Master’s Thesis

The Rise of Python and Qlik Sense in Data Science: Prospects and Limitations

- Two Case Studies from the Aviation Industry

By Felix Scheibe

08/12/2019

Department 3

Project Management and Data Science (MPMD)

First supervisor: Prof. Dr. Tilo Wendler, HTW Berlin

Second supervisor: Prof. Dr. Bertil Haack, TH Wildau

Hochschule für Technik
und Wirtschaft Berlin

University of Applied Sciences

Index

Index of figures

Index of tables

Index of abbreviations

API	Application Programming Interface
BI	Business Intelligence
BI&A	Business Intelligence and Analytics
CRISP-DM	Cross-industry standard process for data mining
ETL	Extract-Transform-Load
EU	European Union
Eurostat	European Statistical Office
GUI	Graphical User Interface
NLP	Natural Language Processing
NUTS	Nomenclature des unités territoriales statistiques
RMSE	Root-mean-square error
SSE	Sum of Squared Errors

1 Introduction

Given the estimation that in today's digital world over 2.5 exabytes (= 2.5 billion gigabytes) of data are created every single day[1], the need for tools to overcome human limitations in information processing becomes obvious. Especially in the field of pattern recognition computers outperform humans significantly[2], however, cognitive scientists have found out that people regularly feel they understand complex phenomena with far greater precision, coherence and depth than they actually do.[3]

This overconfidence is based on people's limited knowledge and their misleading intuitive epistemology and is significantly stronger for explanatory knowledge – knowledge that involves complex causal relations – than for many other categories, such as knowledge for facts or procedures. This phenomenon called the *illusion of explanatory depth* increases with the number of subcomponents in a complex interdependent system: an individual may gain insights into a high level function and, based on this expert knowledge, still falsely assume an understanding of further levels up or down in the hierarchy of causal mechanisms. [4]

In the digital sphere, overcoming the *illusion of explanatory depth* is directly linked to the educational concept of *data literacy,* defined as the skillset that enables individuals to access, interpret, critically assess, manage, handle and ethically use data.[5] A recent study revealed that at high-performing organizations not only the executive team but employees at all levels are on average better educated on data concepts than their competitors.[6]

As the labor market for well-trained 'data workers' is suffering from supply shortages[7], over the recent years several tools have been developed in order to support or even partly replace scarce human resources in the field of data analysis. Tools with advanced analytical functions can fundamentally be grouped into two categories: plain programming languages with comprehensive libraries for data analysis, and ready-to use business intelligence (BI) software with ever expanding features. Both of them provide fact-based insights to support business decisions. Traditionally, BI used to provide new perspectives on previously known things, using some formulas that were already available like KPIs or trend charts. On the other hand, data science

[1] Domo (2018)

[2] John Pavlus (2016)

[3] Mills/Keil (2004), p. 3

[4] Rozenblit/Keil (2002), p. 522-523

[5] Calzada/Marzal (2013), p. 126

[6] Gottlieb/Weinberg (2019)

[7] LinkedIn (2018)

based on programming is supposed to answer data-related questions that nobody ever asked before, developing dedicated algorithms.[8] But state-of-the-art BI platforms are starting to blur this line by implementing ready-to-use data science functionalities like clustering or time series forecasting.

Little research can be found on benchmarking products of one of these two categories, but a total research gap lies in comparing the performance of programming languages against those of BI tools in order to identify areas of relative strength and weakness. Here a systematic methodology must not focus on one single aspect, but on the entire data science workflow according to the CRISP-DM framework.

This thesis tries to answer the research question under which criteria, and in which field a programming language or a BI tool is to be preferred for data analysis. The mentioned supply shortage for highly skilled data experts is currently boosting the demand for self-service BI tools, where more and more tasks can be taken over by the wider field of business analysts.[9] This trend is raising questions like:

- Are BI tools already more efficient in those steps that can be automated or executed by business users, e.g. calculation of KPIs and data visualization?
- Should company's individual needs still better be addressed by highly trained data scientists using programming languages, e.g. for data cleansing?
- Building statistical models used to be the key competence of programmers and is now being challenged as BI platforms are integrating sophisticated but standardized algorithms. How good is their performance when compared to manual coding, e.g. for clustering and forecasting?

In the theoretical part of this thesis analogies from innovation research are applied in order to analyze the current relation of programming languages and BI tools, narrowing down the focus to two market leaders: Python and Qlik Sense. In the second step, evaluation criteria are developed and framed in a utility analysis in order to make these two tools comparable. In the empirical part this utility analysis framework will be applied to two case studies from the aviation industry: one for a clustering model and another one for time series forecasting. Following the CRISP-DM methodology, prospects and limitations for both tools are evaluated for every step, leading to a holistic overview.

[8] Scherbak (2019)
[9] Heller (2017)

Clustering and time series forecasting are chosen as case studies because these two data mining algorithms are the only ones implemented in Qlik Sense so far. The scope of this thesis will be confined by the current state of this tool and the features and measures realized to this day.

Consequently, the focus of the entire thesis will not lie on iterated model fitting, but on comparing and systematically assessing product features of Qlik Sense and their corresponding Python libraries according to criteria pre-defined in the utility analysis framework.

All relevant Python scripts and Qlik Sense files are attached to this thesis. Note that for both tools the datasets are not automatically included in the .ipynb and .qvf files and need to be loaded separately from the local drive. Problems concerning exports and imports from used Qlik Sense extensions like Vizlib have been reported.[10]

[10] QlikTech International AB (2018a)

2 Theory

2.1 Analogies from innovation research: The current stage of Business Intelligence

Technology life cycle and the S-curve model

In innovation research the *S-curve model* (see figure 1) illustrates the typical pathway for innovation, as new technologies begin slowly, accelerate and hit a stalling point that requires a jump to a new curve for a new technology[11]:

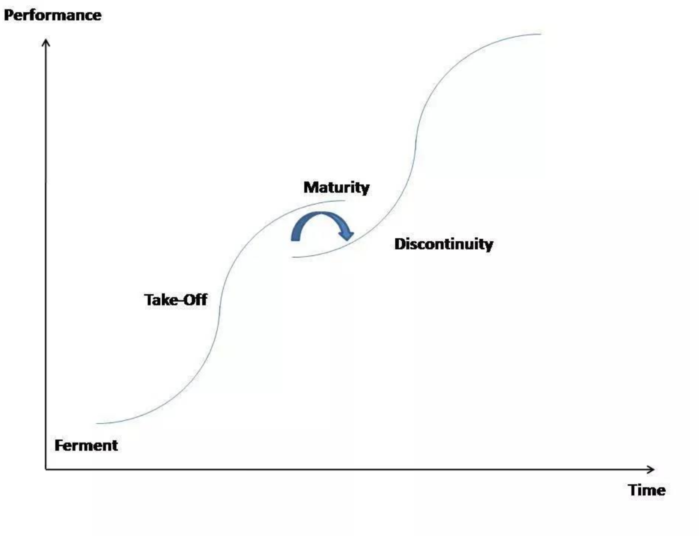


Figure 1: S-curve model[12]

One technology life cycle can contain several product life cycles, and as a relatively young technology the market for data analysis tools and techniques is yet diverse with dozens of competitors fighting for market share.

The need for professional data science and data analytics tools seems to be obvious, with the amount of global data constantly increasing and extremely lucrative business models being built around its exploitation. In the long run the S-curve model might indicate the roadmap to a human-like *general artificial intelligence*[13], but for now suppliers are competing about implementing more and more functionalities and building unrivaled user-friendly products.

[11] Lu/Beamish (2004)
[12] Gal's insights (2015)
[13] Joshi (2019)

At the first glance the S-curve model might seem trivial, but it is a key concept in order to understand the current market developments and the relation between programming languages and BI tools. The next sections will step by step try to explain this relationship in the field of advances analytics and data science.

From innovators to mass market: Diffusion of innovations

Closely linked to the *S-curve model* of the technology life cycle is the so-called *diffusion curve* (see figure 2): empirical research found generic patterns of how new technologies spread. Starting from the small group of innovators (the first 2.5% of the market that adopt a new product), for disruptive technological innovations the most difficult step lies in making the transition between the group of early adopters to the early majority, which is critical for mass-market adoption.[14]

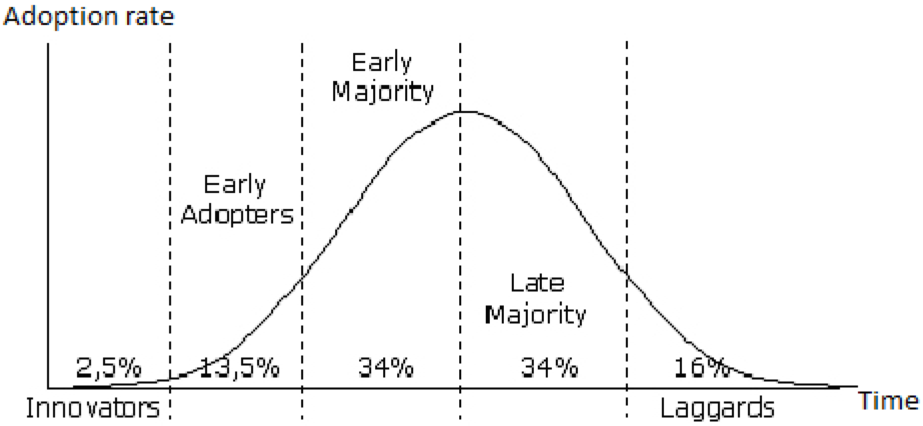


Figure 2: Diffusion of innovations[15]

Current products for advanced data analysis can basically be divided into two categories: BI tools and programming languages. In this context it is crucial to bear in mind that programming languages often follow a free and open source approach, whereas BI tools are regularly sold by license and create a stable revenue stream – especially once they move mainstream and enter the group of early majority in the diffusion curve.

Technology research firm *Gartner* predicts that by 2020, the number of data and analytics experts in business units will grow at three times the rate of experts in IT departments, which supports the hypothesis of data analysis tasks shifting from highly skilled IT experts to the broad mass of business roles in various departments. Furthermore, by 2021 natural language processing (NLP)

[14] Moore (2014), p. 21

[15] Rogers (1995), p. 257

and conversational analytics is expected to boost BI adoption to over 50% of employees, which according to Roger's *diffusion model* (figure 2) clearly means mass market adoption.[16]

Consequently, from an economic perspective BI vendors have a monetary incentive to turn laborious programming tasks into automated easy-to-use BI tools in order to address the skill shortage, enter mass market and create sustainable royalty streams. At the same time the demand for BI tools is likely to rise, due to the growing data volume and the profitability of data-driven business models.[17]

This leads to the question of who is entering the market for BI products and who is likely to achieve the segment of early majority, which is the pivotal point for mass market adoption.

Critical mass theory, network effects and winner-take-all markets

The '*critical mass theory'* attempts to explain diffusion of interactive media, such as telephone, electronic mail or social media. For non-interactive innovations the dependence on other users are generally sequential, which means that the group of early adopters influences the late adopters to use the innovation (see figure 2). On the contrary, for interactive media the interdependence is reciprocal, with both users influencing each other. This phenomenon is based on *network effects*, where the value of a product increases with the number of its users and technology adoption becomes self-sustained after reaching a *critical mass* of users.[18]

In these industries characterized by *network effects*, a single technology standard often rises to dominance, locking out competing technologies and leading to so-called *winner-take-all markets.*[19]

BI tools can be regarded as interactive media for office workers, since data models and analysis can be shared among the users of the same BI platform. This holds true only for users of the same BI platform; an exchange between users of different platforms is not possible due to non-compatible data formats. This means that the bigger your market power, the more likely your BI platform is to be established as a standard office product and to gain monopoly-like profits - comparable to what Microsoft has achieved with its Office Suite for spreadsheet analysis and text processing.[20]

[16] Gartner Inc. (2019), p. 1

[17] Statista (2019b)

[18] Markus (1987), p. 491

[19] Schilling (2002), p. 387

[20] Baseman et al. (1995), p. 12

This is one of the reasons why well-established data-driven companies have started to invest in the market for BI software, with Google's parent company Alphabet acquiring Looker and Salesforce taking over Tableau, both in June 2019.[21] These strategic acquisitions might reflect two expectations:

1) The beginning of mass market adoption for BI products, since they can be handled not only by the small niche of programmers but by the big group of non-technical white-collar workers that are dealing with the ever-growing amount of both internal and external data. Particularly, the graphical user interface (GUI) is likely to address a broader user base and BI platforms are constantly introducing new functionalities which used to be available to programmers only, like built-in clustering and forecasting algorithms.
2) The establishment of BI platforms as the new license-based revenue driver, especially if their mother companies have enough market power to create a critical mass of users. For data-driven companies, BI tools can be offered as a complementary product to their core services, boosting network effects. One BI platform might become a standard office product and turn this new industry into a winner-take-all market.

The same principles of critical mass theory, network effects and winner-takes-all markets also apply to programming languages, however, since they are often free and open source there is no economic incentive to gain market share.

Based on an analysis of BI development stages, the next section will outline why BI might be seen as a new technology cycle, jumping from the programming technology cycle running in parallel (see figure 1).

Definition and development stages of BI

Data as such is raw, random and unorganized. In contrast, information is data that has been organized, structured and processed and can therefore be used to gain knowledge and support decision-making. While data warehousing can be described as the more technical 'back room' where data gets integrated and stored ('*getting data in*'), business intelligence enables access and delivery of information to business users ('*getting information out*'). It is the umbrella term for a company's processes and products that turn data into actionable information, fulfilling the criteria of the 'Five Cs':[22]

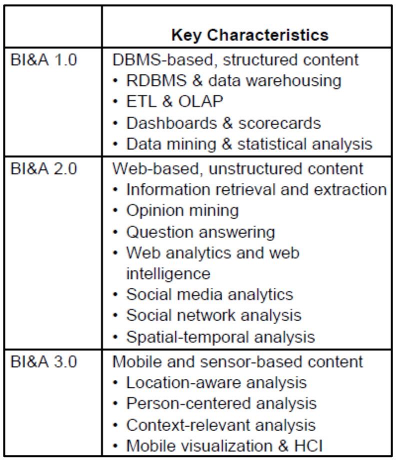
[21] CNBC (2019)

[22] Sherman (2015), p. 8-13

1. **Clean:** no missing items or invalid entries
2. **Consistent:** using the same sources and calculations to avoid different versions of the same data
3. **Conformed:** data needs to be analyzed across common, shareable dimensions
4. **Current:** decisions need to be based on whatever currency is necessary
5. **Comprehensive:** all relevant areas of data should be covered, regardless where it comes from and its level of granularity

A 2018 study among C-suite leaders from 207 large enterprises in North America and Europe found that 60% of those companies already investing in analytics reported significant correlated revenue growth, with 76% of them planning to further expand and/or modernize the underlying IT infrastructure to better support analytics. Here Business Intelligence applications are the top investment priority, taking the lead with 47% before artificial intelligence (40%) and data warehouse (38%).[23]

In scientific literature, there is no common definition of business intelligence, business analytics, big data or data science and these terms are used interchangeably. Taking business intelligence and analytics (BI&A) as a unified term, one can distinguish between several steps of development (see table 1). Under the era of BI&A 1.0, data was mostly structured and stored in traditional relational database management systems. Analysis was limited to simple performance metrics and graphics. BI&A 2.0 centered on web and text analytics for unstructured web content. With the number of mobile phones and tablets surpassing the number of laptops and PCs, BI&A 3.0 evolved and continued to include sensor data:



	Key Characteristics
BI&A 1.0	DBMS-based, structured content • RDBMS & data warehousing • ETL & OLAP • Dashboards & scorecards • Data mining & statistical analysis
BI&A 2.0	Web-based, unstructured content • Information retrieval and extraction • Opinion mining • Question answering • Web analytics and web intelligence • Social media analytics • Social network analysis • Spatial-temporal analysis
BI&A 3.0	Mobile and sensor-based content • Location-aware analysis • Person-centered analysis • Context-relevant analysis • Mobile visualization & HCI

Table 1: Stages of BI&A[24]

[23] Forbes/Cisco (2018), p. 4-10

[24] Hsinchun Chen et al. (2012) p. 5

One might argue that today we are about to enter the era of BI&A 4.0, with modern BI platforms covering the full analytic workflow from data preparation and ingestion to visual exploration and even building of statistical models. After initial setup, the involvement from IT staff should be considerably reduced, emphasizing self-service over various structured and unstructured data sources. Augmented analytics including NLP and machine learning automation are pushing advanced functionalities further towards business users, away from technical experts (see figure 3). The scarcity of data science skills on the labour market has become a significant barrier, and by automating many time-consuming and bias-prone tasks, augmented analytics expands the capabilities of those with more widely available skill sets. This will widely automate data preparation, analytics and model development and democratize insights from analytics to business roles.[25]

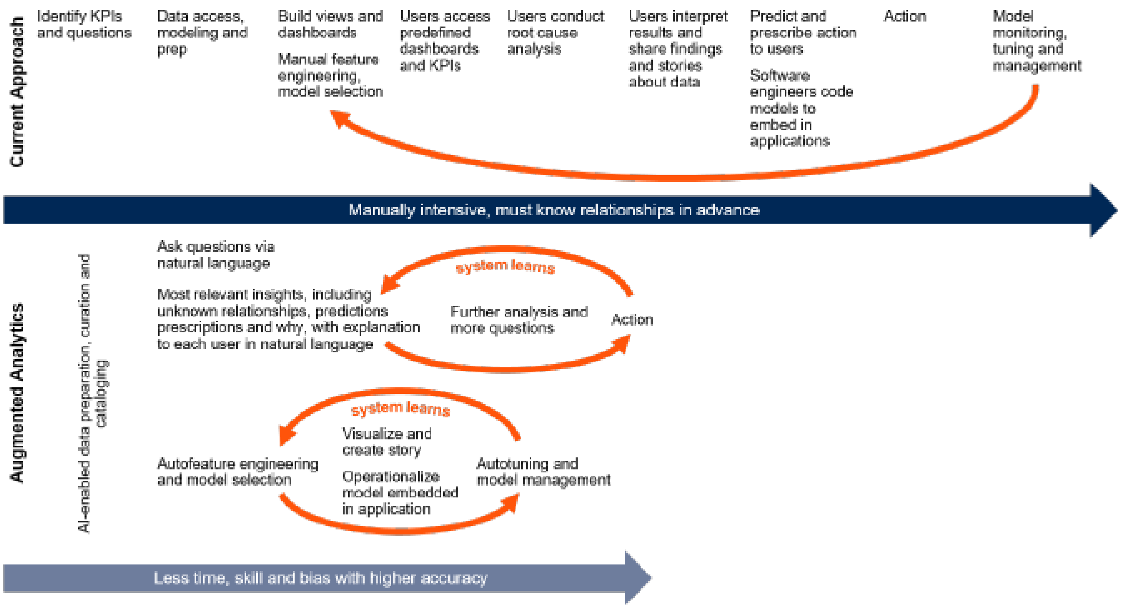

Current Approach
Identify KPIs and questions
Data access, modeling and prep
Build views and dashboards
Manual feature engineering, model selection
Users access predefined dashboards and KPIs
Users conduct root cause analysis
Users interpret results and share findings and stories about data
Predict and prescribe action to users
Software engineers code models to embed in applications
Action
Model monitoring, tuning and management
Manually intensive, must know relationships in advance
Augmented Analytics
AI-enabled data preparation, curation and cataloging
Ask questions via natural language
Most relevant insights, including unknown relationships, predictions prescriptions and why, with explanation to each user in natural language
system learns
Further analysis and more questions
Action
system learns
Visualize and create story
Autofeature engineering and model selection
Operationalize model embedded in application
Autotuning and model management
Less time, skill and bias with higher accuracy

Figure 3: How augmented analytics changes the workflow[26]

After gaining an understanding of how BI developed and how it is expanding to more and more functionalities that used to be within the sphere of competence of programmers and data scientists only, now it is time to analyse the market of both programming languages and BI platforms.

[25] Rita et al. (2018), p. 1

[26] Rita et al. (2018), p. 5

2.2 The rise of Python and Qlik Sense in Data Science

Python

A 2018 survey based on popularity on both the code-sharing platform *GitHub* and the developer community *Stack Overflow* documents the rise in popularity of high-level programming language Python, even outperforming R, which is still well-established for statistical programming (see figure 4):

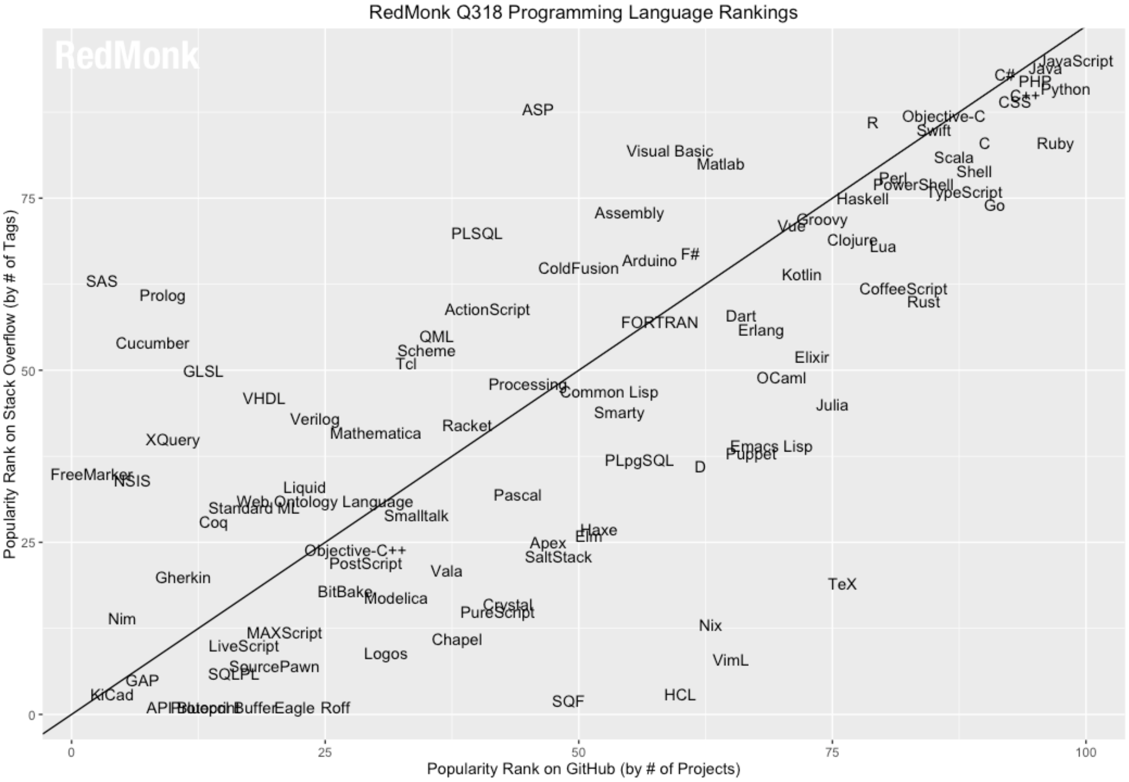
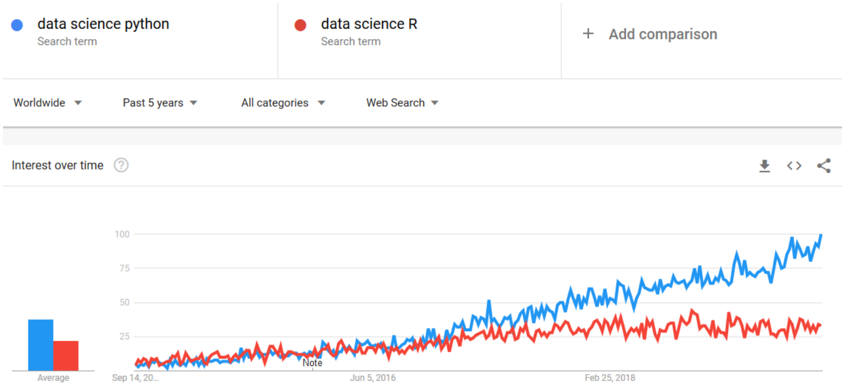


Figure 4: Popularity of Python[27]

Comparing the search string ‘Data Science Python’ to ‘Data Science R’ based on simple worldwide Google trends over the last five years results in a similar picture, underlying the rising popularity of Python (see figure 5):

[27] O'Grady (2018)



data science python
Search term

data science R
Search term

Worldwide

Past 5 years

All categories

Web Search

Figure 5: Python vs. R[28]

Python's success is based on its relatively easy-to-learn syntax, the active user community and its comprehensive functionalities especially in the field of data science.[29] With dozens of dedicated libraries available, the Python universe covers various application areas from programming video games over web crawling to data manipulation. For this thesis, the most relevant libraries can be summarized as follows (see table 2):

Library	Application area
Numpy	Manipulation of arrays and matrices
Pandas	Data analysis
Scipy	Scientific computing, e.g. linear algebra
Matplotlib	Plotting, e.g. histogram, scatter plot
Seaborn	Data visualization
Scikit-learn	Machine learning, data mining

Table 2: Python libraries[30]

Qlik Sense

In a recent study, technology research firm *Gartner* identified Microsoft PowerBI, Tableau and Qlik Sense as the three leading Business Intelligence platforms (see figure 6):

[28] Google Trends (2019)

[29] Pandey (2018)

[30] VanderPlas (2017)

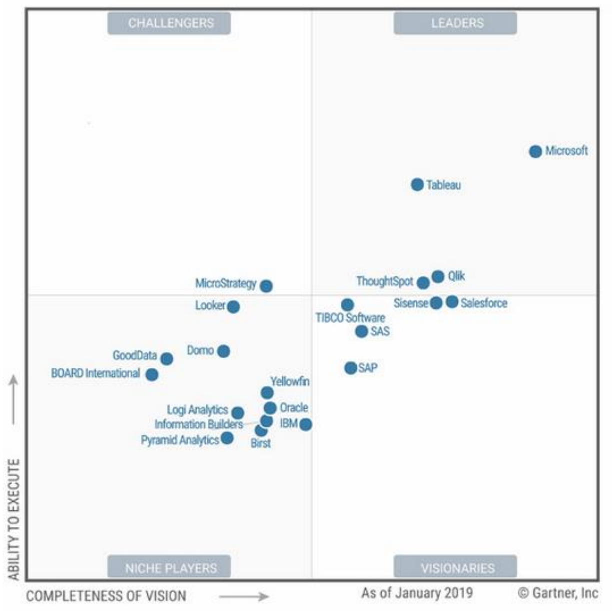


Figure 6: Market for BI platforms[31]

For this master's thesis Qlik Sense shall be compared against programming language Python. Again, this choice doesn't reflect any personal priorities and shall be based on objective criteria: Out of the three top performing BI platforms, Qlik Sense is the only independent product, with PowerBI belonging to the Microsoft universe and Tableau being taken over by Salesforce. The focus of PowerBI and Tableau might thus be the integration into their mother company in order to be compatible to complementary products. As discussed earlier, the market power of Microsoft and Salesforce might be a competitive advantage.

Nevertheless, according to *Gartner* the unique selling proposition of Qlik Sense is the strong performance with big volumes of data. With this volume growing exponentially, Qlik Sense holds a strong position in the market. All product strengths and weaknesses as compared to its competitors can be summarized as follows (see table 3):

Strengths	Weaknesses
1) Product features and extensibility: the in-memory engine supports multiple data sources, complex data models and complex calculations. This leads to scalable, robust and interactive visual applications	**1) Product workflow:** Besides flagship product Qlik Sense, Qlik offers several other analytic tools that need to be implemented separately

[31] Gartner Inc. (2019), p. 5

2) Customer experience: active user community and multiple conferences help to influence the market	**2) Migration experience:** Qlik's product range for analytic tools suffers from functional differences
3) Product vision: clear roadmap for new capabilities like a big data index and augmented analytics features	**3) Lower momentum:** a reduction of headcount in 2018 had negative impacts on customer perception

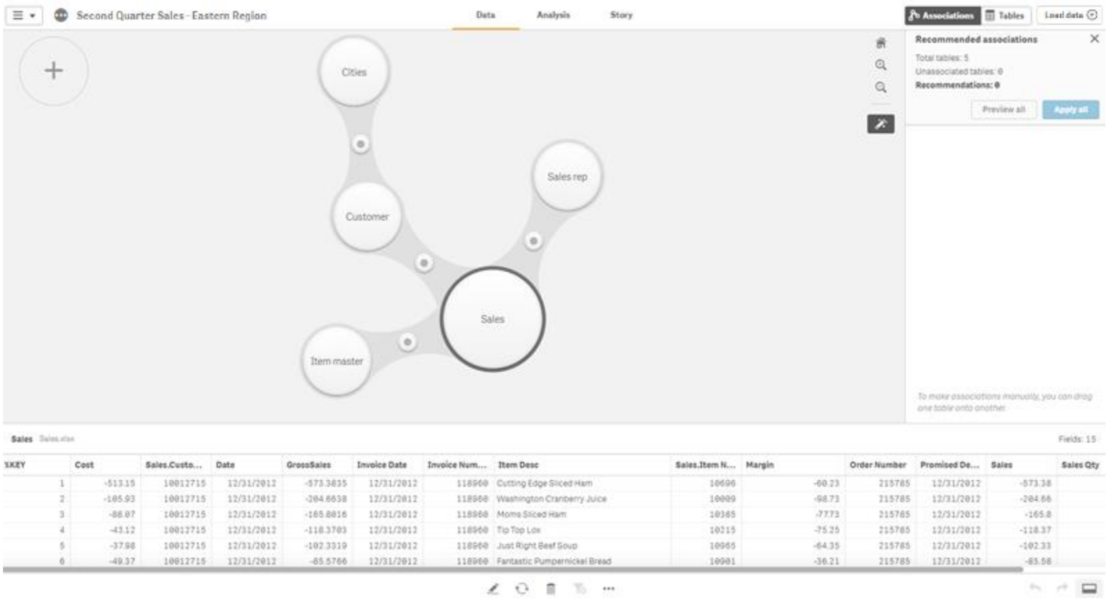
Table 3: Qlik Sense strengths and weaknesses[32]

Qlik Sense is constantly developing new product features and acquiring complementary software, furthermore various extensions are available from external partners such as Vizlib.[33] As a visual-based BI platform, even large amounts of data can be analysed, and patterns can be found in dozens of available visualization types. However, they don't automatically detect statistically significant findings, just visual relationships. Thus, manual interactive exploration using visualisations is the defining feature of visual-based BI platforms. This still requires user interpretation or further statistical analysis to determine which findings are relevant, significant and actionable.[34] Qlik Sense is composed of four major components[35] , whose functionalities will be compared to the relevant Python libraries in the empirical part of this thesis:

1. <u>ETL engine, data manager, script, data model</u>

Qlik Sense includes a built-in **ETL engine** that allows to connect various different data formats and extract data into Qlik Sense. This ETL engine can also be used to transform, manipulate, clean up and create new data. A lot of operations can be done right inside Qlik Sense, without any programming knowledge and strictly following the intuitive user interface called the **data manager**. Here the user can connect to data sources, select data to analyze, pull the data in and link it to other data based on identical data keys (see figure 7):

[32] Gartner Inc. (2019), p. 19-20
[33] Vizlib (2019a)
[34] Rita et al. (2018), p. 7
[35] Labbe et al. (2019), p. 10-23



Recommended associations

Total tables: 5

Unassociated tables: 0

Recommendations: 0

To make associations manually, you can drag one table onto another

Sales Sales.xlsx

Fields: 15

%KEY	Cost	Sales.Custo...	Date	GrossSales	Invoice Date	Invoice Num...	Item Desc	Sales.Item N...	Margin	Order Number	Promised De...	Sales	Sales Qty
1	-513.15	10012715	12/31/2012	-573.3835	12/31/2012	118960	Cutting Edge Sliced Ham	10696	-60.23	215785	12/31/2012	-573.38	
2	-105.93	10012715	12/31/2012	-204.6638	12/31/2012	118960	Washington Cranberry Juice	10009	-98.73	215785	12/31/2012	-204.66	
3	-88.07	10012715	12/31/2012	-165.8016	12/31/2012	118960	Moms Sliced Ham	10385	-77.73	215785	12/31/2012	-165.8	
4	-43.12	10012715	12/31/2012	-118.3703	12/31/2012	118960	Tip Top Lox	10215	-75.25	215785	12/31/2012	-118.37	
5	-37.98	10012715	12/31/2012	-102.3319	12/31/2012	118960	Just Right Beef Soup	10965	-64.35	215785	12/31/2012	-102.33	
6	-49.37	10012715	12/31/2012	-85.5766	12/31/2012	118960	Fantastic Pumpernickel Bread	10001	-36.21	215785	12/31/2012	-85.58	

Figure 7: Qlik Sense data manager[36]

For more advanced users, the Qlik Sense **script** provides the capability to extract and transform data using a scripting language similar to SQL, which allows a more granular control. Since this master's thesis is focussing on Qlik's ready-to-use capabilities compared to manual data analysis in Python, the script functionalities will not be dealt with in greater detail.

The **data model** gives a visual overview of all the data pulled into Qlik Sense, highlighting the relationships and linkages between different sources, including a preview of the data (see figure 8):

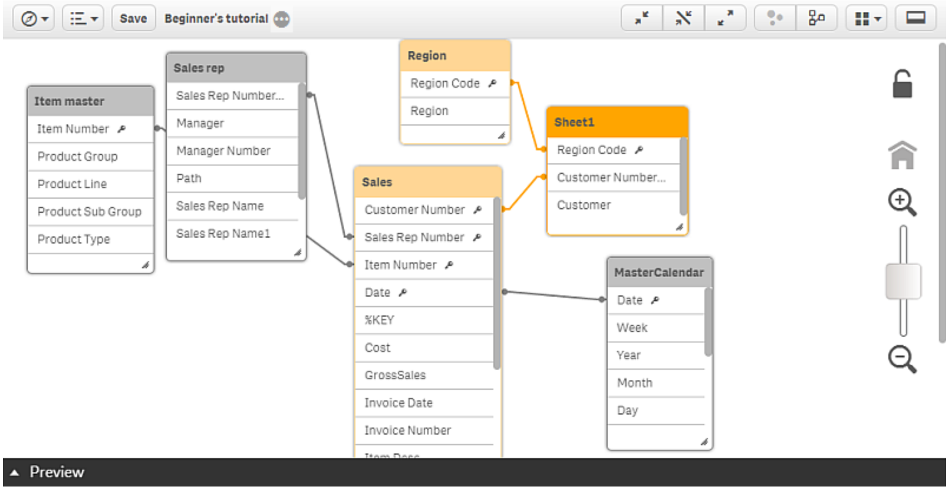


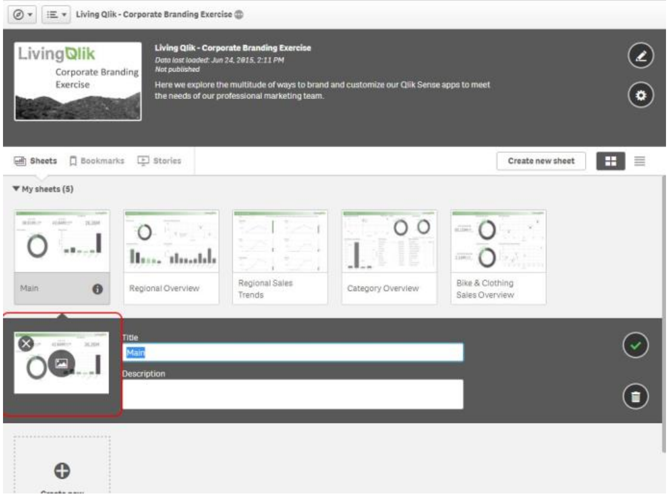
Figure 8: Qlik data model[37]

[36] QlikTech International AB (2019a)

[37] QlikTech International AB (2018b)

2. Visualization platform: the hub, applications, sheets, objects

Once data is loaded, the user can create visualizations on top of that data, e.g. KPIs, bar charts, tables or maps. The **hub** is the starting page when running Qlik Sense, where all existing **applications** are shown and new ones can be created (see figure 9). A Qlik Sense application contains several components, in particular a data model and related sheets with visualizations:


Living Qlik - Corporate Branding Exercise
LivingQlik
Corporate Branding Exercise
Living Qlik - Corporate Branding Exercise
Data last loaded: Jun 24, 2015, 2:11 PM
Not published

Here we explore the multitude of ways to brand and customize our Qlik Sense apps to meet the needs of our professional marketing team.

Sheets
Bookmarks
Stories
Create new sheet
My sheets (5)
Main
Regional Overview
Regional Sales Trends
Category Overview
Bike & Clothing Sales Overview
Title
Main
Description

Figure 9: Application overview[38]

Each **sheet** of an application is a collection of **objects** that can be customized, e.g. KPIs, scatter plots, bar charts, filter panes, maps etc. (see figure 10):

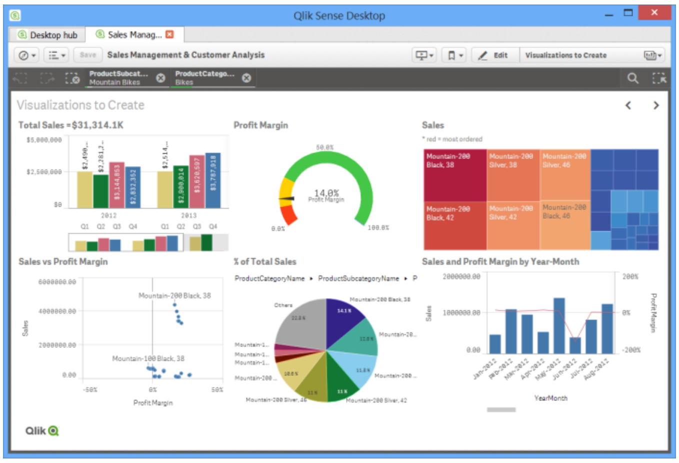


Figure 10: One sheet with several objects[39]

[38] Couron (2016)

[39] Redmond (2014)

3. <u>In-memory associative database, associative engine</u>

Qlik's **in-memory associative database** is the proprietary technology that allows large amounts of data to be compressed, stored in the RAM, and rapidly traversed in the Qlik Sense client. This database houses all the data needed inside the Qlik **associative engine**, which enables the user to not just navigate the data linearly, but also laterally, without a pre-defined path at the beginning of the analysis.

This *'slicing and dicing'* methodology facilitates analytics in the same way human brains work: analytics is a creative process that evolves during performing the analysis. What distinguishes Qlik Sense from many competitors, is the facilitation of the discovery process through a color scheme that highlights filter values throughout the application (see figure 11):

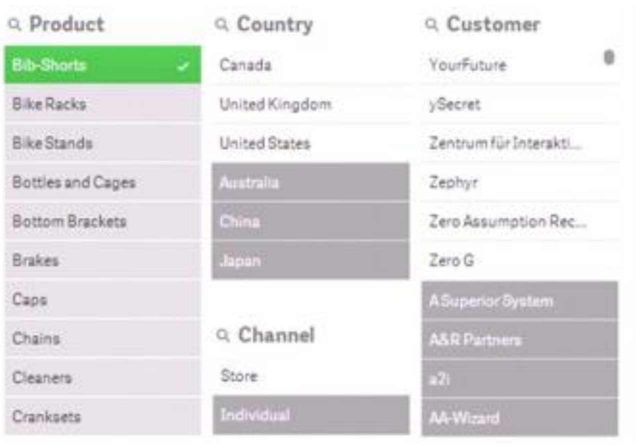


Figure 11: Qlik Sense Associative Engine[40]

In the preceding screenshot, the selected field value (product 'Bib-Shorts') is highlighted in green. The values highlighted in white are *associated with* the selected value, in this case the countries 'Canada', 'United Kingdom' and 'United States', channel 'Store' and six customers. The values highlighted in light grey are called *alternative values* and the ones in dark grey are called *excluded values*, meaning that they are not associated with the selections made in other fields.

4. <u>API and extensibility capabilities</u>

Qlik Sense comes with a comprehensive set of external extensions and APIs that developers can use to enlarge the capabilities and customize them to a company's needs. However, these possibilities are out of this thesis' scope.

[40] QlikTech International AB (2017)

Summary: BI platforms introducing advanced analytics to the mass market

The exponential growth of data volume worldwide has created a demand for new technologies extracting business value by combining and analysing datasets. Technically anything that can be done in a BI platform can be done via coding as well, however, not yet the other way around. Nonetheless, by applying theoretical concepts from the area of innovation research evidence could be found that BI might represent a new technological S-curve following pure manual coding, overcoming technical skill shortages on the labor market and implementing more and more functionalities in user-friendly environments. This goes along with the shift of the user base, away from highly technical data scientists and developers (‘innovators’ and ‘early adopters’) towards business roles (‘early majority’). Well-established data-driven companies are acquiring BI companies, and organizations from all industries are investing into BI products, both clearly leading the way to mass market adoption of BI&A, including advanced algorithms like clustering and forecasting.

So far, theoretical findings and empirical surveys underline the assumption that at least parts of the data science workflow are shifting from labor-intensive programming to partly automated BI-platforms. In order to address the research question under which criteria which tool is to be preferred, in the following chapter the concept of utility analysis will be introduced. The goal is to develop a framework that makes various tools comparable on the basis of predefined criteria.

2.3 Utility analysis for multidimensional evaluation of alternatives

Quantitative research is based on empirical data and statistical techniques, whereas qualitative research contains three main categories of non-standardized information: interviews, documents and observations/fieldwork. For the latter, field notes can be analyzed: detailed descriptions, including the context within which the observations were made.[41]

Due to the research gap and lack of empirical data, the comparison of a programming language with a BI tool can’t be based on quantitative methods only. Single criteria like price over product lifecycle or time needed to build a model may be measurable on a metrical scale. However, often the process and context need to be taken into consideration and can only be measured on a discrete scale, with nominal or ordinal characteristics. That’s why literature has developed so-called ‘mixed methods’, which make quantitative data combinable with qualitative findings.[42]

[41] Patton (2015), p. 14

[42] Merriam/Tisdell (2015), p. 43

Utility analysis is one of these ‘mixed methods’, a framework for multidimensional evaluation of alternatives based on their individual effectiveness. Integrating both quantitative and qualitative criteria, it helps to create a holistic picture for decision theory.[43] Advantages and disadvantages of this method are summarized in table 4:

Advantages	Disadvantages
➢ Several objectives can be pursued	➢ Monetary criteria are difficult to implement
➢ Differentiation of individual quantitative and qualitative criteria	➢ Weighting of criteria should be done by all decision-makers combined
➢ Making the decision transparent and traceable	➢ Data collection can be complex and time-consuming
➢ Different valuation rules grant some degree of flexibility	➢ Evaluation of alternatives can be simplistic

Table 4: Advantages and disadvantages of utility analysis[44]

Both Python and Qlik Sense have a vast number of possible applications, furthermore an almost infinite amount of publicly available datasets can be found on the internet. That’s why this thesis is limited to two case studies, each of them applying one trending algorithm to one dataset. The bounded context is crucial for each case study, making it a suitable research design for this thesis.[45]

Literature has developed a generic 7-step approach for utility analysis, each of which will be introduced and implemented in the following chapters to make Python and Qlik Sense comparable.[46]

1) Definition of objective

In this thesis the objective is defined as the choice between Python and Qlik Sense for data analysis, depending on individual prospects and limitations.

[43] Westermann (2012), p. 36

[44] Nagel (1990), p. 97

[45] Merriam/Tisdell (2015), p. 37

[46] Nagel (1990), p. 88-98

2) Definition of critical requirements

The only critical requirement in this case is that all tools analyzed need to cover the entire cross-industry standard process for data mining (CRISP-DM), which both Python and Qlik Sense do to some extent. This thesis will focus on the four core modules data understanding, data preparation, modeling and evaluation (see figure 12):

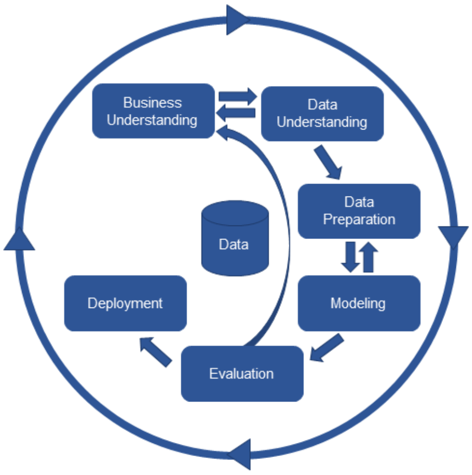
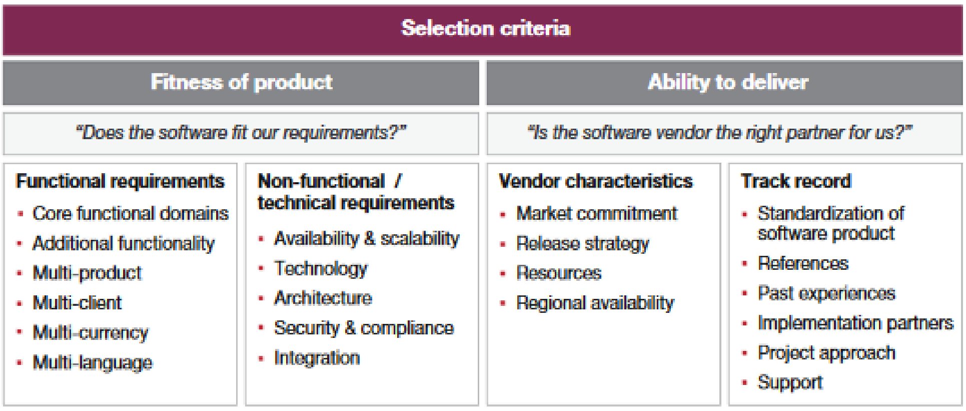


Figure 12: CRISP-DM model[47]

3) Definition of selection criteria

Researchers and practitioners developed several frameworks and templates for software selection processes. For example, IT-consultancy Capgemini distinguishes two main blocks of criteria, ‘Fitness of product’ and ‘Ability to deliver’ (see figure 13):



Selection criteria			
Fitness of product		Ability to deliver	
“Does the software fit our requirements?”		“Is the software vendor the right partner for us?”	
Functional requirements ▪ Core functional domains ▪ Additional functionality ▪ Multi-product ▪ Multi-client ▪ Multi-currency ▪ Multi-language	**Non-functional / technical requirements** ▪ Availability & scalability ▪ Technology ▪ Architecture ▪ Security & compliance ▪ Integration	**Vendor characteristics** ▪ Market commitment ▪ Release strategy ▪ Resources ▪ Regional availability	**Track record** ▪ Standardization of software product ▪ References ▪ Past experiences ▪ Implementation partners ▪ Project approach ▪ Support

Figure 13: Criteria for software selection[48]

[47] Taylor (2017)

[48] Capgemini Consulting (2016), p. 4

The ‘Ability to deliver’ can only be analyzed for a concrete business case, the same counts for the ‘technical requirements’ in the block ‘Fitness of product’. For the generalized approach of this thesis, the ‘functional requirements’ seem most relevant.

For the scope of this thesis, four steps of the CRISP-DM model (data understanding, data preparation, modeling and evaluation) will be broken down to the five functional criteria listed below, which are relevant for any data science project:

i. Speed:

   How long does each step take to be executed in Python and Qlik Sense?

ii. Trustworthiness:

   Are the results reliable?

iii. Flexibility:

   To what degree can this step be adapted to a slightly different situation? Can single parameters be tuned?

iv. Complementary features:

   Are there any additional features besides the core functionality that are easy to build in and bring additional value?

v. Usability:

   Which degree of technical understanding is necessary?

In a real-world business scenario price would always be a relevant criterion, covering the entire life cycle from installation to licenses, maintenance, customer support, etc. However, Python is free, just as a Qlik Sense student license. The total price of Qlik Sense strongly depends on the number of licenses and desired product features, which again can only be calculated on a case by case review.[49]

[49] QlikTech International AB (2019b)

4) Weighting of selection criteria

All criteria are weighted according to individual preferences. In this case all four steps of the CRISP-DM model shall be weighted equally, just as all five selection criteria. In a real-world scenario, the weighting process should be done by all decision-makers from all affected departments combined, since this is one of the crucial steps that affect the final result significantly. Similar to a scenario analysis the outcome of several different weightings can be compared to each other.

5) Compilation of alternatives

This thesis will focus on Python and Qlik Sense as the only two alternatives. This list can be extended, however, all alternatives should cover the critical requirement (see step 2).

6) Evaluation of alternatives

Evaluation of all alternatives requires a suitable scale, taking into consideration the relation between the values. Metrical scales are not applicable in this case, because they require a natural order and quantifiable distances. In principle this would be possible for the ‘speed’ criterion, e.g. how many minutes it takes to perform every task. However, checking the clock for every procedure is not realistic, and all other criteria need to be based on a discrete scale anyway. Trustworthiness, flexibility, complementary features and usability can best be evaluated on an ordinal scale, which facilitates a natural order of the outcome without quantifying the distances between single values. For this thesis all criteria are rated on an ordinal scale from 0 to 10, with 10 being the most favorable outcome.

7) Selection of the best alternative

All five criteria in all four CRISP-DM steps will be rated on an ordinal scale, leading to 20 datapoints for every alternative that in total represent the final result. The best alternative is the one with the highest final score. All seven steps for performing a utility analysis are summarized in table 5.

Summary: Utility analysis as an explorative approach

The 7 steps of utility analysis can be summarized in table 5 below:

Definition of objective 1			Choice of tool for data analysis			
Critical requirements 2			Coverage of CRISP-DM workflow			
			Alternatives 5			
			Qlik Sense		Python	
Selection criteria 3		Weighting 4	Rating	W*R 6	Rating	W*R
Data understanding		**25**				
	Speed	5				
	Trustworthiness	5				
	Flexibility	5				
	C. Features	5				
	Usability	5				
Data preparation		**25**				
	Speed	5				
	Trustworthiness	5				
	Flexibility	5				
	C. Features	5				
	Usability	5				
Modeling		**25**				
	Speed	5				
	Trustworthiness	5				
	Flexibility	5				
	C. Features	5				
	Usability	5				
Evaluation		**25**				
	Speed	5				
	Trustworthiness	5				
	Flexibility	5				
	C. Features	5				
	Usability	5				
Result 7		**100**				

Table 5: Evaluation table for utility analysis[50]

[50] On the basis of Nagel (1990), p. 94

In the following empirical part, the rating for every criterion in every CRISP-DM step will be developed on an ordinary scale from 0 to 10 and multiplied by the weighting factor (step 6 in table 5). In this case the weighting factor is identical for all criteria, but this can easily be adopted to individual requirements. The goal of this explorative approach is to develop general hypotheses under which criteria, and in which field Python or Qlik Sense is to be preferred for data analysis. These hypotheses can be deviated from the final scores (step 7 in table 5). Based on two case studies these hypotheses could then be subject of further research, e.g. via industry surveys.

The focus will not be on iterated model fitting, but on revealing the strengths and weaknesses of both tools over the entire CRISP-DM cycle. As of today, Qlik Sense supports two types of data mining techniques: cluster analysis and time series forecasting. More functionalities are likely to follow, however, this thesis will be based on those two which are also feasible in Python.

3 Empirical evaluation: Two case studies from the aviation industry

3.1 Clustering passengers and air freight over EU regions

I. <u>Business understanding</u>

The European Statistical Office (Eurostat) is responsible for providing harmonized statistical information to the institutions of the European Union in order to enable data-based comparisons and benchmarking between EU countries and regions over time.[51] In the 1970s, Eurostat set up the so-called NUTS classification (from French 'Nomenclature des unites territoriales statistiques'*)* as a coherent framework for dividing up the EU's territory into comparable entities and to produce regional statistics.

NUTS underlies strict quality standards and is updated every few years. The current NUTS 2016 classification lists all 28 EU member states, breaking them down to 104 regions at NUTS 1 level, 281 regions at NUTS 2 level and 1348 regions at NUTS 3 level (see figure 14):

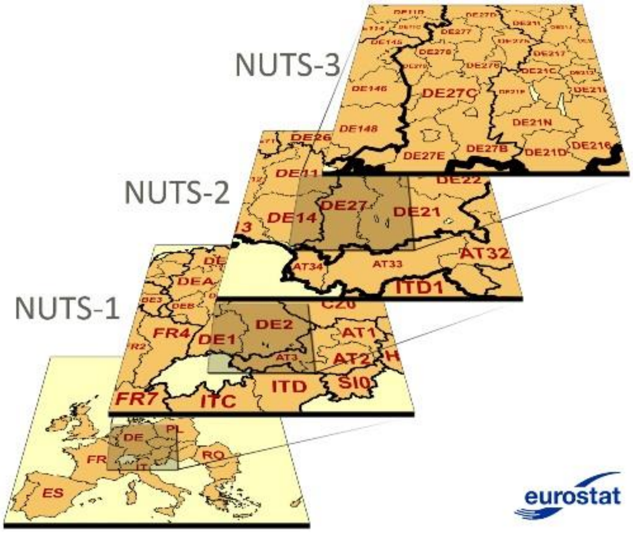
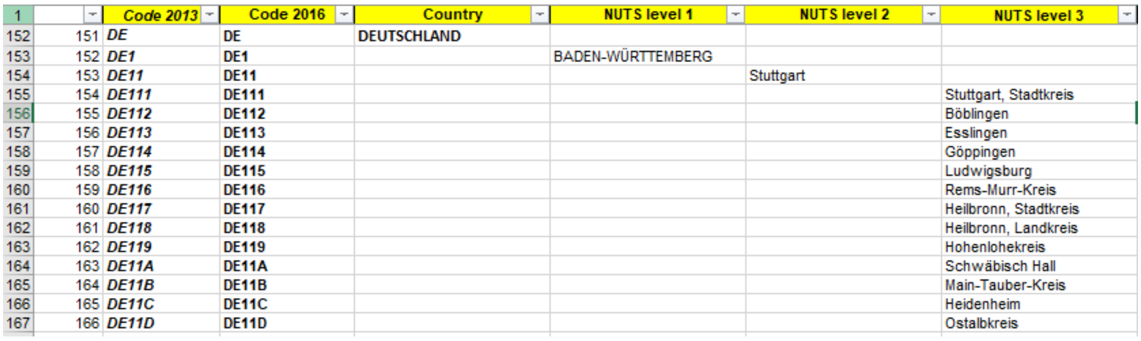


Figure 14: NUTS 2016 classification[52]

This hierarchical classification is the groundwork e.g. for socio-economic analysis of regions and the allocation of EU structural and cohesion funds. For Germany, NUTS level 1 defines the states ('Bundesländer'*)*, level 2 the government regions ('Regierungsbezirke') and level 3 the districts ('Landkreise'). This structure is exemplarily shown in figure 15:

[51] Eurostat (2019c)

[52] Eurostat (2019b)



1		Code 2013	Code 2016	Country	NUTS level 1	NUTS level 2	NUTS level 3
152	151	DE	DE	DEUTSCHLAND			
153	152	DE1	DE1		BADEN-WÜRTTEMBERG		
154	153	DE11	DE11			Stuttgart	
155	154	DE111	DE111				Stuttgart, Stadtkreis
156	155	DE112	DE112				Böblingen
157	156	DE113	DE113				Esslingen
158	157	DE114	DE114				Göppingen
159	158	DE115	DE115				Ludwigsburg
160	159	DE116	DE116				Rems-Murr-Kreis
161	160	DE117	DE117				Heilbronn, Stadtkreis
162	161	DE118	DE118				Heilbronn, Landkreis
163	162	DE119	DE119				Hohenlohekreis
164	163	DE11A	DE11A				Schwäbisch Hall
165	164	DE11B	DE11B				Main-Tauber-Kreis
166	165	DE11C	DE11C				Heidenheim
167	166	DE11D	DE11D				Ostalbkreis

Figure 15: NUTS hierarchy (sample data)[53]

Official data on air transport of passengers[54] and freight[55] can be found online for free in two separate datasets. The relevant NUTS level depends on the research purpose, for analyzing aviation data the aggregation on NUTS 2 level makes sense because some metropolitan areas have more than one airport with different passenger and freight ratios.

Based on two metric dimensions, passengers and air freight, the CRISP-DM workflow in Python will be compared to the one in Qlik Sense. The final goal is to identify three clusters in both tools for low, medium and high frequented regions in aviation. Every step will be evaluated according to the utility analysis table (developed in chapter 2.3), emphasizing relative strengths and weaknesses for both tools. All findings will be summarized at the end of the chapter.

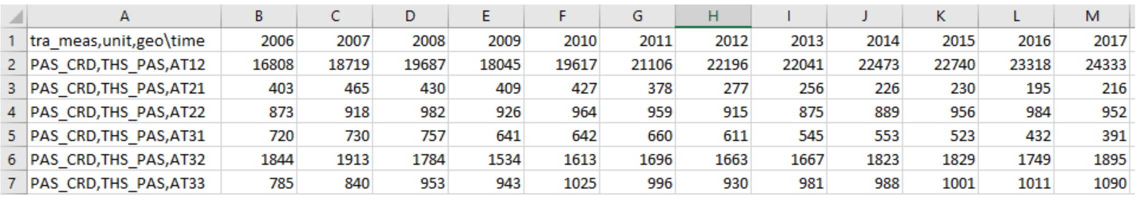
II. Data understanding

Both datasets for passengers and air freight have the same structure: the first column (A) contains spatial coding for the EU regions. The last 4 characters of this column represent the NUTS 2 classification and need to be isolated for better data understanding. All following columns (columns B to M) show the annual number of passengers in thousands (or air freight in thousand tons in the second dataset) per NUTS 2 region (see figure 16). Thus every line of both datasets stands for a time series of one NUTS 2 region, ranging from 2006 until 2017. Being interested in the most recent data, only 2017 is relevant for this clustering approach.

[53] Eurostat (2019a)

[54] EU Open Data Portal (2019b)

[55] EU Open Data Portal (2019a)



	A	B	C	D	E	F	G	H	I	J	K	L	M
1	tra_meas,unit,geo\time	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017
2	PAS_CRD,THS_PAS,AT12	16808	18719	19687	18045	19617	21106	22196	22041	22473	22740	23318	24333
3	PAS_CRD,THS_PAS,AT21	403	465	430	409	427	378	277	256	226	230	195	216
4	PAS_CRD,THS_PAS,AT22	873	918	982	926	964	959	915	875	889	956	984	952
5	PAS_CRD,THS_PAS,AT31	720	730	757	641	642	660	611	545	553	523	432	391
6	PAS_CRD,THS_PAS,AT32	1844	1913	1784	1534	1613	1696	1663	1667	1823	1829	1749	1895
7	PAS_CRD,THS_PAS,AT33	785	840	953	943	1025	996	930	981	988	1001	1011	1090

Figure 16: Passenger sample data[56]

In this first step, data understanding, for both Qlik Sense and Python the following steps will be performed for the passenger and the air freight dataset:

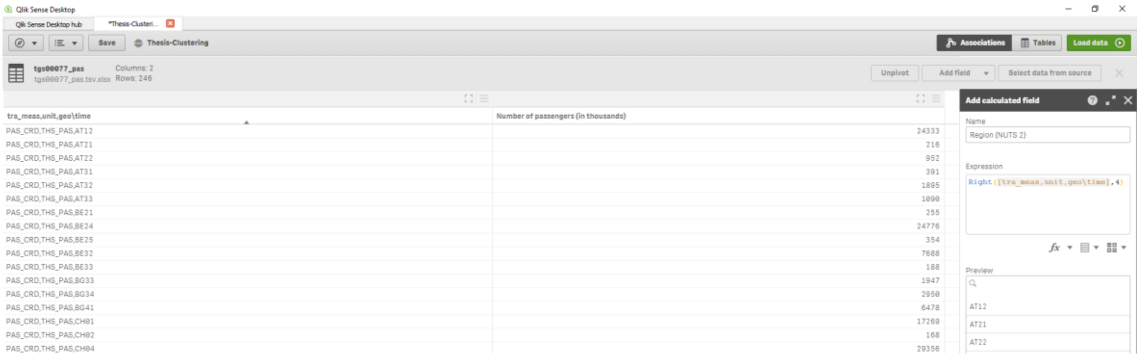
- Deleting all annual columns except the last one for 2017 data
- Renaming the column head to ‘Region (NUTS 2)’ for column A and ‘Number of passengers (in thousands)’ or ‘Air freight (in thousand tons)’ for column B, which now contains 2017 data
- Isolating the last 4 characters of the first columns for NUTS 2 classification
- Merging both datasets (passengers and air freight) based on the isolated NUTS 2 hierarchy (new column A) as key identifier
- Analyzing the merged dataset: missing values, correlation, outliers

The following analysis and figures will focus on Qlik Sense, because most of the operations that can be done here can also be done in Python – but not the other way around. The entire Python code and detailed documentation can be found in the enclosed Python script for replication. Due to the lack of space, only peculiarities will be highlighted.

<u>Qlik Sense</u>

When loading the dataset into Qlik Sense, the undesired columns (years 2006 to 2016) can simply be unselected by mouse click. Subsequently, in the data manager column heads can be renamed (see figure 17). In order to isolate the last four characters of column A as NUTS 2 code, a new column can be created using the function *Right([tra_meas,unit,geo\time],4):*

[56] EU Open Data Portal (2019b)



tgs00077_pas
tgs00077_pas.tsv.xlsx
Columns: 2
Rows: 246

tra_meas,unit,geo\time	Number of passengers (in thousands)
PAS_CRD,THS_PAS,AT12	24333
PAS_CRD,THS_PAS,AT21	216
PAS_CRD,THS_PAS,AT22	952
PAS_CRD,THS_PAS,AT31	391
PAS_CRD,THS_PAS,AT32	1895
PAS_CRD,THS_PAS,AT33	1090
PAS_CRD,THS_PAS,BE21	255
PAS_CRD,THS_PAS,BE24	24776
PAS_CRD,THS_PAS,BE25	354
PAS_CRD,THS_PAS,BE32	7688
PAS_CRD,THS_PAS,BE33	188
PAS_CRD,THS_PAS,BG33	1947
PAS_CRD,THS_PAS,BG34	2950
PAS_CRD,THS_PAS,BG41	6478
PAS_CRD,THS_PAS,CH01	17269
PAS_CRD,THS_PAS,CH02	168
PAS_CRD,THS_PAS,CH04	29356

Name: Region (NUTS 2)

Expression:
```
Right([tra_meas,unit,geo\time],4)
```

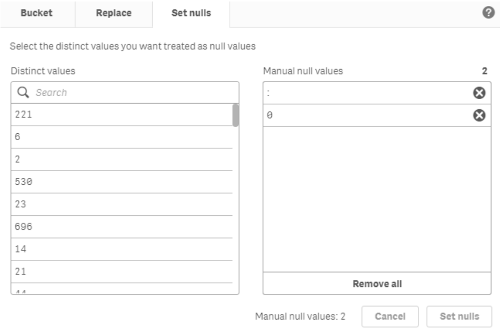
Preview:
AT12
AT21
AT22

Figure 17: Creating the column ‘Region (NUTS 2)’ in the data manager[57]

After repeating these steps for the air freight dataset, both files can be merged in the data manager, using the NUTS 2 code as unique key. For those 11 regions that are contained in the passenger dataset but not in the air freight dataset, Qlik Sense automatically imputes the related air freight values with 0 (and vice versa if applicable, technically a ‘full outer join’). Now analysis and visualizations can be performed for deeper data understanding, especially concerning missing values, correlation and outlier analysis.

For analysis of missing values, at first all observations of the dataset can be displayed in simple tables or bar charts to get an overview. This way, rows with ‘:’ can be identified as missing values, which can then be transformed to null values in the data manager. By using a KPI chart with the *NullCount-function* (for passengers: *NullCount([Number of passengers (in thousands)]),* 37 null values (out of 246) can then be identified in the passenger dataset and 104 (out of 235) in the air freight dataset.

The replacement of ‘:’ values with ‘0’ is important here (see figure 18), because the clustering algorithm needs real values to calculate Euclidian distances and doesn’t work with placeholders (for details see next chapters).



Select the distinct values you want treated as null values

Distinct values: 221, 6, 2, 530, 23, 696, 14, 21

Manual null values 2: :, 0

Manual null values: 2

Figure 18: Setting null values[58]

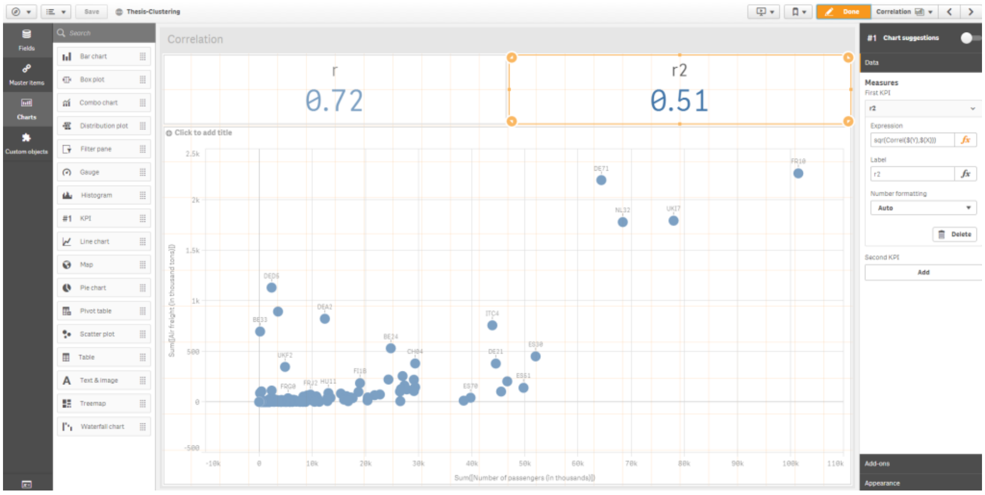
[57] Own figure, see attached Qlik Sense file

[58] Own figure, see attached Qlik Sense file

Missing values could be totally removed in the same step, but given the high quality of Eurostat data they should not be regarded as measurement errors and stay untouched at this point. In this case study missing values just represent regions without any passengers or air freight and deletion or imputation might affect the final cluster structure.

The relationship between both dimensions, passengers and air freight, can be examined by correlation analysis. Qlik Sense doesn't have a dedicated feature for correlation or regression analysis, but a workaround using a scatter plot and KPIs is possible (see figure 19):

,



Correlation

r
0.72

r2
0.51

Measures
First KPI
r2
Expression
sqr(Correl($(Y),$(X)))
Label
r2
Number formatting
Auto
Second KPI

Figure 19: Correlation analysis[59]

After defining the number of passengers as X variable and air freight as Y variable, the R^2 value can be calculated as KPI with the formula *sqr(Correl($(Y),$(X)))*. The moderate value of 0.51 (see figure 19) indicates that building clusters based on these two dimensions might reveal new insights that would remain hidden with univariate clustering only. For datasets with very strong correlation, the two-dimensional cluster structure is likely to be similar to the one-dimensional case.

Finally, the scatter plot in figure 19 proves the existence of outliers for both dimensions. Univariate outlier analysis for every dimension is essential because extreme values can easily cause nonrepresentative distortion of the cluster structure.[60] At this point it gets tricky: classic concepts for outlier analysis like z-value or 3s-rule (also called 68-95-99.7-rule) are only recommended for normally distributed datasets, what is clearly not the case for this right-skewed aviation dataset. Statistical tests like Shapiro-Wilk or Kolmogorov-Smirnov are not feasible in Qlik Sense, but plotting a histogram for both dimensions provides clarity (see figure 20):

[59] Own figure, see attached Qlik Sense file

[60] Liu et al. (2019), p. 1

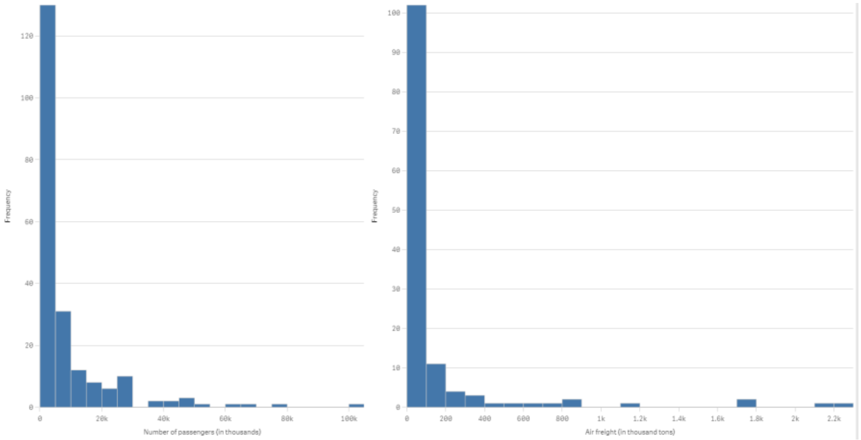


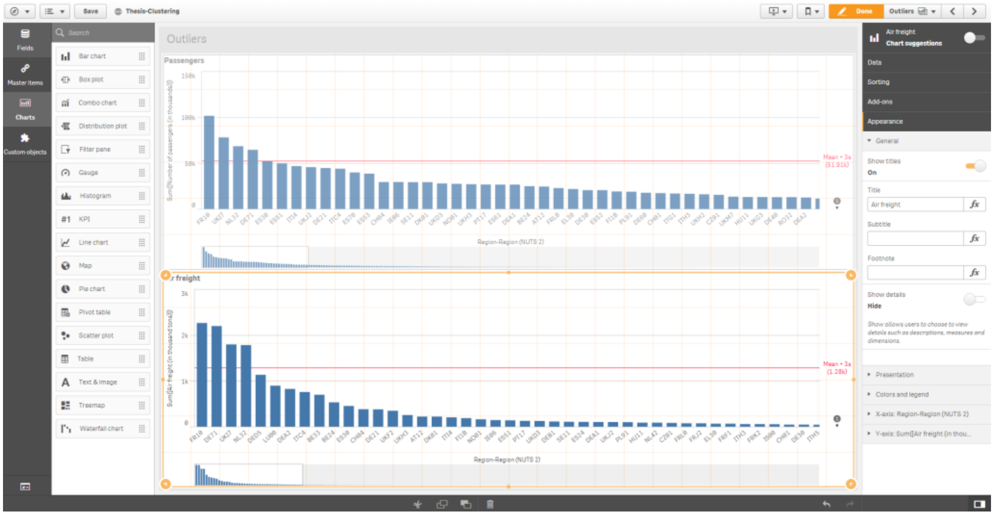
Figure 20: Histograms for both dimensions[61]

Clustering requires outlier analysis, and outlier analysis requires normally distributed values. But normal distribution is not an assumption for clustering[62], and any changes to the original dataset – like transforming all values towards normality - should be avoided in order to not affect the final model or make interpretation more difficult. Besides, moving values towards normality is technically not possible in Qlik Sense, a clear limitation of this tool.

Even though the dataset is right-skewed, the 3s-rule seems applicable here: When plotting the data for both variables, Qlik Sense offers the possibility to fit one or more reference lines (see the red line in figure 21) in a bar chart in order to e.g. highlight all values above the mean plus three standard deviations (Formula for passengers: *Avg([Number of passengers (in thousands)]) + 3*StDev([Number of passengers (in thousands)]))*. Because of the skewness of the datasets and the high standard deviation for both variables there are no outliers for the lower boundary, mean minus three standard deviations.

[61] Own figure, see attached Qlik Sense file

[62] Wendler/Gröttrup (2016), p. 591


Save
Thesis-Clustering
Done
Outliers
Fields
Master items
Charts
Custom objects
Search
Bar chart
Box plot
Combo chart
Distribution plot
Filter pane
Gauge
Histogram
KPI
Line chart
Map
Pie chart
Pivot table
Scatter plot
Table
Text & image
Treemap
Waterfall chart
Outliers
Passengers
150k
100k
50k
0
Sum([Number of passengers (in thousands)])
Mean + 3σ (51.91k)
FR10 UKI7 NL32 DE71 ES30 ES51 ITI4 UKJ2 DE21 ITC4 ES70 ES53 CH04 IE06 SE11 DK01 UKD3 NO01 UKH3 PT17 ES61 DEA1 BE24 AT12 FRL0 EL30 DE30 ES52 FI1B PL91 DE60 CH01 ITG1 ITH3 UKH2 CZ01 UKM7 HU11 UKG3 DE40 RO32 DEA2
Region-Region (NUTS 2)
r freight
3k
2k
1k
0
Sum([Air freight (in thousand tons)])
Mean + 3σ (1.28k)
FR10 DE71 UKI7 NL32 DED5 LU00 DEA2 ITC4 BE33 BE24 ES30 CH04 DE21 UKF2 UKH3 AT12 DK01 ITI4 FI1B NO01 IE06 ES51 PT17 UKD3 DEB1 SE11 ES24 DEA1 UKJ2 PL91 HU11 NL42 CZ01 FRL0 FRJ2 EL30 FRF1 ITH3 FRK2 IS00 CH01 DE30 ITH5
Region-Region (NUTS 2)
Air freight
Chart suggestions
Data
Sorting
Add-ons
Appearance
General
Presentation
Colors and legend
X-axis: Region-Region (NUTS 2)
Y-axis: Sum([Air freight (in thou...

Show titles
On

Title
Air freight

Subtitle

Footnote

Show details
Hide

Show allows users to choose to view details such as descriptions, measures and dimensions.

Figure 21: Outlier analysis[63]

All four outliers from the air freight dataset overlap with the five outliers from the passenger dataset, so in total five regions could be identified as extreme values: FR10 (Paris region), UKI7 (Outer London), NL32 (Amsterdam region), DE71 (Frankfurt region) and ES30 (Madrid region). They might represent an own cluster of exceedingly frequented regions and will be deleted for a clearer picture in the next step, data preparation.

Python

All steps performed in Qlik Sense can also be done in Python, the full script is attached to this thesis. For lack of space, just peculiarities shall be mentioned here:

By default, Python applies an 'inner join' when merging datasets. The passenger dataset contains rows that are not included in the air freight dataset (because some airports only serve passengers, no freight). For clustering we also want to take these regions into consideration, so a 'full outer join' needs to be applied, imputing missing values (by default 'NaN') with '0', because the clustering algorithm only works on real values.[64]

Besides histograms, in Python also statistical tests can easily be performed in order to test both dimensions for normal distribution. The Kolmogorov-Smirnov test shows a p-value of 0.0 both times, so the null-hypothesis of normal distribution can be rejected. In this case this is already obvious from the histograms, but deeper statistical functions are a clear advantage compared to the limited features of Qlik Sense. The same counts for transformation towards normality, which is not applied here but could be done in just a few lines of code.

[63] Own figure, see attached Qlik Sense file

[64] VanderPlas (2017), p. 177

In Python, outliers can be detected by the popular z-score method, subtracting the mean from every value and dividing it by the standard deviation.[65] This is shifting all values to the left and flattening the overall distribution, leading to a standardized distribution with mean of 0 and standard deviation of 1. Interestingly, compared to the 3s-rule applied in Qlik Sense, this method identifies one additional outlier for passengers (ES51: Barcelona region) and one additional outlier for air freight (DED5: Leipzig region). In Qlik Sense these two observations were the closest values right under the 3s-threshold and were just not identified as outliers, probably the small difference is based on internal rounding differences.

To sum up, in the next step (data preparation) 7 outliers will be removed following the Python analysis, versus 5 outliers in Qlik Sense. The current step of data understanding ends at this point, and the ratings for all five criteria can be filled into the utility analysis template developed in chapter 2.3 on an ordinary scale from 1-10. The evaluation in table 6 is based on the subjective experience made in this case study:

Definition of objective			Choice of tool for data analysis			
Critical requirements			Coverage of CRISP-DM workflow			
			Alternatives			
			Qlik Sense		Python	
Selection criteria		Weighting	Rating	W*R	Rating	W*R
Data understanding		**25**	**36**	**180**	**33**	**165**
	Speed	5	10	50	3	15
	Trustworthiness	5	10	50	10	50
	Flexibility	5	3	15	8	40
	C. Features	5	5	25	10	50
	Usability	5	8	40	2	10

Table 6: Evaluation of data understanding in the utility analysis template

For data understanding, Qlik Sense has a competitive edge, however, this is mostly due to the fairly clean datasets used in this case study that can quickly be pulled and visualized. Python on the contrary takes time to get used to, but afterwards the numerous features come as a reward.

[65] Khandelwal (2018)

III. Data preparation

The term “cluster analysis” stands for a set of algorithms developed to find subgroups in a dataset where similarities within each cluster are maximized and similarities between different clusters are minimized. The target value (cluster membership) is not known upfront, making this an unsupervised learning procedure.

Besides outlier removal, overall clustering performance can be improved by standardization and scaling. When looking at the scale of both charts in figure 21, one can see that the measuring unit (passengers in **thousands** versus air freight in **thousand tons**) is different, and the range differs significantly. Even after removing the five outliers in Qlik Sense mentioned above, it goes from 0 to 49.750 for passengers and from 0 to 1.130 for air freight.

In this case study the intention is to identify three clusters for low, medium and high frequented regions, based on two metrical dimensions passengers and cargo. Under these conditions, k-means is the algorithm of choice: the predefined number of clusters is shaped by minimizing the squared Euclidian distances around each cluster’s centroid (see formula in figure 22):

$$J = \sum_{i=1}^{k} \underbrace{\sum_{x_j \in C_i} \| x_j - \mu_i \|^2}_{(C)}$$

Figure 22: K-means minimization function[66]

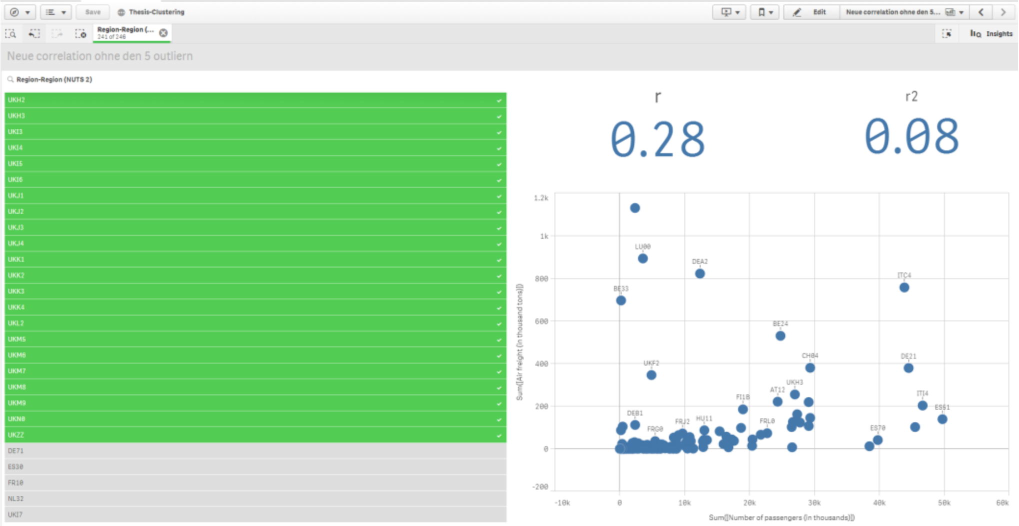
The outer sum runs through all clusters, the inner sum (C) for each element X_j of one cluster C_i represents the squared Euclidian distance from the centroid μ_i for every dimension (see next chapter for details). Function J intends to minimize this pairwise squared deviations of points in the same cluster, known as SSE (Sum of Squared Errors). Volatility and data range of each dimension strongly influence the clustering algorithm, so in a multi-dimensional case dimensions with high variance or outliers have a bigger impact on the model than dimensions with low variance and no outliers. Given the fact that in this case study both dimensions (passengers and air freight) should have the same impact on the model, after the removal of outliers Min-Max-Scaling is an appropriate measure in order to bring all dimensions to the same range.[67]

[66] Frochte (2019), p. 309

[67] Raschka/Mirjalili (2018), p. 357

Qlik Sense

Filtering and selecting values according to ad-hoc requirements is one of the big strengths of Qlik Sense. No operation needs to be repeated or alternated in the data manager, single values can be activated and deactivated right inside the dashboard. By adding a filter pane and deselecting the five outliers, the linked scatter plot and R^2 value immediately change (see figure 23):



Neue correlation ohne den 5 outliern

Region-Region (NUTS 2)

UKH2 ✓
UKH3 ✓
UKI3 ✓
UKI4 ✓
UKI5 ✓
UKI6 ✓
UKJ1 ✓
UKJ2 ✓
UKJ3 ✓
UKJ4 ✓
UKK1 ✓
UKK2 ✓
UKK3 ✓
UKK4 ✓
UKL2 ✓
UKM5 ✓
UKM6 ✓
UKM7 ✓
UKM8 ✓
UKM9 ✓
UKN0 ✓
UKZZ ✓
DE71
ES30
FR10
NL32
UKI7

r
0.28

r2
0.08

Figure 23: Correlation analysis after outlier removal[68]

When comparing this result to the original values (see figure 19), this proves the impact of outlier analysis before clustering.

Just like for transformation towards normality, Qlik Sense does not provide suitable operations for standardization or scaling. For this tool, data preparation ends at this point and clustering will be performed with the current state of the dataset, with 5 outliers removed from the dataset.

Python

Again, for the functions available Qlik Sense is extremely user-friendly. For advanced data preparation methods, however, it seems inappropriate. Python on the other hand requires some training and online research, but then offers plenty of methods for feature engineering. Z-standardization has already been performed for outlier analysis, and the 7 outliers identified should be removed before (!) scaling because the scikit-learn ‘MinMax Scaler’ is sensitive to extreme values. Subsequently, Min-Max-Scaling can be applied in order to bring all dimensions to the same range between 0 and 1, using the formula shown in figure 24:

[68] Own figure, see attached Qlik Sense file

$$x_{i,\ \text{new}} = \frac{x_i - x_{\min}}{x_{\max} - x_{\min}}$$

Figure 24: Formula for Min-Max-Scaling[69]

This step is key to prepare data for clustering and is expected to change results significantly when compared to the Qlik Sense approach (see figure 25):

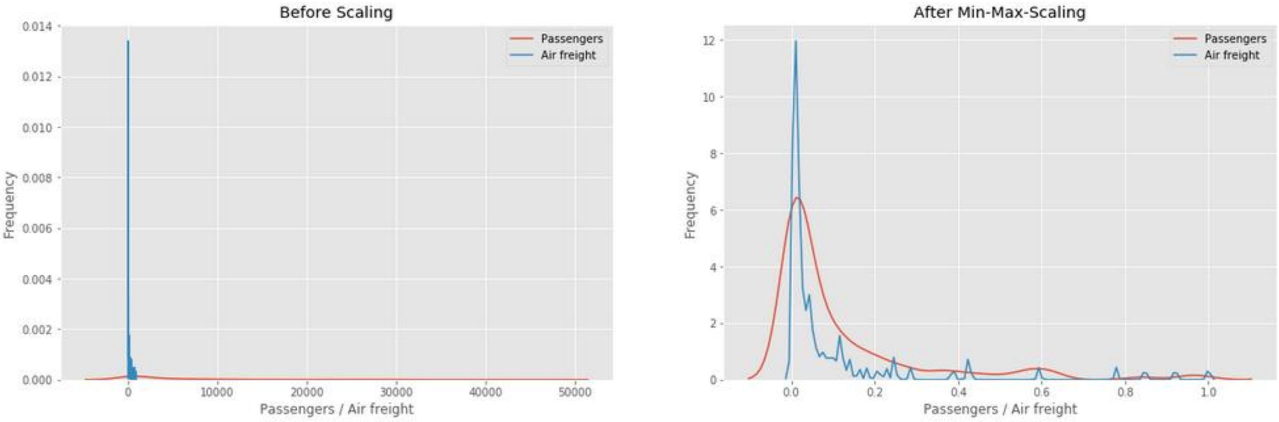


Figure 25: Distribution before and after Min-Max-Scaling[70]

The frequency distribution (absolute values) changes after Min-Max-Scaling, but the scatter plot (relative values) remains the same because both dimensions were scaled to the same range (see figure 26):

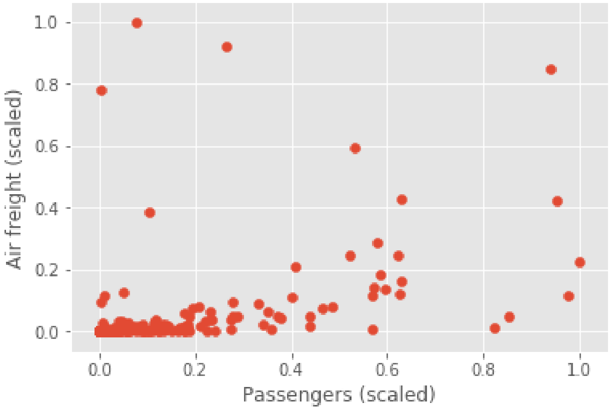


Figure 26: Scatter plot after Min-Max-Scaling[71]

[69] Wendler/Gröttrup (2016), p. 640

[70] Own figure, see attached Python script

[71] Own figure, see attached Python script

[Sidenote: Scaling does NOT affect the R and R^2 values, so the lower the correlation between the dimensions in the data understanding part, the more new insights can be generated by multi-dimensional clustering. In this case with moderate R^2 (0.51) it is still likely to lead to significantly different clusters than in a one-dimensional case.]

In this step of data preparation, the limitations of Qlik Sense become obvious. Outlier removal can be done easily right inside the operating dashboard with just a few mouse clicks, but the fact that standardization and scaling are not possible raises the question of how trustworthy the final clustering results can be. Business roles who are not familiar with essential model assumptions might come to misleading insights. So for the following steps the criterion of trustworthiness certainly plays a distinguished role. The evaluation of the current step, data preparation, can be summarized in table 7:

Definition of objective			Choice of tool for data analysis			
Critical requirements			Coverage of CRISP-DM workflow			
			Alternatives			
			Qlik Sense		Python	
Selection criteria		Weighting	Rating	W*R	Rating	W*R
Data preparation		**25**	**34**	**170**	**38**	**180**
	Speed	5	10	50	3	15
	Trustworthiness	5	6	30	10	50
	Flexibility	5	5	25	10	50
	C. Features	5	5	25	10	50
	Usability	5	8	40	3	15

Table 7: Evaluation of data preparation in the utility analysis template

IV. Modelling

Figure 27 shows the variety of existing clustering algorithms, whereas k-means belongs to the group of so-called iterative algorithms because it is bound to converge to a solution after a certain number of iterations:

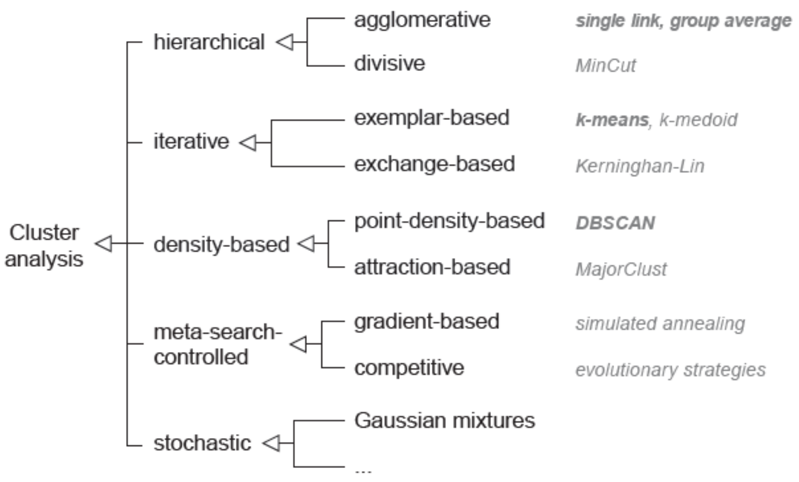


Figure 27: Categories of clustering algorithms[72]

Each category of clustering algorithms is applicable under different assumptions. When the number of clusters is known (here: three, for low, medium and high frequented regions) k-means is the algorithm of choice. The advantages and disadvantages of this particular algorithm are structured in table 8:

Advantages	Disadvantages
➢ Non-normally distributed variables can be used	➢ The number of clusters must be defined upfront by the user
➢ More flexible than agglomerative algorithms, based on the reassignment of the objects to other clusters	➢ Initial clustering is often based on heuristic methods, but initial clusters determine the quality of the final solution
➢ Clusters don’t overlap	➢ The order of cluster definition can’t be visualized in a tree (called Flat-clustering)

Table 8: K-means advantages and disadvantages[73]

After randomly initializing the three cluster centroids μ_1, μ_2, μ_3, the so-called *Expectation-Maximization-Algorithm* proceeds by repeating these two steps until either a maximum number of iterations took place, or a threshold for change in the recalculation of new cluster centers has been obtained:[74]

[72] Frochte (2019), p. 305

[73] Wendler/Gröttrup (2016), p. 591, 641

[74] VanderPlas (2017), p. 492

i. Expectation step: Assigns each observation to the cluster whose centroid has the least squared Euclidian distance.

ii. Maximization step: Calculates the centroids of the observations in the new clusters.

The Euclidian distance between each datapoint and a centroid is calculated as the absolute value of the difference, squared in order to make it more sensitive to extreme values. For every iteration, the distance of all datapoints (Qlik Sense: 246 regions minus 5 removed outliers; Python: 246 regions minus 7 removed outliers) to all three centroids is calculated, constantly reshaping the cluster structure until the breakup criterion is reached. A two-dimensional example after rescaling to the range between 0 and 1 might look like figure 28, visualizing the distance from random datapoint P_1 to centroid μ_1:

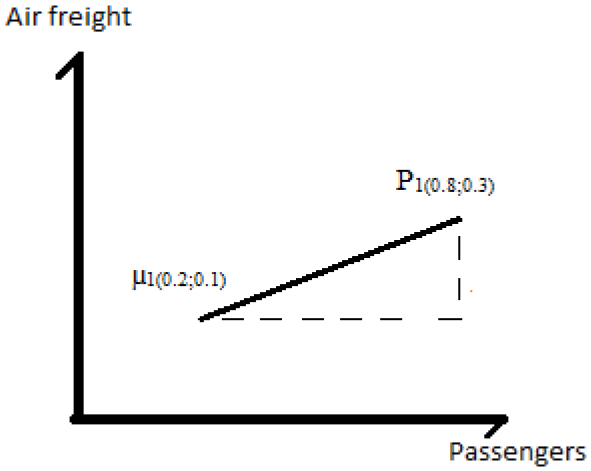


Figure 28: Euclidian distance[75]

The formula for squared Euclidian Distance (*d*) between the centroid μ_1 and datapoint P_1 in this example would be $d = (0.8 - 0.2)^2 + (0.3 - 0.1)^2$.

Every Expectation-Maximization-iteration may improve the result, however, k-means doesn't necessarily find a global optimum because the final cluster structure depends on the randomly initialized centroids. That`s why it is widely recommended to run the algorithm several times with different initial centroids.

Qlik Sense

The model in Qlik Sense is built after removing the identified five outliers, but without any feature scaling because it's technically not possible. Vizlib offers various extensions for Qlik Sense, one of which being a clustering function built into a scatter plot chart. Vizlib also offers the additional

[75] Own figure

feature of creating cluster area groups, with shaded areas indicating the value range for each cluster (see figure 29):

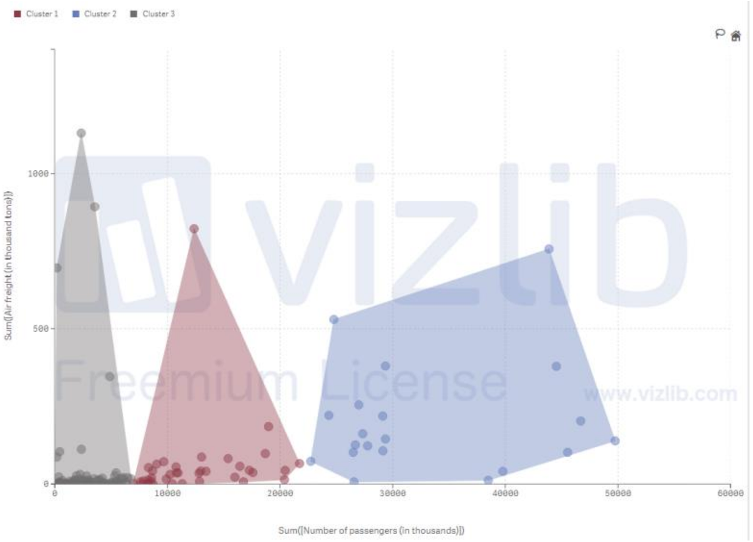


Figure 29: Clustering in the Vizlib extension for Qlik Sense[76]

One limitation of Qlik Sense is that there is no possibility for algorithm tuning at all, for example the number of iterations in k-means can't be set by the user. The clustering function is literally a black box, without any possibility to retrace the result or even modify essential parameters. Running the clustering multiple times does lead to different results in Qlik Sense, so the initial centroids do change (see the changing border between the red and blue cluster in figure 30):

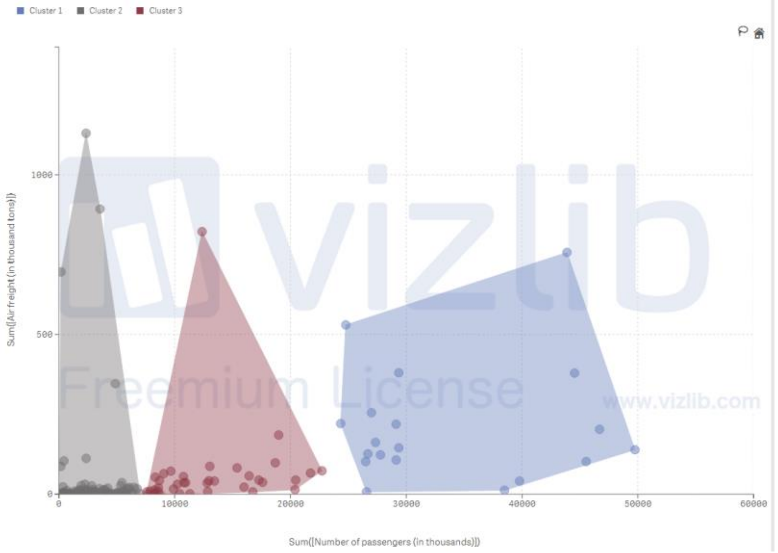


Figure 30: Alternative clustering[77]

[76] Own figure, see attached Qlik Sense file

[77] Own figure, see attached Qlik Sense file

Python

After Min-Max-Scaling both dimensions to a range between 0 and 1, the clustering result is expected to differ significantly from the Qlik Sense approach. In Qlik Sense the number of passengers has a way bigger impact on the cluster structure, because without scaling this dimension has a higher data range and volatility than air freight, which impacts the calculation of squared Euclidian distances in k-means. This can clearly be seen in figure 29, where all three clusters in Qlik Sense are aligned to the x-axis which represents the number of passengers.

In Python, after outlier removal and rescaling both dimensions to the same range, this should no longer be the case. Both dimensions now have the same impact on the model, which is now no longer based on absolute volatility but on relative volatility (see figure 31):

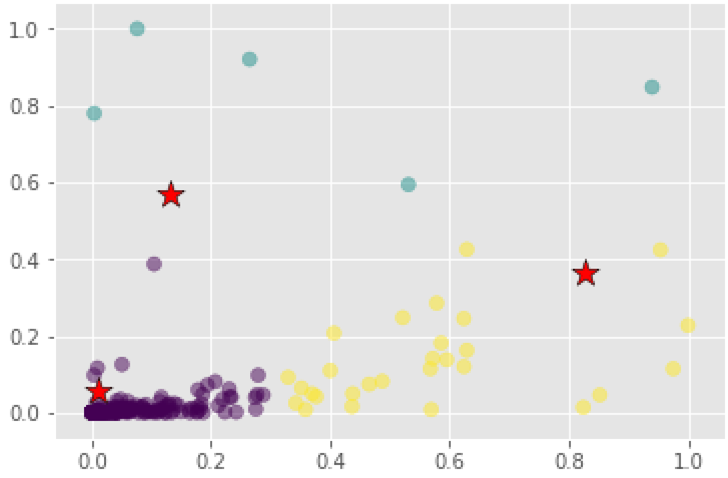


Figure 31: K-means clustering in Python[78]

The algorithm can be highly customized by the user, in this case the algorithm runs 10 times (n_init=10) and every time the initial centroid is randomly placed (init='random'). Within each iteration, the expectation and maximization steps run 100 times (max_iter=100) but automatically stops when converging earlier. The model with smallest Sum of Squared Error (SSE) is automatically chosen as the final model.[79]

Furthermore, the centroids of all three clusters can be highlighted (red stars in figure 31), so when running the algorithm several times changes can at least rudimentary be traced back.

As expected, both approaches lead to totally different cluster structures and can be evaluated in the utility analysis template (see table 9):

[78] Own figure, see attached Python script

[79] Raschka/Mirjalili (2018), p. 356

Definition of objective			Choice of tool for data analysis			
Critical requirements			Coverage of CRISP-DM workflow			
			Alternatives			
			Qlik Sense		Python	
Selection criteria		Weighting	Rating	W*R	Rating	W*R
Modelling		**25**	**22**	**110**	**36**	**180**
	Speed	5	10	50	3	15
	Trustworthiness	5	0	0	10	50
	Flexibility	5	1	5	10	50
	C. Features	5	1	5	10	50
	Usability	5	10	50	3	15

Table 9: Evaluation of modelling in the utility analysis template

In this step trustworthiness and flexibility become the explicit differentiators between Qlik Sense and Python. The results in Qlik Sense totally depend on volatility within every dimension and can't be blindly trusted. The built-in clustering function seems useful at first sight, but in fact can deliver wrong results and have devastating business impacts when economic decisions are based on them. On the other hand, clustering in Python offers a lot of transparency by parameter tuning.

V. Model Evaluation

One way to evaluate a k-means clustering model is the elbow method, which helps to find the optimal number of clusters for a dataset where the sum of squared errors (= 'distortion') gets minimized. Intuitively, this distortion decreases with every additional cluster because the more centroids are in the model, the closer all observations move to one of them. Theoretically, this way distortion could even approach zero, with one centroid for every observation in a one-dimensional case. However, the idea behind the elbow method is to identify a reasonable number of clusters, the point after which the curve descends to a more linear course.[80]

Qlik Sense

Qlik Sense doesn't offer any feature to evaluate the quality of the k-means clustering model.

[80] Raschka/Mirjalili (2018), p. 362

Python

Given that we want to identify 3 clusters for low, medium and high frequented regions, the elbow chart (see figure 32) confirms that three centroids are reasonable for this dataset (after Min-Max-Scaling):[81]

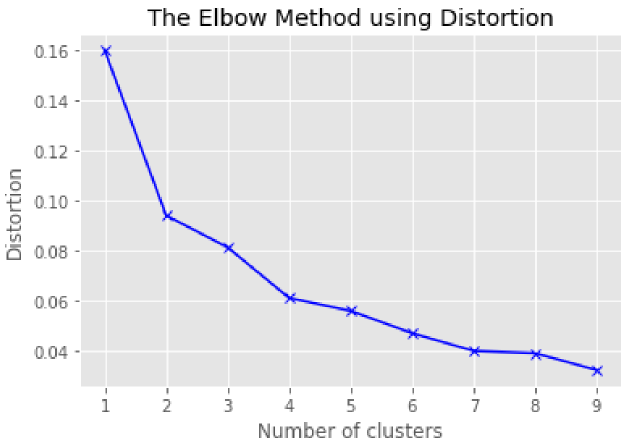


Figure 32: Elbow method to identify optimal number of clusters[82]

Since model evaluation isn't possible at all in Qlik Sense, all values are set to zero in the utility analysis template. Python again scores high with its reliable results and tailored functionalities (see table 10):

Definition of objective			Choice of tool for data analysis			
Critical requirements			Coverage of CRISP-DM workflow			
			Alternatives			
			Qlik Sense		Python	
Selection criteria		Weighting	Rating	W*R	Rating	W*R
Model evaluation		**25**	**0**	**0**	**38**	**190**
	Speed	5	0	0	3	15
	Trustworthiness	5	0	0	10	50
	Flexibility	5	0	0	10	50
	C. Features	5	0	0	10	50
	Usability	5	0	0	5	25

Table 10: Model evaluation in the utility analysis template

[81] GeeksforGeeks (2019)

[82] Own figure, see attached Python script

VI. Summary

The goal of this first case study is not to develop a perfect model or show all possible product features. As the thesis title indicates, the goal is to find prospects and limitations of both tools in order to get an understanding which step of the data science workflow can better be done in Qlik Sense or in Python.

For simple tasks especially in data understanding, Qlik Sense takes the lead because of its ready-to-use visualizations. However, when going into the details of data science, the limitations of this tool become evident. Especially the lack of any data scaling function rings the alarm bell in this case study, because scaling is THE essential step in data preparation for clustering. The results in Qlik Sense are not trustworthy and could put a company in serious economic danger when transformed into business decisions. Clustering can be automated in a dashboard, but business users without substantial knowledge in the field of data science might draw fatal conclusions from the misleading results. It has to be concluded that clustering in BI tools at this stage might do more harm than good. The final utility analysis for this first case study sums up the previous chapters (see table 11):

Definition of objective			Choice of tool for data analysis			
Critical requirements			Coverage of CRISP-DM workflow			
			Alternatives			
			Qlik Sense		Python	
Selection criteria		Weighting	Rating	W*R	Rating	W*R
Data understanding		**25**	**36**	**180**	**33**	**165**
	Speed	5	10	50	3	15
	Trustworthiness	5	10	50	10	50
	Flexibility	5	3	15	8	40
	C. Features	5	5	25	10	50
	Usability	5	8	40	2	10
Data preparation		**25**	**34**	**170**	**38**	**180**
	Speed	5	10	50	3	15
	Trustworthiness	5	6	30	10	50
	Flexibility	5	5	25	10	50
	C. Features	5	5	25	10	50
	Usability	5	8	40	3	15
Modeling		**25**	**22**	**110**	**36**	**180**

	Speed	5	10	50	3	15
	Trustworthiness	5	0	0	10	50
	Flexibility	5	1	5	10	50
	C. Features	5	1	5	10	50
	Usability	5	10	50	3	15
Evaluation		**25**	**0**	**0**	**38**	**190**
	Speed	5	0	0	3	15
	Trustworthiness	5	0	0	10	50
	Flexibility	5	0	0	10	50
	C. Features	5	0	0	10	50
	Usability	5	0	0	5	25
Result			**92**	**460**	**145**	**525**

Table 11: Final utility analysis for the first case study

When comparing how the column ‘weighting*rating’ changes over time, one can see that the performance of Qlik Sense starts high and then constantly decreases (highlighted in red). Python goes the other way around, starting relatively low and then outperforming Qlik Sense more and more. This is a clear indicator that BI tools are a good choice for initial analysis, but advanced functionalities are still dominated by programming languages.

The following second case study follows the same structure: Again, a dataset from the aviation industry will be analyzed, this time implementing a forecasting algorithm in both tools. Qlik Sense recently introduced a built-in predictive function for time-series, which will be compared to the Python approach and summarized in the same utility analysis template. Building this master’s thesis on two case studies from different fields will lead to more representative findings about prospects and limitations for Qlik Sense and Python.

3.2 Forecasting passengers of Munich airport

I. Business understanding

The number of passengers boarded by the global airline industry is constantly growing, 2019 is expected to set a new record with almost 4.6 billion travelers – an increase of 130 percent since

2004.[83] While this growth in major parts stems from the Asia-Pacific region, even central European airports with their omnipresent low-cost carriers are still expanding. Munich is one of those hubs still showing stable growth rates. With existing capacity shortages, the construction of a third runway is a recurring topic, but still in limbo due to political and societal opposition.[84]

By analyzing the trend of passenger transport at Munich airport in the past, the goal of this second case study is to develop a data-based prediction model and to give an outlook for the expectable increase of air traffic for the short-term future. Theoretically, this model might be considered by stakeholders when discussing the necessity and economic viability for the construction of a third runway.

II. Data understanding

A dataset containing the monthly number of passengers over the last years at Munich airport is not available for download, but can be compiled out of the annual and monthly traffic reports published on the airport’s website.[85]

Several steps are similar to the first case study and won’t be discussed again, making this second case study more consolidated. The focus will again be on new findings regarding prospects and limitations of Qlik Sense and Python, not on iterated model fitting.

Qlik Sense

After manually composing the monthly passenger data from January 2015 to September 2019 in Excel and loading the file into Qlik Sense (see first case study), figure 33 shows the resulting time series:

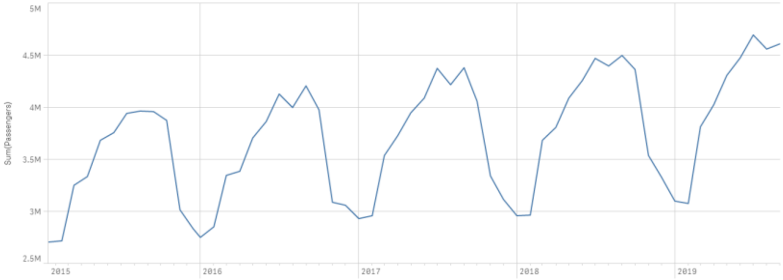


Figure 33: Time series of passengers at Munich airport[86]

[83] Statista (2019a)

[84] Stroh et al. (2018)

[85] Munich airport (2019)

[86] Own figure, see attached Qlik Sense file

Two insights are noteworthy at this point: Firstly, while a moderate growth rate over the years is visible (each monthly value is higher than in the year before), the time series shows a continuous pattern of volatility. The number of passengers clearly underlies seasonalities with annual lows in winter and peaks in summer.

Secondly, the dataset only consists of two columns: 57 monthly timestamps and the related number of passengers. There is no additional information like weather data or economic indicators that might have an impact on travel habits. Forecasting one dependent variable (the number of passengers) with one predictor (the month) based on a time series is a scenario of supervised learning: by analyzing the training data, the algorithm is supposed to learn the mapping function from the input to the output and extrapolate it to the unknown future.

For time series forecasting, the Qlik Sense extension Vizlib implemented the so-called *Holt-Winters-method*[87], which is taking into account three factors: baseline (called ‘level'), trend (here: moderate growth over the years) and seasonality (here: annual increase from January until July, followed by decrease). This method is also available in Python, making this comparison a suitable second case study.

Python

Python offers extensive features for time series analysis and visualization, some of which can be found in the script attached to this thesis. One powerful example is illustrating boxplots aggregated by months over the entire time series in order to get a deeper understanding of the degree of dispersion (see figure 34). July for example shows the highest median but also one of the highest spreads, making interpretation difficult – and maybe even forecasting, as we will see later.

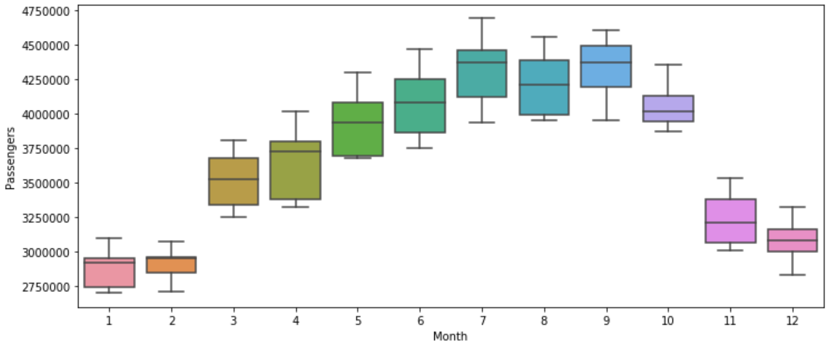


Figure 34: Boxplots by month over entire time series[88]

[87] Vizlib (2019b)

[88] Own figure, see attached Python script

The experience from this first step, data understanding, can be summarized in the well-known utility analysis template (see table 12). Just like in the first case study, for this swift analysis Qlik Sense wins by a narrow margin only because of its ready-to use chart types that make data exploration a child’s play:

Definition of objective			Choice of tool for data analysis			
Critical requirements			Coverage of CRISP-DM workflow			
			Alternatives			
			Qlik Sense		Python	
Selection criteria		Weighting	Rating	W*R	Rating	W*R
Data understanding		**25**	**36**	**180**	**35**	**175**
	Speed	5	10	50	3	15
	Trustworthiness	5	10	50	10	50
	Flexibility	5	3	15	10	50
	C. Features	5	5	25	10	50
	Usability	5	8	40	2	10

Table 12: Evaluation of data understanding in the utility analysis template

Compared to the first case study, only the Python value for ‘Flexibility’ changes from 8 to 10. Python seems incredibly powerful for time series analysis once the user gets familiar with its full potential.

III. Data preparation

Holt-Winters-forecasting requires a time series which is repetitive (here: 12-months cycles) at regular intervals (here: one month each) over several periods (here: 4 full years 2015-2018; one incomplete year January-September 2019).[89] Since the relatively small dataset was created manually it is already clean and fit for purpose. What’s left for this step is thinking about the model parameters.

There are two variations to this method which differ in the nature of the seasonal component: The ‘additive method’ is to be preferred when the seasonal variations are roughly constant through the series, e.g. 100.000 more passengers in February than in January. The ‘multiplicative method’ is

[89] Singh (2018)

to be preferred when the seasonal variations are changing proportionally to the level of the series, e.g. 10% more passengers in February than in January.[90]

Qlik Sense

In Qlik Sense and its Vizlib extension, the parameters can be set right in the GUI by mouse click. Since this tool addresses not just data scientists but a more general audience, the terminology is accordingly[91]:

- ‘Period definition’: Defines the number of datapoints that constitute a period. In this case 12 datapoints represent an annual period.
- ‘Number of training periods’: This is the number of periods the forecast will be based upon (= ‘training data’). Since our dataset covers 4 full years (= 48 out of the 57 available months), this number seems reasonable.
- ‘Number of points to forecast’: The maximum defined by Qlik is the period definition (here: 12) multiplied by the training periods (here: 4) multiplied by 0.5, so in this case 24 months.
- ‘Calculation model’: The forecast is always based on the Holt-Winters-method, but in addition to this the user can choose one out of four grades of intensity. The heavier the model, the slower the calculation time and the smoother the prediction. Unfortunately, there is no further documentation available, so the technical details remain opaque. Due to the relatively small dataset the slowest and most accurate model is chosen for this case study.
- There is no possibility to define a test dataset, as it might be expected for predictive models. There might be an internal procedure, but again no documentation can be found online or in the tool itself.

Python

[Important note: The Python library ‘Statsmodels’ provides classes and functions for implementing statistical models, amongst them exponential smoothing models like Holt-Winters. However, trying to import it in the script might cause an error, because the default version 0.8.0 first needs to be updated manually to version 0.9.0 in the Anaconda terminal. A tutorial can be found on Stackoverflow[92]].

[90] Hyndman/Athanasopoulos (2018)

[91] Vizlib (2019b)

[92] Stackoverflow (2018)

Different to Qlik, the dataset is divided into a training set (4 full years or 48 months) and a test set (remaining 9 months). The forecast is set to 24 months in order to make it comparable to the Qlik model. The ‘grade of intensity’ cannot be chosen in Python, however, the possible tuning parameters and model variations are extensive. Amongst others, the user can choose between the ‘additive’ and ‘multiplicative’ method or transform the input variables, e.g. via Box-Cox-Transformation. Since the focus of this thesis is not on model fitting but on highlighting prospects and limitations of Qlik Sense and Python, just some of these parameters shall be dealt with in detail.

The impression of both tools can be summarized in the utility analysis template (see table 13):

Definition of objective			Choice of tool for data analysis			
Critical requirements			Coverage of CRISP-DM workflow			
			Alternatives			
			Qlik Sense		Python	
Selection criteria		Weighting	Rating	W*R	Rating	W*R
Data preparation		**25**	**34**	**170**	**38**	**180**
	Speed	5	10	50	3	15
	Trustworthiness	5	6	30	10	50
	Flexibility	5	5	25	10	50
	C. Features	5	5	25	10	50
	Usability	5	8	40	3	15

Table 13: Evaluation of data preparation in the utility analysis template

The ratings from the first case study remain unchanged for data preparation, since they exactly reflect the experience made again in this second case study.

IV. Modelling

The Holt-Winters-method is based on triple exponential smoothing, meaning that each of the three underlying components (level, trend and season) are mathematically smoothened by multiplying (1- the smoothing factor α,β or γ) over the time series. The higher each smoothening coefficient, the more weight is given to the most recent observations, and vice versa. Figure 35 shows the composition of the additive method:

$$\begin{aligned} \ell_x &= \alpha(y_x - s_{x-L}) + (1-\alpha)(\ell_{x-1} + b_{x-1}) && \text{level} \\ b_x &= \beta(\ell_x - \ell_{x-1}) + (1-\beta)b_{x-1} && \text{trend} \\ s_x &= \gamma(y_x - \ell_x) + (1-\gamma)s_{x-L} && \text{seasonal} \\ \hat{y}_{x+m} &= \ell_x + mb_x + s_{x-L+1+(m-1)modL} && \text{forecast} \end{aligned}$$

Figure 35: Holt-Winters formulas, additive method[93]

In this notation,

- ‘level’ constitutes the baseline, with α as the smoothing factor whereas the sum of α and (1- α) always equals 1. The series is seasonally adjusted by subtracting the seasonal component. Within each year, the seasonal component adds up to approximately zero.[94]
- ‘trend’ constitutes the long-term orientation, with β as the smoothing factor and β+(1- β) = 1.
- ‘season’ constitutes the volatility within one period, with γ as the smoothing factor, L as the season length and s as the seasonal component. Again, γ+(1- γ) = 1.
- ‘forecast’ combines these three elements, with the index ‘x+m’ indicating the number of points into the future.

The challenge is to find the model that minimizes the root-mean-square error (RMSE, see next chapter for details) by tuning the model parameters like the smoothening factors α, β and γ.

Qlik Sense

After setting the parameters as explained in the last chapter, the resulting line chart (see figure 36) includes the prediction for the 24 months following the last timestamp (September 2019):

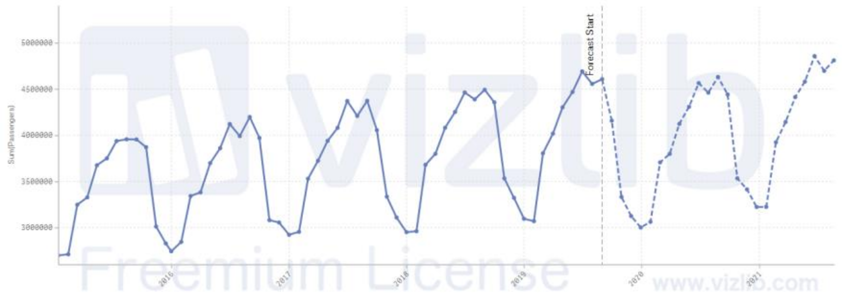


Figure 36: Vizlib forecasting[95]

[93] Trubetskoy (2016)

[94] Hyndman/Athanasopoulos (2018)

[95] Own figure, see attached Qlik Sense file

What strikes the eye is that even though there is a positive trend in the training data, this is not entirely considered in the forecast: the summer peak of 2020 is not higher than for 2019, but then growing again in 2021.

This raises the question how the trend and the smoothing factors are actually calculated, but Qlik Sense and Vizlib don't offer any possibility to trace or tune those parameters. Furthermore, it's also not possible to find out whether the 'additive' or 'multiplicative' method was used in the model. Just like in the first case study the result looks nice at the first glance, but many questions of detail remain unanswered.

Python

Other than in Qlik Sense, in Python it is possible to define train, test and forecast periods and plot all of them in the same graph (see figure 37):

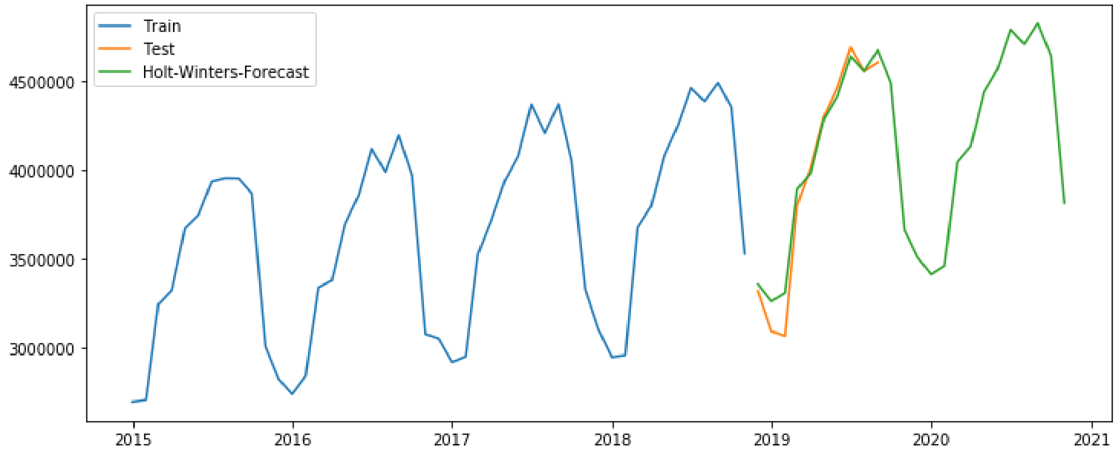


Figure 37: Python forecasting, additive method[96]

In this case the 'additive' method and default parameters for α, β and γ were used, without any further tuning. All of them can be changed manually, but even this result seems promising: the positive trend seems to be extrapolated correctly, and seasonality of test and forecast match to a satisfying degree. Python clearly outperforms Qlik Sense (see values in table 14):

[96] Own figure, see attached Python script

Definition of objective			Choice of tool for data analysis			
Critical requirements			Coverage of CRISP-DM workflow			
			Alternatives			
			Qlik Sense		Python	
Selection criteria		Weighting	Rating	W*R	Rating	W*R
Modelling		**25**	**25**	**125**	**36**	**180**
	Speed	5	10	50	3	15
	Trustworthiness	5	3	15	10	50
	Flexibility	5	1	5	10	50
	C. Features	5	1	5	10	50
	Usability	5	10	50	3	15

Table 14: Evaluation of modelling in the utility analysis template

In the first case study, trustworthiness for Qlik Sense was rated 0 because the clustering model delivered unreliable results. In this second case study, the prediction of Qlik Sense is not entirely wrong but doesn't reflect the positive trend reliably. Consequently, the rating can slightly improve from 0 to 3, with all other values remaining unchanged.

V. Model evaluation

The root-mean-square error (RMSE) represents the square root of the difference between the values predicted by the model and the real values observed, with lower RMSE values indicating a better model fit. This measure can be calculated in order to evaluate the model quality and compare different models used on the same dataset.

Qlik Sense

In this tool the length of the 'training' data can be set manually, but there is no way to define a 'test' dataset. Consequently, RMSE cannot be calculated which makes any comparison and model evaluation impossible.

Python

RMSE can be calculated for those 9 months reserved in the test dataset: When compared to those values predicted by the Holt-Winters model, RMSE of the additive method equals 104,049 (see attached Python script). The value itself can hardly be interpreted and only makes sense when compared to other prediction models trained and tested on the same dataset.

When starting to tune the Holt-Winters algorithm, a ‘damping factor’ can be implemented in order to exponentially reduce the trend over time. One of the limitations of the baseline model is that the trend is assumed to last forever, so this effect can be reduced at each period by a factor Φ. Just like the smoothening factors α, β and γ, the value of Φ is between 0 and 1.[97] After damping the first additive Holt-Winters model, RMSE slightly decreases to 104,018. However, a significant improvement (RMSE = 79,349) can be achieved when damping the multiplicative Holt-Winters model, where the series is seasonally adjusted by dividing through the seasonal component[98] (see figure 38). This indicates that the seasonal changes follow a more relative than absolute pattern over the entire time series, e.g. the increase of passengers from January to February rather resembles plus 10% than plus 100.000.

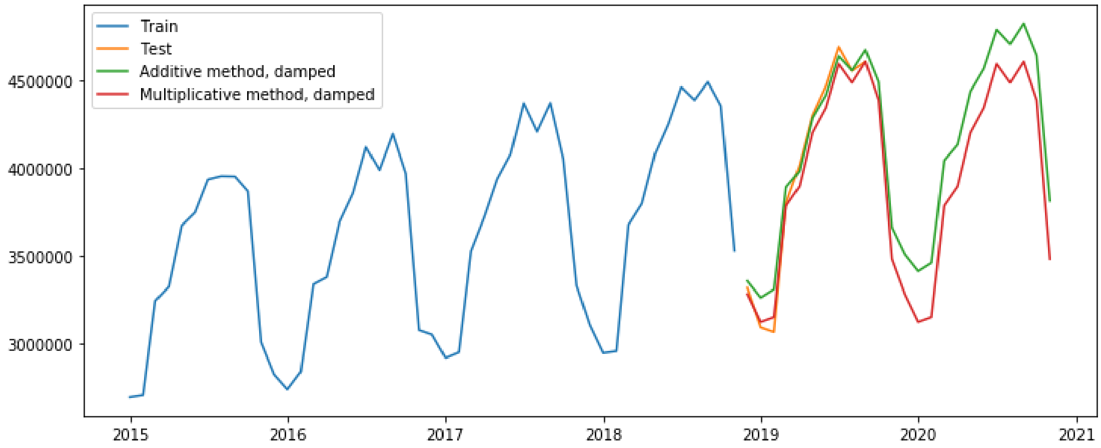


Figure 38: Damped additive method vs. damped multiplicative method

The default values of α, β, γ and Φ could be further tuned manually, but the potential of model evaluation in Python could be outlined and can now be summarized in the utility analysis template (see table 15):

[97] Vandeput (2018)

[98] Hyndman/Athanasopoulos (2018)

Definition of objective			Choice of tool for data analysis			
Critical requirements			Coverage of CRISP-DM workflow			
			Alternatives			
			Qlik Sense		Python	
Selection criteria		Weighting	Rating	W*R	Rating	W*R
Model evaluation		**25**	**0**	**0**	**43**	**215**
	Speed	5	0	0	8	40
	Trustworthiness	5	0	0	10	50
	Flexibility	5	0	0	10	50
	C. Features	5	0	0	10	50
	Usability	5	0	0	5	25

Table 15: Model evaluation in the utility analysis template

Just like in the first case study, Qlik Sense scores 0 because the model can’t be evaluated in any way. For Python only the value for ‘Speed’ improves from 3 to 8, because RMSE can be calculated with only a few lines of code.

VI. Summary

This second case study confirms the findings from the first one: Qlik Sense has its strengths in the area of data understanding thanks to its GUI and the available ready-to-use chart types. But even state-of-the-art BI tools seem inappropriate for advanced analytical functions like forecasting. The model delivers results out of a black box, without major possibilities for the user to interfere. Especially the lack of any model evaluation can be misleading if the user has no deeper knowledge of the techniques used. The final overview in table 16 is quite similar to the one in the first case study:

Definition of objective			Choice of tool for data analysis			
Critical requirements			Coverage of CRISP-DM workflow			
			Alternatives			
			Qlik Sense		Python	
Selection criteria		Weighting	Rating	W*R	Rating	W*R
Data understanding		**25**	**36**	**180**	**35**	**175**
	Speed	5	10	50	3	15
	Trustworthiness	5	10	50	10	50
	Flexibility	5	3	15	10	50
	C. Features	5	5	25	10	50
	Usability	5	8	40	2	10
Data preparation		**25**	**34**	**170**	**38**	**180**
	Speed	5	10	50	3	15
	Trustworthiness	5	6	30	10	50
	Flexibility	5	5	25	10	50
	C. Features	5	5	25	10	50
	Usability	5	8	40	3	15
Modeling		**25**	**25**	**125**	**36**	**180**
	Speed	5	10	50	3	15
	Trustworthiness	5	3	15	10	50
	Flexibility	5	1	5	10	50
	C. Features	5	1	5	10	50
	Usability	5	10	50	3	15
Evaluation		**25**	**0**	**0**	**43**	**215**
	Speed	5	0	0	8	40
	Trustworthiness	5	0	0	10	50
	Flexibility	5	0	0	10	50
	C. Features	5	0	0	10	50
	Usability	5	0	0	5	25
Result			**95**	**475**	**152**	**750**

Table 16: Final utility analysis for the second case study

Again, Qlik Sense wins by a narrow margin in the data understanding section, but the further down the data science workflow, the worse the performance gets. Python works exactly the other way around and can even further expand its leadership when comparing the final score.

In the last part of this thesis all findings are summarized, closing the loop by addressing the initial research questions.

4 Summary and findings

This thesis aimed to systematically compare the performance of programming languages against those of BI tools in order to identify areas of relative strength and weakness along the entire data science workflow according to the CRISP-DM framework.

Based on two empirical case studies, one on clustering and one on forecasting, empirical proof could be found under which criteria and in which field Qlik Sense (representing BI tools) or Python (representing programming languages) is to be preferred for data analysis. Referring to the final utility analysis tables for both case studies (see tables 11 and 16), this research question can be answered as follows: For those areas that are fully covered by Qlik Sense, this tool has clear advantages. Ready-to use chart types and a user-friendly graphical user interface address a broad user base, far bigger than the relatively small community of data scientists and programmers. However, this already contains the constraint: in such a complex and dynamic field as data science, it seems impossible to sufficiently cover all areas of application. High speed and user-friendliness of Qlik Sense can’t compensate for the lack of trustworthiness, flexibility and complementary features necessary for complex datasets and advanced data mining algorithms. Above all, the lack of any possibility to evaluate the model quality constitutes a major drawback when compared to the infinite Python functionalities.

Qlik Sense takes the lead for the first step of data understanding like calculation of KPIs and data visualization and has the potential to add significant value to a broad range of business users. But from the step of data preparation onwards, Python clearly sets the standards. Qlik Sense started integrating statistical models, but for the near future trustworthy results remain within the sphere of data experts: For both clustering and forecasting, the Qlik Sense models are not convincing.

At this point of time BI tools can distinctly not be seen as a new technological S-curve following manual coding. However, the existing labor market shortages for ‘data workers’ will continue to drive innovation, and the latest acquisitions by industry leaders might just be the beginning of a new era of technological disruption. After further research and development efforts BI platforms might become reliable office products, diving deeper and deeper into the data science toolkit and fuel the shift from innovators to the majority of business users.

One needs to highlight that these findings were deduced from two case studies only and can hardly be generalized. They provide a first assessment, but based on this explorative approach further studies (e.g. via industry survey) would be necessary in order to build robust hypotheses.

Bibliography

Baseman, K., Warren-Boulton, F. und Woroch, G. (1995), “MICROSOFT PLAYS HARDBALL. The Use of Exclusionary Pricing and Technical Incompatibility to Maintain Monopoly Power in Markets for Operating System Software”, *Antitrst Bulletin,* Vol. 40 No. 2, ff. 265-315.

Calzada, J. und Marzal, M. (2013), “Incorporating Data Literacy into Information Literacy Programs. Core Competencies and Contents”, *Libri,* Vol. 63 No. 2.

Capgemini Consulting (2016), “Software Selection. Managing the complexity of choosing the right software”.

CNBC (2019), “Salesforce bets on big data with $15.3 billion Tableau buy”, abrufbar unter: https://www.cnbc.com/2019/06/10/salesforce-to-buy-tableau-software-in-an-all-stock-deal.html (letzter Zugriff 3. Oktober 2019).

Couron, A. (2016), “How To: Simple Qlik Sense Branding”, abrufbar unter: http://livingqlikview.com/simple-qlik-sense-branding/ (letzter Zugriff 3. August 2019).

Domo (2018), “Data never sleeps”, abrufbar unter: https://www.domo.com/solution/data-never-sleeps-6 (letzter Zugriff 26. Juli 2019).

EU Open Data Portal (2019a), “Air transport of freight by NUTS 2 regions”, abrufbar unter: http://data.europa.eu/euodp/data/dataset/s3wF94CHaHnypErG74KSQ (letzter Zugriff 11. Oktober 2019).

EU Open Data Portal (2019b), “Air transport of passengers by NUTS 2 regions”, abrufbar unter: http://data.europa.eu/euodp/data/dataset/UdCvojYi5hT1OudPSGKoA (letzter Zugriff 11. Oktober 2019).

Eurostat (2019a), “History of NUTS - Eurostat”, abrufbar unter: https://ec.europa.eu/eurostat/web/nuts/history (letzter Zugriff 11. Oktober 2019).

Eurostat (2019b), “NUTS Background - Eurostat”, abrufbar unter: https://ec.europa.eu/eurostat/web/nuts/background (letzter Zugriff 11. Oktober 2019).

Eurostat (2019c), “Overview - Eurostat”, abrufbar unter: https://ec.europa.eu/eurostat/about/overview (letzter Zugriff 11. Oktober 2019).

Forbes und Cisco (2018), “Advanced Analytics: The key to becoming a data-driven enterprise”.

Frochte, J. (2019), *Maschinelles Lernen: Grundlagen und Algorithmen in Python,* 2. Auflage, Hanser, Carl, München.

Gal's insights (2015), “The Innovation S-Curve”, abrufbar unter: http://www.galsinsights.com/the-innovation-s-curve/ (letzter Zugriff 28. Juli 2019).

Gartner Inc. (2019), “Magic Quadrant for Analytics and Business Intelligence Platforms”, abrufbar unter: https://www.gartner.com/en/documents/3900992 (letzter Zugriff 28. Juli 2019).

GeeksforGeeks (2019), “Elbow Method for optimal value of k in KMeans”, abrufbar unter: https://www.geeksforgeeks.org/elbow-method-for-optimal-value-of-k-in-kmeans/ (letzter Zugriff 30. Oktober 2019).

Google Trends (2019), “Data Science in Python vs. Data Science in R”, abrufbar unter: https://trends.google.com/trends/explore?date=today%205-y&q=data%20science%20python,data%20science%20R (letzter Zugriff 13. September 2019).

Gottlieb, J. und Weinberg, A. (2019), “Catch them if you can: How leaders in data and analytics have pulled ahead”.

Heller, M. (2017), “10 hot data analytics trends — and 5 going cold”, abrufbar unter: https://www.cio.com/article/3213189/10-hot-data-analytics-trends-and-5-going-cold.html (letzter Zugriff 13. September 2019).

Hsinchun Chen, Roger H. L. Chiang und and Veda C. Storey (2012), “Business Intelligence and Analytics: From Big Data to Big Impact”, *MIS Quarterly*, Vol. 36 No. 4, ff. 1165-1188.

Hyndman, R. und Athanasopoulos, G. (2018), “7.3 Holt-Winters’ seasonal method | Forecasting: Principles and Practice”, abrufbar unter: https://otexts.com/fpp2/holt-winters.html (letzter Zugriff 12. November 2019).

John Pavlus (2016), “Computers Now Recognize Patterns Better Than Humans Can”, abrufbar unter: https://www.scientificamerican.com/article/computers-now-recognize-patterns-better-than-humans-can/ (letzter Zugriff 9. September 2019).

Joshi, N. (2019), “How Far Are We From Achieving Artificial General Intelligence?”, abrufbar unter: https://www.forbes.com/sites/cognitiveworld/2019/06/10/how-far-are-we-from-achieving-artificial-general-intelligence/#7921ee906dc4 (letzter Zugriff 28. Juli 2019).

Khandelwal, R. (2018), “Finding outliers in dataset using python”, abrufbar unter: https://medium.com/datadriveninvestor/finding-outliers-in-dataset-using-python-efc3fce6ce32 (letzter Zugriff 27. Oktober 2019).

Labbe, P., Anjos, C., Solanki, K. und DiMaso, J. (2019), *Hands-On Business Intelligence with Qlik Sense: Implement self-service data analytics with insights and guidance from Qlik*

Sense experts / Pablo Labbe, Clever Anjos, Kaushik Solanki, Jerry DiMaso, Packt Publishing, Birmingham.

LinkedIn (2018), "LinkedIn Workforce Report | United States | August 2018", abrufbar unter: https://economicgraph.linkedin.com/resources/linkedin-workforce-report-august-2018 (letzter Zugriff 28. Juli 2019).

Liu, H., Li, J., Wu, Y. und Fu, Y. (2019), *Clustering with Outlier Removal*.

Lu, J.W. und Beamish, P.W. (2004), "International Diversification and Firm Performance. The S-curve Hypothesis", *Academy of Management Journal*, Vol. 47 No. 4, ff. 598-609.

Markus, L. (1987), "Toward a "Critical Mass" Theory of Interactive Media", *Communication Research*, Vol. 14 No. 5, ff. 491-511.

Merriam, S.B. und Tisdell, E.J. (2015), *Qualitative research: A guide to design and implementation / Sharan B. Merriam and Elizabeth J. Tisdell*, Fourth edition, Jossey-Bass, San Francisco.

Mills, C.M. und Keil, F.C. (2004), "Knowing the limits of one's understanding. The development of an awareness of an illusion of explanatory depth", *Journal of Experimental Child Psychology*, Vol. 87 No. 1, ff. 1-32.

Moore, G.A. (2014), *Crossing the chasm: Marketing and selling disruptive products to mainstream customers*, Third edition, HarperBusiness an imprint of HarperCollins Publishers, New York NY.

Munich airport (2019), "Traffic figures", abrufbar unter: https://www.munich-airport.com/traffic-figures-263342 (letzter Zugriff 6. November 2019).

Nagel, K. (1990), *Nutzen der Informationsverarbeitung: Methoden zur Bewertung von strategischen Wettbewerbsvorteilen, Produktivitätsverbesserungen und Kosteneinsparungen*, 2., überarb. und erw. Aufl., Oldenbourg, München.

O'Grady, S. (2018), "The RedMonk Programming Language Rankings: June 2018", abrufbar unter: https://redmonk.com/sogrady/2018/08/10/language-rankings-6-18/?source=post_page--------------------------- (letzter Zugriff 28. Juli 2019).

Pandey, P. (2018), "From 'R vs Python' to 'R and Python'", abrufbar unter: https://towardsdatascience.com/from-r-vs-python-to-r-and-python-aa25db33ce17 (letzter Zugriff 28. Juli 2019).

Patton, M.Q. (2015), *Qualitative research & evaluation methods: Integrating theory and practice / Michael Quinn Patton*, Fourth edition, SAGE Publications, Thousand Oaks, California.

QlikTech International AB (2017), “The Associative Difference™”.

QlikTech International AB (2018a), “Re: VizLib Extensions and Exporting”, abrufbar unter: https://community.qlik.com/t5/QlikView-App-Development/VizLib-Extensions-and-Exporting/m-p/97721 (letzter Zugriff 24. November 2019).

QlikTech International AB (2018b), “Saving Data Model Viewer Layout between loads”, abrufbar unter: https://community.qlik.com/t5/Qlik-Sense-App-Development/Saving-Data-Model-Viewer-Layout-between-loads/m-p/27481 (letzter Zugriff 3. August 2019).

QlikTech International AB (2019a), “Managing data associations – Qlik Sense”, abrufbar unter: https://help.qlik.com/en-US/sense/February2019/Subsystems/Hub/Content/Sense_Hub/LoadData/associating-data.htm (letzter Zugriff 3. August 2019).

QlikTech International AB (2019b), “Qlik Pricing | Qlik Sense – Compare All Editions”, abrufbar unter: https://www.qlik.com/us/pricing (letzter Zugriff 4. Oktober 2019).

Raschka, S. und Mirjalili, V. (2018), *Machine Learning mit Python und Scikit-Learn und TensorFlow: Das umfassende Praxis-Handbuch für Data Science, Deep Learning und Predictive Analytics,* 2., aktualisierte und erweiterte Auflage, mitp, Frechen.

Redmond, S. (2014), “Qlik Tips”, abrufbar unter: https://www.qliktips.com/2014/07/qlik-sense.html (letzter Zugriff 3. August 2019).

Rita, S., Cindi, H. und Carlie, I. (2018), “Augmented Analytics Is the Future of Data and Analytics”.

Rogers, E.M. (1995), *Diffusion of innovations,* 4th ed., Free Press, New York, London.

Rozenblit, L. und Keil, F. (2002), “The misunderstood limits of folk science: an illusion of explanatory depth”, *Cognitive Science,* Vol. 26, ff. 521-562.

Scherbak, M. (2019), “Data Science vs Business Intelligence: same but completely different”, abrufbar unter: https://towardsdatascience.com/data-science-vs-business-intelligence-same-but-completely-different-1d5900c9cc95 (letzter Zugriff 5. Oktober 2019).

Schilling, M.A. (2002), “Technology Success and Failure in Winner-Take-All Markets. The Impact of Learning Orientation, Timing, and Network Externalities”, *Academy of Management Journal*, Vol. 45 No. 2, ff. 387-398.

Sherman, R. (2015), *Business intelligence guidebook: From data integration to analytics,* Elsevier Morgan Kaufmann is an imprint of Elsevier, Amsterdam.

Singh, G. (2018), “7 methods to perform Time Series forecasting”, abrufbar unter: https://www.analyticsvidhya.com/blog/2018/02/time-series-forecasting-methods/ (letzter Zugriff 11. November 2019).

Stackoverflow (2018), “Python - ImportError: cannot import name ExponentialSmoothing”, abrufbar unter: https://stackoverflow.com/questions/48689740/importerror-cannot-import-name-exponentialsmoothing (letzter Zugriff 13. November 2019).

Statista (2019a), “Passenger air traffic each year”, abrufbar unter: https://www.statista.com/statistics/564717/airline-industry-passenger-traffic-globally/ (letzter Zugriff 6. November 2019).

Statista (2019b), “The 100 largest companies in the world by market value in 2019”, abrufbar unter: https://www.statista.com/statistics/263264/top-companies-in-the-world-by-market-value/ (letzter Zugriff 28. Juli 2019).

Stroh, K., Effern, H. und Wittl, W. (2018), “Moratorium für die dritte Startbahn”, abrufbar unter: https://www.sueddeutsche.de/muenchen/flughafen-muenchen-dritte-startbahn-csu-freie-waehler-1.4198035 (letzter Zugriff 6. November 2019).

Taylor, J. (2017), “Four Problems in Using CRISP-DM and How To Fix Them”, abrufbar unter: https://www.kdnuggets.com/2017/01/four-problems-crisp-dm-fix.html (letzter Zugriff 3. August 2019).

Trubetskoy, G. (2016), “Holt-Winters Forecasting for Dummies (or Developers)”, abrufbar unter: https://grisha.org/blog/2016/02/17/triple-exponential-smoothing-forecasting-part-iii/ (letzter Zugriff 8. November 2019).

Vandeput, N. (2018), “Exponential Smoothing with Damped Trend (Python)”, abrufbar unter: https://supchains.com/article/exponential-smoothing-damped-trend-python/ (letzter Zugriff 26. November 2019).

VanderPlas, J. (2017), *Data Science mit Python, Mitp Business*, 2018. Auflage, mitp, Frechen.

Vizlib (2019a), abrufbar unter: https://home.vizlib.com/ (letzter Zugriff 14. September 2019).

Vizlib (2019b), “Vizlib Line Chart - Advanced Analytics: Forecasting”, abrufbar unter: https://community.vizlib.com/support/solutions/articles/35000129484-vizlib-line-chart-advanced-analytics-forecasting (letzter Zugriff 8. November 2019).

Wendler, T. und Gröttrup, S. (2016), *Data mining with SPSS modeler: Theory, exercises and solutions / Tilo Wendler, Sören Gröttrup*, Springer, Switzerland.

Westermann, G. (2012), *Kosten-Nutzen-Analyse: Einführung und Fallstudien, ESV basics*, Schmidt, Erich, Berlin.

Statutory Declaration

I herewith formally declare that I have written the submitted thesis independently. I did not use any outside support except for the quoted literature and other sources mentioned in the paper.

I clearly marked and separately listed all of the literature and all of the other sources which I employed when producing this academic work, either literally or in content.

I am aware that the violation of this regulation will lead to failure of the thesis.

Felix Scheibe

____

Student's name

____

Student's signature

562780

____

Matriculation number

08.12.2019

____

Berlin, date