Mikael Mantere

Development of Project-Based Data Analytics Course Material

Metropolia University of Applied Sciences

Bachelor of Engineering

Industrial Management ICT

Bachelor's Thesis

6 May 2024

Abstract

Author:	Mikael Mantere
Title:	Development of Project-Base Data Analytics Course Material
Number of Pages:	30 pages + 2 appendices
Date:	6 May 2024
Degree:	Bachelor of Engineering
Degree Programme:	Industrial Management
Professional Major:	Information & Communications Technology
Supervisors:	Rakel Peltola, Senior lecturer Sonja Holappa, Senior lecturer

The main objective of this thesis was to design, develop, and validate new online course materials tailored for teaching data analytics at Metropolia UAS, addressing an evident gap between highly guided materials and practical independent data analytics projects. The outcome of this thesis is a two-part assignment that includes the manipulation and analysis of open-sourced data.

The thesis is based on interviews conducted with a Senior Lecturer from Metropolia UAS as well as analysing existing curriculum. Additionally, existing literature was reviewed to establish a basis for the assignments.

The assignments were further developed by surveying students about the assignment's features. The end result was the two-part project-based assignment, and it was validated through lecturer feedback. The assignments were found to effectively meet educational needs.

Keywords: Python, Pandas, Data Analytics, online teaching, project-based learning

Tiivistelmä

Tekijä:	Mikael Mantere
Otsikko:	Projektipohjaisen data-analytiikka -kurssimateriaalin kehittäminen
Sivumäärä:	30 sivua + 2 liitettä
Aika:	6.5.2024
Tutkinto:	Insinööri
Tutkinto-ohjelma:	Tuotantotalous
Ammatillinen pääaine:	Kansainvälinen ICT-liiketoiminta
Ohjaajat:	Rakel Peltola, Lehtori Sonja Holappa, Lehtori

Tämän opinnäytetyön päätavoitteena oli suunnitella, kehittää ja validoida uusia verkkokurssimateriaaleja, jotka on räätälöity data-analytiikan opetukseen Metropolia Ammattikorkeakoulussa. Opinnäytetyössä puututtiin ilmeiseen aukkoon pitkälle ohjattujen materiaalien ja käytännön itsenäisten data-analytiikka projektien välillä. Tämän opinnäytetyön tuloksena on kaksiosainen tehtävä, joka sisältää avoimen datan manipulointia ja analysointia.

Opinnäytetyö perustuu Metropolia Ammattikorkeakoulun luennoitsijoiden kanssa tehtyihin haastatteluihin sekä olemassa olevan opetussuunnitelman analysointiin. Lisäksi käytiin läpi olemassa olevaa kirjallisuutta tehtävänannon pohjan luomiseksi.

Tehtäviä kehitettiin edelleen kartoittamalla opiskelijoita tehtävän ominaisuuksista. Lopputuloksena oli kaksiosainen projektipohjainen tehtävä, ja se validoitiin luennoitsijan palautteen avulla. Tehtävien todettiin täyttävän nykyiset koulutustarpeet.

Keywords: Python, Pandas, Data Analytiikka, verkko-opetus, projektipohjainen oppiminen

---

Contents

List of Abbreviations

IM:	Industrial Management
ICT:	Information and communications technology
SCM:	Supply Chain Management
BA:	Business Analytics
BAT:	Business Analytics Tools
UBA:	Utilizing Business Analytics
BAP:	Business Analytics Project
UAS:	University of Applies Sciences

1 Introduction

In today's digital age, data has become an invaluable asset across all sectors, driving the rapid growth and importance of data analytics. Organizations in business, healthcare, and government increasingly rely on data to make informed decisions, predict trends, and enhance operational efficiency. The rapid increase of data from various sources like social media, sensors, and online transactions has led to an explosion of information, which requires sophisticated tools and methodologies for effective analysis. (McAfee, 2012.)

A Variety of tools and methodologies have been developed to manage and analyse this vast amount of information efficiently. Among the software solutions Microsoft Power BI and Excel are prominent. Additionally, programming languages like R and Python have become indispensable in the data analytics field. R is particularly favoured for statistical analysis and graphical models. Python, with its simplicity and readability, serves as a powerful tool for both data manipulation and analysis. Libraries such as Pandas for data manipulation, NumPy for numerical data, and Matplotlib for data visualization, enhance Python's functionality, making it a preferred choice for data scientists and analysts. Also, in recent years Python has become a prominent tool for creating AI&ML models. (Ozgur, 2017)

Within the scope of Industrial Management (IM) at Metropolia University of Applied Sciences, students are exposed to two major study choices: Information and Communications Technology (ICT) and Supply Chain Management (SCM). Each path offers a unique perspective while keeping a similar core of IM studies. The SCM focus includes courses on Enterprise Resource Planning (ERP), logistics, automation, and process understanding. Conversely, the ICT focus is tailored towards creating a foundation in data-driven business strategies and management aspects. Here, students learn how to harness data analytics to drive business decisions, develop strategic insights, and enhance managerial capabilities. This includes understanding the role of data

governance, data quality management, and how to leverage data analytics to achieve business objectives. (Metropolia UAS)

1.1 Thesis Challenge

In the evolving field of data analytics, the transition from theoretical understanding to practical application remains a significant challenge for many students. Many students find it difficult to initiate their own projects, particularly when it comes to sourcing data that strikes an appropriate balance of complexity and manageability. Furthermore, obtaining data which provides realistic data cleaning challenges often proves difficult.

1.2 Objective, Outcome and Scope

This thesis aims to develop in-depth understanding of the current curriculum and relevant theory to bridge the gap between curriculum and students needs by developing a new comprehensive online course for students. The proposed course will focus on real-world-like data cleaning and project-based learning. The overarching goal is to enhance students' abilities to independently manage and execute analytics projects. The scope of this thesis is the current situation and prevailing need in Industrial Management (IM) at Metropolia University of Applied Sciences. Other Universities and courses are excluded outside the research.

2 Methods and Materials

This chapter outlines the workflow, methods and materials of the thesis. The thesis is loosely following the GATE model which is used in Metropolia IM studies. The GATE model is a structured framework used to guide the thesis-writing process by dividing it into seven stages. This approach is employed to ensure that students systematically acquire the necessary methodology and research skills to effectively execute a business project for their case and successfully produce their thesis. (Holappa, 2020)

2.1 Research Design

The roadmap to the final the proposal is divided into five stages that can be seen in Figure 1. Throughout these stages data from existing courses, surveys and interviews are utilized to support the development of the course proposal.

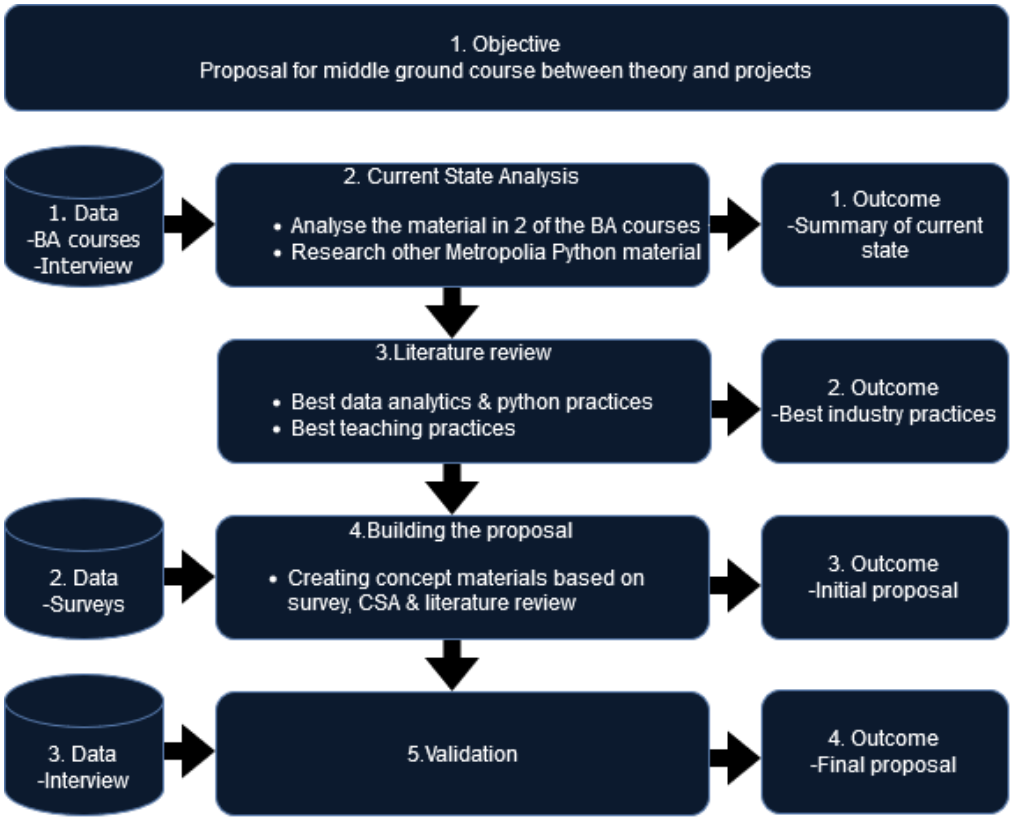


Figure 1. Research design for developing a Python & Pandas based data analytics course.

Objective

A clear goal is established at the outset of the research: drafting a proposal for a Python & Pandas data analytics course.

Current State Analysis

An examination of Metropolia's current data analytics courses is undertaken, focusing on the materials from BA courses and other Python resources used within Metropolia. The findings from this analysis will identify the strengths and weaknesses of the existing curriculum, which will inform the development of the new course.

Literature Review

Existing academic and industry knowledge is systematically reviewed, encompassing scholarly articles, whitepapers, and other relevant publications on Python, Pandas, data analytics, and education. The outcomes from this review are expected to lay a theoretical foundation and identify best practices and tools in course design.

Building the Proposal

Combining insights from the current state analysis and literature review are applied to shape the framework for the course proposal. The surveys with students are carried out to build upon the proposals framework and the thesis author’s personal experiences are used to refine it.

Validation

The initial proposal undergoes a critical validation phase, including the collection of feedback from lecturers and potentially students through surveys and other validation methods. The data gathered during this phase is used to refine the course proposal.

2.2 Collected data

The data is collected at 4 stages of this thesis. In table 1 the collected data is defined.

Table 1. Table of data collected for this thesis.

	Stage	Participants	Data type	Description	Documented as
1.	Objective	Lecturer	Online interview	Discussion about the current challenges	Field notes
2.	CSA	Lecturer	Online interview	Current state of Python & data analytics teaching	Field notes
3.	Building the proposal	Targeted for IM ICT students	Online Survey	Google forms, about project features	Google forms, Field notes
4.	Validation	Lecturer		Feedback on the proposal	Field notes

The conducted interviews as seen in Table 1, were carried out online with the Senior Lecturer currently teaching business analytics curriculum.

3 Current State Analysis

This chapter explores the current state of data analytics teaching at Metropolia, particularly focusing on the integration of Python ecosystem within the bachelor's program in ICT Industrial Management. The primary objective is to leverage these insights and find weaknesses to develop the proposal for an online course centred around data analytics with Python. The current state analysis was carried out by finding relevant courses and going through all the course material & assignments to determine exactly what skills were taught. Also, an interview with a Business Analytics lecturer was conducted to go more in depth about the state of teaching. The scope of this CSA is to only explore courses taught for Metropolia IM students. CampusOnline courses were not addressed.

3.1 Business Analytics trilogy

As part of the bachelor's program in ICT Industrial Management, Metropolia currently has three mandatory courses specifically about data analytics within the Python & Pandas environment. These courses are known as Business Analytics (BA) courses, and they are a part of the 3rd year studies. When students embark on these courses they have had a Python basics course during their first year.

In the interview it was found that the primary goal of BA courses is to equip students with a foundational understanding of the data landscape, encompassing data processing and analytical methods. It was noted that while previous courses significantly focused on theoretics of analytics, this course series is designed to equip students with the essential skills needed for a deeper practical understanding of the field. The interviewed lecturer clarified that these courses are not intended to turn students into expert coders. However, developing a basic proficiency in coding is crucial because it opens doors to hands-on tasks.

Two of the BA courses contain exercises from the DataCamp platform. DataCamp is a paid online learning platform that specializes in data science and analytics, offering interactive courses on a wide range of topics such as data manipulation, programming in Python and R, statistical analysis, and machine learning.

3.1.1 Business Analytics Tools

The Business Analytics Tools (BAT) course begins with the basics of importing data in Python, teaching students how to gather data from various sources including APIs. The curriculum then moves into data cleaning, where students learn to enhance data quality by addressing issues such as missing values and duplicates. Advanced topics include both supervised and unsupervised learning, where students explore predictive modelling and clustering techniques. Python's powerful libraries such as Pandas, NumPy, Matplotlib, Seaborn, and Scikit-learn are utilized throughout the course. The course includes and is graded based on a DataCamp section & six practical assignments that apply these concepts, helping students develop skills to solve real-world data analysis challenges.

Week			
1	Tuesday	22.8.2023	Kick-off, DataCamp, Python-recap
2	Monday	28.8.2023	DC: Introduction to importing Data in Python
	Tuesday	29.8.2023	Assignment 1
3	Monday	4.9.2023	DC: Intermediate Importing Data in Python
	Tuesday	5.9.2023	Assignment 2
4	Monday	11.9.2023	RECAP
	Tuesday	12.9.2023	RECAP
5	Monday	18.9.2023	DC: Cleaning Data In Python
	Tuesday	19.9.2023	Assignment 3
6	Monday	25.9.2023	DC - Supervised Learning w Python Ch1&Ch2
	Tuesday	26.9.2023	Lesson & Assignment 4
7	Monday	2.10.2023	DC - Supervised Learning w Python Ch3&Ch4
	Tuesday	3.10.2023	Lesson & Assignment 5
8	Monday	9.10.2023	DC - Unsupervised learning
	Tuesday	10.10.2023	Clustering + Assignment 6

Figure 2. Business Analytics Tools TX00CN88-3006 Course schedule.

3.1.2 Utilizing Business Analytics

Utilizing Business Analytics (UBA) course continues from the learnings of the BAT course. In the first three weeks of the course, advanced topics such as unsupervised learning, natural language processing, and sentiment analysis are introduced through DataCamp. In the interview it was found that these topics must be pushed from BAT to UBA due to a strict schedule, though it was not seen as a major issue. Later in the course students are engaged in a practical project where both supervised and unsupervised learning techniques are applied to data of their choice, ranging from web data to CSV files. Detailed planning is required for the project, outlining methods and expected outcomes. Google Colab or Jupyter Notebooks are recommended for the analysis and visualization phases. Challenges encountered during the project are addressed, enhancing problem-solving skills. This educational approach prepares students for complex data analysis tasks by teaching them practical application of theoretical knowledge.

	Thursday	Plan
1	26.10.2023	Kickoff
2	2.11.2023	DC1-dl: 5.11.
3	9.11.2023	DC2-dl: 12.11.
4	16.11.2023	DC3-dl: 19.11.
5	23.11.2023	Plan-dl 26.11.
6	30.11.2023	
7	7.12.2023	Presentations
8	14.12.2023	Presentations

Figure 3. Utilizing Business Analytics TX00CN87-3006 Course schedule.

3.1.3 Business Analytics Project

The BA course trilogy ends by combining a heavy business perspective with the learnings of previous courses. The Business Analytics Project (BAP) course consists of a small essay and a group project where CRISP-DM methodology is used. CRISP-DM is a data mining model for projects to improve business understanding and knowledge-based management. (Smart Vision Europe, 2017)

3.2 Other courses

This chapter contains an overview about other Metropolia courses concerning Python & Pandas. For IM studies an optional study path of 2 advanced Python courses ”Python-ohjelmoinnin jatkokurssi TX00DZ62-3002" & ”Tekoäly ja koneoppiminen liiketoiminta-asiantuntijalle TU00FH43-3004” exists with the prerequisite of Python basics.

The first courses contain the basics of NumPy, Pandas, Matplotlib, Scikit-learn & Jupyter notebooks. The second course is more focused on AI with Python, the course is quite advanced in its nature. The advanced course explores key concepts of artificial intelligence, including how to build and refine predictive models. It covers the basics of machine learning, pattern recognition, and decision-making processes, using real-world applications such as image recognition to illustrate these concepts.

3.3 Summary

The current state of data analytics teaching with Python is comprehensive. But there remains a significant gap between the basic Python course and the Business Analytics courses, though according to the conducted interviews, efforts are underway to bridge this gap. Also there exists an overlap between the BA courses and the optional Python courses. This is something that is not in the scope of this thesis. Figure 4 displays the overall structure of Python related teaching in Metropolia.

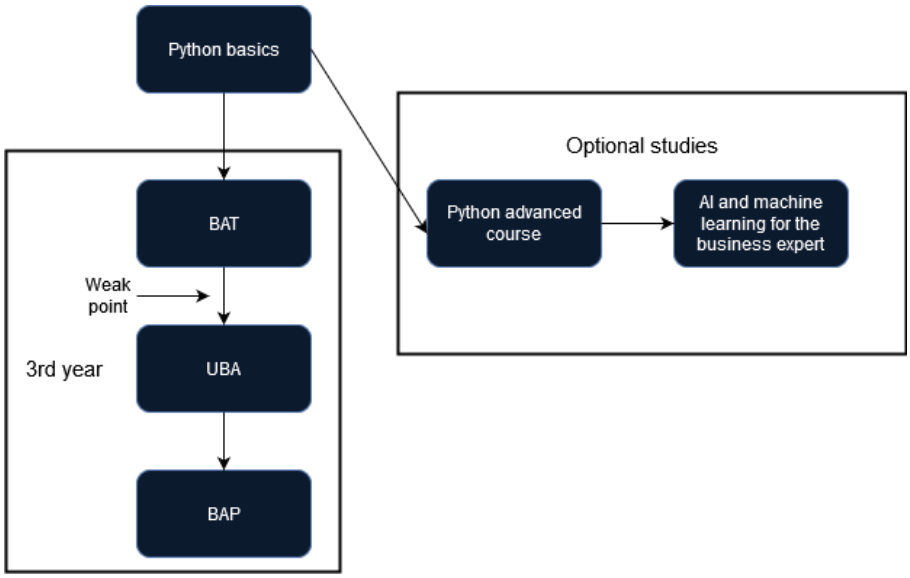


Figure 4. Metropolia Python & data analytics teaching landscape.

In analyzing the current state analysis findings in relation to the proposal, key strengths and weaknesses can be identified to inform the course design.

Table 2. Metropolia IM data analytics teaching strengths and weaknesses.

Strengths	Weaknesses
- The trilogy format of the BA courses is logical, enhancing learning	- Significant transition to project work in UBA course. Topics are only covered once. Weak point noted in Figure 4
- Students have strong foundational knowledge of data analytics with Python from the BAT course	- Insufficient complexity in data cleaning exercises. Provided data is clean, lacking real-world messiness
- Advanced topics are covered, including machine learning with	-

This chapter concludes the current state analysis. The next chapter explores relevant literature based on the findings of the CSA.

4 Literature Review on Best Practices for Building the Course

The objective of this literature review is to gather information on the tools and best practices to be used in the proposal. The literature review is specifically conducted on the tools found in the CSA.

4.1 Python

This chapter introduces Python, a popular programming language known for its simplicity and readability. It discusses key features and explores its extensive libraries. The goal is to create a baseline understanding of why Python is a preferred choice for data science & analytics.

4.1.1 Python skills in context of data science

A DataCamp blog article by Nakamoto lists several essential Python skills. At the foundation of using Python for data science is a solid grasp of fundamental programming concepts. This includes understanding syntax, data types and notably object-oriented programming (OOP). These elements provide the basis for more complex data structures and algorithms.

Python’s strengths lie in its vast number of libraries that are useful for many aspects of data science. Libraries like Pandas for data manipulation and NumPy for numerical operations are particularly important. These libraries offer pre-written functionalities that expedite routine data handling tasks, reducing development time, and minimizing potential for error. By focusing on these tools, data scientists can focus on analysis rather than the nuances of programming.

Proficiency in data manipulation and analysis is invaluable, allowing for the extraction of insights from data. Equally important is the ability to visually represent data, using Python's visualization tools to create understandable charts and graphs.

As skills develop, venturing into advanced techniques like web scraping and web frameworks expands a programmer's ability to handle diverse projects, from data collection to sharing insights in web formats. These capabilities ensure a comprehensive skill set in Python programming in the context of data analytics. (Nakamoto, 2023)

4.1.2 Libraries in Python

Python is known for its rich ecosystem of libraries that cater to various needs in programming and data science. Some popular Python libraries include: NumPy, Pandas, Matplotlib, SciPy, Scikit-learn, TensorFlow, Keras, PyTorch, Flask & Django. (Thangaraja, 2020)

Understanding libraries, modules & packages

- Library: A library is a collection of packages or modules. It is not limited to a single namespace, and you can have multiple libraries, each with multiple packages.
- Package: Is a collection of Python modules under a common namespace. This is typically implemented by having directory with a special '__init__.py' file, containing one or more modules.
- Modules: A module is a single Python file that contains executable code, including Python classes, functions or variables.

Importing libraries

Python libraries are imported using the 'import' statement. Libraries can be imported as a whole or different parts, importing specific parts helps optimise the program. The importing can be done by using the following syntax:

```
import math                           # Imports 'math' as a whole
from math import sqrt                 # Imports sqrt function, can pass lists
from math import *                    # Imports all math functions & classes
from pandas import DataFrame          # Imports a DataFrame class from Pandas
from sklearn.metric import r2_score   # Imports r2_score from subpackagage
```

Listing 1. Python code for different methods of import

Before additional Python packages can be imported, they have to be installed. Installation usually happens using pip. Pip is the Python package installer that allows the user to install and manage additional libraries that are not a part of Python's standard libraries. Once the package is installed using pip it is placed in a venv or a lightweight "virtual environment" with their own directories.
(Koidan, 2021)

4.1.3 Jupyter notebook

Jupyter Notebook is an open-source web application that allows users to create and share documents that contain live code, visualisations and narrative text. It is particularly popular among data scientists and researchers for its ease of use in data analysis and machine learning projects. Jupyter Notebooks are highly valuable in this educational context due to their versatility and interaction nature. Jupyter notebook also supports more programming languages than just Python, for example HTML. HTML is especially useful in an educational context because it allows for integration of interactive elements and multimedia, making content more engaging and visually appealing. (Kluyver, 2016)

4.1.4 Advanced Python methods

Advanced methods could be beneficial for the students for a competitive edge on the job market. Sarkar highlights that understanding advanced methods like, web technologies is important because the end goal is to solve business or scientific problems, and being able to communicate solutions effectively is crucial. Different web frameworks allow interactive ways of sharing information. (Sarkar, 2022)

4.2 Pandas ecosystem

Since its introduction by Wes McKinney in 2008, Pandas has become an indispensable tool for data scientists and analysts. This chapter is going to explore the most important functionalities of Pandas and supporting libraries. As seen in 4.1.2 some other most popular data science libraries include NumPy & Matplotlib, which often work in conjunction with Pandas to provide comprehensive data manipulation and visualizations. (Kopf, 2017.) (McKinney, 2011.)

4.2.1 What is Pandas?

Pandas is a Python library for data analysis that offers data structures and operations for manipulating numerical tables and times series. The primary data structure in Pandas is the DataFrame, a two-dimensional labelled structure where each column can hold data of a different type, much like an Excel spreadsheet or an SQL table. Below is an example how to create and view a DataFrame. (PyData.org)

```
import pandas as pd
data = {'col1': [1, 2], 'col2': [3, 4]}
df = pd.DataFrame(data=data)
print(df)

   col1  col2
0     1     3
1     2     4
```

Listing 2. Python code to create a Pandas DataFrame

4.2.2 Enhancing Pandas with supporting libraries

Pandas integrates seamlessly with NumPy and Matplotlib, forming a robust toolkit for data analysis and visualisation. NumPy is known for its efficient array operations underlies the Pandas data structures, boosting computational performance and providing many basic and advanced mathematical functions.

Matplotlib on the other hand serves as a great visualisation tool allowing for detailed graphical representations of data managed through pandas. Using the DataFrame created in 4.2.1 with Matplotlib function in Listing 3, its content can be visualised using for example a scatter plot function, the visualisation can be seen in Figure 5. (McKinney, 2011.)

```
import matplotlib.pyplot as plt

plt.scatter(df.iloc[:, 0], df.iloc[:, 1])
plt.show()
```

Listing 3. Using Matplotlib to create a scatter plot, with DataFrame from Listing 2.

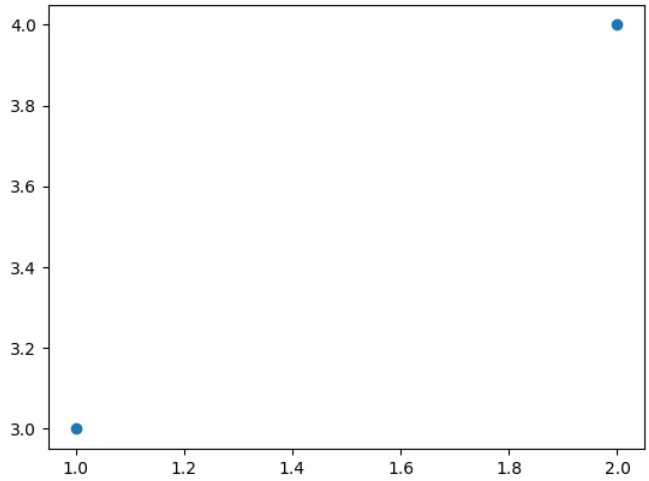


Figure 5. Matplotlib scatter plot created by code from Listing 3.

4.3 The nature of real-word data

Real-world data often comes in formats and conditions unsuitable for immediate analysis. This chapter delves into the various types of data highlighting the challenges of its messiness and discusses the transformation required to make it usable for analysis. Understanding how to create authentic real-world like data is important for the proposal's material to prepare students for practical data handling and analysis in complex real-world scenarios.

4.3.1 Types of data & characteristics of real-world data

An IBM article defines that data can be categorized into three primary types. Table 3 shows the three categorization types of data.

Table 3. Data categories.

Type	Description
Structured	Highly organized and formatted data. Typically, in relational databases with defined datatypes for each column.
Unstructured	Data that does not have a recognisable structure or format. Examples include text files and multimedia content.
Semi-structured	This type of data falls between the previous two. Examples include JSON and XML

Understanding the types of data is just the beginning of understanding what the features of authentic data are. Once the data is categorized the characteristics must also be recognized. Various papers characterize qualities or messy data as seen in Table 4. (Erhard, 2000.) (Murty, n.d.)

Table 4. Characteristics of data in real-world.

Characteristic	Description
Incomplete	Often lacks full entries or specific sections
Inconsistent	Discrepancies in data arise from entry errors & lack of uniformity
Duplicative	Multiple copies or slight variations of the same measurement
Noisy	Frequently filled with error or irrelevant information that can obscure meaningful patterns.

4.3.2 Dealing with messy data / preparation

To effectively manage the issues of incomplete, inconsistent, duplicative and noisy data as identified in Table 4 of section 4.3.1, various techniques are applied. Addressing incomplete data, techniques such as replacing missing values with substituted values, or algorithms that can handle missing data directly. For inconsistent data validation rules or automated scripts are utilised to identify discrepancies. Duplicative data is managed by identifying and removing duplicate records using automated tools, while being careful to avoid removing non-duplicates. Noisy data is handled through data smoothing techniques like rolling averages or median filtering and outlier detection are applied to identify anomalies in the data. (McKinney. 2011) (Erhard, 2000)

4.4 Best practices in online data analytics education

This chapter explores the process of creating engaging learning materials for an online data analytics course, addressing challenges such as accessibility and student engagement. It discusses innovative teaching practices and interactive content. It also discusses optimal assignment complexity.

4.4.1 Making learning interesting

To foster an engaging learning environment, it is essential to mitigate technical difficulties and to increase the complexity of assignments gradually to prevent student frustration. Research, such as that conducted in Jaggars' study, found that students show a preference for courses that integrate suitable learning technologies, favoring these over courses that heavily rely on text-based content. This suggests that the inclusion of interactive elements is crucial. Additionally, findings from Aparicio's study underscore the significance of gamification in the success of online courses. The implementation of game-like elements can significantly increase student engagement. (Jaggars, 2016) (Aparicio, 2018)

4.4.2 Difficulty and complexity of the assignments

A study by Xu focuses on teaching Python to undergraduate information systems students. Information systems studies have similar aspects as industrial engineering ICT studies, which is beneficial for the context of this thesis. The study found that student's views on the usefulness and relevance of topics predict their perceptions of learning outcomes. Additionally, the perceived difficulty of homework assignments correlates with their actual performance. (Xu, 2021)

4.4.3 Gamification – for data analytics assignments

Gamification integrates game design elements into non-game contexts, such as data analytics to increase engagement. It uses storytelling, teamwork, competition and rewards to make data exploration interactive and enjoyable. For example, participants might solve a crime using data or optimise resources in a simulated city, thereby applying analytical skills in a narrative setting. Studies like those by (Hamari, 2014) show that gamification can significantly enhance engagement and learning outcomes, though effectiveness depends on implementation.

4.5 Summary

This summary chapter concludes the key findings from the reviewed literature. The key findings are twofold, firstly the technical skills that should be taught, and secondly the features that the assignment should have. The summary serves as a foundation for proposing the new curriculum.

Good to know technical skills according to literature:

- Library utilisation
- Fundamental programming skills
- Data manipulation & analysis
- Visualisation techniques

Features of a good data analytics assignment according to literature:

- Gamification
- Appropriate level of complexity
- Brings something unique
- Gradual complexity

5 Building the Online Course

In this chapter the proposal for the online course is created, drawing from insights gathered during the current state analysis (CSA) and literature review.

The course design is built with the expectation that students will have acquired the specific skills from the Business Analytics Tools (BAT) course, as identified during the CSA. The chapter also outlines the various challenges faced in the initial course design.

5.1 Educational objectives & outcomes

The overarching objectives of the proposed course are derived from the challenge, CSA, and literature. These objectives are twofold: to give students a practical glimpse into the types of projects they might encounter in the future, thereby also inspiring future school projects, and secondly to build on the knowledge they have acquired during BAT as outlined in 3.1.1.

The technical side of the educational objective focuses on equipping students with key data analysis skills found from the literature review. This includes data cleaning, exploratory data analysis (EDA), and building machine learning models. The goal is to enable students to conduct research and to answer specific questions using their data.

5.2 Survey for narrowing the projects topics

A survey was conducted among Metropolia IM ICT students with experience in the BA courses to establish a foundation for developing the projects. The survey, which received 14 responses, aimed to gather insights directly from the students about their preferences and experiences related to project topics in business analytics courses.

The survey asked students to identify topics that interested them for upcoming projects. A notable finding was that students preferred pre-selected topics, with a gamification-based project involving solving a crime, as discussed in chapter 4.4.3, emerging as the most popular choice. Conversely, projects involving open data and financial data analysis were least favoured.

In terms of workload distribution, there was no significant preference among the students, indicating flexibility in how they manage project tasks. When it came to presenting their findings, the students clearly favoured easy-to-understand visualisations, suggesting a preference for simplicity and clarity in data presentation.

These insights from the survey have been instrumental in shaping the proposed projects for the course, ensuring that the topics not only engage students but also align with their educational needs and preferences for data visualisation and project management.

5.3 Concept Projects

The initial stage in developing the projects involves creating conceptual projects to assess both their development practicality and educational value. The primary objective during this stage is to rapidly develop various concepts to identify any shortcoming early in the process, adhering to the “fail fast” approach. Table 5 below outlines the concepts with their ultimate outcome.

Table 5. Table of concepts that went through evaluation of feasibility.

Concept idea	Description	Educational value (1-5)	Practicality (1-5)	Outcome
Air quality relationship with meteorological data	Introduction level project about investigating the correlation between air quality and meteorological data	4	5	A conceptual project with educational value
Detective project	An intermediate level gamified project about solving a crime using data	2	1	Concept was deemed impractical with little educational value
Image processing and analysis	Advanced level extracting text and numbers from food labels using OCR	4	2	Concept was deemed too impractical and technically challenging
Driving patterns and weather correlation	Intermediate level open-source data for car behaviour and weather data	4	4	A conceptual project with educational value

The two project concepts, the detective project, and the image processing project, were evaluated and found to be infeasible. The detective project, initially designed to be gamified, was considered too simplistic for the skill level the students possessed, reducing its educational merit. Although a more extended development cycle could potentially make such a gamified approach feasible, it was not practical with the current constraints. On the other hand, the image processing and analysis project appeared promising in its initial stages. However, the unavailability of sufficient data and the students' limited tool set rendered the project impractical for implementation.

5.4 Building the project assignments

The development of project assignments in the proposed course was based on concepts identified as feasible in section 5.3. For the proposal course it was necessary to at least build two projects, as it was found in 4.4.1 that increasing the complexity gradually is beneficial for learning. The first project is designed to familiarise students with the process of constructing analytics projects, while the second project increases in difficulty, requiring students to apply what they have learned with less guidance but in a format similar to the first project. Both assignments can be found as appendices at the end of the thesis.

5.4.1 Assignment project 1

In the first project students are tasked to examine the relationship between Air quality index and meteorological observations. The data is gathered from HSY and Finnish Meteorological Institute respectively and provided ready for the students. Table 6 shows the structure and notable steps in the first assignment.

Table 6. Proposed assignment's overall structure.

1. Importing data	Tasks: - Importing data from CSV and Excel files. - Familiarise with the basic structure of the datasets. - Handle missing values
2. Cleaning and reformatting	- Identify and correct illegal datetime values in the dataset. - Combine time columns in to one Pandas DateTime object column
3. Exploratory Data Analysis	- Calculate descriptive statistics such as mean, minimum and maximum. - Correlation analysis to find relationships between variables. - Create basic visualisations (e.g. scatter plots, line plots)
4. Prediction models	- Test different machine learning models to assess their effectiveness in predicting air quality

5.4.2 Assignment project 2

The second project combines traffic data with the meteorological data from the first project. The traffic data is downloaded from Fintraffic website, the data is gathered using Traffic measurement stations (TMS).

In this project, students are instructed to go download the dataset themselves, ensuring they select data that aligns with the timeline used in the meteorological dataset. This is to help students to find and select data, addressing the challenge highlighted in chapter 1.1.

The structure of the second assignment is mostly similar to the first one that is seen in Table 6. In this assignment the difficulty is increased, and the amount of assistance given to the student is very limited.

5.4.3 Implementation methods

The proposed assignments are designed for immediate implementation using Google Colab, a cloud-bassed platform that supports Jupyter notebooks. Google colab provides a collaborative environment for students, eliminating the need for local software installation. It integrates seamlessly with Google Drive simplifying the management and submission of assignments. With the proposals building chapter thus completed, the next chapter validates the proposal.

6 Validation

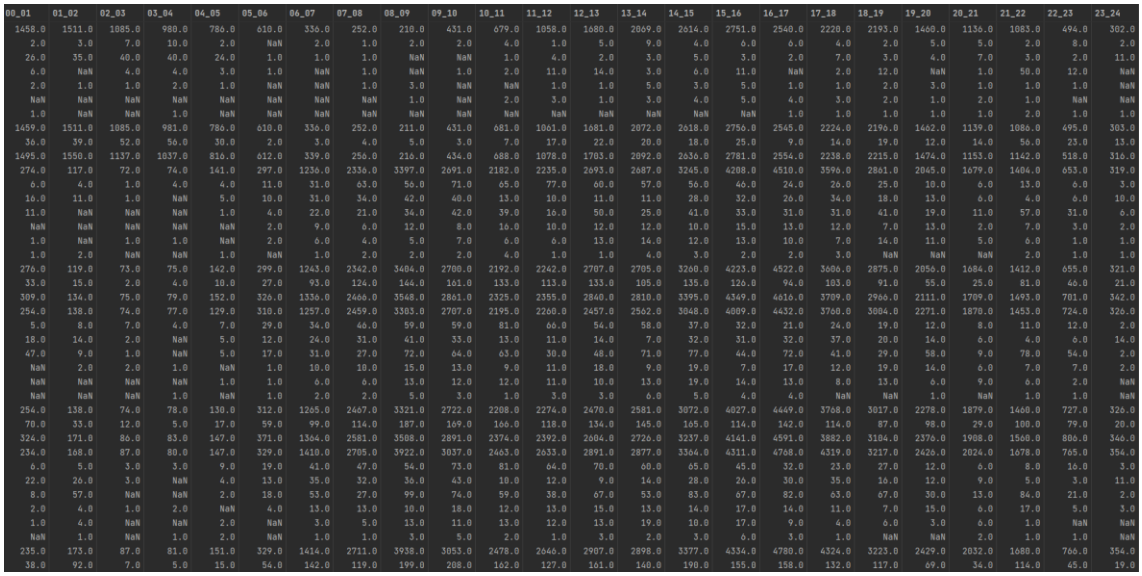
In the validation phase the proposed assignments were thoroughly reviewed by the lecturer, who assessed their alignment with the set objectives. The assignments were found to meet their intended goals, bridging the gap between highly guided tasks from the Business Analytics (BAT) course and the more independent tasks of the Utilizing Business Analytics (UBA) course.

The lecturer highlighted the suitability of the assignments for course integration as they effectively form a cohesive whole, covering an interesting and concrete topic. The flexibility of the assignment template was also noted, allowing the adaptation of different datasets and potential future enhancements.

7 Summary & Conclusions

7.1 Conclusions

This thesis set out to bridge the gap between theoretical and practical application of data analytics at Metropolia UAS by developing a project-based course for the challenges outlined in chapter 1.1. The developed course effectively achieved the stated objectives by introducing progressively challenging assignments that incorporate open-sourced real-world messy data, as seen in assignment 2's dataset in Figure 6. This approach aligns with the good practices identified in chapter 4.3 regarding data qualities ensuring students engage in realistic assignment scenarios.



00_01	01_02	02_03	03_04	04_05	05_06	06_07	07_08	08_09	09_10	10_11	11_12	12_13	13_14	14_15	15_16	16_17	17_18	18_19	19_20	20_21	21_22	22_23	23_24
1458.0	1511.0	1085.0	980.0	786.0	610.0	336.0	252.0	210.0	431.0	679.0	1058.0	1680.0	2069.0	2614.0	2751.0	2540.0	2220.0	2193.0	1460.0	1136.0	1083.0	494.0	302.0
2.0	3.0	7.0	10.0	2.0	NaN	2.0	1.0	2.0	2.0	4.0	1.0	5.0	9.0	4.0	6.0	6.0	4.0	2.0	5.0	5.0	2.0	8.0	2.0
26.0	35.0	40.0	40.0	24.0	1.0	1.0	1.0	NaN	NaN	1.0	4.0	2.0	3.0	5.0	3.0	2.0	7.0	3.0	4.0	7.0	3.0	2.0	11.0
6.0	NaN	4.0	4.0	3.0	1.0	NaN	1.0	NaN	1.0	2.0	11.0	14.0	3.0	6.0	11.0	NaN	2.0	12.0	NaN	1.0	50.0	12.0	NaN
2.0	1.0	1.0	2.0	1.0	NaN	NaN	1.0	3.0	NaN	NaN	1.0	1.0	5.0	3.0	5.0	1.0	1.0	2.0	3.0	1.0	1.0	1.0	NaN
NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	1.0	NaN	2.0	3.0	1.0	3.0	4.0	5.0	4.0	3.0	2.0	1.0	2.0	1.0	NaN	NaN
1.0	NaN	NaN	1.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	1.0	1.0	1.0	1.0	1.0	2.0	1.0	1.0
1459.0	1511.0	1085.0	981.0	786.0	610.0	336.0	252.0	211.0	431.0	681.0	1061.0	1681.0	2072.0	2618.0	2756.0	2545.0	2224.0	2196.0	1462.0	1139.0	1086.0	495.0	303.0
36.0	39.0	52.0	56.0	30.0	2.0	3.0	4.0	5.0	3.0	7.0	17.0	22.0	20.0	18.0	25.0	9.0	14.0	19.0	12.0	14.0	56.0	23.0	13.0
1495.0	1550.0	1137.0	1037.0	816.0	612.0	339.0	256.0	216.0	434.0	688.0	1078.0	1703.0	2092.0	2636.0	2781.0	2554.0	2238.0	2215.0	1474.0	1153.0	1142.0	518.0	316.0
274.0	117.0	72.0	74.0	141.0	297.0	1236.0	2336.0	3397.0	2691.0	2182.0	2235.0	2693.0	2687.0	3245.0	4208.0	4510.0	3596.0	2861.0	2045.0	1679.0	1484.0	653.0	319.0
6.0	4.0	1.0	4.0	4.0	11.0	31.0	63.0	56.0	71.0	65.0	77.0	60.0	57.0	56.0	46.0	24.0	26.0	25.0	10.0	6.0	13.0	6.0	3.0
16.0	11.0	1.0	NaN	5.0	10.0	31.0	34.0	42.0	40.0	13.0	10.0	11.0	11.0	28.0	32.0	26.0	34.0	18.0	13.0	6.0	4.0	6.0	10.0
11.0	NaN	NaN	NaN	1.0	4.0	22.0	21.0	34.0	42.0	39.0	16.0	50.0	25.0	41.0	33.0	31.0	31.0	41.0	19.0	11.0	57.0	31.0	6.0
NaN	NaN	NaN	NaN	NaN	2.0	9.0	8.0	12.0	8.0	16.0	10.0	12.0	12.0	10.0	15.0	13.0	12.0	7.0	13.0	2.0	7.0	3.0	2.0
1.0	NaN	1.0	1.0	NaN	2.0	6.0	4.0	5.0	7.0	6.0	6.0	13.0	14.0	12.0	13.0	10.0	7.0	14.0	11.0	5.0	6.0	1.0	1.0
1.0	2.0	NaN	NaN	1.0	NaN	1.0	2.0	2.0	2.0	4.0	1.0	1.0	4.0	3.0	2.0	2.0	3.0	NaN	NaN	NaN	2.0	1.0	1.0
276.0	119.0	73.0	75.0	142.0	299.0	1243.0	2342.0	3404.0	2700.0	2192.0	2242.0	2707.0	2705.0	3260.0	4223.0	4522.0	3606.0	2875.0	2056.0	1684.0	1412.0	655.0	321.0
33.0	15.0	2.0	4.0	10.0	27.0	93.0	124.0	144.0	161.0	133.0	113.0	133.0	105.0	135.0	126.0	94.0	103.0	91.0	55.0	25.0	81.0	46.0	21.0
309.0	134.0	75.0	79.0	152.0	326.0	1336.0	2466.0	3548.0	2861.0	2325.0	2355.0	2840.0	2810.0	3395.0	4349.0	4616.0	3709.0	2966.0	2111.0	1709.0	1493.0	701.0	342.0
254.0	138.0	74.0	77.0	129.0	310.0	1257.0	2459.0	3383.0	2707.0	2195.0	2260.0	2457.0	2562.0	3048.0	4089.0	4432.0	3760.0	3004.0	2271.0	1870.0	1453.0	724.0	326.0
5.0	8.0	7.0	4.0	7.0	29.0	34.0	46.0	59.0	59.0	81.0	66.0	54.0	58.0	37.0	32.0	21.0	24.0	19.0	12.0	8.0	11.0	12.0	2.0
18.0	14.0	2.0	NaN	5.0	12.0	24.0	31.0	41.0	33.0	13.0	11.0	14.0	7.0	32.0	31.0	32.0	37.0	20.0	14.0	6.0	4.0	6.0	14.0
47.0	9.0	1.0	NaN	5.0	17.0	31.0	27.0	72.0	64.0	63.0	30.0	48.0	71.0	77.0	44.0	72.0	41.0	29.0	58.0	9.0	78.0	54.0	2.0
NaN	2.0	2.0	1.0	NaN	1.0	10.0	10.0	15.0	13.0	9.0	11.0	18.0	9.0	19.0	7.0	17.0	12.0	19.0	14.0	6.0	7.0	7.0	2.0
NaN	NaN	NaN	NaN	1.0	1.0	6.0	6.0	13.0	12.0	12.0	11.0	10.0	13.0	19.0	14.0	13.0	8.0	13.0	6.0	9.0	6.0	2.0	NaN
NaN	NaN	NaN	1.0	NaN	1.0	2.0	2.0	5.0	3.0	1.0	3.0	3.0	6.0	5.0	4.0	4.0	NaN	NaN	1.0	NaN	1.0	1.0	NaN
254.0	138.0	74.0	78.0	130.0	312.0	1265.0	2467.0	3321.0	2722.0	2208.0	2274.0	2470.0	2581.0	3072.0	4027.0	4449.0	3768.0	3017.0	2278.0	1879.0	1460.0	727.0	326.0
70.0	33.0	12.0	5.0	17.0	59.0	99.0	114.0	187.0	169.0	166.0	118.0	134.0	145.0	165.0	114.0	142.0	114.0	87.0	98.0	29.0	100.0	79.0	20.0
324.0	171.0	86.0	83.0	147.0	371.0	1364.0	2581.0	3508.0	2891.0	2374.0	2392.0	2604.0	2726.0	3237.0	4141.0	4591.0	3882.0	3184.0	2376.0	1988.0	1560.0	806.0	346.0
234.0	168.0	87.0	80.0	147.0	329.0	1410.0	2705.0	3922.0	3037.0	2463.0	2633.0	2891.0	2877.0	3364.0	4311.0	4768.0	4319.0	3217.0	2426.0	2024.0	1678.0	765.0	354.0
6.0	5.0	3.0	3.0	9.0	19.0	41.0	47.0	54.0	73.0	81.0	64.0	70.0	60.0	65.0	45.0	32.0	23.0	27.0	12.0	6.0	8.0	16.0	3.0
22.0	26.0	3.0	NaN	4.0	13.0	35.0	32.0	36.0	43.0	10.0	12.0	9.0	14.0	28.0	26.0	30.0	35.0	16.0	12.0	9.0	5.0	3.0	11.0
8.0	57.0	NaN	NaN	2.0	18.0	53.0	27.0	99.0	74.0	59.0	38.0	67.0	53.0	83.0	67.0	82.0	63.0	67.0	30.0	13.0	84.0	21.0	2.0
2.0	4.0	1.0	2.0	NaN	4.0	13.0	13.0	10.0	18.0	12.0	13.0	15.0	13.0	14.0	17.0	14.0	11.0	7.0	15.0	6.0	17.0	5.0	3.0
1.0	4.0	NaN	NaN	2.0	NaN	3.0	5.0	13.0	11.0	13.0	12.0	13.0	19.0	10.0	17.0	9.0	4.0	6.0	3.0	6.0	1.0	NaN	NaN
NaN	1.0	NaN	1.0	2.0	NaN	1.0	1.0	3.0	5.0	2.0	1.0	3.0	2.0	3.0	6.0	3.0	1.0	NaN	NaN	2.0	1.0	1.0	NaN
235.0	173.0	87.0	81.0	151.0	329.0	1414.0	2711.0	3938.0	3055.0	2478.0	2646.0	2907.0	2898.0	3377.0	4334.0	4780.0	4324.0	3223.0	2429.0	2032.0	1680.0	766.0	354.0
38.0	92.0	7.0	5.0	15.0	54.0	142.0	119.0	199.0	208.0	162.0	127.0	161.0	140.0	190.0	155.0	158.0	132.0	117.0	69.0	34.0	114.0	45.0	19.0

Figure 6. Assignment 2 data hourly traffic measurements, in a difficult format to access.

The implementation of gamification did not meet the initial expectations, various concepts were explored but they did not fully come to fruition. The extensive coding required for integrating gamification at university level proved significant, limiting its application in this thesis. However, this experience highlights an important area for future research, suggesting the need to investigate alternative, less labour-intensive methods to incorporate gamification effectively.

7.2 Summary

The purpose of the thesis was to find out how to help students transition from theory to independently conducting data analytics projects in Metropolia UAS. The exact need for this was identified during the current state analysis (CSA). Two clear weaknesses were identified in the CSA, first the need for a guided project such as an assignment which would give students inspiration for future independent projects. Secondly significant overlap of curriculum topics was found between Business analytics courses and optional courses.

From the literature review the main findings revealed that getting students engaged and interested with the content was important. The tool that was researched for this was Jupyter Notebooks, allowing the incorporation of HTML elements to make the content interactive. A significant finding was the idea of gamifying assignments to keep students interested even in challenging topics.

The outcome of the thesis is a proposal for a small project-based course, split between two assignments. The proposal's assignments rely on open-source data, something that students most likely will work with during their future independent projects. The final proposal found in chapter 5, can be considered a success due to the validation by the lecturer in chapter 6.

References

Harvard (author-date) system:

Aparicio, M., Oliveira, T., Bacao, F. and Painho, M. (2018). Gamification: A key determinant of massive open online course (MOOC) success. *Information & Management*.

Erhard Rahm (2000). Data Cleaning: Problems and Current Approaches. *Data Engineering Bulletin*.

Hamari, J., Koivisto, J. and Sarsa, H. (2014). Does Gamification Work? -- a Literature Review of Empirical Studies on Gamification. *2014 47th Hawaii International Conference on System Sciences*.

Hamari, J. (n.d.). Gamification Motivations & Effects. [online] Available at: https://aaltodoc.aalto.fi/server/api/core/bitstreams/e2bc4939-4e34-4b98-bbce-a69e17271c77/content [Accessed 5 April 2024].

Holappa, S. (2020). Benefits of Systematic Degree Program Design 2/3: The GATE model. [Online] Available at: https://blogit.metropolia.fi/hiilta-ja-timanttia/2020/11/02/benefits-of-systematic-degree-program-design-2-3-the-gate-model/ [Accessed 20 March 2024].

IBM Cloud Education (2021). Structured vs. Unstructured Data: What's the Difference? [online] IBM Blog. Available at: https://www.ibm.com/blog/structured-vs-unstructured-data/ [Accessed 2 April 2024].

Jaggars, S.S. and Xu, D. (2016). How do online course design features influence student performance? *Computers & Education*.

Kluyver, T., Ragan-Kelley, B., Pérez, F., Granger, B., Bussonnier, M., Frederic, J., Kelley, K., Hamrick, J., Grout, J., Corlay, S., Ivanov, P., Avila, D., Abdalla, S., Willing, C. and Development Team, J. (2016). Jupyter Notebooks—a publishing format for reproducible computational workflows.

Koidan, K. (2021). Python Modules, Packages, Libraries and Frameworks. Master [Online] Available at: https://learnpython.com/blog/python-modules-packages-libraries-frameworks/ [Accessed 2 April 2024].

Kopf, D. (2017). Meet the man behind the most important tool in data science. [online] Quartz. Available at: https://qz.com/1126615/the-story-of-the-most-important-tool-in-data-science [Accessed 3 April 2024].

McAfee, A. and Brynjolfsson, E. (2012). Big Data: The Management Revolution. *Harvard Business Review*. Available at: https://hbr.org/2012/10/big-data-the-management-revolution [Accessed 18 March 2024].

Metropolia UAS, Tuotantotalous insinööri (AMK) päiväopiskelu. [Online] Available at: https://www.metropolia.fi/fi/opiskelu-metropoliassa/amk-tutkinnot/tuotantotalous [Accessed 18 March 2024].

Murty, M. Ramakrishna. (n.d.). CHAPTER-5 DATA PREPROCESSING References. Available at: http://dataminingzone.weebly.com/uploads/6/5/9/4/6594749/ch_5_data_preprocessing.pdf [Accessed 5 April 2024].

Nakamoto, T. (2023). 10 Essential Python Skills All Data Scientist Should Master [Online] Available at: https://www.datacamp.com/blog/essential-python-skills-all-data-scientists-should-master [Accessed 1 April 2024].

Ozgur, C., Colliau, T., Rogers, G., Hughes, Z., Myer-Tyson, E. (2017). 'MatLab vs. Python vs R', *Journal of Data Science*.

Pydata.org. (2019). pandas: powerful Python data analysis toolkit — pandas 0.25.3 documentation. [online] Available at: https://pandas.pydata.org/pandas-docs/stable/index.html [Accessed 1 April 2024].

Sarkar, T. (2022). Other Useful Skills to Master. In: Productive and Efficient Data Science with Python. Apress.

Smart Vision Europe (2017). What is the CRISP-DM methodology? [Online] Available at: https://www.sv-europe.com/crisp-dm-methodology/ [Accessed 1 April 2024].

Thangaraja, V., Sayeth, S. (2020). Popular Python libraries and their application domains. *ResearchGate*.

Wes McKinney. (2011). Python for data analysis: data wrangling with pandas, NumPy, and Jupyter.

Xu, J. and Frydenberg, M. (2021). Python Programming in an IS Curriculum: Perceived Relevance and Outcomes. *Information Systems Education Journal (ISEDJ)*, [online]. Available at: https://files.eric.ed.gov/fulltext/EJ1310052.pdf [Accessed 1 April 2024].

Appendix 1 Assignment_1.ipynb

Introduction level project

First data: Ilmanlaatu

Contains hourly data for air quality index measured at Mannerheimintie, Helsinki.

The air qulity index is used to describe the air quality in simple terms. The index takes into account the concentrations of sulphur dioxide (SO_2), nitrogen dioxide (NO_2), respirable particles (PM10), fine particles ($PM_{2.5}$), ozone (O_3) and the total reduced sulphur compounds (TRS). The measured concentrations are compared with the current air quality guidelines. https://www.ilmatieteenlaitos.fi/ilmanlaatuindeksi

Second data: Meteorological observations

Contains hourly meteorological observations gathered from Kaisaniemi weather station.

- **Havaintoasema:** The identifier of the observation station from which the data is collected.
- **Vuosi:** The year of the observation.
- **Kuukausi:** The month of the observation.
- **Päivä:** The day of the month for the observation.
- **Aika [Paikallinen aika]:** The time at which the observation was recorded, based on local time.
- **Ilman lämpötila keskiarvo [°C]:** The average air temperature.
- **Keskituulen nopeus keskiarvo [m/s]:** The average wind speed.
- **Puuskanopeus keskiarvo [m/s]:** The average gust speed.
- **Tuulen suunta keskiarvo [°]:** The average wind direction in degrees from true north. (180 in data means wind is blowing from south to north)
- **Vallitseva sää:** The prevailing weather conditions observed.
- **Näkyvyys keskiarvo [m]:** The average visibility in meters.
- **Lumensyvyys keskiarvo [cm]:** The average snow depth in centimeters.
- **Sademäärä keskiarvo [mm]:** The average amount of precipitation in millimeters.
- **Suhteellinen kosteus keskiarvo [%]:** The average relative humidity in percentage.
- **Pilvisyys [1/8]:** The cloud cover.
- **Ilmanpaine merenpinnan tasolla keskiarvo [hPa]:** The average sea-level air pressure in hectopascals.
- **Kastepistelämpötila keskiarvo [°C]:** The average dew point temperature in degrees Celsius.

Data sources: https://www.hsy.fi/ymparistotieto/avoindata/avoin-data---sivut/paakaupunkiseudun-ja-muun-uudenmaan-ilmanlaatuindeksit/
https://www.ilmatieteenlaitos.fi/havaintojen-lataus

Research questions

What are the top variables that effect air quality in the datasets? What outliers exist and what could be the reason?

```
# Import pandas, numpy, matplotlib.pyplot, seaborn
# Read in the 2 files ("data/filename")
df_1 =
# Reading excel files might require openpyxl library
df_2 =
```

Cleaning & Reformatting

Cleaning

```
# Take a look at the first DataFrame, what columns, datatypes, remove any NaNs, etc
# Repeat the same for the second DataFrame, pay attention especially to the date formats which are needed to merge the data
# Second DataFrame contains a lot of measurements for air quality index, pick mannerheimintie which should be the closest to Kaisaniemi, which is the measurement point for first DataFrame
# You can return here later and try a different air quality measurement point if interested

df_2 = df_2[['Aikaleima (loppuleimattu)', '???']]
```

Reformatting

As you could see both DataFrames have different formats of storing time. To make analysis later on possible a datetime column should be created now.

```
# Combine different columns to create Pandas DateTime objects
df_1['date_time'] = pd.to_datetime(df_1['Vuosi'].astype(str) + '-' +
                                   df_1['Kuukausi'].astype(str) + '-' +
                                   df_1['Päivä'].astype(str) + ' ' +
                                   df_1['Aika [Paikallinen aika]'],
                                   format='%Y-%m-%d %H:%M')
# Now create an identical format datetime column to the second DataFrame, the .loc is used to avoid pandas warning
df_2.loc[:, 'date_???'] = pd.to_datetime(df_2['???'], format='???')
```

Well that didn't work...

Pandas datetimes are a type of data structure is specifically designed to handle date and time information efficiently. They are useful because they allow for easy manipulation, formatting, and arithmetic operations with dates and times, making it simpler to analyze time series data or perform date-based filtering and grouping in data analysis tasks.

The second DataFrame contains illegal times, so pd.to_datetime won't work automatically. The ISO8601 standard expects hours to fall between 0-23. The hour 24 is ambiguous, meaning that it is not clear if it should be the current or next day.

```
# Fix the 24:00 hour format so that pd.to_datetime method can be used.
def correct_datetime(s):
    """
    This function replaces the illegal 24:00 format values with 00:00 and adds 1 day
    You don't need to touch this function. Just use the Pandas .apply() method to the
desired DataFrame.
    :param s:
    :return:
    """
    if s.startswith('24:00'):
        date_part = s.split(' ')[1]
        date_corrected = pd.to_datetime(date_part, format='%d.%m.%Y') +
pd.Timedelta(days=1)
        return '00:00 ' + date_corrected.strftime('%d.%m.%Y')
    return s

# Use .apply() to run a function on each cell. df['column/s'].apply(function)
df_2.loc[:, 'Aikaleima (loppuleimattu)'] =

# Finally convert the DateTime to the same format as the first DataFrame, hint copy the
line from the cell that gave error before...

# The DataFrame should now have a new column at the end
# This cell should output True if everything is correct.
# The reason for different indexes is because of the non-ISO8601 standard format of
keeping time.
# First DataFrame starts counting from hour 0 and second from hour 1. This shouldn't cause
any major bias ??????
df_1['date_time'][1] == df_2['date_time'][0]
```

Merge & clean again

Goal here is to prepare the dataset for proper analysis.

```
# Combine the data from df_1 and df_2 into a single dataframe, merged_df. Only include
rows where the 'date_time' values match in both dataframes.

merged_df = pd.merge(df_1, df_2, on='date_time', how='inner')
```

The merged_df should now have 20 columns and 8782 rows. Take a look at the columns and delete the ones that you deem unnecessary. You can always come back and add or delete columns!

```
merged_df = merged_df # drop columns = that are not needed
```

Exploratory Data Analysis

Are the data types what you would expect them to be? Trends in different variables? Correlations?

```
merged_df.dtypes
```

```
# Convert data types to a format that is suitable for calculations, dtype: object, is not suitable
# For data type conversion use pd.to_numeric(DataFrame, errors='') errors can be handled with raise, coerce, ignore
# When converting data make sure date_time is not converted to anything, it should be datetime64[ns]



# Double check that all .dtypes are correct
# Also good to replace any possible NaNs created now
# To create visualisations that are easier to interpret, we can resample the data to calculate mean values
# Resampling requires the index to be datetime
merged_df.set_index('date_time', inplace=True)
# Without resampling basic plots look like this
plt.plot(merged_df.index, merged_df['Mannerheimintien indeksi'])
# Resampling could also be done for months by using 'M' instead of 'D'
daily_data = merged_df.resample('D').mean()
```

Look at possible trends

Try to get a decent understanding of the data with different plots and correlations with air quality.

```
fig, axs = plt.subplots(2, 2, figsize=(12, 10))

axs[0, 0].plot(daily_data.index, daily_data['Mannerheimintien indeksi'])
axs[0, 0].set_title('Mannerheimintien indeksi')
axs[0, 0].set_ylabel('Air quality')

axs[0, 1].plot(daily_data.index, daily_data['???'])
axs[0, 1].set_title('???')
axs[0, 1].set_ylabel('???')

axs[1, 0].plot(daily_data.index, daily_data['???'])
axs[1, 0].set_title('???')
axs[1, 0].set_ylabel('???')

axs[1, 1].plot(daily_data.index,  daily_data['???'])
axs[1, 1].set_title('???')
axs[1, 1].set_ylabel('???')
# Create a correlation matrix using pandas .corr()

# Correlation matrix can be neatly visualised using seaborn heatmap
```

Machine learning

Since there doesn't seem to be significant correlation between air quality index & any other variable in the dataset, machine learning can be leveraged to identify if more complex relationships exist between the variables.

```
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error


from sklearn.model_selection import GridSearchCV
```

```
from sklearn.ensemble import RandomForestRegressor
X = merged_df.drop('Mannerheimintien indeksi', axis=1)
y = merged_df['Mannerheimintien indeksi']

# Hint: look at the imports

# Split training data X_train, X_test ...
# Scale, scaler = StandardScaler()
# Pick a model and fit to training data
# Predictions

# Evaluate the model
mae = mean_absolute_error(y_test, predictions)
print(f"Mean Absolute Error: {mae}")
# Try another model
# Optional model tuning
```

Before moving on

Try to get the Mean Absolute Error below 7.

First things to try:

Try using GridSearchCV

Consider what variables should be added or removed A MAE of ~6.65 should be achievable with columns: ['Kuukausi', 'Aika [Paikallinen aika]', 'Ilman lämpötila keskiarvo [°C]', 'Keskituulen nopeus keskiarvo [m/s]', 'Puuskanopeus keskiarvo [m/s]', 'Tuulen suunta keskiarvo [°]', 'Näkyvyys keskiarvo [m]', 'Lumensyvyys keskiarvo [cm]', 'Sademäärä keskiarvo [mm]', 'Suhteellinen kosteus keskiarvo [%]', 'Ilmanpaine merenpinnan tasolla keskiarvo [hPa]', 'Kastepistelämpötila keskiarvo [°C]', 'Mannerheimintien indeksi']
A specific hour column can be created with: merged_df['Aika [Paikallinen aika]'] = merged_df['Aika [Paikallinen aika]'].str.split(':').str[0].astype(int)

Predict current air quality

Go to https://www.foreca.fi/Finland/Helsinki, fill out the values below and test your best model. Are the results what you expected? If you can't find a certain value, use the mean

```
merged_df.columns
weather_now = {
    'Kuukausi' : [5],
    'Aika [Paikallinen aika]': [14],
    'Ilman lämpötila keskiarvo [°C]': [14.4],
    'Keskituulen nopeus keskiarvo [m/s]': [4],
    'Puuskanopeus keskiarvo [m/s]': [5],
    'Tuulen suunta keskiarvo [°]': [180],
    'Näkyvyys keskiarvo [m]': [37000],
    'Lumensyvyys keskiarvo [cm]': [0],
    'Sademäärä keskiarvo [mm]': [0],
    'Suhteellinen kosteus keskiarvo [%]': [53],
    'Ilmanpaine merenpinnan tasolla keskiarvo [hPa]': [1020.8],
```

```
    'Kastepistelämpötila keskiarvo [°C]': [5]
}
new_data_df = pd.DataFrame(weather_now)
new_data_scaled = scaler.transform(new_data_df)

#predicted_values = model.predict(new_data_scaled)
predicted_values = best_rf_model.predict(new_data_scaled)
print("Predicted 'Mannerheimintien indeksi':", predicted_values[0])
```

Check current air quality, https://www.hsy.fi/ilmanlaatu-ja-ilmasto/ilmanlaatu-nyt/ and compare your predicted value to the table below
Possible reasons for results: -Model is overfitting -Temporary construction work (Mannerheimintie is undergoing major renovations) -Consider how close would the prediction be accounting MAE -For this project the day current weekday was not considered, for example monday might have more traffic than sunday -etc.

	Indeksin väri	**BC**	**LDSA**	**PNC**
Erittäin huono (≥151)	Violetti	≥12	≥120	>101
Huono (101-150)	Punainen	7-12	80-120	61-100
Välttävä (76-100)	Oranssi	3-7	40-80	31-60
Tyydyttävä (51-75)	Keltainen	1-3	20-40	16-30
Hyvä (≤50)	Vihreä	≤1	≤20	<15

Source: https://www.hsy.fi/ilmanlaatu-ja-ilmasto/mika-on-ilmanlaatuindeksi/ For specific air quality number: https://waqi.info/fi/#/c/60.183/24.925/10.5z
https://aqicn.org/city/finland/helsinki/mannerheimintie/m/
https://www.hsy.fi/ilmanlaatu-ja-ilmasto/ilmanlaatu-nyt/

Appendix 2 Assignment_2.ipynb

Intermediate project

Traffic data collected with LAM

What is LAM

The LAM works by electromagnetic induction of a loop embedded in the pavement, where the metallic mass of the vehicle causes a change in the magnetic field of the loop. The LAM consists of two induction loops and a data acquisition unit in each lane. The LAM device registers the vehicles crossing the point, providing for each vehicle the time of the pass, direction of travel, lane, speed, vehicle length, time difference between successive vehicles and vehicle category. There are more than 450 stations in Finland. source: https://www.digitraffic.fi/tieliikenne/lam/

Download time matching data or folder

Material: Liikennemäärät Report: Tuntiliikenneraportti Time: 2023-01-01 - 2023-12-31 LAM: vt1_Munkkiniemi

https://tie.digitraffic.fi/ui/tms/history/ https://www.ilmatieteenlaitos.fi/havaintojen-lataus

https://www.digitraffic.fi/tieliikenne/lam/#lam-raakadata 1 HA-PA (henkilö- tai pakettiauto) 2 KAIP (kuorma-auto ilman perävaunua) 3 Linja-autot 4 KAPP (kuorma-auto ja puoliperävaunu) 5 KATP (kuorma-auto ja täysperävaunu) 6 HA + PK (henkilöauto ja peräkärry) 7 HA + AV (henkilöauto ja asuntovaunu) 8 MP (Moottoripyörät ja mopot) 9 HCT (High Capacity Truck)

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt

import hashlib

def hash_answer(answer):
    return hashlib.sha256(answer.encode()).digest().hex()

def verify_answer(provided_answer, stored_hashed_answer):
    return hash_answer(provided_answer) == stored_hashed_answer

solution_1 = 'df7e70e5021544f4834bbee64a9e3789febc4be81470df629cad6ddb03320a5c'
solution_2 = '559aead08264d5795d3909718cdd05abd49572e84fe55590eef31a88a08fdffd'
solution_3 = '204ac66a8856d0cc4eca88d8a2560079f64a9aaaa840308817934574949e04ee'
```

Cleaning & reformatting

1. Read in the data
2. Drop unnecessary columns
3. Remove NaNs, reformat time & etc
4. Merge DataFrames

```
df_1 = 'Weather'
df_2 = 'Traffic'
# df_1 cleaning and reformatting
# df_1 drop columns

# df_1 is ready
# df_2 replace NaNs, drop columns, 'yhteensa' should atleast be dropped
# df_2 reformatting, pvm to '%Y%m%d' format


# Melting the DataFrame to convert hour columns to rows using .melt()
#df_2 = df_2.***(id_vars=['ajoneuvoluokka', 'pvm'], var_name='Hour',
value_name='Measurement')
# Extract the hour and convert to 0 based time
#df_2['Hour'] = df_2['Hour'].str.extract('(\d+)').astype(int)
# Create identical date_time for df_2 to enable merging
pd.to_datetime(df_2['pvm']) + pd.to_timedelta(df_2['Hour'], unit='h') # Combines a date
and hour to create...
# Merge df_1 and df_2 just like in assignment 1
# Set the date_time as the index, inplace=False

# Code below will only work if index is set correctly. It creates a Weekday column to
indicata Monday, Tuesday... as numbers 1, 2...
#merged_df['Weekday'] = merged_df.index.to_frame()['date_time'].dt.dayofweek + 1

# After this all datetimes should be ready, column 'pvm' can be dropped
```

EDA

1. Data types
2. Resampling
3. Plots to answer questions below
4. Correlations

```
# Just like in previous project check dtypes and do the appropriate conversions
# And just like in the previous project resample the data
# Using the code below the desired vehicle type can be selected for resampled DataFrame
#daily_df = merged_df[merged_df['ajoneuvoluokka'] == 'kaikki'].drop('ajoneuvoluokka',
axis=1)
# Code below can create a neat looking scatter plot with points colored by the day of the
week, but it's missing parts
#sns.scatterplot(data=, x=daily_df.index, y='y', hue='Weekday', palette='Paired')
# Check correlations, again using heatmap makes this easy
# Can help you answer question 3
for i in merged_df['ajoneuvoluokka'].unique():
    corr = merged_df[merged_df['???'] == i]['Measurement'].corr(merged_df['???'])
    print(f'Correlation of {corr} for {i}')
```

Questions

Question 1: What are the 2 days that on average have the least traffic?

- A) Monday & Sunday
- B) Saturday & Sunday
- C) Monday & Saturday

Question 2: What single weekday has had the lowest amount of traffic?

- A) Saturday
- B) Monday
- C) Tuesday
- D) Sunday

Question 3: What vehicle type (ajoneuvoluokka) is most effected by snow?

- HA - PA
- KAIP
- Linja-autot
- KAPP
- KATP
- HA + PK
- HA + AV
- Kevyet
- Raskaat
- kaikki
- MP

Question 4: There are 2 major drops in the data, when are these and what do you think the reason is?

```
# Replace ? with choice A,B,C...
question_1_answer = "?"
question_2_answer = "?"
question_3_answer = "?" # Vehicle type, for ex. : "kaikki"

print('Answer to question 1:', verify_answer(question_1_answer, solution_1))
print('Answer to question 2:', verify_answer(question_2_answer, solution_2))
print('Answer to question 3:', verify_answer(question_3_answer, solution_3))
```

Machine learning

Goal here is to look under the hood of a RandomForest model using feature importances.

feature_importances_ in machine learning indicates how much each feature (like time or weather) contributes to the predictions made by a model, especially in tree-based

models like decision trees. It helps identify which features are most influential, though it doesn't show whether their impact is positive or negative on the prediction.

Alternative to feature_importances_ is sklearn.inspection.permutation_importance it should provide a more reliable measure of a feature's impact on model accuracy.

Code is provided ready, you only have to change the type of vehicle. Take a look at the plot and see how feature importances change based on the data selected.

```
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_absolute_error
from sklearn.preprocessing import StandardScaler

from sklearn.ensemble import RandomForestRegressor
#
hourly_df = merged_df[merged_df['ajoneuvoluokka'] == 'kaikki'] # Change type here
hourly_df = hourly_df.drop(['ajoneuvoluokka', 'Weekday'], axis=1) # Change also the
columns
hourly_df = hourly_df.fillna(0)
# TEST BOTH HOURLY AND DAILY
X = hourly_df.drop('Measurement', axis=1)
y = hourly_df['Measurement']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=73)
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
random_forest = RandomForestRegressor(n_estimators=100, random_state=42)

random_forest.fit(X_train, y_train)

predictions = random_forest.predict(X_test)
mae = mean_absolute_error(y_test, predictions)
print(f"Mean Absolute Error with Random Forest: {mae}")
feature_importance = random_forest.feature_importances_
plt.barh(range(len(feature_importance)), feature_importance, align='center')
plt.yticks(range(len(feature_importance)), X.columns)
plt.xlabel("Feature Importance")
plt.title("Feature Importance")
plt.show()
```