silesia in Katowice, Poland

Piet Kommers | Helix5, Netherland

**Graphics**

Ľubomír Benko | Constantine the Philosopher University in Nitra, Slovakia

David Sabol | Constantine the Philosopher University in Nitra, Slovakia

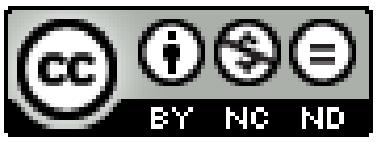
Erasmus+ FITPED

Work-Based Learning in Future IT Professionals Education

Project 2018-1-SK01-KA203-046382

Co-funded by the
Erasmus+ Programme
of the European Union








**ISBN 978-80-558-1794-1**

# Table of Contents

# Knowledge Discovery

Chapter 1

# 1.1 Knowledge discovery process

## 🕮 1.1.1

The huge amount of data that is recorded not only when visiting websites has little informative value. For this reason, the **concept of Knowledge Discovery** was created. The term knowledge discovery can be understood as a process that includes **data selection, data preprocessing, data transformation, data analysis** and **interpretation of results.**

One of the best-known areas of knowledge discovery is Knowledge Discovery in Databases (KDD), which could be defined as the non-trivial retrieval of implicit, previously unknown and potentially useful information from data. The data sources in this case are production databases and data warehouses. By analogy, the discovery of knowledge from databases includes the acquisition of data from texts, and this process is called **text mining**, i.e. Knowledge Discovery in Texts (KDT). As with KDD, statistical methods and machine learning methods are used to analyze data, while the biggest differences are in the data preparation itself, ie the representation of the text.

## 1.1.2

Choose the correct answer.

What is the area for discovering knowledge from the web?

- Web Mining
- Knowledge Discovery in Databases
- Knowledge Discovery in Texts

## 🕮 1.1.3

Today the Internet is the most dynamically developing source of information and an important source of data. From the need to analyze data from this source, a related area of discovering knowledge from databases was created - **discovering knowledge from the web** (*Web Mining, WM*). The definition of WM could be understood as the extraction of interesting and potentially useful knowledge and information from web-related activities. Based on the examined type of data in the extraction process, WM is categorized into three types:

- **discovery of knowledge based on web structure mining,**
- **discovery of knowledge based on web content** (*Web Content Mining*),

- **discovery of knowledge based on web usage** (*Web Log Mining or the term Web Usage Mining is also used*).

The biggest differences between the areas of knowledge discovery, in the process management by CRISP-DM methodology, are in **the phase of data preparation** (*Data Preparation*), while data preparation is **the most time-consuming phase** in the whole process of knowledge discovery. One of the most demanding data sources in terms of data preparation is the web server log file. The main reason is the large amount of irrelevant data collected and its inaccuracy or incompleteness.

## 1.1.4

Choose the correct answer.

What is the data source for web servers?

- Log file
- Cloud
- FTP

## 1.1.5

Websites are a source of information for visitors. However, a website can also serve as a source of information about visitors, such as a visitor's interests, needs, and behavior on a web portal.

The goal of **Web-based knowledge discovery** is to analyze user browsing behavior.

The issue of **Web Usage Mining** (*WUM*) and the study of the behavior of web users is part of several types of research. WUM involves understanding the behavior of users when using a website. A similar philosophy can be used for users of information systems, whose behavior in the system can reveal possible errors or contribute to the improvement of the system.

**Log files** are used to record tracks, whether on websites or in information files. Examining the log files reveals not only the behavior but also the habits of the users.

Since mainly anonymous data are recorded in the log files, it is necessary to process them and prepare them for analysis, using **the methods of data preprocessing**. Data preprocessing is an important part of WUM and a number of preprocessing techniques have been proposed for this purpose.

## 1.1.6

Choose the correct answers.

What can be the goal when examining the behavior of e-shop visitors from the log file?

- Shopping cart analysis
- The most visited categories
- Purchase satisfaction questionnaire
- Shopping in competing e-shops

## 1.1.7

The principle of **discovering knowledge** can be approached by **discovering patterns of behavior of web users**. Knowledge discovery can be understood as a process that includes data selection, data preprocessing, data transformation, data analysis and interpretation of results.

Web usage data is recorded in a web server log file. From a large amount of data, we can obtain information that will help us better understand the researched data. This information includes, for example, **statistics on the number of accesses for a given time period**, **the number of visits** or **the average length of visits to the web**, etc.

**The result of sequence analysis is sequence rules**, which represent the acquired knowledge, while the found rules are expected to be clear and useful. From the point of view of the application, it is possible to use only a part of the discovered knowledge (patterns of behavior of web users), the remaining rules are trivial in terms of usefulness, resp. inexplicable, that is, unusable because they do not bring any new knowledge. Based on **useful rules** is possible to identify navigation errors, edit links and other inaccuracies, respectively. identify the behavior of web portal users.

## 1.1.8

Choose the correct answers.

What different information can we get by examining the log file from the webserver?

- Numbers of accesses
- Average visiting
- Information about used devices (pc, mobile, ...)

- Information about the hardware used
- Browsing history of all sites

### 🕮 1.1.9

The **CRISP-DM methodology** provides a uniform and universal procedure for solving various tasks in the field of knowledge discovery (it represents the standardization of the process). The methodology consists of a sequence of steps - **business understanding, data understanding, data preparation, modelling, evaluation of results, deployment of results.** The order of the individual phases is not fixed and the process is cyclical.

It was primarily created for project management in the field of discovering knowledge from databases, but it can also be used for T*ext Mining* and *Web Mining*.

### 1.1.10

Sort the individual phases of the CRISP-DM methodology in the correct order.

- Evaluation
- Deployment
- Data preparation
- Modelling
- Data understanding
- Business understanding

## 1.2 Analytic methods

### 🕮 1.2.1

The core of the whole process of discovering knowledge is the application of analytical methods in order to acquire new knowledge. This phase is often referred to in English terms like ***Data Mining***, ***Modeling*** or ***Analyze***. The most common translation of the term *Data Mining* is an in-depth analysis, modelling, i.e. data mining. Any method that helps to gain more knowledge from the data is useful, i.e. *Data Mining* methods are a highly heterogeneous group.

The input to the analytical procedures is preprocessed or transformed data and the output is **knowledge**. The following phase of evaluation of results i.e. acquired knowledge is closely linked to the use of basic statistical methods. It is common practice to return to the modelling phase when the knowledge gained is not of the

required quality. Subsequently, it is necessary to change the parameters of the model, i.e. use of another valid method.

The choice of analytical method depends on the purpose for which the model is intended. There are usually a number of different methods for solving a problem. It is recommended to **use several different methods and combine their results.**

The creators of the **CRISP-DM** methodology (*Cross-Industry Standard Process for Data Mining*) list six types of problems - tasks and the methods recommended for them. The use of a particular method depends on the specific problem, and the list of recommended methods is not complete, but rather a list of the most commonly used methods to solve the problems. Also, problem types are not disjoint groups.

### 🕮 1.2.2

## Data description and summary

*Suitable methods:*

- descriptive statistics,
- data visualization,
- OLAP data analysis.

*KDD example:*

A business chain can use OLAP analysis to create a data cube that summarizes sales volume by time (year, quarter, month, week), product (product type, category, subcategory, product name) and geographic (region, county, district, city) dimension.

*Example WUM:*

Using the descriptive statistics application, the portal operator can find out basic characteristics about the use of the portal, such as the total number of accesses, the share of accesses from NAT and proxy devices, the share of accesses by search engine robots (Google, Bing, Yahoo, etc.), the share of internal and external access, the average length of identified sequences/visits, number of identified sequences, etc. Even a simple visualization of interaction frequencies can reveal, e.g. the dependence between the content of the portal viewed and accesses from inside and outside the organization's network.

### 1.2.3

Choose the correct answer.

Which methods would you use for data description and summaries?

- Data visualization
- Neural networks
- Clustering
- Decision trees
- Genetic algorithms
- Regression analysis
- Association rules
- Decision rules
- Discriminant analysis

### 🕮 1.2.4

## Segmentation

*Suitable methods:*

- clustering techniques,
- neural networks,
- data visualization.

*KDD example:*

A car sales company regularly collects information about its customers focused on socio-economic characteristics such as salary, age, gender, profession, etc. By applying cluster analysis, a company can segment its customers into several understandable groups and analyze the structure of each of them, which will allow them to develop specific marketing strategies for each group separately.

*WUM example:*

By applying cluster analysis, the portal operator can segment site users, identifying groups of visitors with similar needs. Subsequently, in combination with other knowledge, it can adapt the portal to the identified homogeneous groups.

### 1.2.5

Choose the right answers.

Which methods would you use for segmentation?

- Data visualization
- Neural networks

- Clustering
- Decision trees
- Genetic algorithms
- Regression analysis
- Association rules
- Decision rules
- Discriminant analysis

### 🕮 1.2.6

## Description of concepts

*Suitable methods:*

- decision rules,
- clustering of concepts.

*KDD example:*

By using data on new car owners and applying decision rules to that data, a car company can obtain rules that describe the loyalty and disloyalty of its customers. Below are examples of decision rules:

```
If GENDER = male and AGE> 51 then CUSTOMER = loyal
If GENDER = female and AGE> 21 then CUSTOMER = loyal
If PROFESSION = manager and AGE <51 then CUSTOMER = disloyal
If STATUS = single and AGE <51 then CUSTOMER = disloyal
```

*Example WUM*:

By applying decision rules to data on the use of the e-commerce system in connection with the user profile, the e-commerce operator can obtain rules that describe potential applicants for the offered product categories, i.e. services.

### 1.2.7

Choose correct answers.

Which methods would you use to describe concepts?

- Data visualization
- Neural networks
- Clustering

- Decision trees
- Genetic algorithms
- Regression analysis
- Association rules
- Decision rules
- Discriminant analysis

## 🕮 1.2.8

# Classification

*Suitable methods:*

- discriminant analysis,
- decision rules,
- decision trees,
- neural networks,
- k-nearest neighbours/methods based on analogy,
- case judgment / analogy-based methods,
- genetic algorithms.

*KDD example:*

Banks generally have information on the payments of their clients who repay the loan. By combining this financial information with other information about the client, such as gender, age, salary, etc., it is possible to create a system for classifying new clients (credit risk when accepting a client can be low or high).

*Example WUM:*

After segmenting the site visitors and applying the decision trees, the portal operator can clearly characterize the identified groups based on the user profile.

## 1.2.9

Choose correct answers.

Which methods would you use for classification?

- Data visualization
- Neural networks
- Clustering
- Decision trees
- Genetic algorithms

- Regression analysis
- Association rules
- Decision rules
- Discriminant analysis

### 🕮 1.2.10

## Prediction

*Suitable methods:*

- regression analysis,
- regression trees,
- neural networks,
- k-nearest neighbours/methods based on analogy,
- Box-Jenkins methodology/time series,
- genetic algorithms.

*KDD example:*

The annual income of an international company correlates with other attributes, such as advertising expenses, exchange rate, inflation, etc. Using these values (or their reliable estimates for next year), the company can predict expected income for the next year.

*Example WUM:*

By applying a multinominal logit model, the portal operator can model user behavior over time. **The multinominal logit model** can be used to model the probabilities of access to web parts of the portal depending on time, while by introducing artificial variables it can distinguish days of the week, weekend and working week, internal and external accesses, etc. depending on the hour of the day. This knowledge can be used for maintenance planning, modifications, portal downtime, i.e. its parts. If the portal serves as an interface (gateway) for access to other systems of the organization, the acquired knowledge can be useful for their administrators.

### 1.2.11

Choose correct answers.

Which methods would you use for prediction?

- Data visualization
- Neural networks

- Clustering
- Decision trees
- Genetic algorithms
- Regression analysis
- Association rules
- Decision rules
- Discriminant analysis

### 🕮 1.2.12

## Dependency analysis

*Suitable methods:*

- correlation analysis,
- regression analysis,
- association rules,
- Bayesian networks,
- inductive logic programming,
- data visualization.

*KDD example:*

By applying regression analysis, a commercial production company can find a significant relationship between the total sales of its products, their price and the total amount spent on advertising. By discovering this knowledge, a company can achieve the desired level of sales by changing the price and/or advertising costs.

*WUM example*:

By applying association and sequence rules, the portal operator can extract patterns of behavior of site users, search for associations between visited pages in order to restructure the pages according to the way they are viewed.

### 1.2.13

Choose correct answers.

Which methods would you use for dependency analysis?

- Data visualization
- Neural networks
- Clustering
- Decision trees

- Genetic algorithms
- Regression analysis
- Association rules
- Decision rules
- Discriminant analysis

# Data Acquisition

Chapter 2

# 2.1 Business understanding

## 🕮 2.1.1

The first phase of the process is the **phase of business understanding**. This phase is focused on understanding the objectives of the problem formulated from the point of view of the client and determining the task of discovering knowledge - the type of problem from the point of view of data modelling. In this initial step, it is very important to work with an expert on data from a given application area.

Knowledge discovery tasks:

- **data description and summarization,**
- **segmentation,**
- **description of concepts,**
- **classification,**
- **prediction,**
- **dependency analysis.**

## 📝 2.1.2

Choose correct answers.

Where can visits by visitors to the web portal be recorded?

- Log file
- Database
- PHP file
- Web browser
- Operating system

## 🕮 2.1.3

Most often we can meet with traffic analysis, restructuring, personalization and maintenance planning, where these applications can correspond to the following knowledge discovery tasks:

- traffic analysis - **dependency analysis,**
- restructuring - **dependency analysis,**
- personalization - **segmentation, classification, dependency analysis**,
- maintenance planning - **prediction.**

Dependency analysis is used mainly in the analysis of portal traffic (website), where we do not examine access to portal pages, but access to wider categories of portal content. Similarly, e.g. in the case of virtual learning environments, we do not examine access to course pages, but approaches to course activities. The aim is to **identify associations** between broader categories of portal/system content.

## 2.1.4

Choose correct answers.

If we wanted to focus on improving the user interface of our website, what methods would we use for analysis?

- Segmentation
- Classification
- Dependency analysis
- Prediction

## 2.1.5

In the case of the restructuring of the portal, we would also solve the task of dependency analysis, where based on the found patterns of behavior of web users, we can regroup links, identify navigation errors and inaccuracies on the site.

For example, personalizing the portal could consist of the following tasks - identifying groups of users with the same needs, describing homogeneous groups, and searching for patterns of behavior within the identified homogeneous groups.

In the case of maintenance planning, we would solve the prediction task, where we can model the probabilities of access to the web parts of the portal depending on time and other explanatory variables. This knowledge can be used for planned modifications, outages of the portal or its parts.

The first phase in the process of discovering knowledge is **business understanding**. The task is to understand the goals of the problem formulated in terms of data modelling.

Knowledge discovery tasks include:

- **data description,**
- **summary,**
- **segmentation,**
- **description of concepts,**
- **classification,**

- **prediction,**
- **dependency analysis.**

## 2.2 Data understanding - Data sources

### 🕮 2.2.1

**Web usage data** is collected from a variety of sources. They represent patterns of navigation for various segments of total web traffic, from single users and single-page browsing behaviors to patterns of multi-user and multi-page access.

The web server log file may not always contain enough information to infer client-side behavior because it relates to pages provided by the webserver. Data can be collected from:

- web servers,
- proxy servers,
- web clients.

### 🕮 2.2.2

From the point of view of what we want to analyze, we can have several **data sources**. Whether we want to research data regarding the use of a web portal, information system or electronic documents. Data sources for different approaches:

**Database:**

- production databases,
- data warehouses,
- public databases.

**Text:**

- collections of electronic documents (qualification works, open answers in questionnaires, text contributions ...).

**Web:**

- web content (websites, text posts, discussions, ...),
- site map,
- log files.

## 2.2.3

Choose correct answers.

What information does the sitemap provide us?

- Web portal structure
- The path to a specific page
- Information about the pages visited on the web portal
- Path to the log file

## 2.2.4

Let's take a closer look at the resources in **databases**. The advantage of the source in the database is mostly the possibility of editing using **SQL statements**. Due to the fact that not all data that are recorded are necessary for the analysis of sources, it is necessary to pre-process them in subsequent phases. **Working with a database reduces the need to create or use a data preparation tool.**

In the case of web portals, the ***MySQL*** database is most often used, nowadays ***MariaDB*** or ***PostgreSQL***, which are freely available database tools. A paid alternative is an ***MS SQL server***, or the possibility of using various cloud repositories (e.g. *Firebase*).

The preparation of data from databases is affected by the complexity of the database and the structure itself.

## 2.2.5

Choose correct answers.

Freely available database systems include:

- MariaDB
- MiSQL
- PostgreSQL
- OpenSQL
- SQLopen
- MySQL
- MS SQL Server Express

## 📖 2.2.6

For web servers, the main data source is a log file that is stored on a disk. The so-called ***Common Log File standard***.

```
127.0.0.1 user-identifier frank [10 / Oct / 2000: 13: 55: 36 -
0700] "GET /apache_pb.gif HTTP / 1.0" 200 2326
```

This type of file includes information:

- IP address,
- user identifier (in the case of web portals with mandatory authentication),
- date and time of access,
- Web server requests (that is, the requested page)
- return code (or server response),
- response size in bytes.

Some web servers also record so-called **referrer**, i.e. the website from which the user came to the page and also the so-called **user agent**, i.e. information about the web browser used and the operating system. All this information represents a significant added value for the analysis of the behavior of visitors to the web portal.

## 📝 2.2.7

Choose the correct answer.

What is a referrer?

- Information from where the user came to the page
- Friend on the phone
- Information regarding the recommendation of goods for purchase
- User information

## 📖 2.2.8

The ***Common Log File* format** is supported by most analytics tools, but the information about each transaction with the server is fixed. In many cases, it is desirable to record more information. Sites that are sensitive to privacy issues may want to skip certain data. In addition, ambiguities arise when analyzing the log file, because some delimiter characters may appear in some fields. The ***Extended Log File* Format** is designed to meet the following needs:

- allow control over the recorded data,

- support for the needs of representatives, clients and servers in a classic format,
- providing a robust solution to character problems
- allow the exchange of demographic data
- allow the expression of aggregated data.

Example of records in extended log file format:

```
#Version: 1.0
#Date: 12-Jan-1996 00:00:00
#Fields: time cs-method cs-uri
00:34:23 GET /foo/bar.html
12:21:16 GET /foo/bar.html
12:45:52 GET /foo/bar.html
12:57:34 GET /foo/bar.html
```

## 📖 2.2.9

When working with the Web, several **log files** are generally used, some **on the server-side** and others **on the client-side**. If we look at the server-side, there are several types of physical locations of the webserver.

1. Direct connection to the **Internet**, in which case the log file is created by the web server itself and this file is unique.
2. Proxy connection. The **proxy server** is connected to the Internet on the one hand and to the web server on the other. The role of a proxy is to reduce the load on the webserver by caching individual web pages, and if multiple clients request the same page, it offers it from memory and no longer forwards the request to the webserver. The proxy log file contains all access data, while the web server log file is no longer consistent.
3. Connection via a **load balancer**. It can be a stand-alone device or just using a Domain Name System (DNS) configuration. In this case, several web servers with the same content are involved, and clients connect to one of these servers depending on their configuration (either request are distributed sequentially as they arrive, or so that each server is equally busy). In this case, it may happen that each server stores its own log file, and in order to get complete, these files must be merged.

## 📝 2.2.10

Choose the correct answer.

What is a load balancer?

- Connecting multiple web servers with the same content
- Weight balancer
- One server that redirects users to another website
- User type

## 📖 2.2.11

From the **client's side**, we consider only two ways of connection:

1. the client has a direct connection to the **Internet** and thus its communication is directly between it and the server (even in the case of using a NAT device)
2. the client is connected through a **proxy**. In this case, the role of the proxy is to reduce the load on the Internet connection in such a way that if multiple clients request the same page, the connection to the server is made only once and each client is provided with a cached version. This results in the absence of requests in the webserver log files.

## 📖 2.2.12

Another necessary source of data in data preparation is the current **site map**. The sitemap contains information about whether there is a link between the pages, that is, whether there is a hyperlink between the pages from one page to another.

The most common way to obtain a site map is through **web crawling** implemented in data mining tools. Due to the fact that web portals are dynamic and constantly changing, it can be problematic to obtain historical data corresponding to the examined log file. Therefore, an alternative method is to generate a site map from the log file itself.

## 📝 2.2.13

Is this statement true?

The old site map can be used in combination with a newer log file even if the structure of the web portal has changed in the meantime.

- False
- True

# 2.3 Data acquisition (example)

## 🕮 2.3.1

The aim of the following project will be to complete all phases of work with the log file and analysis of data that are in the log file.

We will focus on the behavior of visitors to the university's web portal. The university portal consists of several pages, where visitors have access to important information about current and future studies. Orientation on the website or the availability of information may prevent visitors from accessing some information, or we may reveal that their information is a priority for other information.

## 🕮 2.3.2

As a source of data, we used automatically stored data on the use of the website, which are stored in a common standard structure, in text format, or in its own structure, most often organized in a relational database (for example, in the case of virtual learning environments).

We distinguish two types of web portals:

- web portal with **anonymous access**,
- web portal with **mandatory authentication**.

The difference between the types of web portals is significant because in the latter case we have information about the user directly in the log file.

## 📝 2.3.3

Choose the correct answer.

What type of portal is the *https://www.ukf.sk* website in terms of browsing the portal?

- Web portal with anonymous access
- Web portal with mandatory authentication

## 🕮 2.3.4

From the given sources it is enough to monitor the attributes:

- *IP address*,
- *date and time of access*,
- *URL*.

For portals/systems that require user authentication, the *user ID* must also be monitored. If our source is data on the use of the virtual learning environment, we are also interested in the *activity* attribute, which categorizes the course pages into individual activities. From these attributes, we further create variables that are used directly for modelling, i.e. in the data preparation phase.

## 📝 2.3.5

Choose correct answers.

What information can a standard log file from a web portal contain?

- IP address
- Date and time of access
- Access page
- Personal information
- Login passwords
- Web browser information

## 🕮 2.3.6

Another source of data that is needed primarily for data preparation is the current *site map*. Sometimes it is necessary to analyze user behavior from historical data, and a site map from that period may not be available.

For example, we would be interested in the behavior of users of the bank's portal in the pre-crisis period, i.e. in the period before the adoption of the euro. The adoption of the euro brought not only a new currency and new rules but also a new portal with a new structure for most banks. In this case, a *referrer* entry in the log file can be used to reconstruct the activities of site users.

## 📝 2.3.7

Choose correct answers.

We are researching the university web portal during the period in which the University hosts the *Open Day* event. What can we find out from the web portal log file?

- Has there been an increased interest in the study programs offered?
- Were visitors interested in the possibility of accommodation in the dormitory?
- What other universities are students still considering?
- What sites on the university portal did students visit?

# Data Preprocessing

Chapter 3

# 3.1 Data cleaning

## 📖 3.1.1

To perform a good data analysis, it is necessary to have quality data. Log files are typical in that they contain a considerable amount of irrelevant data that can corrupt the analysis of the data, so it is necessary to delete this data already in the data preparation phase.

The following methods can be used to examine the behavior of users or visitors to the Web portal:

- **sample survey** - answers to specific items of the questionnaire are surveyed and the website visitor is aware of the subject of research,
- **web usage mining** - the log file of the webserver is analyzed, which contains information about access to the pages of the web portal, the visitor is not aware of this research, and his data is to some extent anonymous.

## 📝 3.1.2

Choose the right answers.

What information is recorded in web server log files?

- IP address
- E-mail
- Referrer
- Browsing history
- Site map
- Browser information

## 📖 3.1.3

Preparation of data for further analysis is, in addition to the collection of quality data, one of the prerequisites for quality work with data. The ***data preparation phase*** (or ***data preprocessing phase***) is one of the most time and resource consuming in the process of discovering knowledge. One reason is the amount of irrelevant data in the log files.

The researchers who analyzed the data preparation in *WUM* concluded that in the field of web analysis, data preparation is a very important phase that requires the use of tools typical for data preparation, and these tools cannot be used in other domains.

Other researchers have come up with their own modification of the log file, in the case of virtual learning environment (*VVP*) portals. With this adjustment, they managed to minimize the need for data preparation and directly extract all the necessary data for analysis. However, this solution cannot be used in the case of portals with anonymous access, where it is necessary to follow the classic process of data preparation.

## 3.1.4

Choose correct answers.

What is considered unnecessary data in the log file?

- Access to images
- Access to websites
- Access to files
- Access to javascripts
- Access to icons

## 3.1.5

**Cleaning data** from unnecessary data is one of the first steps in data preparation and is specific to each web portal or data source. The aim of data cleaning is to delete records, i.e. links that are not essential to the behavior of web users. Such links mainly include approaches to:

- picture,
- flash videos,
- cursor icons
- javascript,
- style.

The usual procedure for identifying such records involves **identification based on the extension** (** .jpg, * .jpeg, * .bmp, * .png, * .gif, * .css, * .js, * .flw, * .swf, * .cur , * .rss, * .ico, * .xml*, fonts and the like). Even if only one page is loaded, all these requests are written to the log file.

## 3.1.6

Choose the correct answer.

On what basis can we identify accesses to unnecessary data in the log file?

- File extensions
- Special designation in the log file
- Prepositions in the file name

## 🕮 3.1.7

In addition to the **GET** request, other HTTP protocol requests are written to the log file, such as 4xx / 5xx return or status codes, which identify the client/server error that needs to be cleared.

The **HTTP status code** is part of the server response header for the client request. Specifies how the response was processed by the server - whether the request was processed positively, negatively, or an error occurred. The next step is for the client to interpret and respond to the response status code.

The response header, along with the status code, includes a **status message**, which is an English verbal description of the status code. Status codes are divided according to the nature of the response into five categories:

- informational,
- successful,
- redirect,
- bad request (client error),
- server error.

The **status code** is three decimal numbers, where the first number specifies the category of the answer and the remaining numbers specify it in more detail:

- **1xx Information**
- **2xx Successful**
- **3xx Redirect**
- **4xx Client error**
- **5xx Server error**

## 3.1.8

Choose correct answers.

Which status codes are not needed to examine log files?

- Informational
- Successful

- Redirect
- Client error
- Server error

## 📖 3.1.9

The next step in data preparation is to clean the data from **accesses by search engine robots** such as *Google, Yahoo, Bing,* etc. Because robots access the web portal sequentially, it is not appropriate to include their activity in the study of user behavior.

The robots are detected either on the basis of their identification in the *User Agen*t field or on the basis of an IP address that can be compared with the robot database, which can be found, for example, at *www.robotstxt.org*.

Identification of search engine robots can be performed using:

- keyword ***bot, spider, crawl, robot,***
- ***hidden link*** access,
- ***robots.txt*** accesses.

## 📝 3.1.10

Choose correct answers.

Which keywords can we use to identify the search engine robot?

- bot
- slot
- crawl
- spider
- bee
- sniper
- crouch

## 📖 3.1.11

In the case of data cleaning, we can proceed based on the following algorithm:

1. Loading a log file into the program;
2. Creating a new text file;

3. Creating tokens based on which redundant accesses will be identified (*.jpg, *.jpeg, *.bmp, *.png, *.gif, *.css, *.js, *.flw, *.swf, *.cur , *.rss, *.ico, *.xml, ...) supplemented by search engine robot tokens, where the IP addresses of the robots are extracted based on access to the *robots.txt* file;
4. Searching for log file entries, if the entry does not contain any of the searched tokens, then the entry is written to a new file;
5. The new file then contains the cleaned up original log file.

## 3.2 Data cleaning (example)

### ⌨ 3.2.1 Cleaning from unnecessary access I. - images

Remove accesses to **jpg* and **jpeg* images from the log file. Remember that the log file may also contain uppercase extensions. The filename is given by input and you should import it into the dataframe.

Print the shape of the cleaned data frame.

### ⌨ 3.2.2 Cleaning from unnecessary access II. - images

Remove accesses to **png*, **bmp*, and **gif* images from the log file. Remember that the log file may also contain uppercase extensions. The filename is given by input and you should import it into the dataframe.

Print the shape of the cleaned file.

### ⌨ 3.2.3 Cleaning from unnecessary access III. - videos

Remove access to**flv*, **swf* videos from the log file. Remember that the log file may also contain uppercase extensions. The filename is given by input and you should import it into the dataframe.

Print the shape of the cleaned data frame.

### ⌨ 3.2.4 Cleaning from unnecessary accesses IV. - icons

Remove access to **ico*, **cur* icons from the log file. Remember that the log file may also contain uppercase extensions. The filename is given by input and you should import it into the dataframe.

Print the shape of the cleaned data frame.

### ⌨ 3.2.5 Cleaning from unnecessary approaches V. - styles, javascript

Remove accesses to styles and javascripts **css*, **js* from the log file. Remember that the log file may also contain uppercase extensions. The filename is given by input and you should import it into the dataframe.

Print the shape of the cleaned data frame.

### ⌨ 3.2.6 Cleaning from unnecessary accesses VI.- fonts

Remove access to the **svg*, **woff*, **eot* fonts from the log file. Remember that the log file may also contain uppercase extensions. The filename is given by input and you should import it into the dataframe.

Print the shape of the cleaned data frame.

### ⌨ 3.2.7 Cleaning from unnecessary accesses VII. - other

Remove accesses to **rss*, **xml*, **json* from the log file. Remember that the log file may also contain uppercase extensions. The filename is given by input and you should import it into the dataframe.

Print the shape of the cleaned data frame.

### ⌨ 3.2.8 Cleaning from unnecessary accesses VIII. - status codes

Remove accesses from the log file that contain server or client error status codes. The filename is given by input and you should import it into the dataframe.

Print the shape of the cleaned data frame.

### ⌨ 3.2.9 Cleaning from unnecessary access IX. - requirements

Remove accesses that contain *POST* or *HEAD* requests from the log file. The filename is given by input and you should import it into the dataframe.

Print the shape of the cleaned data frame.

### ⌨ 3.2.10 Identification of robots I.

Identify *robots.txt* accesses and delete entries from these IP addresses. The filename is given by input and you should import it into the dataframe.

Print the shape of the cleaned data frame.

### ⌨ 3.2.11 Identification of robots II.

Identify the accesses of the search engine crawlers in the *User Agent* field and delete the records. The filename is given by input and you should import it into the dataframe.

Print the shape of the cleaned data frame.

## 3.3 Variable creation

### 🕮 3.3.1

In this section, we will focus on the **dummy variables** that we will use in the next phases of data preparation. These are mainly time-based variables, or other artificial variables that are needed for the later phase of data analysis (for example, distinguishing user access, etc.).

### 🕮 3.3.2

The first essential variable is a **variable representing the date and time of access** to the web portal. Thanks to this variable, we can later identify individual user sessions. It usually has a date and time format, and because the webserver can handle different format settings, the formats may differ. For example, the source data date fields are in the format YYYY / MM / DD and the target date fields are in the format MM-DD-YYYY. It is necessary to unify these formats. To unify the formats, we will use the so-called ***unix time*** and use the transformation to convert the date field of the source data to the corresponding target format.

**Unixtime** (also known as Epoch time, POSIX time, seconds since Epoch, or UNIX Epoch time) is a time point description system. It is the number of seconds that

have elapsed since the Unix era, minus the leap seconds; the Unix epoch is dated **January 1, 1970 00:00:00 UTC**; leap seconds are ignored, with a leap second having the same Unix time as the second before it, and **each day** is considered to be exactly **86 400 seconds** long. Thanks to this approach, Unix time is not a true expression of UTC.

Unix time is widely used in operating systems and file formats. On *Unix*-like operating systems, the ***date*** command prints or sets the current time. By default, it prints or sets the time in the system time zone, but with the *-u* flag, it prints or sets the time in UTC and with the TZ environment variable set to reference a specific time zone, it prints or sets the time in that time zone.

## 3.3.3

Write correct answer.

What date and time are marked as the beginning of the era? Write the answer in the form *dd.month yyyy hh:mm:ss*

## 3.3.4

We can use different approaches to transform the date and time to Unix time. In *Python*, we can use the ***datetime*** library.

```
import datetime
```

The *datetime* library ensures that the date and time are read correctly, regardless of the format. All we have to ensure is the correct distribution of the log file (i.e. the correct identification of the year, month, day, hour, minute and second).

```
dt = datetime.datetime(year, month, day, hour, minute, second)
```

The obtained ***timestamp*** can be converted to Unix time by the following notation:

```
unix = dt.timestamp()
```

## 3.3.5

Choose the correct answer.

What would happen if we did not define time values for the *datetime* function?

- It would work with the time 00:00:00

- The program would return an error
- It would work with a time of 12:00:00

### 🕮 3.3.6

Another dummy variable that we will need is the time spent on the page, i.e. length of time spent on-page. It is mainly used to identify sessions and it is used to refer to this variable as ***length***. When creating the *length* variable, it is necessary to start from the Unix time stamp and have a log file sorted according to the following fields:

- IP address,
- UserAgent,
- unixtime.

This will ensure sequential follow-up of the approaches of individual visitors (user identification will be discussed in more detail in the next chapter).

### 📝 3.3.7

Arrange the log file correctly by columns.

- ip address
- unix time
- user agent

### 🕮 3.3.8

We will create the ***length*** variable by going through the whole log file and comparing two consecutive records. If there are equal ***IP addresses*** and also a ***User Agent*** in two consecutive records, we can read the Unix access times between these two records. In this way, we get the time spent on the page of the first record.

The time spent on the site is **always positive!** It is advisable to choose the upper limit of the so-called ***time window***, we assume that if the time between two visited pages is greater than, for example, 1 hour, then it will be a **new visit**. We can choose the size of the time window according to the needs of the web portal. For example, when researching time spent on an educational web portal when the lesson time is 90 minutes, this limit can be selected.

## 3.4 Variable creation (example)

### ⌨ 3.4.1 Date and time transformation

Divide the date and time from the log file into separate variables: *year, month, day, hour, minute, second.* The filename is given by input and you should import it into the dataframe.

Print the date in format for the given row of the dataframe, e.g. 24.12.2021 21:20:12

### ⌨ 3.4.2 Creating unixtime

Transform the date and time from the log file to Unix time and insert it into the log file as a new variable named *unixtime*. The filename is given by input and you should import it into the dataframe.

Print the unixtime for the given row of the dataframe, e.g. 1582089949.0

### ⌨ 3.4.3 Arrangement

Arrange the entries in the log file by *IP address, user agent* and *Unix time.* The filename is given by input and you should import it into the dataframe.

Print the IP address for the given row of the dataframe, e.g. 192.168.1.160

## 3.5 User identification

### 🕮 3.5.1

In the previous chapters, we introduced two types of web portals, web portals with anonymous access and web portals with mandatory authentication. When working with a web portal with anonymous access, it **is necessary** to complete the phase of data preparation - **user identification**. In the case of working with a web portal with mandatory authentication, this phase **does not need to be** completed. This information is already in the log file.

## 📖 3.5.2

The log file primarily records anonymous user data, but there is a problem with uniquely identifying the site visitor. In the analysis, it is not necessary to know the specific identity of the user but to be able to **distinguish between individual users**. The assumption that an *IP address* is sufficient to identify a user is incorrect because there can be multiple users behind one *IP address*.

## 📝 3.5.3

Is this statement true?

The *IP address* is sufficient as a sufficient identifier for user access to the web portal.

- False
- True

## 📖 3.5.4

Because the *IP address* is not a sufficient parameter to identify the user, it is necessary to combine several methods, such as using the *Cookie* field, or a combination of the *IP address* with the User Agent field.

The *cookie* stores some parameter values on the client-side. For each request with the same web server browser, the *cookie* information is sent along with the request, and the webserver recognizes that it is the same user and therefore delivers the requested page without recreating it.

Several heuristic methods mainly use a **combination of an *IP address* with a *User Agent*** field. If the *IP address* changes, it is clear that it is a new user. If the *IP address* is the same, the User Agent field is compared, and if there is a change, a **new user** is identified, otherwise, it is the **same user**.

If the *IP address* and *User Agent* are the same, the provider **URL** and site topology is checked. If the requested page is not directly accessible from any of the pages visited by the user, then the user is marked as a **new user** with the same address.

The caching issue can be fixed by giving HTML pages a short expiration time and forcing the browser to load each page from the server.

### 📝 3.5.5

Choose correct answers.

By combining which information from the log file can we identify users?

- IP address
- Unixtime
- Return code
- URL
- Cookie
- User agent
- Referrer

### 🕮 3.5.6

In the case of a portal with mandatory user registration, i.e. by logging in, user identification is simplified because the entry is already in the log file.

A more detailed analysis of session identification options was provided by several authors, and one of the other session identification options was the **analysis of the sequence of site visits** to the web portal. It is assumed that if there is **no link** to the next web page from the current page, it must be a **new user**.

## 3.6 User identification (example)

### ⌨ 3.6.1 User identification

Create a new variable in the log file to indicate the user ID. Identify users by a combination of *IP address* and *User Agent*. The filename is given by input and you should import it into the dataframe. Print the userID for the given row of the dataframe

### ⌨ 3.6.2 The length of time spent on the page

Create a *length* variable representing the length of time spent on the page. Select 60 minutes as the top border. The filename is given by input and you should import it into the dataframe. Print the length for the given row of the dataframe

# 3.7 Session identification

## 📖 3.7.1

A user can visit a specific page multiple times, in which case a multiple session (visit) is recorded for each user in the log file. However, to work with data, we need to **distinguish individual sessions**, to divide the individual approaches of each user into separate sessions. This is done by **session identification**, which is one of the most important steps in data preprocessing.

## 📝 3.7.2

Is this statement true?

We can identify a session without knowing which user it is.

- False
- True

## 📖 3.7.3

A ***session*** can be defined as:

- the sequence of steps that lead to the fulfilment of a certain task,
- the sequence of steps that lead to the achievement of a certain goal.

Structure-oriented heuristics, time-oriented heuristics, as well as combinations of these two approaches, are used to identify sessions. In the English literature, the identification of a user's session is also referred to as a ***user activity record***, which reveals the sequence of logged-in activities of one user.

Identification of users is challenging. The most common way to distinguish the uniqueness of visitors is the use of ***cookies*** on the client's side. However, not all sites use *cookies* and may also be disabled by the client for privacy reasons.

## 📖 3.7.4

Sessions can also be distinguished by time. The simplest method is if we consider a session to be a series of clicks over a period of time - a ***time window***, e.g. in 10

minutes, 30 minutes, etc. **The duration of the session must not exceed the value of the time window.**

However, a more effective method is to divide the session into **several sessions**. In the event that we find such two records of page views, when the time between views was greater than the selected time window, e.g. 30 minutes, the session is split.

The real-time value for a session can be obtained from empirical data. Estimation of the time window based on the value of the ***average time on site*** statistics obtained, e.g. using Google Analytics, which represents the **average time a user has on a website**.

## 3.7.5

Assign a session number to the log file entries based on a 60-minute time window. Number the sessions from 1 ...

```
_____ 178.41.1.187 - - [16/Apr/2021:18:16:26 +0100] "GET "...
_____ 178.41.1.187 - - [16/Apr/2021:18:18:42 +0100] "GET
/oznamy"...
_____ 178.41.1.187 - - [16/Apr/2021:20:55:00 +0100] "GET
/studium"...
_____ 178.41.1.187 - - [16/Apr/2021:21:05:12 +0100] "GET
/stipendium"...
_____ 178.41.1.187 - - [16/Apr/2021:21:05:19 +0100] "GET "...
_____ 172.45.7.160 - - [16/Apr/2021:17:17:55 +0100] "GET
/fakulty"...
_____ 172.45.7.160 - - [16/Apr/2021:17:58:20 +0100] "GET "...
```

## 3.7.6

The ***time window*** is usually 30 minutes long. This value is based on research by *Catledge* and *Pitkow*, who calculated the average time spent on the page plus the standard deviation value. Specifically, 9.3 minutes was the time spent on the page, plus 1.5 times the standard deviation of the time spent on the website. The result was a *Session Timeout Threshold* (STT) of 25.5 minutes.

Although a 30-minute time window has become the most widely used, this does not mean that it is the most appropriate in all circumstances.

## 3.7.7

Choose the correct answer.

What is the most commonly used time window length for identifying sessions?

- 30 minutes
- 60 minutes
- 90 minutes
- 120 minutes
- 10 minutes
- 30 seconds
- 60 seconds
- 120 seconds
- 10 seconds

## 3.7.8

An alternative to a fixed time window is an **estimate based on a quartile range** that is not affected by extreme values, e.g. *Q3 + 1.5Q* where *Q3* is the ***upper quartile*** (75th percentile) and *Q* is the ***quartile range*** (mean 50% of the values). In other words, if we consider the time spent on the site to be a remote value, a new session begins.

## 3.7.9

Choose the correct answer.

How do we calculate the quartile range Q?

- Q3-Q1
- Q4-Q2
- Q1+Q3
- Q1-Q3
- Q2+Q4
- Q1+Q2+Q3+Q4

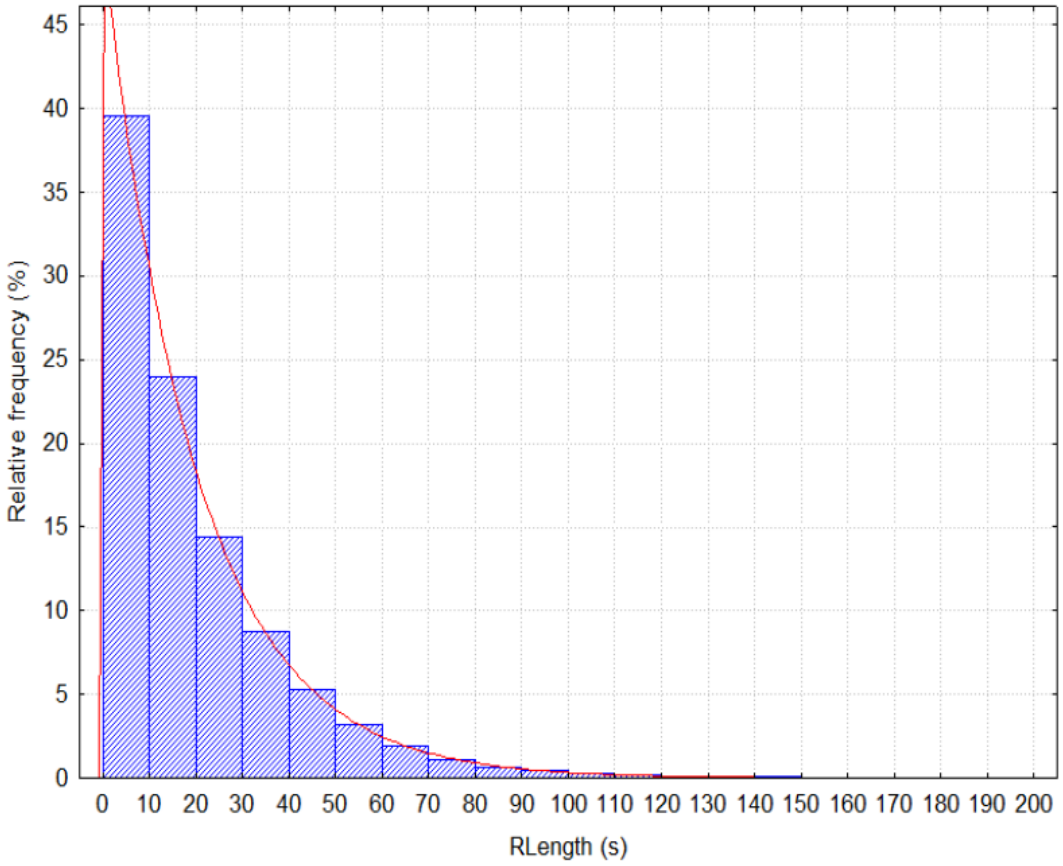
## 3.7.10

Another example of a method combining these two approaches is the ***Reference Length*** method. The identification of sessions using the *Reference Length* method

is based on the assumption that the length of time a user spends on a page is related to whether the page is classified as ***content*** or ***auxiliary***.

Suppose a user navigates through several pages while visiting a site to find the content they are looking for. These pages are called ***auxiliary*** and the user "only" navigates through them. A ***content*** page is a page where a user ends a search and spends more time on it than on *auxiliary* pages.

The figure below shows a histogram showing the distribution of the *Length* variable, which represents the time spent on the pages of the university portal. We assume that the **variance of time** spent on auxiliary pages is small, as the user "only" moves to the destination of their search. The auxiliary pages form the left part of the graph. The length of time spent on content pages is more varied and forms the right part of the graph.




## 3.7.11

Choose correct answers.

In the *Reference Length* method, we classify web pages into:

- Auxiliary
- Content
- Tracking
- Redirect
- Activation
- Pop-up
- Blocking

## 🕮 3.7.12

The *Reference Length* session identification method is typical in that to use it, we need to calculate the ***cut-off time C***, which determines the end of the session and the beginning of a **new session**. To calculate the cut-off time C, it is necessary to know the share of navigation pages of the examined web portal. This value is used to estimate subjectively by the web portal administrator.

Based on the assumption of the exponential distribution of the variable, it is possible to calculate the *cut-off time C* as follows:

$$C = -\ln(1 - p) / \lambda, \text{ for } 0 <= p < 1$$

If ***p*** is the relative number of navigation pages, we can use the quantile function to estimate the *cut-off time C*. The maximum plausible estimate of the parameter ***λ*** (average intensity of events) is:

$$\lambda = 1 / RLength_{avg},$$

where $RLength_{avg}$ is the observed average length of visits.

If we have an estimated time limit, the session can be identified by comparing each time spent on the page with the time limit, with the time limit dividing the pages into navigation and content according to the length of time spent on a particular page.

## 🕮 3.7.13

If we have an estimated *cut-off time*, the session can be identified by comparing each time spent on the page with *cut-off time*, with the *cut-off time* dividing the pages into navigation and content according to the length of time spent on a particular page.

After estimating the *cut-off time C*, the session will be represented by a sequence of visited pages with a ***timestamp*** to which the following applies:

$$RLengthi <= C,$$

where *1 <= i < k* and for the last page of the session:

***RLengthk> C,***

where **another session** is defined from the page with this property.

## 3.7.14

Is this statement true?

The following approaches, with a *cut-off time C* of 40 seconds, will be in one session.

```
178.41.1.187 - - [16/Apr/2021:18:21:42 +0100] "GET /oznamy"...
178.41.1.187 - - [16/Apr/2021:18:22:23 +0100] "GET
/studium"...
```

- False
- True

## 3.7.15

From the page to which the property applies

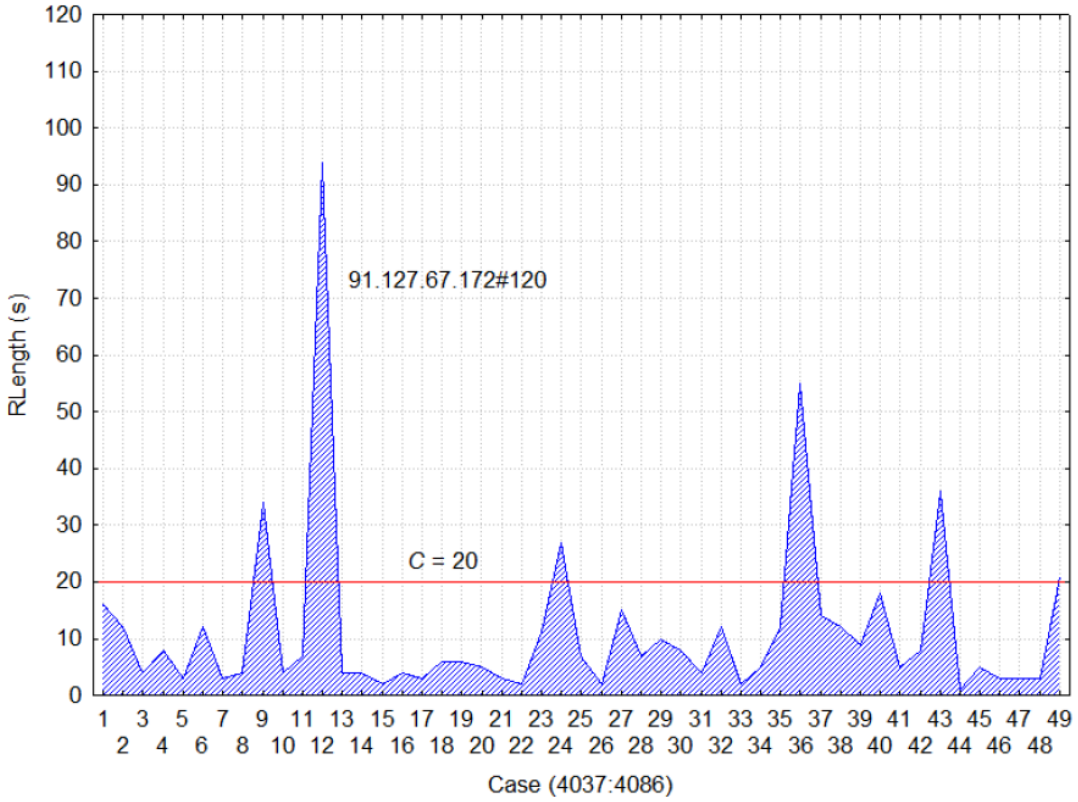
***RLength$_i$ > C***

a **new session** is defined, with the first *k - 1* pages classified as navigation pages and the last *k* - page classified as content.

The time spent on navigation pages is less than or equal to *cut-off time C* and the time spent on the content page is greater than *cut-off time C*.

The **x-axis** in the image represents the sequence of visited pages from a given IP address and from a given agent arranged according to the access time, the **y-axis** represents the time spent on the page, the estimated time limit is 20 seconds.

The first session consists of a sequence of pages with sequence numbers 1 to 9, the first eight being classified as navigation pages, the ninth being classified as a content page. The next session consists of a sequence of pages with sequence numbers 10 to 12.




## 3.7.16

Is this statement true?

The following approaches, with a *cut-off time C* of 120 seconds, in one session.

```
178.41.1.187 - - [16/Apr/2021:18:20:31 +0100] "GET /oznamy"...
178.41.1.187 - - [16/Apr/2021:18:21:00 +0100] "GET
/studium"...
178.41.1.187 - - [16/Apr/2021:18:22:17 +0100] "GET
/fakulty"...
178.41.1.187 - - [16/Apr/2021:18:22:33 +0100] "GET /"...
```

- False
- True

## 3.7.17

The length of time spent on a page is calculated as the difference between the time of the next page and the current page. Naturally, the time spent on the last page of the sequence cannot be calculated. The *Reference Length* method **assumes that all**

**recent pages are content pages**. However, this assumption can cause errors. For example, in case the user interrupts the session, e.g. by phone call or lunch break, the navigation page may be incorrectly classified as content. However, it is unlikely that the error will appear regularly on the same page.

Also important is the fact that dividing a page into a specific type in terms of user model may be different for each user. For example, a navigation page for one user may be a content page for another user and vice versa.

## 3.8 Session identification (example)

### ⌨ 3.8.1 STT average I.

Calculate the average length of the session.

The filename is given by input and you should import it into the dataframe. Print the average.

### ⌨ 3.8.2 STT average II.

Based on the average time window estimate from the previous entry (*STT_MEAN*), identify the sessions and create a new variable in the log file.

The filename is given by input and you should import it into the dataframe. Print the STT_Mean for the given row of the dataframe

### ⌨ 3.8.3 STT quartile estimate I.

Calculate the value of the quartile estimate for the given log file.

The filename is given by input and you should import it into the dataframe. Print the estimate.

### ⌨ 3.8.4 STT quartile estimate II.

Based on the time window quartile estimate (*STT_Q*), identify the sessions and create a new variable in the log file.

The filename is given by input and you should import it into the dataframe. Print the STT_Q for the given row of the dataframe

### ⌨ 3.8.5 STT 10 minutes

Based on a 10-minute estimate of the session length (*SLength*), identify the sessions and create a new variable in the log file.

The filename is given by input and you should import it into the dataframe. Print the *SLength* for the given row of the dataframe

### ⌨ 3.8.6 Reference Length - cutoff time

Calculate the cutoff time for the Reference Length method.

### ⌨ 3.8.7 Reference Length

Based on the cutoff time estimate from the previous entry, identify the sessions and create a new variable in the log file based on the Reference Length method.

The filename is given by input and you should import it into the dataframe. Print the RLength for the given row of the dataframe

## 3.9 Path completion

### 🕮 3.9.1

The next step in pre-processing the data is to **path completion** that reconstructs the visitor's activity on the website. It is implemented mostly after the identification of the session, but it is no longer one of its methods. The purpose of path completion is to determine if there are significant web accesses that are not recorded in the log file.

This may be the case, for example, if a user returns to the previous page during the same session, in which case a second attempt to access the page is likely to end up displaying a previously downloaded version of the page. Researchers *Taucher* and *Greenberg* have shown that more than 50% of web accesses are a move backwards.

Another reason for not completing paths may be the browser's cache. When moving back, sometimes the query to the webserver is not executed, so there is no record in the log file.

The solution to this problem is **path completion**. By adding a path, we add these missing lines to the log file.

## 3.9.2

Is this statement true?

It is possible to add paths without identifying sessions.

- False
- True

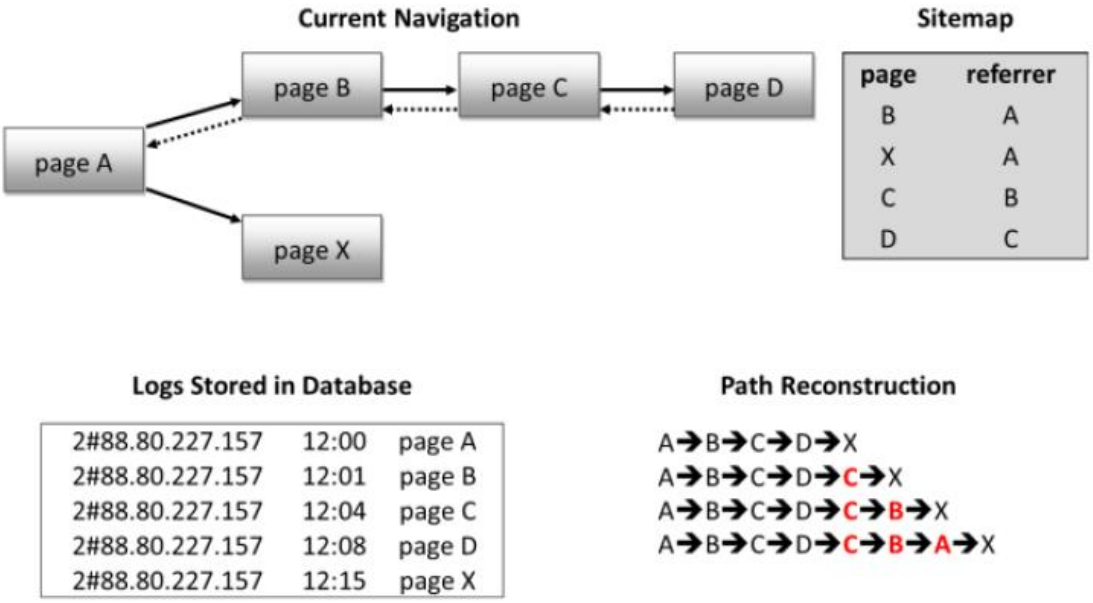
## 3.9.3

A simple example of path completion is illustrated in the figure below.

The graph represents the structure of the web portal and the arrows move the user through the pages. The dashed arrows represent the page navigation caused by the Back button. We see that the user returned from page *D* to page *C*, from there to page *B*, and finally to page *A*, where they clicked on the link to page *X*.

Links to pages *C*, *B*, and *A* do not appear in the log file because they were already saved in the previous steps on the client-side. When recording activity, there is a mismatch between the actual activity of the user and its recording in the log file from page *D* to page *X*.

Keep in mind that if there is a link from page *B* or page *C* to page *X*, and we only have a log file, there is probably an unlimited number of possible path completions.




## 🕮 3.9.4

For the backfilling of the path, a **site map** is of great importance, which contains information on whether there is a **link** between the individual pages, i. whether there is a hyperlink from one page to another. We obtain a **web map** using web crawling methods implemented in most data mining tools.

After sorting the records by *IP address*, we will find out if there is a link between successive pages. The sequence for the selected *IP address* can look like e.g. $A \rightarrow B \rightarrow C \rightarrow D \rightarrow X$. Based on the **sitemap**, it can be determined that there is no hyperlink from page *D* to page *X*. We, therefore, assume that the user came to this page using the Back button from one of the previous pages. By looking back, we then determine which of the previous pages there is a link to page *X*.

In our example, the algorithm finds that if there is no hyperlink from page *C* to page *X*, page *C* is inserted into the sequence, i. the sequence will look like this $A \rightarrow B \rightarrow C \rightarrow D \rightarrow C \rightarrow X$.

Similarly, he finds that there is no hyperlink even from page *B* to page *X*, so we also add it to the sequence, i. $A \rightarrow B \rightarrow C \rightarrow D \rightarrow C \rightarrow B \rightarrow X$.

Finally, the algorithm detects that page *A* contains a hyperlink to page *X*, and the sequence will look like $A \rightarrow B \rightarrow C \rightarrow D \rightarrow C \rightarrow B \rightarrow A \rightarrow X$ after the return path analysis is complete. This means that the user used the Back button to switch between pages *D* to *C*, *C* to *B*, and *B* to *A*.

## 3.9.5

Choose the correct answer.

What is the role of a sitemap when completing roads?

- Represents the structure of a web portal
- Contains information about the number of visits to the web portal
- Contains information about pressing the back button in the browser

## 3.9.6

One of the tasks of a ***site map*** is to provide the structure of a web portal. Due to the dynamic nature of the web portal, the site map may not always be up to date or may not always be available. One option is to extract the sitemap directly from the log file, which can result in an incomplete sitemap.

There are cases where the site map cannot be created, such as when the log file contains historical data. Then it is possible to proceed on the basis of the ***Referrer*** field, using a certain type of backtracking, i.e. recursion. It is possible to use different approaches, e.g. a sequential search of the log file based on the session, and if the request does not match the previous ***Referrer***, then there is **no link** between the two pages.

Subsequently, it is possible to search back the previous accesses from the given session and search for the page from which the user got to the required page (*Referrer* vs. *URL*). If we don't come across the requested page in the sequence, it means that the user has probably left the web portal and returned later, or the path cannot be completed.

The absence of a site map, or leaving the web portal on the user's side, may represent certain limitations for calculating the estimated share of navigation pages for the needs of the *Reference Length* method.

## 3.9.7

Choose the correct answer.

An old site map of a newer log file can also be used to examine web portal traffic.

- Yes, provided that the structure of the site has not changed
- Yes, the timeliness of the site map does not matter
- No

# Data Transformation

Chapter 4

# 4.1 Data transformation

## 🕮 4.1.1

In this section, we will focus on **working with the log file**, which we preprocessed in the previous phase of data preparation. As part of the **data preparation**, redundant accesses to images, *javascripts*, styles, etc. were removed in the log file. The accesses of search engine robots were also identified in the file, which was subsequently removed. Finally, users and sessions were identified using the *Reference Length* method.

## 🕮 4.1.2

Although we cleaned the log file during the data cleanup phase, it is still necessary to **transform** the data, because, in the case of web portals, the log file **still contains too much information**. Specifically, these are **time data and access** to the portal pages.

If the web portal has an extensive structure and contains several subpages, or it is an e-shop, the log file contains access to a large number of pages. The number of these approaches will be a small or negligible number of the total.

It works similarly with access time. It is very impractical for web portals to distinguish between accesses in seconds or minutes. For this reason, the so-called **data transformation**, ie the creation of new variables.

We will show the data transformation on the example of a prepared log file of a university web portal with anonymous access.

## 🗎 4.1.3

Choose the correct answer.

What is data transformation?

- Merge information into categories for better data analysis
- Delete a structure-based log file
- Delete a small number of site accesses

## 📖 4.1.4

Before a deeper analysis of user behavior can be performed, it is necessary to create ***artificial variables*** that will be used for closer analysis:

- *category* - websites classified into categories;
- *internal* - takes values 0 or 1, depending on whether it is accessed from inside the network or from outside;
- *student* - takes the values 0 or 1, depending on whether it is a student's access to the web portal;
- *employee* - acquires the values 0 or 1, depending on whether it is an employee's access to the web portal;
- *hour* - takes values 0-24, which represent the hour of access to the web portal;
- *square hour* - the square of the hour of access;
- *day* - takes values 1-7, which represent the days of the week;
- *individual days* (Monday - Saturday) - take the values 0 or 1, depending on the day of access of the record.

## 📝 4.1.5

Choose correct answers.

Which of the following artificial variables is appropriate to create by transforming data?

- Week - takes values 0-53
- Year - takes on the values of the given year - 2020
- Second - takes values 1-60
- Day - takes values 1-7

## 📖 4.1.6

The web portal pages you visit are divided into several **categories**, based on the structure of the web. The URLs of some pages may contain additional characters that may depend on the user's activity on the website (e.g. search, access to various announcements, subpages, etc.). With the help of categories, it is possible to simplify the evaluation of portal traffic.

In our example, 13 categories were created:

- úvod (home),
- štúdium (study),
- oznamy (announcement),

- o_fakulte (faculty),
- informácie_pre (information_for),
- informácie_o (information_about),
- predpisy_vyhlášky (statutes),
- dokumenty (documents),
- veda_výskum (research),
- ostatné (other),
- vyhľadávanie (search),
- udalosti (events),
- konferencie (conferences).

## 4.1.7

Choose correct answers.

How can we categorize web portal pages?

- Structures - categories based on subpages
- Traffic - categories for the most visited sites
- Polls - visitors choose their favorite sites by poll

## 4.1.8

Another variable that needs to be created is information on whether it is ***internal or external access***.

Based on the *IP address* from the log file, we can divide access to *student* access and *employee* access. IP addresses are divided as follows:

- 10.160.0.xxx - internal student access,
- 10.160.1.xxx - internal student access,
- 10.160.2.0xx - internal student access,
- 10.160.2.1xx - internal student access,
- 10.160.2.2xx - internal employee access,
- 10.160.3.xxx - internal student access,
- 10.160.xxx.xxx - internal employee access,
- others - external access.

In the case of the *IP address* of the internal variable, the value is set to **1**. In the case of another IP address, it is external access and the value is set to **0**.

For external accesses, we will not distinguish between student and employee access, because it is not possible to determine the type of user based on the IP address. If we were to examine the login portal, where each user would be

categorized, it would be possible to **distinguish between student access and employee access.**

In our example of a university portal, it is possible to distinguish between student access and employee access by logging in via an e-mail address, as an e-mail address was set up for each student in the form of *xxxx@student.ukf.sk* and for employees in the form of *xxxx@ukf.sk*.

### 🕮 4.1.9

The next step is to **transform the date-time variable**, which is divided into a specific *hour* and date corresponding to the *day*. The variable *hour* takes values from 0 to 23 and the variable *day* takes values from 1 to 7, where 1 represents Monday and 7 represents Sunday.

### 4.1.10

Choose the correct answer.

How could the *IP addresses* of university portal visitors who are part of the university (*student/employee*) be identified in the case of external access?

- Based on access to the page accessible only to students / employees.
- Based on the regularity of access to the web portal.

## 4.2 Data transformation (example)

### 4.2.1 Data transformation I.

Create an *internal* variable that contains information about whether the access is from the university's internal network (1) or not (0). Identify accesses based on IP addresses:

- 10.160.xxx.xxx - internal access,
- others - external access.

The filename is given by input and you should import it into the dataframe. Print the internal value for the given row of the dataframe

### 4.2.2 Data transformation II.

Create a *student* variable that contains information about whether it is student access (1) or not (0). Identify accesses based on IP addresses:

- 10.160.0.xxx - internal student access,
- 10.160.1.xxx - internal student access,
- 10.160.2.0xx - internal student access,
- 10.160.2.1xx - internal student access,
- 10.160.2.2xx - internal employee access,
- 10.160.3.xxx - internal student access,
- 10.160.xxx.xxx - internal employee access,
- others - external access, ie 0.

The filename is given by input and you should import it into the dataframe. Print the student value for the given row of the dataframe

### 4.2.3 Data transformation III.

Create an *employee* variable that contains information about whether it is employee access (1) or not (0). Identify accesses based on IP addresses:

- 10.160.0.xxx - internal student access,
- 10.160.1.xxx - internal student access,
- 10.160.2.0xx - internal student access,
- 10.160.2.1xx - internal student access,
- 10.160.2.2xx - internal employee access,
- 10.160.3.xxx - internal student access,
- 10.160.xxx.xxx - internal employee access,
- others - external access, ie 0.

The filename is given by input and you should import it into the dataframe. Print the employee value for the given row of the dataframe

### 4.2.4 Data transformation IV.

Create an *hour* variable that will contain information in the form 0-23, depending on when the website was accessed. At the same time, create the *hour_var* variable, which is needed for the data analysis phase and is equal to the square of the access hour.

The filename is given by input and you should import it into the dataframe. Print the hour value for the given row of the dataframe

### ⌨ 4.2.5 Data transformation V.

Create a *day* variable that will contain information in the forms of 1-7. The day is calculated based on the date of access to the website.

The filename is given by input and you should import it into the dataframe. Print the day value for the given row of the dataframe

### ⌨ 4.2.6 Data transformation VI.

Create variables representing individual days of the week (*Mon, Tue, Wed, Wed, Fri, Sat, Sun*), which will contain information in the form 0-1. The day is calculated based on the date of access to the website.

The filename is given by input and you should import it into the dataframe. Print the wed value for the given row of the dataframe

# Data Exploration

Chapter 5

# 5.1 Python for Data Science

## 🕮 5.1.1

*Python* is an excellent language for performing data analysis, mainly because of the fantastic ecosystem of *Python* libraries. The most common libraries for working with data include:

- *Pandas*
- *Scipy.stats*
- *Numpy*
- *Statsmodels*
- *Scikit-learn*
- *Researchpy*

## 🕮 5.1.2

The ***Pandas*** library is used to work with data files and greatly simplifies the import and analysis of data, so we will work with it.

The library has functions for analyzing, cleaning, exploring, and manipulating data. The name "***Pandas***" has a reference to both "*Panel Data*", and "*Python Data Analysis*" and was created by *Wes McKinney* in 2008.

***Pandas*** library allows us to **analyze big data and make conclusions based on statistical theories**. ***Pandas*** can clean messy data sets, and make them readable and relevant. Relevant data is very important in data science.

*Pandas* library gives you answers about the data. Like:

- Is there a correlation between two or more columns?
- What is the average value?
- Max value?
- Min value?

*Pandas* are also able to delete rows that are not relevant, or contains wrong values, like empty or NULL values. This is called ***cleaning the data***.

## 📝 5.1.3

Choose the correct answer.

What is *Pandas* library used for?

- Data analysis
- Data generating

## 📖 5.1.4

In *Python* alias are an alternate name for referring to the same thing. Library *Pandas* is usually imported under the ***pd*** alias.

```
import pandas as pd
```

Now we can the *Pandas* library referred to as ***pd*** instead of *Pandas*.

*Example*

```
import pandas as pd

mydataset = {
 'page': ["home", "study", "about"],
 'visits':   [354, 74, 85]
}

data = pd.DataFrame(mydataset)

print(data)
```

## 📝 5.1.5

Write the correct answer.

In what relation to the *Pandas* library is *pd* in the said declaration?

```
import pandas as pd
```

## 📖 5.1.6

A *Pandas* ***Series*** is like a column in a table. It is a **one-dimensional array** holding data of **any type**. It's easy to transform a list into a ***Series*** in *Python*.

```
import pandas as pd

session = ["home", "home", "study"]
```

```
data = pd.Series(session)

print(data)
```

The result will be following:

```
0     home
1     home
2     study
```

## 5.1.7

Choose the correct answer.

How many dimensions does a *Series* structure have?

- One-dimensional
- Two-dimensional
- Three-dimensional

## 5.1.8

You can also use a key/value object, like a dictionary, when creating a *Series*.

```
import pandas as pd

session = {"ses1":"home", "ses2":"home", "ses3":"study"}

data = pd.Series(session)

print(data)
```

The keys of the dictionary become the labels:

```
ses1     home
ses2     home
ses3    study
```

The ***index*** argument is used to select only some of the entries in the dictionary, specifying only the entries to be included in the *Series*.

```
import pandas as pd

session = {"ses1":"home", "ses2":"home", "ses3":"study"}
```

```
data = pd.Series(session, index=["ses1","ses3"])

print(data)
```

Output:

```
ses1      home
ses3     study
```

## 5.1.9

Choose the correct answer.

Which of the following best describes the structure of the dictionary?

- Key / value pair
- One-dimensional
- String only

## 5.1.10

Data sets in *Pandas* are usually multidimensional tables called **DataFrames**. The series represents one column of the table, and the *data frame* **is like the entire table**. A Pandas ***DataFrame*** is a two-dimensional data structure, such as a two-dimensional array, or a table with rows and columns.

```
data = pd.read_csv("log.csv",sep=";")
print(data)
```

Output:

```
       CATEGORY    HOUR
0          home       8
1         study       8
2          home       8
3   information       8
```

## 5.1.11

Choose the correct answer.

What is the difference between the *Series* and *DataFrame*?

- Series is like column of a table and DataFrame is the whole table
- They are the same but DataFrame is more effective
- DataFrame is like column of a table and Series is the whole table

## 📖 5.1.12

A *DataFrame* is like a table with rows and columns. *Pandas* library provides a unique method for retrieving rows from a *DataFrame*. The ***DataFrame.loc[]*** method is a method that **returns one or more specific rows.**

The *Pandas* Indexer ***loc*** can use a *DataFrame* for two different use cases:

- selection of rows by label/index
- boolean/conditional search row selection

```
print(data.loc[0])
```

The result of this command will be the output of the first item from the *DataFrame*:

```
CATEGORY    home
HOUR           8
```

## 📝 5.1.13

Choose the correct answer.

What is the **loc** used for in DataFrame?

```
print(data.loc[0])
```

- It returns one or more specified rows of the DataFrame
- It returns one or more specified columns of the DataFrame
- It returns the whole table of the DataFrame

## 📖 5.1.14

One of the most commonly used methods for getting a quick overview of a *DataFrame* is the ***head()*** method. The method returns the top *n* rows of a *DataFrame* or *Series*, where *n* is the input value from the user. The default value of the parameter is 5. The method is used to quickly test whether a certain object has the correct data type.

```
data = pd.read_csv("log.csv",sep=";")
```

```
print(data.head())
```

The output will be following:

```
      CATEGORY    HOUR    INTERNAL
0         home       8           1
1        study       8           1
2         home       8           1
3  information       8           1
4   university       8           1
```

With negative values of n, the ***head ()*** function returns all rows except the last n rows, equivalent to df ***[: - n]***.

## 📖 5.1.15

The opposite of the *head ()* function is the ***tail()*** function, which is used to get the last *n* rows. The function returns the last *n* rows of an object based on position. Used to quickly verify data, such as after sorting or joining rows.

```
data = pd.read_csv("log.csv",sep=";")
print(data.tail())
```

With negative values of *n*, this function returns all rows except the first *n* rows, equivalent to df ***[n:]***.

## 5.1.16

Choose the correct answer.

What is the difference between the function *head()* and *tail()*?

- Function head() returns the first 5 rows of the DataFrame and tail() the last 5 rows
- Function tail() returns the first 5 rows of the DataFrame and head() the last 5 rows
- Function head() returns the first row of the DataFrame and tail() the last row
- Function tail() returns the first row of the DataFrame and head() the last row

### 📖 5.1.17

The first step in getting to know the data is to discover the different types of data they contain. Even if anything is inserted into the list, the columns of the data frame contain values of a specific data type. The ***dataframe.info()*** function is used to obtain a brief summary of the data frame (*DataFrame*) and to display the data types for the individual columns.

```
data = pd.read_csv("log.csv",sep=";")
print(data.info())
```

The function returns information about the *DataFrame*, including index and column types, non-zero values, and memory usage.

**Empty values or Null** values in rows can cause problems when parsing data, and consideration should be given to eliminating them. Ideal in the pre-processing phase.

Of course, the **model definition phase** can identify some of these values that we did not discover during pre-processing and can **solve the problem**.

### 📝 5.1.18

Choose the correct answer.

What function do you use to obtain more knowledge about the examined dataset (like columns type and row count)?

- info()
- head()
- tail()
- information()
- getinfo()
- getinformation()

## 5.2 Model definition

### 📖 5.2.1

Errors can occur when moving data from one software environment to another. Also, an interpretation of the new data set may lead to an error or cause further

misunderstandings. For this reason, it is necessary to set aside time to **verify the data**. This can save you time later and avoid any mistakes.

One way to verify the data is to **calculate basic statistics** and **compare them with published results**. The *Series* class provides a ***value_counts*** method that counts the number of occurrences of each value that results in a *Series*; ***sort_index()***. The *Series* is sorted by index so that the values are displayed in order.

## 5.2.2

Choose the correct answer.

What is the *sort_index()* command for?

- Sort data by index
- Sort data alphabetically

## 5.2.3

In the model definition phase, the **individual variables** and their **interdependence are examined**. Using contingency tables, coefficients and significance tests, it will be determined which variables can be used in the model, e.g. whether it is necessary to distinguish between approaches from inside and outside the network, approaches from students and staff, or whether it is necessary to distinguish between days of the week.

***Contingency Tables*** are a necessary foundation when working with categorical data. A contingency table is one of the techniques for examining two or even more variables. It is basically the sum of the numbers between two or more categorical variables. The size of the contingency table is given by the number of categorical variables, the more categorical variables, the more complicated the table and the more difficult it is to determine the mutual relations.

Estimates such as mean, median, standard deviation are quite suitable tools for the analysis of one-dimensional data. When **comparing two variables**, a ***correlation*** is possible.

***Correlation*** is a measure of the dependence between two or more variables. The correlation coefficient can range from -1 to +1. A value of -1 represents the highest negative and +1 the highest positive correlation. A value of 0 indicates no correlation.

## 5.2.4

Choose the correct answer.

What is the PivotTable not used for?

- Research of two or more variables
- Research of one - dimensional data

## 5.2.5

The analysis of the log file, cleaned of irrelevant data, can be divided into several steps. In the next section, we will go through each of them step by step.

To begin with, we can save the log file to a *DataFrame* and load it using the following command:

```
data = pd.read_csv("log.csv", sep = ";")
```

First, we need to review the log file during model definition. To do this, we can use the *DataFrame* structures of the *Pandas* library and their properties:

```
data.head()
```

As we know, the ***head()*** function returns information about the first n elements. In our case, since the value of n is not set and the default value is 5, the function returns information in the form of the first 5 lines of the *DataFrame*, ie 5 lines of our log file. This way we can take a close look at what our data looks like in the log file.

## 5.2.6

Choose the correct answer.

What will be the result of the *head()* function with the defined parameter?

```
df.head(10)
```

- Instead of the first 5 lines, n lines defined as a function parameter are displayed
- Instead of the first 5 lines, every n lines defined as a function parameter are displayed
- The last n rows defined as a function parameter are displayed

## 📖 5.2.7

From the previous lesson, we know another method for the *DataFrames* object, ***info()***, which provides more information about network data.

```
print(data.info())
```

The output is following:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 36443 entries, 0 to 36442
Data columns (total 13 columns):
CATEGORY     36443 non-null object
INTERNAL     36443 non-null int64
STUD         36443 non-null int64
EMPL         36443 non-null int64
HOUR         36443 non-null int64
HOUR_SQR     36443 non-null int64
DAY          36443 non-null int64
MO           36443 non-null int64
TU           36443 non-null int64
WE           36443 non-null int64
TH           36443 non-null int64
FR           36443 non-null int64
SA           36443 non-null int64
dtypes: int64(12), object(1)
memory usage: 3.6+ MB
```

The result tells there are 36443 rows and 13 columns:

```
RangeIndex: 36443 entries, 0 to 36442
Data columns (total 13 columns):
```

And the name of each column, with the data type:

```
CATEGORY     36443 non-null object
INTERNAL     36443 non-null int64
STUD         36443 non-null int64
...
```

## 📝 5.2.8

Choose the right answers.

What methods can be used to initially examine our data set?

- head()
- tail()
- info()
- about()
- asc()
- desc()

## 📖 5.2.9

After the first examination of the data in the data set, we can look at the basic **statistical parameters**. Use the ***description()*** function to display some basic statistical details of a *DataFrame* or *Series* of numeric values, such as percentile, average, default, and more.

```
data.describe()
```

The results show that, for example, the average value of the variable *hour* is 14.1932, which could lead to the assumption that most accesses were during the afternoon. The min and max values of the *hour* variable indicate that the data is correct because min is equal to 0 and max is 23.

## 5.2.10

Choose correct answers.

What statistical details can be obtained from a DataFrame using the *description()* function?

- mean
- std
- min
- max
- low
- high
- error rate
- difference

## 📖 5.2.11

One of the best ways to describe a variable is by distributing the variable.

A **variable distribution** is a mathematical function that consists of the possible values of a given variable and the frequency of each group of values. For example, the number of inhabitants and their age classification.

Usually, these values are arranged in order from smallest to largest and presented graphically. The most common graphical representation of the distribution is a ***histogram***.

## 5.2.12

Choose the correct answer.

What is a histogram?

- A graph that shows the frequency of each value
- A graph that shows the mean of each value
- A graph that shows the std of each value
- A graph that shows the min and max of each value

## 5.2.13

Another library offered by the *Python* programming language that we can use to work with data is ***researchpy***. This library creates *Pandas DataFrames*, which contain the relevant statistical information about testing that is commonly required for academic research. The information is returned as *Pandas DataFrames*, allowing you to quickly and easily export results to any format/method that works with a traditional *Pandas DataFrame*.

```
import researchpy as rp
```

***Researchpy*** combines various packages such as *Pandas, Scipy.stats* and *Statsmodels* to obtain all the standard required information in one method. Because the analyzes were not available in these packages, code was developed to fill the gap.

## 5.2.14

Choose the correct answer.

What is used the *researchpy* library for?

- Obtaining statistical testing information

- Obtaining research information
- Obtaining specific knowledge from data

## 🕮 5.2.15

The Contingency Table tool, introduced in the previous lesson, can also be used in *Python* with the ***crosstab()*** function.

The *Pandas* library offers several options for grouping and summarizing data, but this variety of options can sometimes be confusing. All of these approaches are powerful tools for analyzing data, but it can be confusing to know whether to use a grouping, a Contingency Table, or a crosstab to build a summary table. Therefore, it is better to use the ***crosstab()*** function to analyze categorical data and create a frequency table.

A cross-tabulation is a two- or multidimensional table that records the number (frequency) of a particular variable that has the specific properties described in the table cells. In the literature, we can also find the name contingency table, *Chi-square*, *data tabulation*.

## 🕮 5.2.16

*Cross-tabulation* tables provide a wealth of information about the **relationship between variables**.

***Chi-square*** statistics are the primary statistics used to test the **statistical significance of a multi-table**. ***Chi-square*** tests determine whether two variables are independent or not. If the variables are **independent** (have no relationship), then the result of the statistical test will be "insignificant" and we assume that there is no relationship between the variables. If the variables are **dependent**, then the result of the statistical test will be "statistically significant" and we can state that there is a certain relationship between the variables.

## 🕮 5.2.17

To find out the significance of individual variables, we will use the ***contingency tables*** mentioned in the last lesson and compare the various variables in our data set.

Let's focus on internal and category variables. The internal variable tells us whether the web portal was accessed from inside or outside the university network. To find

out the contingency coefficients, we will use the ***researchpy*** library and the ***crosstab()*** function as follows:

```
import researchpy as rp

crosstab, test_results, expected =
rp.crosstab(data["CATEGORY"], data["INTERNAL"],
                                                    test= "chi-
square",
                                                    expected_freqs=
True,
                                                    prop= "cell")
```

The input to the function is two examined variables, in our case *category* and *internal*.

## 🕮 5.2.18

The test selection follows. For our example, we will use the ***Chi-square independence test***.

The ***Chi-square test of independence*** is a statistical test of hypotheses that are used to determine whether two categorical or nominal variables are likely to be related or not.

To get the parameters, look at the values of the *test_results* variable.

```
test_results
```

| | Chi-square test | results |
|---|---|---|
| 0 | Pearson Chi-square ( 12.0) = | 2814.6017 |
| 1 | p-value = | 0.0000 |
| 2 | Cramer's V = | 0.2779 |

The ***Cramer contingency coefficient V*** is the best-known contingency coefficient and represents the most appropriate degree of association between two nominal variables. Gets values from 0 (no relationship) to 1 (perfect relationship). When interpreting the contingency coefficient, the scale for the correlation coefficient may be used:

- Cramer's V> 0.50: Strong dependence,
- Cramer's V> 0.30 & <0.50: Medium dependence,
- Cramer's V> 0.10 & <0.30: Weak dependence,
- Cramer's V <0.10: Trivial dependence.

Based on the results of the contingency, it can be seen that there is a weak dependence between the variables *internal* and *category*.

In the project, we will focus on comparing several variables of our data file.

## 5.2.19

Choose the correct answer.

How can we find the relationship between two categorical variables?

- Contingency coefficient
- Average value

## 5.2.20

When comparing two variables and determining their interdependence for analysis, we use a ***contingency coefficient***.

The ***contingency coefficient - C***. is used to calculate the relationship between the two variables and is based on *Pearson's* ***Chi-square test***, which measures the difference between the **actual and expected frequencies** in the cells of the contingency table. Gets values from the interval <0,1>, where 0 means absolute independence.

However, monitoring contingency coefficients is not enough for analysis. It is also advisable to monitor the **number of records** for each category.

In our example of a university portal, we look at the frequency of visits to individual web categories by users who access from within the network and by users who access from the outside. In this case, we will use a ***histogram***.

## 5.2.21

Based on the observation of the frequency of access to individual categories, we can say that for most categories, the examination of access would not be relevant due to the low number of visits. Only researching the categories *home*, *study, announcement* and *faculty* could bring interesting results.

We can say that distinguishing between internal and external approaches is important only in the case of the above-mentioned web categories.

## 🕮 5.2.22

The aim of the application of analytical methods is to acquire new knowledge. The input to the analytical tools is pre-processed or modified data and the output is **knowledge**. The choice of analytical method depends on the purpose for which the model is intended.

The basic concept for a correct understanding of the acquisition of new knowledge is the ***discovery of knowledge***, which describes the process of finding knowledge and useful information from a large amount of data that help us make decisions in various situations.

Over time, various methodologies have emerged to standardize the knowledge discovery process, such as *SEMMA, 5A, ASUM-DM* or *CRISP-DM*.

## 🕮 5.2.23

Among the mentioned methodologies, one of the best known is ***CRISP-DM*** (*Cross Industry Standard Process for Data Mining*), which covers the complete process of data mining tasks. It is a standardized and freely available form of a suitable approach to solving data mining problems and is independent of the software tools used.

***CRISP-DM*** consists of 6 consecutive phases. The result achieved in one phase will influence the choice of step in the following phases. It is often necessary to return to some steps and phases (eg data preparation, modelling).

Phases of the ***CRISP-DM*** method:

- business understanding phase
- data understanding phase
- data preparation phase
- modelling phase
- evaluation phase
- the deployment phase of the acquired models

## 🕮 5.2.24

One method of solving prediction and classification problems is ***logistic regression***, which describes the dependence of a qualitative dependent variable on one or more quantitative independent variables. In ***logistic regression***, the probability that a variable has a specific value is modelled depending on the combination of the values of the independent variables.

The results of ***logistic regression*** are well interpretable, but on the other hand, they require thorough data preparation. ***Logistic regression*** is one of the generalized linear models and, like linear regression, is based on statistical distribution. The difference is that the dependent variable is not continuous, but is discrete. In order to use regression, the dependent variable is **transformed into a continuous** value, which is a function of the probability of the event occurring.

# Data Modeling

## Chapter 6

# 6.1 Modelling

## 🕮 6.1.1

We will use the ***Multinominal Logit Model*** to model the behavior of web users, which is a special case of a generalized linear model and is used to model the distribution of a categorical variable. Specifically for modelling the probabilities of access to the web parts of the portal as a function of time. The ***multinominal logit model*** is practically identical to logistic regression, except that instead of the one you have more possible results.

*Example*

*Young people's career choices can be influenced by their parents' occupations and their own level of education. We can study the relationship between choosing a profession and the level of education and profession of, for example, father. The choice of occupation will be the resulting variable, which consists of occupational categories.*

When modelling the behavior of web users, we use a pre-processed log file, which is cleaned of unnecessary data and accesses by search engine robots.

Subsequently, it is necessary to select the correct method by which we identify user sessions so that it is possible to properly reconstruct the path of all users. Before determining the model, we will create a variable, in our case an independent variable hour, which will take values from 0 to 23 (which represents the hour of the day of access).

All the above-mentioned steps, phases have been explained in the previous chapters. We will continue to explore the university portal log file.

Other options of data analysing include the **generation of association** and **sequence rules** or **cluster analysis**.

## 6.1.2

Choose correct answers.

Which phases must be followed before analyzing the log file?

- Cleaning data from unnecessary data
- User identification
- Seat identification
- Path completion
- Cluster analysis

- Transformation of variables
- Defining the model
- Sequence analysis
- Association analysis

## 🕮 6.1.3

The use of specific methods for analysis - data modelling, affects the structure as well as the nature of data on the web use. In the preprocessed data from the log file, **patterns of behavior** of web users are most often searched for.

For this purpose, ***Sequence Rule Analysis*** is used to extract the sequence rules. Through these rules, we predict sequences of user visits to various web parts.

The method was derived from the **association rules** and is an example of a method that takes into account the peculiarities of the Internet. The difference between association and sequence analysis is that in the case of *Association Rule Analysis*, we do not analyze sequences, but **transactions**, i.e. the analysis does not include the time variable.

## 🕮 6.1.4

Based on the model determination, a **multinominal logit model** was chosen for our example of a university log file.

After creating artificial variables and defining the model, the data file is ready to analyze and estimate the parameters needed to estimate the logits and probabilities of access to the individual web pages of the web portal.

## 🕮 6.1.5

Based on the contingency results for each variable, we decided to examine only the *home, announcement, study, and faculty* websites. On this website, it was interesting to examine the different approaches of students, i.e. employees during all days of the week.

To examine only the required data, it was necessary to apply a filter to the examined data file. In our case, we focused on examining the behavior of employees on the university web portal.

We also filtered the time period of the day, between 7 am and 10 pm due to the fact that the university is physically closed at night and it is not possible for someone from inside the network to access the researched web portal.

Sunday, when the university is physically closed all day, was also excluded from the analysis.

## 6.2 Association Analysis

### 🕮 6.2.1

Association rules are one of the most popular methods of in-depth analysis (data mining). They are successfully applied in the analysis of the shopping cart, financial data, etc. Association rules allow us to get a picture of the relationships between elements in a given data set, while the results are very easy to interpret. The rules represent the construction of IF THEN, which can be found in all programming languages and can be expressed in natural language. Agraval popularized the rules of the association in the early 1990s in connection with the analysis of the shopping cart. Association rules are among the symbolic methods of machine learning and, like decision trees, are tailored to qualitative/categorical data. Quantitative/numerical data require discretization - replacing numerical values with intervals of values, which are then treated as categorical variables.

### 📝 6.2.2

To which programming structure could we compare the generated association rules?

- IF THEN
- FOR
- WHILE
- SWITCH CASE

### 🕮 6.2.3

An example of an association rule might be a single purchase, with all purchases over a period of time representing an entire set. The result of the analysis are rules of the form IF condition THEN consequence (IF body THEN head). The rules are determined by the frequency with which the condition and consequence occur in the data.

Example of an association rule from a shopping cart analysis:

*{bread, cheese} => {butter}, confidence = 57%, support = 21%*

The interpretation of the rule is as follows: Customers who buy bread and cheese in one purchase are 57% more likely to buy butter, with 21% of purchases containing bread, cheese and butter.

## 6.2.4

Example of an association rule from a shopping cart analysis:

*{computer, keyboard} => {mouse}, confidence = 68%, support = 30%*

How could we interpret the following rule?

- Customers who buy a computer and keyboard for one purchase are also 68% likely to buy a mouse, with 30% of purchases including a computer, keyboard, and mouse.
- Customers who buy a computer and keyboard for one purchase are also 68% likely to buy a mouse, with 30% of purchases containing only a mouse.
- Customers who buy a computer and a keyboard for one purchase are unlikely to buy a mouse, with 30% of purchases containing only a mouse.

## 6.2.5

The aim of the association analysis is to find all rules whose rates are greater than or equal to the specified support (minimum *support*) and reliability (minimum *confidence*). These two measures indicate the frequency of occurrence of the rule in the database (*support*) and the strength of the rule (*confidence*). A *K*-itemset with support greater than a specified minimum is called a frequented/large/frequently recurring itemsets. The whole process of obtaining rules, defined later, consists of two steps, where all frequented itemsets are first obtained, from which the rules themselves are then generated.

## 6.2.6

What does the support rule represent in the association analysis?

- frequency of occurrence of the rule in the database
- the strength of the rule in the database

## 6.2.7

What does the confidence of the rule represent in the association analysis?

- frequency of occurrence of the rule in the database
- the strength of the rule in the database

## 6.2.8

**Apriori** is an algorithm for frequent itemset mining and association rule learning over relational databases. It proceeds by identifying the frequent individual items in the database and extending them to larger and larger item sets as long as those item sets appear sufficiently often in the database. The frequent itemsets determined by Apriori can be used to determine association rules which highlight general trends in the database: this has applications in domains such as market basket analysis.

## 6.2.9

What is the name of the algorithm used to generate the association rules?

- apriori algorithm
- FP-growth algorithm
- AdaBoost
- ID3 algorithm

## 6.2.10

Apriori algorithm is the perfect algorithm to start with association analysis as it is not just easy to understand and interpret but also to implement. Python has many libraries for apriori implementation. One can also implement the algorithm from scratch. But as there are many solutions, we will use the library **mlxtend**. This library has a beautiful implementation of apriori and it also allows to extract of association rules from the result.

```
from mlxtend.frequent_patterns import apriori,
association_rules
```

## 6.2.11

Insert the missing library functions to get the apriori algorithm working in Python:

```
from mlxtend.frequent_patterns import _____, _____
```

## 6.2.12

Apriori module from mlxtend library provides fast and efficient apriori implementation.

```
apriori(df, min_support=0.5, use_colnames=False, max_len=None,
verbose=0, low_memory=False)
```

Parameters

- **df** : One-Hot-Encoded DataFrame or DataFrame that has 0 and 1 or True and False as values
- **min_support**: Floating point value between 0 and 1 that indicates the minimum support required for an itemset to be selected.
- # of observation with item / total observation# of observation with the item / total observation
- **use_colmnames**: This allows to preserve column names for itemset making it more readable.
- **max_len**: Max length of itemset generated. If not set, all possible lengths are evaluated.
- **verbose**: Shows the number of iterations if >= 1 and low_memory is True. If =1 and low_memory is False, shows the number of combinations.
- **low_memory**: If True, uses an iterator to search for combinations above min_support. Note that while low_memory=True should only be used for large datasets if memory resources are limited because this implementation is approx. 3–6x slower than the default.

The model will generate frequent itemsets. The output is a data frame with support for each itemset.

## 6.2.13

Let's look another time at the support, confidence and lift that are associated with the association analysis.

**Support** is the relative frequency that the rules show up. In many instances, you may want to look for high support in order to make sure it is a useful relationship.

However, there may be instances where low support is useful if you are trying to find “hidden” relationships.

Suppose there a set of transactions with page1 --> page2. So, support for page1 will be defined by n(page1) / n (total transactions).

## 📝 6.2.14

Is it necessary to look also at items with low support?

- Yes
- No

## 🕮 6.2.15

**Confidence** is a measure of the reliability of the rule. Confidence of 0.5 would mean that in 50% of the cases where one page occurs in the session, then the session also included another page. For product recommendation (e.g. e-shop), a 50% confidence may be perfectly acceptable but in a medical situation, this level may not be high enough.

Suppose there is a set of transactions with page1 --> page2. Confidence on the other hand is defined as, n (page1 & page2) / n(page1). So, confidence tells us the strength of the association and support tells us the relevance of the rule. Because we don’t want to include rules about pages that are seldom visited, or in other words, have low support.

## 📝 6.2.16

What is confidence?

- measure of the reliability of the rule
- measure of the support of the rule

## 🕮 6.2.17

**Lift** is the ratio of the observed support to that expected if the two rules were independent. The basic rule of thumb is that a lift value close to 1 means the rules were completely independent. Lift values > 1 are generally more “interesting” and could be indicative of a useful rule pattern. It can be said that the rules with higher

lift mean that they are more often together in the session. On the other hand, 0 < lift < 1 means that the pair is more often not together in the session. The lift value of an association rule can be understood also as the ratio of the confidence of the rule and the expected confidence of the rule.

Lift is Confidence/Support. The higher the lift, the more the significance of applying the Apriori algorithm to determine the rule.

## 6.2.18

What is the interval of the lift of association rule?

- 0 - ∞
- -∞ - ∞
- -∞ - 0
- 1 - ∞
- -1 - 1

## 6.2.19

**F-P Growth** (or frequent-pattern growth) algorithm is another popular technique in association analysis. It produces the same results as the Apriori algorithm but is computationally faster due to a mathematically different technique (divide and conquer).

F-P Growth follows a two-step data preprocessing approach:

1. First, it counts the number of occurrences of each item in the transactional dataset.
2. Then, it creates a search tree structure using the transactions.

## 6.2.20

What is the F-P Growth algorithm based on?

- divide and conquer algorithm
- apriori algorithm
- monte-carlo algorithm

## 📖 6.2.21

Unlike the Apriori algorithm, F-P Growth sorts items within each transaction by their frequency from largest to smallest before inserting them into a tree. This is where it has a substantial computational advantage over the Apriori algorithm since it does the frequency sorting early on. Items that don't meet minimum support (frequency) requirements (that we can set) are discarded from the tree.

Another advantage is that frequent itemsets that repeat will have the same path (unlike the Apriori algorithm, where each itemset has a unique path).

## 📝 6.2.22

What is the computational advantage of F-P Growth over the Apriori algorithm?

- it sorts items within each transaction by it's frequency
- it uses binary trees to find the rules
- it sorts based on alphabetical order
- it uses bubble sort to find rules

## 📖 6.2.23

Once again we can use the **mlxtend** library to obtain F-P Growth results.

First, we import the F-P growth algorithm function from the library.

```
from mlxtend.frequent_patterns import fpgrowth
```

Then we apply the algorithm to our data to extract the itemsets that have a minimum support value of 0.01 (this parameter can be tuned on a case-by-case basis).

```
frequent_itemsets_fp=fpgrowth(data, min_support=0.01,
use_colnames=True)
```

## 📝 6.2.24

Fill in the code with the correct library to use F-P Growth.

```
from _____.frequent_patterns import _____
```

### 🕮 6.2.25

What you can observe if you examine the results of the Apriori algorithm and F-P Growth algorithm is, that regardless of the technique you used, you arrived at the identical itemsets and support values. The only difference is the order in which they appear. You should notice that the output from F-P Growth appears in descending orders, hence the proof of what we mentioned in the theoretical part about this algorithm.

### 🕮 6.2.26

In the final step, we will find the association rules for the frequent itemsets which were calculated in the previous microlection.

First, we import the required function from the page to determine the association rules for a given dataset using some set of parameters.

```
from mlxtend.frequent_patterns import association_rules
```

Then we apply it to the two frequent item datasets which we created in the previous microlections to compare the results of the different approaches to the frequent itemset generation.

```
rules_ap = association_rules(frequent_itemsets_ap,
metric="confidence", min_threshold=0.8)
rules_fp = association_rules(frequent_itemsets_fp,
metric="confidence", min_threshold=0.8)
```

The “metric” and “min_threshold” parameters can be tuned on a case-by-case basis, depending on the business problem requirements.

The results will show us that both algorithms found identical association rules with the same coefficients, just presented in a different order.

## 6.3 Market Basket Analysis (example)

### 🕮 6.3.1

Let's summarize what is the most important about the association analysis. Association analysis is relatively light on the math concepts and easy to explain to non-technical people. It is a good start for certain cases of data exploration and can point the way for a deeper dive into the data using other approaches. Market basket

analysis is just one application of association analysis and will serve as an illustrative example.

**Support** is the relative frequency that the rules show up. In many instances, you may want to look for high support in order to make sure it is a useful relationship. However, there may be instances where low support is useful if you are trying to find “hidden” relationships.

**Confidence** is a measure of the reliability of the rule. Confidence of 0.5 would mean that in 50% of the cases where two items were purchased, the purchase also included another item. For product recommendation, a 50% confidence may be perfectly acceptable but in a medical situation, this level may not be high enough.

**Lift** is the ratio of the observed support to that expected if the two rules were independent. The basic rule of thumb is that a lift value close to 1 means the rules were completely independent. Lift values > 1 are generally more “interesting” and could be indicative of a useful rule pattern.

One final note, related to the data. This analysis requires that all the data for a transaction be included in 1 row and the items should be 1-hot encoded.

## 6.3.2

Let's start with our example where we will look at the market basket analysis and compare the results of apriori and F-P Growth algorithms. Get our pandas and MLxtend code imported and read the data from csv and show the first five rows:

```
import _____ as pd
from _____.frequent_patterns import apriori
from mlxtend.frequent_patterns import _____
from _____.frequent_patterns import association_rules

df = pd._____('basket.csv')
df._____()
```

## 6.3.3

The results of the previous code can be seen here:

| | Invoice | Description | Quantity | InvoiceDate | UnitPrice | CustomerID | Country |
|---|---|---|---|---|---|---|---|
| 0 | 536389 | CHRISTMAS LIGHTS 10 REINDEER | 6 | 1.12.2020 10:03 | 8,5 | 12431.0 | Czechia |
| 1 | 536389 | VINTAGE UNION JACK CUSHION COVER | 8 | 1.12.2020 10:03 | 4,95 | 12431.0 | Czechia |
| 2 | 536389 | VINTAGE HEADS AND TAILS CARD GAME | 12 | 1.12.2020 10:03 | 1,25 | 12431.0 | Czechia |
| 3 | 536389 | SET OF 3 COLOURED FLYING DUCKS | 6 | 1.12.2020 10:03 | 5,45 | 12431.0 | Czechia |
| 4 | 536389 | SET OF 3 GOLD FLYING DUCKS | 4 | 1.12.2020 10:03 | 6,35 | 12431.0 | Czechia |

First, we need to do a little data preprocessing. There are some spaces in the descriptions that could be removed and also we don't need the rows without the invoice number. On the other hand, there are some rows that contain negative quantity as this is the credit transaction. We can see that it contains a letter C in the invoice, so we can use it to remove those rows.

## 6.3.4

Remove the spaces in the *Description* column. Drop the rows that don't have invoice numbers and remove the credit transactions (those with invoice numbers containing C).

```
df['Description'] = df['_____'].str.strip()
df.dropna(axis=_____, subset=['Invoice'], inplace=True)
df['Invoice'] = df['_____'].astype('str')
df = df[~df['Invoice'].str.contains('_____')]
```

## 6.3.5

After the cleanup, we need to consolidate the items into 1 transaction per row with each product 1 hot encoded. For the sake of keeping the data set small, we will be only looking at sales for Slovakia and compare these results to sales from Czechia. Further country comparisons would be interesting to investigate.

## 6.3.6

Select the data for Slovakia and group it by Invoice and Description. The index will be set based on the Invoice number as this number is unique for each transaction.

```
basket = (df[df['_____'] =="_____"]
          .groupby(['_____', '_____'])['Quantity']
          .sum().unstack().reset_index().fillna(0)
          .set_index('_____'))
```

## 📖 6.3.7

Let's look at some of the rows.

| Description | 10 COLOUR SPACEBOY PEN | 12 COLOURED PARTY BALLOONS | 12 PENCIL SMALL TUBE WOODLAND | 12 PENCILS SMALL TUBE SKULL | 12 PENCILS TALL TUBE POSY | 12 PENCILS TALL TUBE RED RETROSPOT | 12 PENCILS TALL TUBE SKULLS | 12 PENCILS TALL TUBE WOODLAND | 12 RED ROSE PEG PLACE SETTINGS | 15CM CHRISTMAS GLASS BALL 20 LIGHTS | ... | WRAP MAGIC FOREST | WRAP PAISLEY PARK | WRAP PINK FAIRY CAKES |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Invoice | | | | | | | | | | | | | | |
| 536858 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... | 0.0 | 0.0 | 0.0 |
| 539488 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... | 0.0 | 0.0 | 0.0 |
| 541518 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... | 0.0 | 0.0 | 0.0 |
| 541569 | 0.0 | 0.0 | 24.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... | 0.0 | 0.0 | 0.0 |
| 542586 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... | 0.0 | 0.0 | 0.0 |

5 rows × 987 columns

There are a lot of zeros in the data but we also need to make sure any positive values are converted to a 1 and anything less the 0 is set to 0. This step will complete the one-hot encoding of the data and remove the postage column (since that charge is not one we wish to explore).

## 📝 6.3.8

Create the method one_hot_encode that will convert all positive values to 1 and anything less than 0 to 0.

```
def _____(n):
    if _____:
        return 0
    if _____:
        return 1
```

## 📝 6.3.9

Apply the created method one_hot_encode to our data saved in the basket. Drop the POSTAGE column.

```
basket_sets = _____.applymap(_____)
basket_sets.drop('_____', inplace=True, axis=_____)
```

## 📖 6.3.10

Now that the data is structured properly, we can generate frequent itemsets that have the support of at least 7% (this number was chosen so that we could get enough useful examples).

## 6.3.11

Generate the frequent itemsets with the minimal support 0.07 using the apriori algorithm.

```
frequent_itemsets = _____(basket_sets, min_support=_____,
use_colnames=True)
```

## 6.3.12

We got frequent itemsets generated using the apriori algorithm. Let's look at the first rows using head().

```
frequent_itemsets.head()
```

| | support | itemsets |
|---|---|---|
| 0 | 0.092593 | (5 HOOK HANGER RED MAGIC TOADSTOOL) |
| 1 | 0.092593 | (6 RIBBONS RUSTIC CHARM) |
| 2 | 0.074074 | (ALARM CLOCK BAKELIKE GREEN) |
| 3 | 0.092593 | (ASSORTED BOTTLE TOP MAGNETS) |
| 4 | 0.092593 | (ASSORTED COLOUR BIRD ORNAMENT) |

## 6.3.13

Next, let's look at the F-P Growth algorithm and generate the frequent itemsets again for the same input and with the same min support.

```
frequent_itemsets_fp = _____(basket_sets, _____,
use_colnames=True)
```

## 6.3.14

Now let's look at the first five rows of the itemsets generated by the F-P Growth.

```
frequent_itemsets_fp.head()
```

| | support | itemsets |
|---|---|---|
| 0 | 0.314815 | (ROUND SNACK BOXES SET OF4 WOODLAND) |
| 1 | 0.314815 | (PLASTERS IN TIN WOODLAND ANIMALS) |
| 2 | 0.203704 | (RED RETROSPOT MINI CASES) |
| 3 | 0.148148 | (RED TOADSTOOL LED NIGHT LIGHT) |
| 4 | 0.092593 | (PINK POLKADOT BOWL) |

As was mentioned in the theoretical lessons, one of the advantages of the F-P Growth is that it orders the itemsets based on the support. This offers us already some insight into the obtained knowledge. We can see that the highest support was identified for snack boxes and plasters. That is a very strange combination but let's see how the rules will work out.

## 📖 6.3.15

The last step is to generate the rules based on the specified support, confidence or lift. Let us look at the rules based on the lift, where we would like to see the rules that have the minimum lift of 1. This will generate rules that create itemset pairs that are more often together than separate in the basket.

## 📝 6.3.16

Identify the association rules based on the minimum lift of 1.

```
rules = _____(frequent_itemsets, metric="_____",
min_threshold=_____)
```

## 📖 6.3.17

Let's have a look at the identified rules using head().

```
rules.head()
```

| | antecedents | consequents | antecedent support | consequent support | support | confidence | lift | leverage | conviction |
|---|---|---|---|---|---|---|---|---|---|
| 0 | (5 HOOK HANGER RED MAGIC TOADSTOOL) | (PLASTERS IN TIN SPACEBOY) | 0.092593 | 0.333333 | 0.074074 | 0.800000 | 2.400000 | 0.043210 | 3.333333 |
| 1 | (PLASTERS IN TIN SPACEBOY) | (5 HOOK HANGER RED MAGIC TOADSTOOL) | 0.333333 | 0.092593 | 0.074074 | 0.222222 | 2.400000 | 0.043210 | 1.166667 |
| 2 | (5 HOOK HANGER RED MAGIC TOADSTOOL) | (PLASTERS IN TIN WOODLAND ANIMALS) | 0.092593 | 0.314815 | 0.074074 | 0.800000 | 2.541176 | 0.044925 | 3.425926 |
| 3 | (PLASTERS IN TIN WOODLAND ANIMALS) | (5 HOOK HANGER RED MAGIC TOADSTOOL) | 0.314815 | 0.092593 | 0.074074 | 0.235294 | 2.541176 | 0.044925 | 1.186610 |
| 4 | (6 RIBBONS RUSTIC CHARM) | (ASSORTED COLOUR BIRD ORNAMENT) | 0.092593 | 0.092593 | 0.074074 | 0.800000 | 8.640000 | 0.065501 | 4.537037 |

We can see the obtained rules with support, confidence and lift values. At a better look at the rules (using the command *info()*), we have identified 2830 rules for the Slovak customers. It is time to generate the rules from the F-P Growth algorithm.

## 6.3.18

Identify the association rules based on the minimum lift of 1 and obtained from the frequent itemsets identified based on F-P Growth in the previous assignment.

```
rules_fp = _____(frequent_itemsets_fp, _____, _____)
```

## 6.3.19

Once again let us look at the head of the obtained rules.

```
rules_fp.head()
```

| | antecedents | consequents | antecedent support | consequent support | support | confidence | lift | leverage | conviction |
|---|---|---|---|---|---|---|---|---|---|
| 0 | (PLASTERS IN TIN SPACEBOY) | (ROUND SNACK BOXES SET OF4 WOODLAND) | 0.333333 | 0.314815 | 0.166667 | 0.500000 | 1.588235 | 0.061728 | 1.370370 |
| 1 | (ROUND SNACK BOXES SET OF4 WOODLAND) | (PLASTERS IN TIN SPACEBOY) | 0.314815 | 0.333333 | 0.166667 | 0.529412 | 1.588235 | 0.061728 | 1.416667 |
| 2 | (PLASTERS IN TIN WOODLAND ANIMALS) | (ROUND SNACK BOXES SET OF4 WOODLAND) | 0.314815 | 0.314815 | 0.166667 | 0.529412 | 1.681661 | 0.067558 | 1.456019 |
| 3 | (ROUND SNACK BOXES SET OF4 WOODLAND) | (PLASTERS IN TIN WOODLAND ANIMALS) | 0.314815 | 0.314815 | 0.166667 | 0.529412 | 1.681661 | 0.067558 | 1.456019 |
| 4 | (PLASTERS IN TIN SPACEBOY) | (PLASTERS IN TIN WOODLAND ANIMALS) | 0.333333 | 0.314815 | 0.240741 | 0.722222 | 2.294118 | 0.135802 | 2.466667 |

Now we can see some differences but it is only in order. We have identified the same number of rules as using the apriori algorithm. The order stayed the same as in the frequent itemsets. The last step is the most important and that is to extract the knowledge from the obtained results.

## 6.3.20

The tricky part is figuring out what this tells us. For instance, we can see that there are quite a few rules with a high lift value which means that it occurs more frequently than would be expected given the number of transaction and product combinations. We can also see several rules where the confidence is high as well. This part of the analysis is where the domain knowledge should be implemented. This means to get an expert of the examined domain or administrator of the web portal to better interpret the results.

## 6.3.21

We will now focus on the most important rules obtained from the data. Fill in the code to look for a large lift, high confidence and support higher than 0.05.

```
rules[ (rules['_____'] >= 10) &
       (rules['_____'] >= 0.8) &
       (rules['_____'] >= 0.09)]
```

## 6.3.22

The results show us 10 rules that could be the most interesting for the final interpretations.

| | antecedents | consequents | antecedent support | consequent support | support | confidence | lift | leverage | conviction |
|---|---|---|---|---|---|---|---|---|---|
| 1042 | (PLASTERS IN TIN SPACEBOY, RED TOADSTOOL LED N... | (RED RETROSPOT SHOPPER BAG) | 0.092593 | 0.092593 | 0.092593 | 1.0 | 10.8 | 0.084019 | inf |
| 1047 | (RED RETROSPOT SHOPPER BAG) | (PLASTERS IN TIN SPACEBOY, RED TOADSTOOL LED N... | 0.092593 | 0.092593 | 0.092593 | 1.0 | 10.8 | 0.084019 | inf |
| 1837 | (PLASTERS IN TIN SPACEBOY, RED TOADSTOOL LED N... | (RED RETROSPOT SHOPPER BAG) | 0.092593 | 0.092593 | 0.092593 | 1.0 | 10.8 | 0.084019 | inf |
| 1840 | (PLASTERS IN TIN SPACEBOY, RED TOADSTOOL LED N... | (RED RETROSPOT SHOPPER BAG, ROUND SNACK BOXES ... | 0.092593 | 0.092593 | 0.092593 | 1.0 | 10.8 | 0.084019 | inf |
| 1845 | (RED RETROSPOT SHOPPER BAG, ROUND SNACK BOXES ... | (PLASTERS IN TIN SPACEBOY, RED TOADSTOOL LED N... | 0.092593 | 0.092593 | 0.092593 | 1.0 | 10.8 | 0.084019 | inf |
| 1848 | (RED RETROSPOT SHOPPER BAG) | (PLASTERS IN TIN SPACEBOY, RED TOADSTOOL LED N... | 0.092593 | 0.092593 | 0.092593 | 1.0 | 10.8 | 0.084019 | inf |
| 2120 | (WATERING CAN PINK BUNNY, RED TOADSTOOL LED NI... | (WATERING CAN BLUE ELEPHANT, ROUND SNACK BOXES... | 0.092593 | 0.092593 | 0.092593 | 1.0 | 10.8 | 0.084019 | inf |
| 2122 | (WATERING CAN PINK BUNNY, ROUND SNACK BOXES SE... | (RED TOADSTOOL LED NIGHT LIGHT, WATERING CAN B... | 0.092593 | 0.092593 | 0.092593 | 1.0 | 10.8 | 0.084019 | inf |
| 2123 | (RED TOADSTOOL LED NIGHT LIGHT, WATERING CAN B... | (WATERING CAN PINK BUNNY, ROUND SNACK BOXES SE... | 0.092593 | 0.092593 | 0.092593 | 1.0 | 10.8 | 0.084019 | inf |
| 2125 | (WATERING CAN BLUE ELEPHANT, ROUND SNACK BOXES... | (WATERING CAN PINK BUNNY, RED TOADSTOOL LED NI... | 0.092593 | 0.092593 | 0.092593 | 1.0 | 10.8 | 0.084019 | inf |

We can see that in most of the rules is a shopper bag and it is in combination with various plasters. This is an interesting find as from the frequent itemsets analysis we saw the high support of plasters but not the bag. But we can also see that many customers bought also the snack boxes together with plasters. Of course, the domain expert would find more interesting results from this data and maybe raise some questions.

Now we can repeat the same experiment for another country - Czechia and compare the findings.

## 6.3.23

Fill in the code and repeat the experiment for the country Czechia.

```
#import the neccessary libraries
import pandas as _____
```

```
from _____ #import the apriori algorithm
from _____ #import association rules

dfBS = _____('EshopSKCZ.csv', sep=';') #read the csv using
pandas
```

### 6.3.24

Fill in the code and repeat the experiment for the country Czechia.

```
dfBS['Description'] = _____ #remove the spaces
dfBS._____(_____, subset=['Invoice'], _____) #drop empty rows
dfBS['Invoice'] = dfBS['Invoice'].astype('str')
dfBS = dfBS[~dfBS['Invoice'].str.contains('C')]
```

### 6.3.25

Fill in the code and repeat the experiment for the country Czechia.

```
#filter the country Czechia
basketCZ = (dfBS[_____]
          .groupby(['Invoice', 'Description'])['Quantity']
          .sum().unstack().reset_index().fillna(0)
          .set_index('Invoice'))
```

### 6.3.26

Fill in the code and repeat the experiment for the country Czechia.

```
basket_setsCZ = _____ #use the one_hot_encode function
basket_setsCZ.drop('POSTAGE', inplace=True, axis=1)
```

### 6.3.27

Fill in the code and repeat the experiment for the country Czechia.

```
# use apriori algorithm to extract the frequent itemsets with
support >= 0.07
frequent_itemsetsCZ = _____(basket_setsCZ, _____,
use_colnames=True)
```

```
# generate the association rules based on min lift = 1
rulesCZ = _____
```

## 📖 6.3.28

The results have shown that we have identified only 78 rules for Czech customers. That is a big difference. This could be because of a big difference in the input data. So we can examine whether we compare two various sizes of samples. Using the info() function we can see that there were 1967 entries for Slovakia and 1210 entries for Czechia. This does not seem like such a great difference that would generate such opposing results.

This could suggest that the customers from Czechia are buying more various items and there are little patterns in their market basket. We could lower the lift min threshold to obtain more rules but let us stick with this setting as we wanted to compare similar results.

## 📖 6.3.29

The results of the association analysis of Czechia customers generated 78 rules. When we wanted to look at the most interesting rules we had to make a few changes to our conditions:

```
fin_rulesCZ = rulesCZ[ (rulesCZ['lift'] >= 10) &
       (rulesCZ['confidence'] >= 0.8) &
       (rulesCZ['support'] >= 0.05)]
```

Higher support and lift would not produce any rules that are the reason we had to use lower values in contrast to the Slovakian rules. We obtained 4 rules now:

| | antecedents | consequents | antecedent support | consequent support | support | confidence | lift | leverage | conviction |
|---|---|---|---|---|---|---|---|---|---|
| 68 | (SPACEBOY LUNCH BOX, REGENCY CAKESTAND 3 TIER) | (DOLLY GIRL LUNCH BOX, ROSES REGENCY TEACUP AN... | 0.084746 | 0.084746 | 0.084746 | 1.0 | 11.8 | 0.077564 | inf |
| 70 | (SPACEBOY LUNCH BOX, ROSES REGENCY TEACUP AND ... | (REGENCY CAKESTAND 3 TIER, DOLLY GIRL LUNCH BOX) | 0.084746 | 0.084746 | 0.084746 | 1.0 | 11.8 | 0.077564 | inf |
| 71 | (REGENCY CAKESTAND 3 TIER, DOLLY GIRL LUNCH BOX) | (SPACEBOY LUNCH BOX, ROSES REGENCY TEACUP AND ... | 0.084746 | 0.084746 | 0.084746 | 1.0 | 11.8 | 0.077564 | inf |
| 73 | (DOLLY GIRL LUNCH BOX, ROSES REGENCY TEACUP AN... | (SPACEBOY LUNCH BOX, REGENCY CAKESTAND 3 TIER) | 0.084746 | 0.084746 | 0.084746 | 1.0 | 11.8 | 0.077564 | inf |

The rules tell us that Czech customers were more interested in the lunch boxes and teacups.

### 🕮 6.3.30

We can wrap this lesson once again. The really nice aspect of association analysis is that it is easy to run and relatively easy to interpret. If you did not have access to MLxtend and this association analysis, it would be exceedingly difficult to find these patterns using basic Excel analysis. With python and MLxtend, the analysis process is relatively straightforward and since you are in python, you have access to all the additional visualization techniques and data analysis tools in the python ecosystem.

Our results showed different behaviour for two different groups of customers and also showed us that for analysis we need to tweak the parameters of support, confidence and lift to obtain various results. Do not forget that we focused only on lift higher than 1 and there are a lot of other findings when we would look at lift smaller than 1 or even 0.

## 6.4 Cluster analysis

### 🕮 6.4.1

Clustering is basically a type of *unsupervised learning method*. An unsupervised learning method is a method in which we draw references from datasets consisting of input data without labelled responses. Generally, it is used as a process to find meaningful structure, explanatory underlying processes, generative features, and groupings inherent in a set of examples.

**Clustering** is the task of dividing the population or data points into a number of groups such that data points in the same groups are more similar to other data points in the same group and dissimilar to the data points in other groups. It is basically a collection of objects on the basis of similarity and dissimilarity between them.

### 📝 6.4.2

What type of method is clustering?

- unsupervised learning method
- supervised learning method
- learning method with a teacher

## 🕮 6.4.3

Clustering is very much important as it determines the intrinsic grouping among the unlabeled data present. There are no criteria for good clustering. It depends on the data analyst, what is the criteria they may use which satisfy their need. For instance, we could be interested in finding representatives for homogeneous groups (data reduction), in finding “natural clusters” and describe their unknown properties (“natural” data types), in finding useful and suitable groupings (“useful” data classes) or in finding unusual data objects (outlier detection). This algorithm must make some assumptions that constitute the similarity of points and each assumption make different and equally valid clusters.

## 🕮 6.4.4

In practice, clustering helps identify two qualities of data:

- Meaningfulness - clusters expand domain knowledge, e.g. market basket analysis can show clusters of most bought items together in groups (clusters), similar to association rules but clusters are based on visualisation.
- Usefulness - clusters, on the other hand, serve as an intermediate step in a data pipeline, e.g. businesses use clustering for customer segmentation. The clustering results segment customers into groups with similar purchase histories, which businesses can then use to create targeted advertising campaigns.

## 🕮 6.4.5

We can see some differences between clustering and cluster analysis.

Clustering is generally, a group of abstract objects made into classes of similar objects. The cluster of data objects is treated like one group. While doing cluster analysis, we first partition the set of data into groups. That is based on data similarity and then assign the labels to the groups. The main advantage of over-classification is that it is adaptable to changes. And helps single out useful features that distinguish different groups.

On the other hand cluster analysis in data mining means finding such groups of objects where the objects in a group will be like one another. And different from the objects in other groups.

## 📖 6.4.6

But why is there a need for clustering in data mining? Let's look at the requirements and characteristics that help us better understand the usage of clusters.

**Scalability** - in data mining as we mentioned earlier we work with large data matrices and log files. The amount of data contains also a big amount of information. For that reason, highly scalable clustering algorithms are needed to work with this amount of data.

## 📝 6.4.7

What does scalability mean in clustering?

- highly scalable clustering algorithms
- big data methods
- low cost solutions to data analysis

## 📖 6.4.8

As you have seen the log files contain not only a lot of records but also contain many variables. The algorithms used in clustering should be capable to be applied to any kind of data such as interval-based (numerical) data, categorical, and binary data. The clustering algorithm should be also capable of detecting clusters of arbitrary shape. They should not be bounded by only distance measures. That tends to find a spherical cluster of small sizes.

## 📝 6.4.9

Select other requirements for the clustering algorithms.

- Ability to deal with different kinds of attributes
- Discovery of clusters with attribute shape
- Ability to deal with one type of attributes
- Discovery of clusters with only distance measures

### 📖 6.4.10

Let's focus on the last requirements of the clustering algorithms. The clustering algorithm should not only be able to handle low-dimensional data. Although, they need to handle also the high dimensional space. On the other hand, log files contain noisy, missing or erroneous data. Some algorithms are sensitive to such data and may lead to poor quality clusters. So the algorithm must know how to deal with this type of data. The last important requirement is that the clustering results should be interpretable, comprehensible, and usable.

### 📝 6.4.11

Select other requirements for the clustering algorithms.

- High dimensionality
- Ability to deal with noisy data
- Interpretability
- Low dimensionality
- Smoothness

### 📖 6.4.12

Data Mining clustering methods can be divided into the following categories:

- hierarchical
- partitioning
- density-based
- model-based
- grid-based
- constraint-based

### 📖 6.4.13

## Hierarchical method

The hierarchical method creates a hierarchical decomposition of the given set of data objects. We can classify methods on the basis of how the hierarchical decomposition is formed. There are two approaches:

- Agglomerative approach
- Divisive approach

Agglomerative approach

This approach is also known as the bottom-up approach. In this, we start with each object forming a separate group. It keeps on merging the objects or groups that are close to one another. It keeps on doing so until all of the groups are merged into one or until the termination condition holds.

Divisive approach

This approach is also known as the top-down approach. In this, we start with all of the objects in the same cluster. Then, in the continuous iteration, a cluster is split up into smaller clusters. Also, it is down until each object in one cluster or the termination condition holds. Hence, this method is rigid, i.e., once a merging or splitting is done, it can never be undone.

## 🕮 6.4.14

These methods produce a tree-based hierarchy of points called a dendrogram. Similar to partitional clustering, in hierarchical clustering, the number of clusters (*k*) is often predetermined by the user. Clusters are assigned by cutting the dendrogram at a specified depth that results in *k* groups of smaller dendrograms.

## 🕮 6.4.15

Unlike many partitional clustering techniques, hierarchical clustering is a deterministic process, meaning cluster assignments won't change when you run an algorithm twice on the same input data.

The strengths of hierarchical clustering methods include the following:

- They often reveal the finer details about the relationships between data objects.
- They provide an interpretable dendrogram.

The weaknesses of hierarchical clustering methods include the following:

- They're computationally expensive with respect to algorithm complexity.
- They're sensitive to noise and outliers.

## 6.4.16

Which one of the hierarchical method approaches is known as bottom-up?

- Agglomerative approach
- Divisive approach

### 📝 6.4.17

Which one of the hierarchical method approaches is known as top-bottom?

- Divisive approach
- Agglomerative approach

### 🕮 6.4.18

## Partitioning method

Suppose we are given a database of *n* objects. And the partitioning method constructs the *k* partition of data. Each partition will represent a cluster and $k \leq n$. It means that it will classify the data into *k* groups. That must need to satisfy the following requirements:

- Each group contains at least one object.
- Each object must belong to exactly one group.

### 🕮 6.4.19

Two examples of partitional clustering algorithms are ***k*-means** and *k*-medoids. These algorithms are both non-deterministic, meaning they could produce different results from two separate runs even if the runs were based on the same input.

Partitional clustering methods have several strengths:

- They work well when clusters have a spherical shape.
- They’re scalable with respect to algorithm complexity.

They also have several weaknesses:

- They’re not well suited for clusters with complex shapes and different sizes.
- They break down when used with clusters of different densities.

### 📝 6.4.20

What is the base of the partitioning method?

- it will classifies the data into specified number of groups
- it is based on divide and conquer

### 🕮 6.4.21

## Density-based method

It is based on the notion of density where the idea is to continue growing the given cluster until it is exceeding as long as the density in the neighbourhood threshold. For each data point within a given cluster, the radius of a given cluster has to contain at least a number of points.

Unlike the other clustering categories, this approach doesn't require the user to specify the number of clusters. Instead, there is a distance-based parameter that acts as a tunable threshold. This threshold determines how close points must be to be considered a cluster member.

### 🕮 6.4.22

Examples of density-based clustering algorithms include Density-Based Spatial Clustering of Applications with Noise or DBSCAN, and Ordering Points To Identify the Clustering Structure or OPTICS.

The strengths of density-based clustering methods include the following:

- They excel at identifying clusters of nonspherical shapes.
- They're resistant to outliers.

The weaknesses of density-based clustering methods include the following:

- They aren't well suited for clustering in high-dimensional spaces.
- They have trouble identifying clusters of varying densities.

📖 6.4.23

## Grid-based method

In the case of this method, the objects form a grid together. The object space is quantized into a finite number of cells that form a grid structure. The major advantage of this method is its fast processing time. It is dependent only on the number of cells in each dimension in the quantized space.

📖 6.4.24

## Model-based method

This method is specified by a model that is hypothesized for each cluster to find the best fit of data for a given model. Also, this method locates the clusters by clustering the density function. Thus, it reflects the spatial distribution of the data points. This method also provides a way to determine the number of clusters. That was based on standard statistics, taking outlier or noise into account. It, therefore, yields robust clustering methods.

📖 6.4.25

## Constraint-based method

The clustering is performed by the incorporation of user or application-oriented constraints. A constraint refers to the user expectation. Constraints provide us with an interactive way of communication with the clustering process. Constraints can be specified by the user or the application need.

📝 6.4.26

Based on what you have learned, is this example of dividing students into different registration groups, by the last name, appliable to solve using clustering?

- No
- Yes

# 6.5 K-means clustering (example)

## 🕮 6.5.1

Let's focus on a practical application of clustering. There's a robust implementation of ***k*-means** clustering in Python from the machine learning package **scikit-learn**. We will look at how to write a practical implementation of the *k*-means algorithm using the scikit-learn version of the algorithm.

## 🕮 6.5.2

Let's look at how the algorithm works. To process the learning data, the k-means algorithm in data mining starts with the first group of randomly selected centroids, which are used as the beginning points for every cluster, and then performs iterative (repetitive) calculations to optimize the positions of the centroids

It halts creating and optimizing clusters when either:

- The centroids have stabilized — there is no change in their values because the clustering has been successful.
- The defined number of iterations has been achieved.

## 🕮 6.5.3

Let's see the steps on how the K-means machine learning algorithm works using the Python programming language.

We'll use the Scikit-learn library and some random data to illustrate a K-means clustering simple explanation.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
```

## 📝 6.5.4

Import the right library for k-means algorithm.

```
from _____ import KMeans
```

## 📖 6.5.5

Let's generate random data to work with the k-means algorithm.

```
r= -2 * np.random.rand(100,2)
r1 = 1 + 2 * np.random.rand(50,2)
r[50:100, :] = r1
```

## 📝 6.5.6

Insert the right code to generate random numbers:

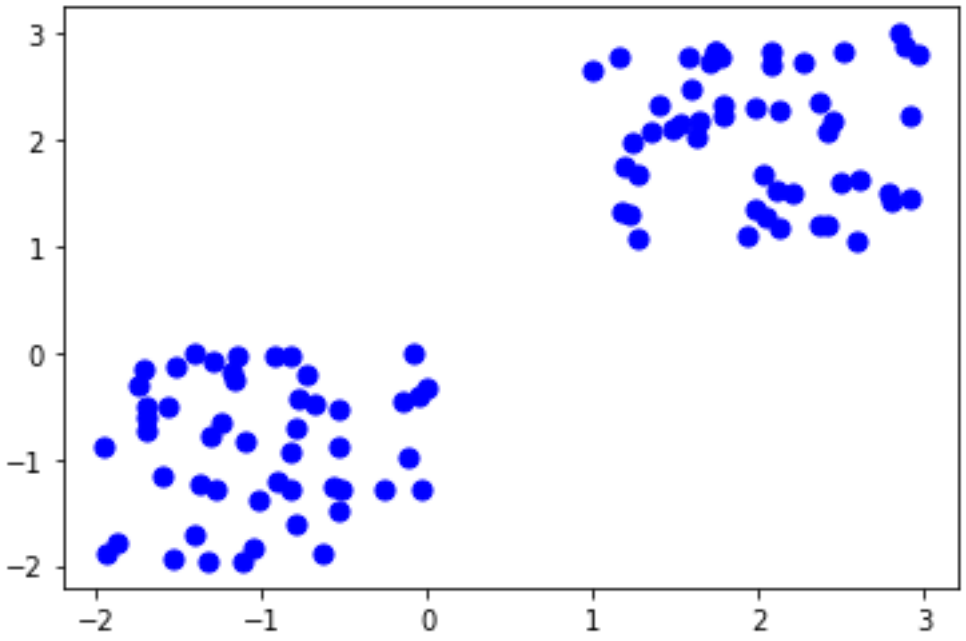
```
r= -2 * _____(100,2)
```

## 📖 6.5.7

After the generation of random two-dimensional numbers, we can look at the visualisation. A total of 100 data points has been generated and divided into two groups, of 50 points each.

```
plt.scatter(r[ : , 0], r[ :, 1], s = 50, c = 'b')
plt.show()
```

Let's look at the output figure:



## 6.5.8

Fill in the recommended type of plot to visualise the random two-dimensional values.

```
plt._____(r[ : , 0], r[ :, 1], s = 50, c = 'b')
plt._____
```

## 6.5.9

We will use some of the available functions in the Scikit-learn library to process the randomly generated data.

```
k_mean = KMeans(n_clusters=2)
k_mean.fit(r)
```

We have decided to gave *k* (n_clusters) an arbitrary the value 2.

```
KMeans(algorithm='auto', copy_x=True, init='k-means++',
max_iter=300,
 n_clusters=2, n_init=10, n_jobs=1,
precompute_distances='auto',
 random_state=None, tol=0.0001, verbose=0)
```

## 6.5.10

The next step in the process is to find the centroid of the clusters.

```
k_mean.cluster_centers_
```

For our random numbers we get the following centroids:

```
array([[-1.02458808, -0.87651595],
       [ 1.99234672,  2.04440745]])
```

## 6.5.11

Fill in the code to obtain the values of centroids:

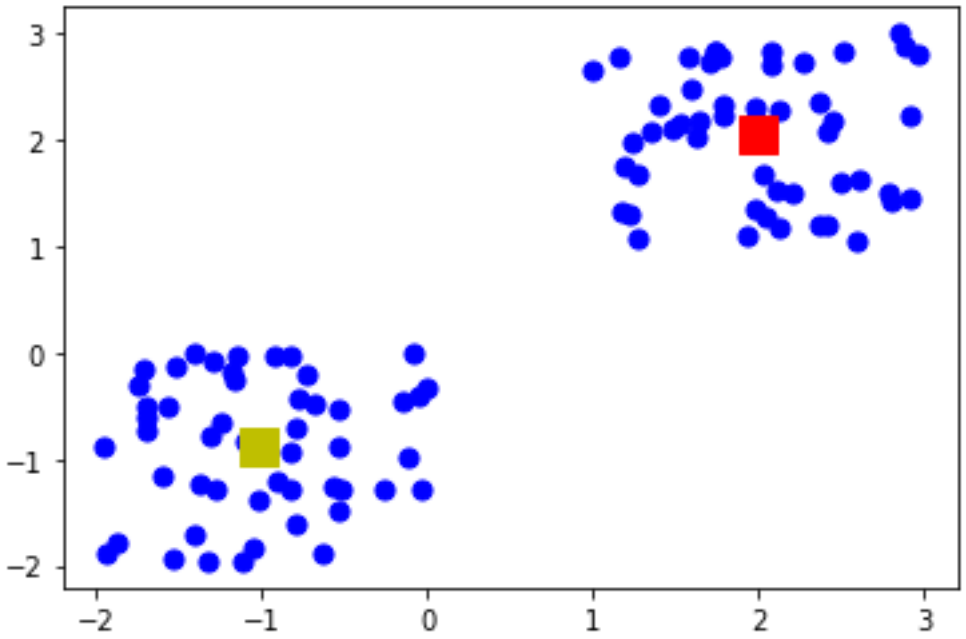
```
k_mean._____
```

## 🕮 6.5.12

Next, let us look at the visualisation of the centroids. As we did not save the centroid values into a variable, we just wrote the results and created two squares.

```
plt.scatter(r[ : , 0], r[ : , 1], s =50, c='b')
plt.scatter(-1.02458808, -0.87651595, s=200, c='y',
marker='s')
plt.scatter(1.99234672, 2.04440745, s=200, c='r', marker='s')
plt.show()
```

The output is following:




## 🕮 6.5.13

Let us return a step back and focus on one important note. It is the decision to choose the number of clusters for our data. There are two methods that are commonly used to evaluate the appropriate number of clusters:

1. The elbow method
2. The silhouette coefficient

## 6.5.14

What methods are used to choose the number of clusters?

- the elbow method
- the silhouette coefficient
- the arm method
- the right hand rule
- the knee method

## 🕮 6.5.15

These are often used as complementary evaluation techniques rather than one being preferred over the other. To perform the **elbow method**, run several *k*-means, increment *k* with each iteration, and record the sum of square errors (SSE) :

```
kmeans_kwargs = {"init": "random", "n_init": 10, "max_iter":
300, "random_state": 42 }

sse = []
for k in range(1, 11):
    kmeans = KMeans(n_clusters=k, **kmeans_kwargs)
    kmeans.fit(scaled_features)
    sse.append(kmeans.inertia_)
```

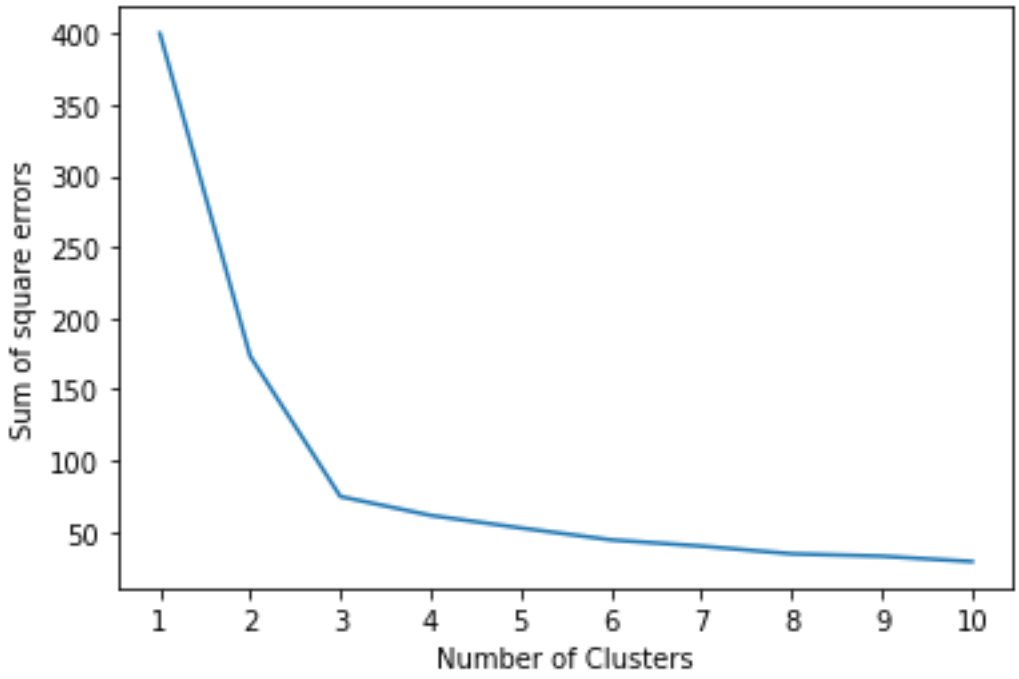
## 🕮 6.5.16

When you plot SSE as a function of the number of clusters, notice that SSE continues to decrease as you increase *k*. As more centroids are added, the distance from each point to its closest centroid will decrease.

There's a sweet spot where the SSE curve starts to bend known as the **elbow point**. The x-value of this point is thought to be a reasonable trade-off between error and the number of clusters. In this example, the elbow is located at *x=3*:

```
plt.plot(range(1, 11), sse)
plt.xticks(range(1, 11))
plt.xlabel("Number of Clusters")
plt.ylabel("Sum of square errors")
plt.show()
```

Let us look at the output graph:




## 🕮 6.5.17

The **silhouette coefficient** is a measure of cluster cohesion and separation. It quantifies how well a data point fits into its assigned cluster based on two factors:

1. How close the data point is to other points in the cluster
2. How far away the data point is from points in other clusters

Silhouette coefficient values range between -1 and 1. Larger numbers indicate that samples are closer to their clusters than they are to other clusters.

In the scikit-learn implementation of the silhouette coefficient, the average silhouette coefficient of all the samples is summarized into one score. The **silhouette score()** function needs a minimum of two clusters, or it will raise an exception.

## 🕮 6.5.18

Let us loop through the values of *k* again. This time, instead of computing SSE, we will compute the silhouette coefficient:

```
from sklearn.metrics import silhouette_score

silhouette_coefficients = []
```

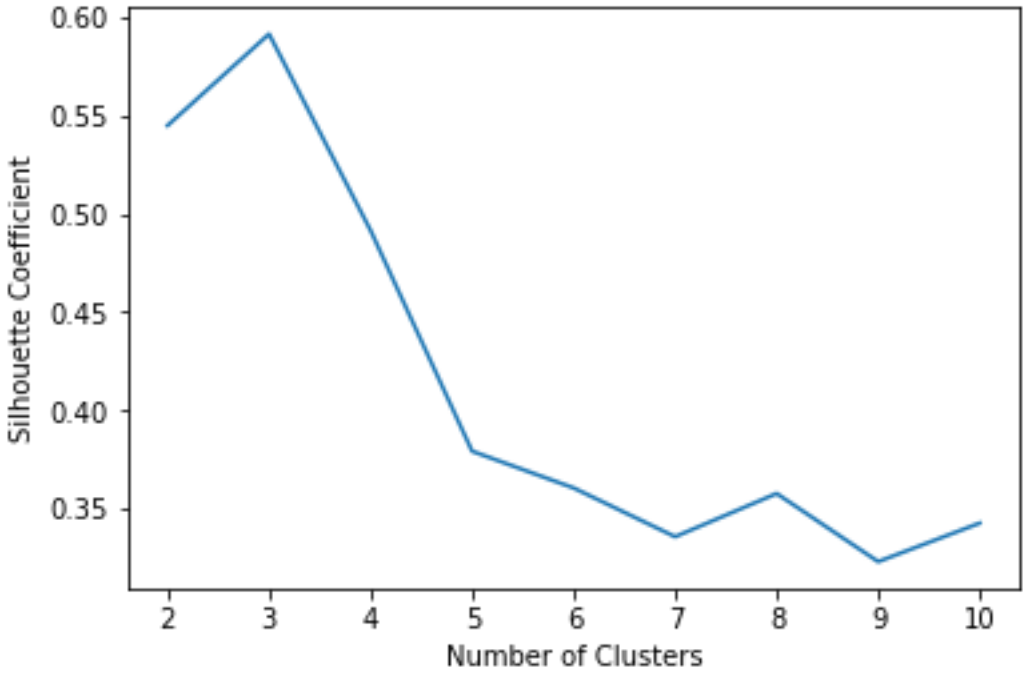
```
for k in range(2, 11):
    kmeans = KMeans(n_clusters=k, **kmeans_kwargs)
    kmeans.fit(scaled_features)
    score = silhouette_score(scaled_features, kmeans.labels_)
    silhouette_coefficients.append(score)
```

### 🕮 6.5.19

Plotting the average silhouette scores for each *k* shows that the best choice for *k* is 3 since it has the maximum score:

```
plt.plot(range(2, 11), silhouette_coefficients)
plt.xticks(range(2, 11))
plt.xlabel("Number of Clusters")
plt.ylabel("Silhouette Coefficient")
plt.show()
```

And the output looks like this:




Ultimately, your decision on the number of clusters to use should be guided by a combination of domain knowledge and clustering evaluation metrics.

# 6.6 Multinomial Logit Model

## 🕮 6.6.1

Logit Regression describes the dependence of a qualitative/categorical dependent variable on one or more quantitative independent variables. In logit regression, we do not directly model the variable *Y* but the probability that this variable has a specific value depending on the combination of the values of the independent variables **X**. It is based on the concept of conditional chance.

That means that the dependent variable is transformed into quantitative/continuous, which is a function of the probability of occurrence of the event. Classification and prediction are methods that use variables to predict the result, which takes the form of categories. This method is very effective when used correctly and the results can be interpreted relatively well. On the other hand, it requires thorough data preparation and fulfilment of application requirements.

## 📝 6.6.2

What methods are used to predict the results using variables, which takes the form of categories?

- prediction
- classification
- division
- retrospection
- logistics

## 🕮 6.6.3

Data on the use of portals or systems represent time data. Nevertheless, modelling the behavior of web users over time is absent in this application area. In part, the time variable, most often in the form of a "unixtime" integrating the date and time, is used in the extraction of sequential rules but is used only to determine the order of visited web parts during individual sessions.

In order to model the behavior of web users as a function of time, we use a multinomial logit model, which is a special case of the Generalized Linear Model (GLM).

The goal will be to model the behavior of web users based on time - hours of the day and other explanatory variables. In our case, the examined variable is the web

part of the portal, specifically the variable whose levels represent the content categories of the portal.

## 📖 6.6.4

In a given application area, the condition for using the LR test is often violated. The researched variable usually has a considerable number of levels, which represent the web parts of the portal (sites, content categories, activities, etc.). This results in a breach of the LR test condition, i.e. the expected numbers are not large enough. For this reason, we use alternative techniques to evaluate the model - visualization of differences of empirical and theoretical frequencies, identification of extremes, comparison of the distribution of empirical relative frequencies of approaches and estimated probabilities of the web part selection *j* in time *i* and visualization of empirical and theoretical logits for individual web parts, except reference web part.

## 📝 6.6.5

What are the three layers that will be examined using the multinominal logit model?

- logites
- probabilities
- frequencies
- levels
- clusters
- rules

## 📖 6.6.6

When modelling data, we proceed as follows, provided that we are based on data where individual accesses to the web parts of the portal are recorded.

1. Model definition
2. The estimation of the model's parameters
3. The estimation of logits
4. Probability estimation of accesses in time for reference web category
5. Probability estimation of accesses in time for other web categories
6. Visualization of the probabilities of web categories in time

## 6.6.7

Reorder the methodology used.

- Probability estimation of accesses in time for other web categories
- Visualization of the probabilities of web categories in time
- The estimation of logits
- The estimation of the model's parameters
- Model definition
- Probability estimation of accesses in time for reference web category

## 6.6.8

Model definition

Probability distribution of accesses $Y_{ij}$ in time *i* for the category *j* with observations $y_{ij}$ if the count of access is given

$$n_i = \sum_j y_{ij}$$

in time *i* is multinomial

$$P[Y_{i1} = y_{i1}, Y_{i2} = y_{i2}, \dots, Y_{iJ} = y_{iJ}] = \frac{n_i!}{y_{i1}!\, y_{i2}! \dots y_{iJ}!} \pi_{i1}^{y_{i1}} \pi_{i2}^{y_{i2}} \dots \pi_{iJ}^{y_{iJ}}$$

Since

$$\sum_{j=1}^{J} \pi_{ij} = 1$$

it is necessary to estimate *J-1* of unknown probabilities. The estimates are calculated using the Maximum Likelihood method. In the logarithmic function of likelihood (without constants)

$$\sum_i \sum_{j=1}^{J} y_{ij} \ln \pi_{ij}$$

is denoted a logit transformation

$$\eta_{ij} = \ln \frac{\pi_{ij}}{\pi_{iJ}}$$

where the last category is chosen as the reference category

$$\eta_{iJ} = 0$$

and it is assumed that the logits are linear functions of the independent variables

$$\eta_{ij} = \alpha_j + \boldsymbol{x}_i^T \boldsymbol{\beta}_j$$

Using inverse transformation, it is denoted

$$\pi_{iJ} = \frac{1}{1 + \sum_{j=1}^{J-1} e^{\eta_{ij}}}, \pi_{ij} = e^{\eta_{ij}} \pi_{ij}, j = 1, 2, \dots, J-1$$

respectively

$$\pi_{iJ} = \frac{1}{1 + \sum_{j=1}^{J-1} e^{\alpha_j + x_i^T \boldsymbol{\beta}_j}}, \pi_{ij} = \frac{e^{\alpha_j + x_i^T \boldsymbol{\beta}_j}}{1 + \sum_{j=1}^{J-1} e^{\alpha_j + x_i^T \boldsymbol{\beta}_j}}, j = 1, 2, \dots, J-1$$

The logarithm function is a likelihood function with unknown parameters after substituting such expressed into the first equation. After determining the model, it is necessary to identify the type of dependence for determining the degree of the polynomial and the selection of predictors, including dummy variables.

## 🕮 6.6.9

The estimation of the model's parameters by maximizing the logarithm of the multinomial likelihood function. The significance of parameters was tested using the Wald test. The estimated parameters are used to calculate estimates of logits and from logits can be calculated probabilities of selection of specific categories at a given time.

The logit model provides us with an estimate of the probabilities at the output. However, knowing the parameters of the model is also useful. Their absolute size can give us information on which predictors the variable under study depends the

most. The high absolute value of the parameter indicates a large dependence. A positive (negative) value speaks of a directly (indirectly) proportional dependence.

### 🕮 6.6.10

The estimation of logits for all values of independent variables:

$$\hat{\eta}_{ij} = a_j + \boldsymbol{x}_i^T \boldsymbol{b}_j, j = 1, 2, \dots, J-1$$

We use the multinominal logit model to model the distribution of a categorical variable. The categorical variable examined is the *Category* variable, the levels of which are the content categories of the portal. We created a model for staff, students, and other visitors, using the day of the week and hour of the day as predictive variables. In our example, we present a model for students.

### 🕮 6.6.11

Probability estimation of accesses in time *i* for reference web category *J*

$$\hat{\pi}_{iJ} = \frac{1}{1 + \sum_{j=1}^{J-1} e^{\hat{\eta}_{ij}}}$$

### 🕮 6.6.12

Probability estimation of accesses in time *i* for web category

$$\hat{\pi}_{ij} = e^{\hat{\eta}_{ij}} \hat{\pi}_{iJ}, j = 1, 2, \dots, J-1$$

### 🕮 6.6.13

Visualization of the probabilities of web category *j* in time *i*, where *j = 1, 2, ... , J*.

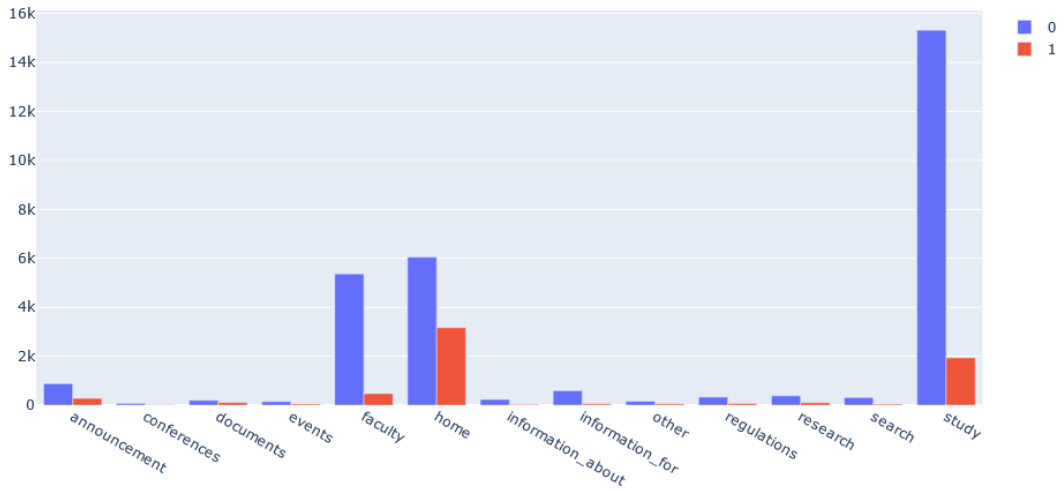
# 6.7 Web User behaviour (example)

## 🕮 6.7.1

The log file that was prepared in the first phase of this course will be now analyzed. We will look into a practical example of the multinomial logit model. First, we had to estimate the parameters for which we used a statistical method.

| Effect | Intercept | HOUR | HOUR_SQR | MO | TU | WE | TH | FR | SA |
|---|---|---|---|---|---|---|---|---|---|
| home_STUD | 1.34835 | 0.34182 | -0.01563 | -0.63353 | 1.10307 | 0.11376 | -0.49911 | 0.68682 | 0 |
| study_STUD | -2.84924 | 0.38448 | -0.00867 | 0.95518 | 3.02822 | 1.54827 | 1.22148 | 1.20767 | 0 |
| announcement_STUD | 1.51459 | -0.38342 | 0.01248 | 1.65098 | 3.04913 | 1.59056 | 0.14253 | -1.24192 | 0 |
| home_EMP | 5.58384 | -0.68231 | 0.02633 | 0.16019 | 0.00955 | -0.76405 | -0.98585 | 0 | 0 |
| study_EMP | 0.92959 | 0.06409 | -0.00612 | 0.38236 | 0.82227 | -0.1773 | -0.00245 | 0 | 0 |
| announcement_EMP | -3.91218 | 0.40371 | -0.02517 | 2.31863 | 2.96682 | 0.57197 | -0.49513 | 0 | 0 |

The results show us the estimates that we can use in the model. Based on the model definition we will analyse only four web categories: home, study, announcement and faculty (this one was selected as the reference web category).




## 🕮 6.7.2

The general approach to estimate the logits for all values of independent variables is based on:

$$\hat{\eta}_{ij} = a_j + x_i^T b_j, j = 1,2, \dots, J-1$$

Let us apply this to our example and examine the behaviour of the employees of the university on the web portal during a week. The web categories home, study and

announcement will be used to calculate the values for the reference category faculty. Estimation of the logits for specific categories during the weekdays will be denoted:

$$\hat{\eta}_{ij} = a_j + t * hour + t_2 hour^2 + day$$

where *a* is the parameter estimate for the web category, *t* is the parameter estimate for an hour, $t_2$ is the parameter estimate for hour_sqr, *hour* is for the value of hour (has value 7-22 because the university is open during these hours) and *day* that is the parameter estimate of the day. Because during the weekend the employees do not visit the university, we have a low number of accesses and we will estimate the logits only for the working days. This way the last examined day has 0 value and the equation looks like this:

$$\hat{\eta}_{ij} = a_j + t * hour + t_2 hour^2$$

### 📖 6.7.3

Let us look at Monday and how to get the logits from the estimates. We loop through hours and then calculate the logits for each category home, study and announcement.

```
for i in range(7,23):
    estH = data.loc[data["Effect"]=="home_EMP"]
    logitH =
estH.iloc[0]['Intercept']+estH.iloc[0]['HOUR']*i+estH.iloc[0][
'HOUR_SQR']*pow(i,2)+estH.iloc[0]['MO']
    estS = data.loc[data["Effect"]=="study_EMP"]
    logitS =
estS.iloc[0]['Intercept']+estS.iloc[0]['HOUR']*i+estS.iloc[0][
'HOUR_SQR']*pow(i,2)+estS.iloc[0]['MO']
    estA = data.loc[data["Effect"]=="announcement_EMP"]
    logitA =
estA.iloc[0]['Intercept']+estA.iloc[0]['HOUR']*i+estA.iloc[0][
'HOUR_SQR']*pow(i,2)+estO.iloc[0]['MO']
    logitsMO_EMP.loc[i]=[i,logitH,logitS,logitA]
```

### 📝 6.7.4

Fill in the code for Wednesday (WE):

```
for i in range(_____):
    estH = data.loc[data["Effect"]=="home_EMP"]
    logitH =
estH.iloc[0]['Intercept']+estH.iloc[0]['HOUR']*i+estH.iloc[0][
'HOUR_SQR']*pow(i,2)+estH.iloc[0]['_____']
    _____ = data.loc[data["Effect"]=="study_EMP"]
    logitS =
estS.iloc[0]['Intercept']+estS.iloc[0]['HOUR']*i+estS.iloc[0][
'HOUR_SQR']*_____(i,2)+estS.iloc[0]['_____']
    estA = data.loc[data["Effect"]=="announcement_EMP"]
    _____ =
estA.iloc[0]['Intercept']+estA.iloc[0]['_____']*i+estA.iloc[0]
[_____]*pow(i,2)+estO.iloc[0]['WE']
    logitsWE_EMP.loc[i]=[i,_____,_____,logitA]
```

## 🕮 6.7.5

The output of the code will be following:

```
Logites Monday Employee
    hour     home    study  announcement
7    7.0  2.25803  1.46070      -0.00091
8    8.0  1.97067  1.43299       0.02525
9    9.0  1.73597  1.39304       0.00107
10  10.0  1.55393  1.34085      -0.07345
11  11.0  1.42455  1.27642      -0.19831
12  12.0  1.34783  1.19975      -0.37351
13  13.0  1.32377  1.11084      -0.59905
14  14.0  1.35237  1.00969      -0.87493
15  15.0  1.43363  0.89630      -1.20115
16  16.0  1.56755  0.77067      -1.57771
17  17.0  1.75413  0.63280      -2.00461
18  18.0  1.99337  0.48269      -2.48185
19  19.0  2.28527  0.32034      -3.00943
20  20.0  2.62983  0.14575      -3.58735
21  21.0  3.02705 -0.04108      -4.21561
22  22.0  3.47693 -0.24015      -4.89421
```

And for Wednesday

| Logites | Wednesday | Employee | |
|---|---|---|---|
| hour | home | study | announcement |
| 7 | 7.0 | 1.33379 | 0.90104 | -1.74757 |
| 8 | 8.0 | 1.04643 | 0.87333 | -1.72141 |
| 9 | 9.0 | 0.81173 | 0.83338 | -1.74559 |
| 10 | 10.0 | 0.62969 | 0.78119 | -1.82011 |
| 11 | 11.0 | 0.50031 | 0.71676 | -1.94497 |
| 12 | 12.0 | 0.42359 | 0.64009 | -2.12017 |
| 13 | 13.0 | 0.39953 | 0.55118 | -2.34571 |
| 14 | 14.0 | 0.42813 | 0.45003 | -2.62159 |
| 15 | 15.0 | 0.50939 | 0.33664 | -2.94781 |
| 16 | 16.0 | 0.64331 | 0.21101 | -3.32437 |
| 17 | 17.0 | 0.82989 | 0.07314 | -3.75127 |
| 18 | 18.0 | 1.06913 | -0.07697 | -4.22851 |
| 19 | 19.0 | 1.36103 | -0.23932 | -4.75609 |
| 20 | 20.0 | 1.70559 | -0.41391 | -5.33401 |
| 21 | 21.0 | 2.10281 | -0.60074 | -5.96227 |
| 22 | 22.0 | 2.55269 | -0.79981 | -6.64087 |

## 🕮 6.7.6

Probability estimation of accesses in time *i* for reference web category *J* that is in our example the category *faculty:*

$$\hat{\pi}_{iJ} = \frac{1}{1 + e^{\hat{\eta}_{i,home}} + e^{\hat{\eta}_{i,study}} + e^{\hat{\eta}_{i,announcement}}}$$

## 🕮 6.7.7

Now let us try to calculate the probability of access to the reference web category *faculty* using the logits of other web categories.

```
prob_ref_EMP = pd.DataFrame(columns=('hour','MO'))
for i in range(7,23):
    mon = 1/(1+exp(logitsMO_EMP.iloc[i-
7]['home'])+exp(logitsMO_EMP.iloc[i-
7]['study'])+exp(logitsMO_EMP.iloc[i-7]['announcement']))
    prob_ref_EMP.loc[i]=[i,mon]
```

This will return the following output:

| | hour | MO |
|---|---|---|
| 7 | 7.0 | 0.063003 |
| 8 | 8.0 | 0.074670 |
| 9 | 9.0 | 0.085451 |
| 10 | 10.0 | 0.095406 |
| 11 | 11.0 | 0.104604 |
| 12 | 12.0 | 0.112909 |
| 13 | 13.0 | 0.119849 |
| 14 | 14.0 | 0.124561 |
| 15 | 15.0 | 0.125861 |
| 16 | 16.0 | 0.122511 |
| 17 | 17.0 | 0.113688 |
| 18 | 18.0 | 0.099560 |
| 19 | 19.0 | 0.081598 |
| 20 | 20.0 | 0.062282 |
| 21 | 21.0 | 0.044227 |
| 22 | 22.0 | 0.029279 |

## 6.7.8

Try to fill in the gaps for all of the examined days:

```
prob_ref_EMP =
pd.DataFrame(columns=('hour','MO','TU','WE','TH','FR'))
for i in range(7,23):
    mon = 1/(1+exp(logitsMO_EMP.iloc[i-
7]['home'])+exp(logitsMO_EMP.iloc[i-
7]['study'])+exp(logitsMO_EMP.iloc[i-7]['announcement']))
    tue = 1/(1+exp(logitsTU_EMP.iloc[i-
7]['home'])+exp(logitsTU_EMP.iloc[i-
7]['study'])+exp(logitsTU_EMP.iloc[i-7]['announcement']))
    wed = 1/(1+exp(logitsWE_EMP.iloc[i-
7]['home'])+exp(logitsWE_EMP.iloc[i-
7]['study'])+exp(logitsWE_EMP.iloc[i-7]['announcement']))
    thu = 1/(1+exp(logitsTH_EMP.iloc[i-
7]['home'])+exp(logitsTH_EMP.iloc[i-
7]['study'])+exp(logitsTH_EMP.iloc[i-7]['announcement']))
    fri = 1/(1+exp(logitsFR_EMP.iloc[i-
7]['home'])+exp(logitsFR_EMP.iloc[i-
7]['study'])+exp(logitsFR_EMP.iloc[i-7]['announcement']))
    prob_ref_EMP.loc[i]=[i,mon,_____,_____,_____,_____]
```

## 🕮 6.7.9

When we print the dataframe then there will be the following output:

```
Probability estimation for the reference web category faculty for employees
    hour        MO        TU        WE        TH        FR
7    7.0  0.063003  0.056095  0.134558  0.142188  0.082057
8    8.0  0.074670  0.063938  0.155735  0.161420  0.099268
9    9.0  0.085451  0.071188  0.174599  0.178423  0.115201
10  10.0  0.095406  0.078250  0.191457  0.193770  0.129381
11  11.0  0.104604  0.085419  0.206611  0.207926  0.141511
12  12.0  0.112909  0.092762  0.220066  0.221047  0.151248
13  13.0  0.119849  0.100025  0.231364  0.232866  0.158044
14  14.0  0.124561  0.106560  0.239496  0.242590  0.161085
15  15.0  0.125861  0.111269  0.242897  0.248816  0.159373
16  16.0  0.122511  0.112682  0.239596  0.249531  0.151977
17  17.0  0.113688  0.109275  0.227664  0.242339  0.138486
18  18.0  0.099560  0.100106  0.206044  0.225162  0.119510
19  19.0  0.081598  0.085565  0.175563  0.197438  0.096915
20  20.0  0.062282  0.067656  0.139460  0.161325  0.073474
21  21.0  0.044227  0.049337  0.102668  0.121740  0.051981
22  22.0  0.029279  0.033255  0.069968  0.084582  0.034355
```

We can already see the probability of access to the *faculty* web category during the week. The values are from the interval <0,1>. The higher the number the higher the probability that the employee accessed the web category during the specific hour on a specific day.

## 🕮 6.7.10

The estimation of probabilities of access to the other web parts is based on the logit estimate for a specific web category and at the same time on the estimation of probabilities of access to the reference web category.

$$\hat{\pi}_{ij} = e^{\hat{\eta}_{ij}}\hat{\pi}_{iJ}, j = 1,2,\ldots,J-1$$

## 🕮 6.7.11

Now let us try to calculate the estimates for the *study* web category:

```
prob_STUDY_EMP =
pd.DataFrame(columns=('hour','MO','TU','WE','TH','FR'))
```

```
for i in range(7,23):
  moSTUDY = exp(logitsMO_EMP.iloc[i-
7]['study'])*prob_ref_EMP.iloc[i-7]['MO']
  tuSTUDY = exp(logitsTU_EMP.iloc[i-
7]['study'])*prob_ref_EMP.iloc[i-7]['TU']
  weSTUDY = exp(logitsWE_EMP.iloc[i-
7]['study'])*prob_ref_EMP.iloc[i-7]['WE']
  thSTUDY = exp(logitsTH_EMP.iloc[i-
7]['study'])*prob_ref_EMP.iloc[i-7]['TH']
  frSTUDY = exp(logitsFR_EMP.iloc[i-
7]['study'])*prob_ref_EMP.iloc[i-7]['FR']
  prob_STUDY_EMP.loc[i]=[i,moSTUDY ,tuSTUDY ,weSTUDY ,thSTUDY
,frSTUDY ]
```

The new data frame will produce the following results:

```
Probability estimation for other web categories for employees
    hour         MO         TU         WE         TH         FR
7    7.0   0.602574   0.461475   0.510700   0.432310   0.668645
8    8.0   0.535793   0.394628   0.443451   0.368206   0.606862
9    9.0   0.484887   0.347461   0.393161   0.321851   0.556940
10  10.0   0.451274   0.318364   0.359370   0.291361   0.521393
11  11.0   0.434732   0.305356   0.340750   0.274703   0.501067
12  12.0   0.434595   0.307115   0.336136   0.270472   0.495993
13  13.0   0.450342   0.323291   0.344992   0.278159   0.505957
14  14.0   0.481623   0.354404   0.367479   0.298181   0.530657
15  15.0   0.527849   0.401392   0.404248   0.331725   0.569459
16  16.0   0.587425   0.464741   0.455896   0.380350   0.620848
17  17.0   0.656937   0.543135   0.522049   0.445158   0.681781
18  18.0   0.730790   0.632045   0.600174   0.525394   0.747384
19  19.0   0.801969   0.723359   0.684731   0.616867   0.811527
20  20.0   0.863940   0.807243   0.767676   0.711385   0.868333
21  21.0   0.912674   0.875754   0.840764   0.798632   0.913908
22  22.0   0.947474   0.925660   0.898497   0.870104   0.947185
```

## 6.7.12

Fill in the code for other days to estimate the probabilities to access the other web categories:

```
prob_STUDY_EMP =
pd.DataFrame(_____=('hour','MO','TU','WE','TH','FR'))
prob_HOME_EMP = pd.DataFrame(columns=(_____))
prob_ANN_EMP =
pd._____(columns=('hour','MO','TU','WE','TH','FR'))
```

```
for i in range(7,23):
    moHOME = exp(logitsMO_EMP.iloc[i-
7]['home'])_____prob_ref_EMP.iloc[i-7]['MO']
    moSTUDY = _____(logitsMO_EMP.iloc[i-
7]['study'])*prob_ref_EMP.iloc[i-7]['MO']
    moANN = exp(logitsMO_EMP.iloc[i-
7]['_____'])*prob_ref_EMP.iloc[i-7]['MO']
    tuHOME = exp(logitsTU_EMP.iloc[i-7]['home'])*_____[i-
7]['TU']
    tuSTUDY = exp(logitsTU_EMP.iloc[i-
7]['_____'])*prob_ref_EMP.iloc[i-7]['TU']
    _____ = exp(logitsTU_EMP.iloc[i-
7]['announcement'])*prob_ref_EMP.iloc[i-7]['TU']
    weHOME = exp(logitsWE_EMP.iloc[i-
7]['_____'])*prob_ref_EMP.iloc[i-7]['WE']
    weSTUDY = exp(logitsWE_EMP.iloc[i-
7]['study'])*prob_ref_EMP.iloc[i-7]['WE']
    weANN = exp(logitsWE_EMP.iloc[i-
7]['announcement'])*prob_ref_EMP.iloc[i-7]['_____']
    frHOME =
exp(logitsTH_EMP.iloc[_____]['home'])*prob_ref_EMP.iloc[i-
7]['TH']
    thSTUDY = exp(logitsTH_EMP.iloc[i-
7]['study'])*prob_ref_EMP._____[i-7]['TH']
    thANN = exp(logitsTH_EMP.iloc[i-
7]['announcement'])*prob_ref_EMP.iloc[i-7]['TH']
    frHOME = exp(_____.iloc[i-7]['home'])*prob_ref_EMP.iloc[i-
7]['FR']
    frSTUDY = exp(logitsFR_EMP.iloc[i-
7]['study'])*prob_ref_EMP.iloc[i-7]['_____']
    frANN = exp(logitsFR_EMP.iloc[i-
7]['_____'])*prob_ref_EMP.iloc[i-7]['FR']
    prob_HOME_EMP.loc[i]=[i,_____]
    _____.loc[i]=[i,moSTUDY,tuSTUDY,weSTUDY,thSTUDY,frSTUDY]
    prob_ANN_EMP.loc[i]=[i,moANN,tuANN,weANN,thANN,frANN]
```

## 📖 6.7.13

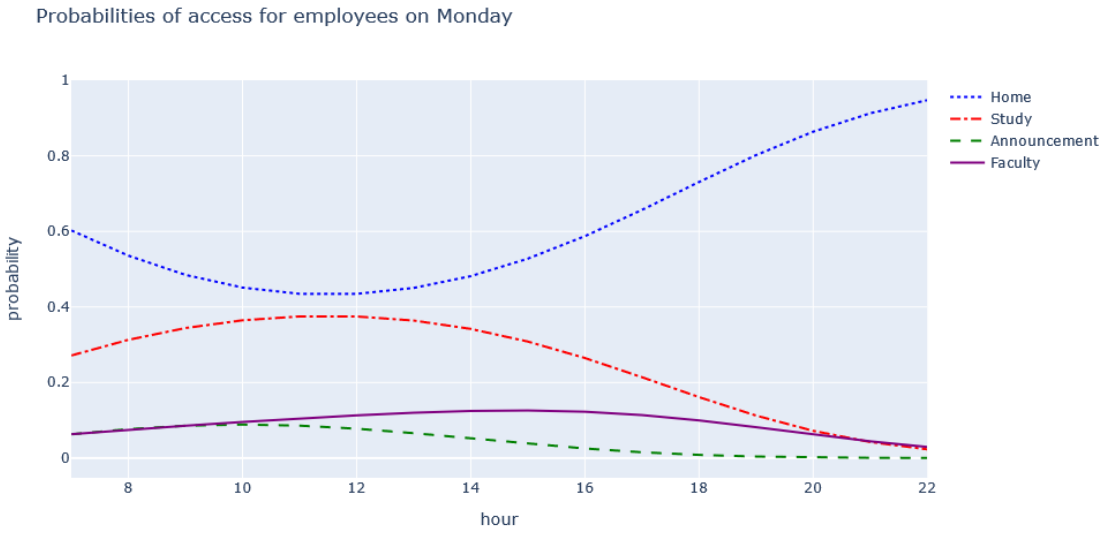
After estimating the probabilities of access for individual web categories, we can take a closer look at the behavior of users on a given web portal. We focused on examining the behavior of university staff. We identify employees as approaches that are implemented from within the network, and only at the time when the university is accessible (from 7 am to 10 pm).

We can use an interactive library *plotly* to visualize the probabilities. The code can look like this:

```
figMO = go.Figure()

figMO.add_trace(go.Scatter(x=prob_HOME_EMP['hour'],
y=prob_HOME_EMP['MO'], mode='lines', name='Home', line =
dict(color='blue', width=2, dash='dot')))

figMO.add_trace(go.Scatter(x=prob_STUDY_EMP['hour'],
y=prob_STUDY_EMP['MO'], mode='lines', name='Study', line =
dict(color='red', width=2, dash='dashdot')))

figMO.add_trace(go.Scatter(x=prob_ANN_EMP['hour'],
y=prob_ANN_EMP['MO'], mode='lines', name='Announcement', line
= dict(color='green', width=2, dash='dash')))

figMO.add_trace(go.Scatter(x=prob_ref_EMP['hour'],
y=prob_ref_EMP['MO'], mode='lines', name='Faculty', line =
dict(color='purple', width=2)))

figMO.update_layout(title='Probabilities of access for
employees on Monday', xaxis_title='hour',
yaxis_title='probability')
figMO.show()
```

## 🕮 6.7.14

The resulting graphs look like this:



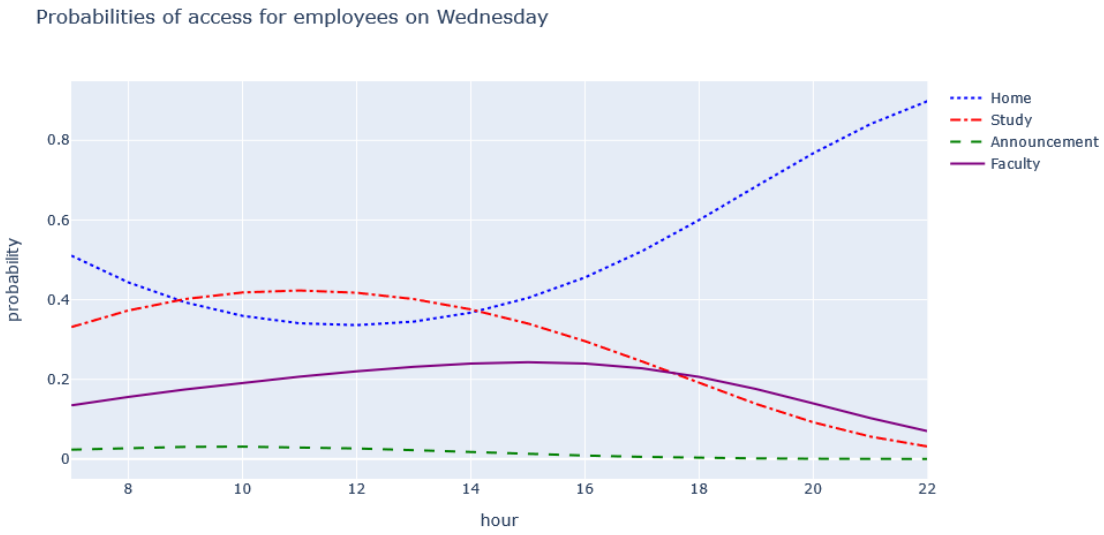


As can be seen in the graphs of visualizations of access probability estimates, during the first two days of the week (Monday, Tuesday), the behavior of employees on the *home* and *study* web categories is similar. The working hours of university employees start at 7 am, which can be seen on the graph as a higher probability of visiting the web category *home*. During working hours, ie from 7 am to 3 pm, we observe a declining trend. An interesting behavior is the increasing tendency of the probability of accessing the web category *home* in the evening. The reason may be incorrect data cleaning or an improperly selected model.

On the contrary, in the case of estimating the probability of access to the web category of the *study*, we observe a decreasing probability of access in the evening. Also, the probability of access estimates for the *faculty* web category and *announcement* have a decreasing probability of access in the evening.

## 6.7.15

Now it is your turn to interpret the results of the analysis. Look at the data for Wednesday and choose the correct statement.




- the home category is visited as first in the morning
- the announcement category is less visited
- the announcement category is the most visited
- the study category has a decreasing trend in the morning
- the study category has a decreasing trend after the noon

# Data Evaluation

Chapter 7

# 7.1 Multinomial Logit Model

## 🕮 7.1.1

Based on the assumption that the expected counts are big enough (they are not zero and no more than 20% from observations for a specific category in a specific time is less than 5) to compare the actual model with the saturated model that is used to predict the probabilities independently for *i=0,1,...,23* then the statistics (deviance, likelihood ratio) can be used.

In the examined application field is often the condition of using the LR test/ Pearson statistics violated. Usually, the examined variable has a considerable number of levels that are web parts of the portal or system (pages, content categories, activities, etc.). This results in the violation of using the LR test/Pearson statistics- the expected counts are not large enough. For this reason, are used alternative methods to evaluate the model - visualization of empirical and theoretical counts differences, extreme identification, comparison of the distribution of empirical relative counts of accesses and estimated probabilities of the examined web part *j* in time *i*, and empirical and theoretical logit visualization of each web part, except the reference web part.

## 🕮 7.1.2

The created model in the last chapter was evaluated based on the following steps:

1. Empirical counts determination
2. Theoretical counts estimation
3. Visualization of differences in the empirical and theoretical counts of accesses
4. Extreme values identification
5. Calculation of relative empirical counts of accesses
6. Comparison of the distribution of the relative empirical counts of accesses with the estimated probabilities of the selected web category *j* in time *i*. To test the zero hypotheses, dividing the pair differences is symmetric around zero, a Wilcoxon pair test will be used
7. Calculation of empirical logits
8. Visualization of empirical and theoretical logits for individual web categories except for the reference one.

## 📝 7.1.3

Reorder the methods to evaluate our created model:

- Visualization of differences in the empirical and theoretical counts of accesses
- <|br>
- Theoretical counts estimation
- Extreme values identification
- Comparison of the distribution of the relative empirical counts of accesses with the estimated probabilities
- Calculation of empirical logits
- Calculation of relative empirical counts of accesses
- Visualization of empirical and theoretical logits for individual web categories except for the reference one
- Empirical counts determination

## 🕮 7.1.4

First, we need to extract from the log file the actual (empirical) number of accesses to individual web categories at the time *i*. We will use a contingency table to extract these values from the log file and store them in a dataframe.

## 🕮 7.1.5

Based on the estimation of the probabilities of accesses, it is possible, with a combination of empirical frequencies of accesses to web categories, to estimate the theoretical frequencies of accesses to the investigated web categories. The estimation of the theoretical number of accesses in time *i* and on the web category *j* is as follows:

$$\hat{y}_{ij} = \hat{\pi}_{ij} \sum_{j} y_{ij}$$

where Pi(ij) is the estimate of probabilities of access in time *i* on the web category *j* and y(ij) is the empiric count of accesses in time *i* on web category *j*.

The result is an estimate of the theoretical number of accesses over time and to the investigated web categories, which in the next step we will compare with the estimate of empirical frequencies of accesses over time and to the investigated web categories.

## 🕮 7.1.6

In order to determine the correctness of the proposed model, it is necessary to compare the empirical, that means the actual observed counts, and the theoretical counts, which we calculated using an estimate of the probability of access to individual web parts. For comparison, we calculate the differences between empirical and theoretical frequencies as follows:

$$d_{ij} = y_{ij} - \hat{y}_{ij}$$

where it is the difference of empirical counts of theoretical counts in time *i* of the web category *j*. Based on the differences calculated in this way for individual web categories, we can visualize for individual days the differences in the number of estimated and observed.

## 🕮 7.1.7

The next step is to identify the extreme values that may signal to us at which points (in our case hours) the deviation of the estimated frequencies was significantly higher or lower from the observed frequencies. To identify extreme values, we use the following equation:

$$d_{ij} < \bar{d}_j - 2s \ \wedge \ d_{ij} > \bar{d}_j + 2s$$

where *d* is the average value of the differences of counts for web category *j* and *s* is the standard deviation.

## 🕮 7.1.8

Just as it was possible to compare empirical and theoretical counts, it is also possible to compare the empirical relative frequencies of access to web parts and the estimated probabilities of access to web parts. In principle, empirical relative numbers are the actual probabilities of access to a given web part. Therefore, these values are calculated as follows:

$$p_{ij} = \frac{y_{ij}}{\sum_j y_{ij}}$$

where *y* is the empirical count int time *i* and of web category *j*.

## 🕮 7.1.9

Next is to determine the suitability of the selected model at the level of estimated probabilities. We compare them with empirical relative frequencies, which are in principle the real share of access to the investigated web parts. We will compare the distributions of probabilities of empirical relative frequencies of accesses and estimated probabilities of selection of the web category *j* at the time *i*:

$$r_{ij} = p_{ij} - \pi_{ij},\ H0: F(-r) = 1 - F(r)$$

To test the null hypothesis, the distribution of pair differences is symmetric around zero, the Wilcoxon pair test can be used.

## 🕮 7.1.10

Subsequently, we can proceed to the evaluation of theoretical and empirical logits. In this case, we observe whether our estimated theoretical logits fit (model) empirical logits calculated from empirical relative counts:

$$h_{ij} = \ln\left(\frac{p_{ij}}{p_{iJ}}\right), j = 1,2, \dots, J-1$$

# PRISCILLA

priscilla.fitped.eu