



ANNA PELTOMAA

Data analysis and data visualization tools for sales and sales predicta- bility on E-commerce platform

DEGREE PROGRAMME IN DATA ENGINEERING
2024

CONTENTS

1 INTRODUCTION	5
2 PURPOSE OF THE STUDY.....	6
3 METHODS.....	7
3.1 Case study	7
3.2 Software development process models	8
3.2.1 Waterfall model	8
3.2.2 Software Development Life Cycle (SDLC).....	9
3.2.3 Spiral model	9
3.2.4 Agile development.....	10
3.3 Requirements analysis.....	11
3.4 Thesis schedule	11
4 KEY CONCEPTS	11
4.1 Data analytics	11
4.2 Data visualisation.....	12
5 PROGRAMMING AND OTHER LANGUAGES.....	12
5.1 Python.....	12
5.2 CSS (Cascading Style Sheet).....	13
5.3 HTML (HyperText Markup Language)	13
6 LIBRARIES AND ENVIRONMENTS	13
6.1 Pandas.....	13
6.2 Plotly	14
6.3 Jupyter lab	14
6.4 Anaconda Distribution.....	14
6.4.1 Anaconda Navigator.....	14
6.5 Visual Studio Code	15
7 FRAMEWORKS	15
7.1 Django.....	15
7.2 Dash.....	15
7.3 Streamlit.....	16
8 VERSION CONTROL.....	16
8.1 Git	16
8.2 GitHub.....	17
9 FILE FORMAT.....	17
9.1 CSV (Comma separated value).....	17
10 DEVELOPING THE SOFTWARE.....	18

10.1 Requirements for the software.....	18
10.2 First steps, Inspecting the data.....	19
10.3 Creating the Django app.....	19
10.4 Testing visualisations.....	20
10.5 Creating the app with Dash.....	23
10.5.1 Dash HTML.....	25
10.5.2 Dash Core Components module.....	25
10.5.3 React.....	25
10.6 Streamlit framework, yet another possibility	25
10.7 Connecting e-commerce platform and app via Api.....	26
10.8 Cron	26
11 TESTING THE SOFTWARE	27
12 CONCLUSION.....	28
REFERENCES.....	30

ABSTRACT

Anna Peltomaa: Data analysis and data visualisation tools for sales on E-commerce platform

Bachelor's thesis

Degree programme: Data Engineering

August 2024

Number of pages: 33

This practice-based case study which included a good part of requirements analysis for a software company Phenomenal Development Oy was conducted as a project. The purpose of this study was to take the first steps with data-analysis for the company and for the writer of the thesis to gain more understanding about software development and data-analysis in real life settings.

The planning started by creating a schedule for the project and it was planned to take place between autumn 2023 – May 2024. Development begun by gathering requirements for the software and doing research about the subject. Finding the right software development model and finding the key concepts to clarify the direction of the thesis was done in the early stages of planning the thesis. Few meetings were held with the representative of the company to plan the software.

When developing the software, it became clear that some of the tools/choices made in the beginning to develop the software were not the best ones even though they were suitable, and some changes were made to tools to create the software with. The first plan was to create the software with Django and Cufflinks and at end, Dash was used to create the software with Plotly.

From the beginning it was self-explanatory, that creating the software can lead to changing the software used and planned tools and it was recognised to be part of the process and learning when getting more information about software and possibilities available. In the beginning the idea was to also develop a prediction model for the sales, but it was decided to leave out at the end, because the project got delayed by several months till august 2024. Finally, the software was created, and it is now running in its own server. This offers also a very good point for future development.

Keywords: Data-analysis, Software development, Visualisations

1 INTRODUCTION

There is a lot of value in data analytics everywhere, where data is gathered. Visualisations are a helpful way for data scientists and data analysts to find for example missing data and draw statistical conclusions, but it has another kind of purpose for business owners when viewing sales statistics.

Data collection and analysis has been known for a long time. One example of data analysis is from mid 1800 hundreds when the pioneer of ocean navigation Mathew Fontaine Maury studied meteorology, gathered ships logs and created charts. It revolutionized the understanding of oceanography and marine navigation in 1920's Benjamin Graham and David Dodd created a security analysis which involved analysis of financial data to inform investment decisions.

Today's modern data analysis was enabled by computing and the credit goes many times to John Tukey for writing a paper "The Future of Data analytics" in 1962. In the paper he described procedures for analysing data, interpreting results, planning to gather the data and the statistics to be applied. In 1974 Peter Naur published a paper where he used consistently term data science, but many give the credit to William Cleveland for his publication in 2001 where he concluded, the data science should include multidisciplinary investigations, models and methods for data, computational systems, pedagogy for education, evaluation of tools and theoretical foundations (Liebowitz. pp.2,3).

This thesis is practise -based case study and is conducted for a technology company Phenomenal Development Oy.

2 PURPOSE OF THE STUDY

The purpose of this study is to create a foundation for data analytics and data visualization tools utilizing to existing e-commerce platform as data source.

First step is to create tools for company's internal use and later to customer use so, that it adds value to customers, who are using the e-commerce platform for their business purposes. As a result of this thesis, the plan is reviewed and implemented to existing software and one of the goals all in all is for this to be a first step of developing data analysis and visualisations for the company.

The results of this thesis are meaningful and mostly important for the company and its customers but at the same time it takes part in advancing and using data and analytics. Customers can view their sales and other data produced, can get a better understanding of their business through visualizations and at the same time, elevate customer experience as well as the versatility of the E-Commerce platform.

This thesis is conducted for Phenomenal Development Oy. Phenomenal development Oy is a technology start-up company founded 2020 in Rauma Finland and the focus of the company has been developing custom software. The E-commerce platform is the main product and a need for adding data visualisations as part of the existing software and not using existing software's such as Tableau, Microsoft Power Bi or Qlik, was recognised. Instead of implementing existing software from other companies, there was a need to create an application purely and only for the companies use. This way, the software is not dependent on other parties and the functionalities of the software can be altered as needed.

3 METHODS

This thesis was conducted as a case study and it will be carried out as a project. The programming language used in the software was Python. Several different libraries were considered as well as used. It was expected the planned libraries to be used might change, when development starts depending on which of the choice's suite the best for the purpose. In the beginning these libraries and frameworks were for example considered: Pandas, Pyspark, Matplotlib, Seaborn, decision trees and possibly others such as Plotly and cufflinks, Django as a framework. At the moment, the plan was to embed or transfer data between servers via API from created application to existing software. Visual studio code is going to be used, Git and GitHub for version control.

This thesis is a case study with a good part of requirements analysis in it. First the theoretical part of chosen methods is written, and subject is introduced. Secondly the tools/python libraries and visualisation tools are chosen for the software to create the overall picture of the thesis. Finally, the software will be created, tested, and implemented.

3.1 Case study

Case study in software engineering is a practical way of investigating a phenomenon in a real-life settings like new technologies or project risks and failure factors. Case study is in the contrast to specified population, a study of very specific case. Software engineering research has high level objectives to increase understanding of how and why software engineering should be started and with this knowledge to improve the engineering process and the resulting software products (Runeson, 2012 p. 3,7)

3.2 Software development process models

There are different development models which could be used in the software development process. Below there are few of the well-known models.

3.2.1 Waterfall model

Waterfall model as in Figure 1. was designed to meet the needs for large software development. Waterfall begins with requirements gathering followed by development steps finishing to maintenance in which the development process ends.

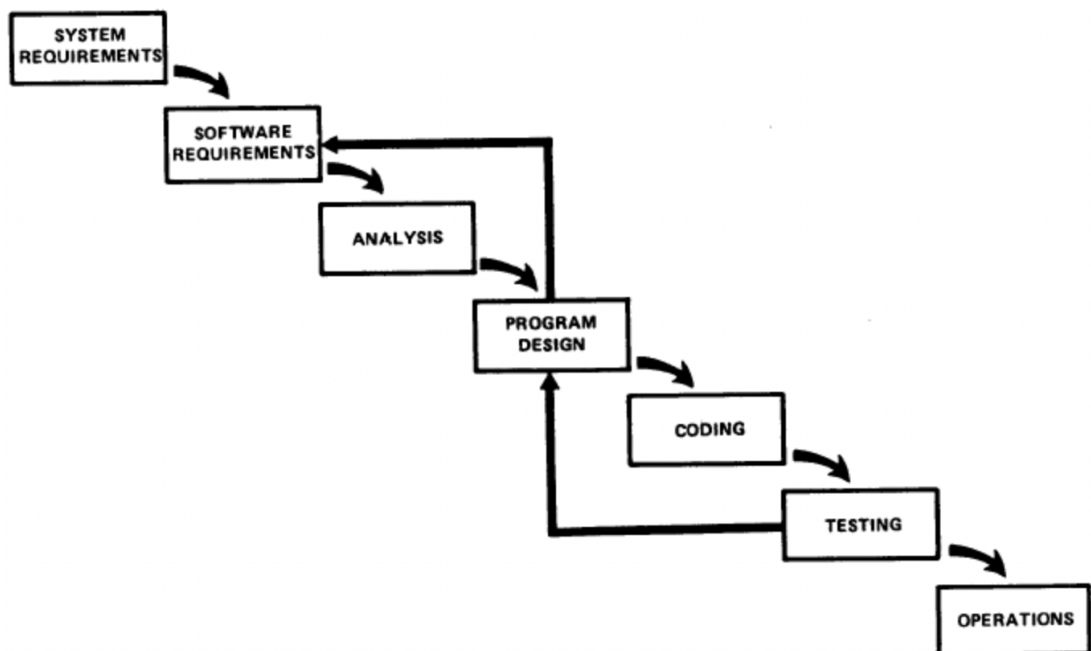


Figure 1. (Kemp Peter 2010, Wikipedia).

3.2.2 Software Development Life Cycle (SDLC)

SDLC as in Figure 2. introduces a systematic approach to software development by creating structure for developers. Its aim is to create and deliver high quality software to meet the requirements of a customer and at the same time to be cost effective.



Figure 2. Development life cycle (Ample WebSol 2018).

3.2.3 Spiral model

Spiral development model combines Waterfall model and SDLC and gets its name from the spiral figure as in Figure 3. Model was first introduced in 1986 by Barry Boehm and it's used for large and complex software.

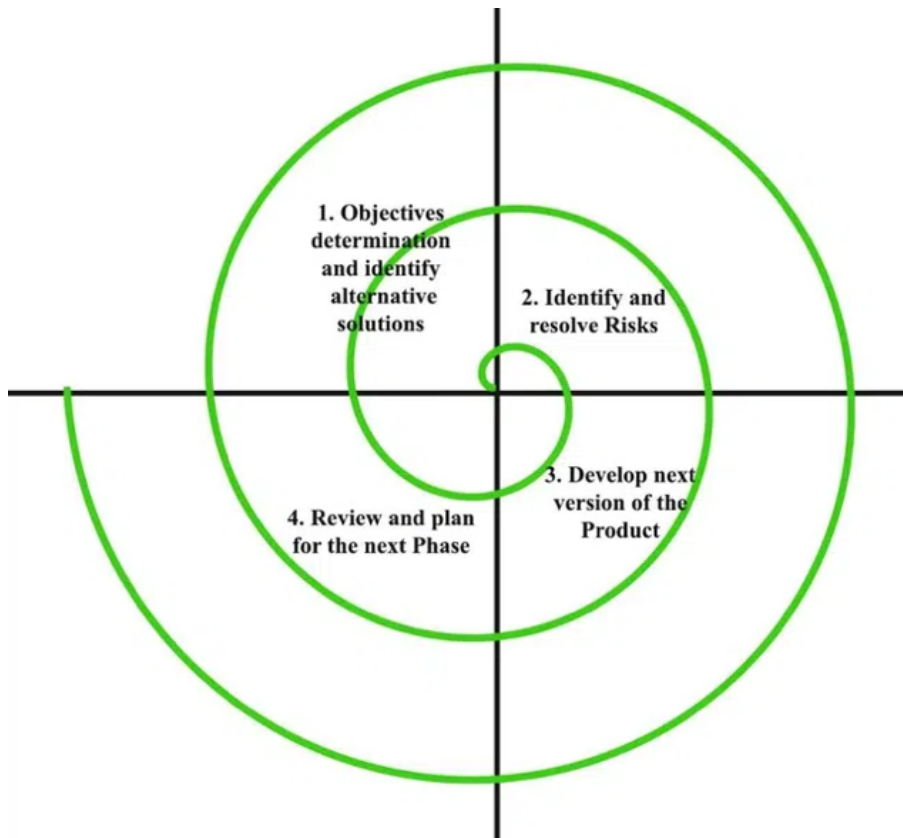


Figure 3. Spiral model (GeegsForGeeks 2024)

3.2.4 Agile development

Agile software development model aims to bring practical flexibility to a software production process and working in collaboration rather than contract negotiation. Testing is done continuously throughout the process which ensures high quality and provides an opportunity to make changes as needed. Agile model has six steps: concept, inception, iteration/construction, release, production, retirement and can be visualized like in Figure 4.

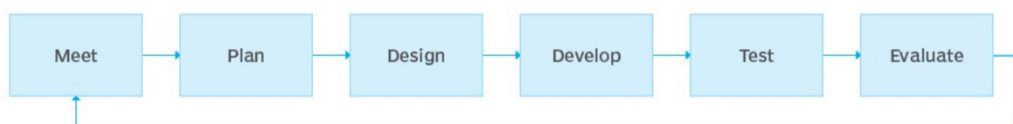


Figure 4. TechTarget (Brush,Silverthorne 2024)

3.3 Requirements analysis

Requirement analysis is a process where the principles and preconditions are documented (Tonder 2013, 103). Requirements describe what the software needs to fulfil to be successful (Arnowits, Arent & Berger 2007, 574)

Requirements of the software were discussed in detail in 05/2024. First it was brought to attention what kind of data is gathered at the moment for it will be the foundation for this project such as purchase date and time, amount of sold products, product id and name, price among others.

3.4 Thesis schedule

Autumn 2023

- Getting familiarized with the subject

Spring 2024

- February - March, thesis plan
- March - April developing the application
- April – May testing the application
- May-June, finalizing the thesis

4 KEY CONCEPTS

4.1 Data analytics

Data analytics is a broad field. The primary fields are predictive, diagnostic, descriptive and prescriptive analytics. Descriptive analysis analyzes from multiple data sources, and it gives valuable insight of the past. Predictive analysis tells us what is likely to happen. Diagnostic analysis gives us insights into a problem and why the problem is happening. Prescriptive analysis tells us what

we need to do in the future if problem occurs based on the information about what has happened (Liebowitz. p.23).

4.2 Data visualisation

Data science is rising, and it has fuelled data visualisation techniques. It has led to many new data visualisation tools and libraries being developed. Programming environments like Python has these open-source libraries embedded like Matplotlib. Python especially has libraries available for high-performance computing, numerical calculations, regression modelling and visualisations. These are also regularly maintained (Shirowzhan, 2022 p. 90)

Data visualisation can feature for example basic charts, plots such as scatter plots, line charts, bar charts and bubble charts. More advanced techniques are chord diagrams, horizon graphs, heat maps, scatter plot matrices and multi-dimensional techniques like parallel coordinates and radar charts. Some existing visualisation techniques have not been integrated into any tool or application yet, such as time nets, data vases and people garden (Shirowzhan, 2022 p. 98,99).

5 PROGRAMMING AND OTHER LANGUAGES

There are many options to choose from when developing a software. It depends on a purpose and the needed functionalities, which languages, libraries and frameworks are being used. Here are the libraries and Frameworks considered when developing the software.

5.1 Python

Python is widely used, easy to learn, high-level interpreted programming language. It was created by Guido Van Rossum and first released in 1991 and

since then there have been thousands of contributors making it possible to spread all over the world. Python is a mature language, and it is a great choice for data analytics and machine learning, software development, web development and many others (Python Institute, 2024).

5.2 CSS (Cascading Style Sheet)

CSS is commonly used and known styling method. It is used to control how the page looks. CSS can be used to specify that a typeface should be certain type such as large bold Arial type and the background should be light green. CSS can be used to control which items appear on the page (Larsen 2013, XXXIII).

5.3 HTML (HyperText Markup Language)

A web page has a certain structure and HTML is needed to explain the structure. It contains the information about which words form a heading, where paragraphs begin and end and if there are lists or other bullet points. It also specifies links between different web pages and where images should appear, location of the videos and forms for entering text (Larsen 2023, XXXIII).

6 LIBRARIES AND ENVIRONMENTS

6.1 Pandas

Pandas is one of the Python packages. It is a tool for real world practical data-analysis in python. It provides fast, flexible and expressive data structures and designed to work with relational or labeled data easily (PyPi 2024).

6.2 Plotly

Plotly Python library is an interactive open-source plotting library, that supports over 40 unique chart types covering a wide range of statistical, financial, geographic, scientific, and 3-dimensional use-cases. It enables creation of web-based visualizations (Plotly 2024).

6.3 Jupyter lab

Jupyter Lab is web-based and interactive environment for notebooks, code and data. It allows to configure workflows in data-science, scientific computing and machine learning and is available for multiple programming languages (Jupyter).

6.4 Anaconda Distribution

Anaconda Distribution is open-source software, and it is suitable for data visualizations among others. It features over 8000 data science and machine learning packages (including Pandas) for all major operating systems and architectures (Anaconda, 2024).

2012 Anaconda was founded by Peter Wang and Travis Oliphant. There was a need to bring Python into business data analytics and since that, Python has become popular language. Anaconda provides Data science tools, Machine learning Ops and data & model management (Anaconda, 2024).

6.4.1 Anaconda Navigator

Anaconda Navigator is a graphical user interface (GUI) for desktop and it includes Anaconda® Distribution. It ables you to launch applications and manage conda packages, environments, and channels without using command line interface (CLI) commands (Anaconda Inc. 2024).

6.5 Visual Studio Code

Visual Studio Code is a source code editor. It runs on desktop, and it is available for Windows, macOS and Linux. It has wide range of extensions for other languages including Python (Visual Studio Code, 2024).

7 FRAMEWORKS

7.1 Django

Django is an open-source Python framework which has built-in support for database querying, URL mapping and template rendering (web development with Django 2021 p. 2-3)

Django was designed to be fast. Application can be launched from concept in only hours. It is very secure and helps to avoid common security issues like SQL injections, cross-site scripting, cross-site request forgery, clickjacking and user authentication system makes it secure to manage user passwords and accounts. Scalability is good and it can manage heavy traffic (Djangoproject, 2024)

Django was invented by Lawrence Journal-Worlds web developers in 2003 and it became public in 2005. It was developed to meet the short deadlines in the newspaper (w3schools, 2024)

7.2 Dash

Dash is a low-code framework for rapidly building data apps in Python using Plotly. There is a free community edition available for non-commercial use, Dash enterprise for commercial use and it contains more features. Dash enables interactive use of graphs and there is no need for CSS, HTML, JavaScript,

unless you want to use them. Styling is recommended to be done with Dash Enterprise Design kit, which requires purchases (Plotly 2024).

Dash is built on Flask which takes care of the backend and React taking care of the frontend. Good documentation and tutorials on communal site helps a long way providing lots of information and examples

7.3 Streamlit

Streamlit is an open-source Python framework for creating dynamic data apps with only a few lines of code, no frontend coding experience is required.

Streamlit is owned by Snowflake, a fully managed SaaS (software as a service). It provides a single platform for data warehousing, data lakes, data engineering, data science, data application development, and secure sharing and consumption of real-time / shared data (Narayan, 2023).

With Streamlit its possible to transfer Pythons scripts to web apps fast. It is good for fast prototyping, and it is fully free. Apps can be shared Via Streamlits community cloud platform (Snowflake Inc 2024)

8 VERSION CONTROL

8.1 Git

Git is a distributed version control system. It is open source and designed from small to large projects fast and effectively. It enables the use of multiple branches locally, making it easy to try development ideas and delete unwanted branches easily (Git, 2024).

8.2 GitHub

GitHub and Git can be used to collaborate on work. You can store, share and work writing code in a cloud-based Platform GitHub. GitHub allows you to save your work to repositories and this enables you to share your work, track and manage changes, let others review your code and collaborate on projects (GitHub, 2024).

9 FILE FORMAT

File formats are used by Apps or programs and provide mechanisms to store data in a structured way. File formats can organize metadata as well as and data such that the specific App can use (Hummert 2022, p.128).

9.1 CSV (Comma separated value)

CSV is a format of text file where commas separate the values where typically each line represent one data record. It is common data exchange format and at the same time widely supported. Most of the database programs export files as CSV and most of the spreadsheet programs read CSV files. It can be used for storing data and Pandas, data science tool, includes the option to export data to CSV for long term storage (Wikipedia, 2024).

10 DEVELOPING THE SOFTWARE

10.1 Requirements for the software

Meeting to discuss the software's requirements was conducted in may 2024. Although there had been few brief discussions earlier on, this was the meeting where all the requirements were discussed in more detail.

It became clear, this software should be separate from the main software for many reasons one of them being securing the functionality of the main software.

Some of the requirements for the app was:

- Items sold
- Prices
- Dates
- Daily sales as currency
- Number of sales
- Possibility to export visualizations and data

Visualization should be categorized and at least shown as:

- Lines
- Pie Charts
- Downloadable files of visualizations

Data could be gathered from the main software by fetching data time to time and analytics software does the analysis Reports and visualizations should have its own server. Having its own server always ensures the functionality of the main software so, that the new piece of software doesn't compromise the existing software in any circumstance. The data-analysis and visualization software shouldn't be active all the time but to be able to run the software every so often and do the calculations and visualization only if there has been a change in the data.

The colors used will be gathered from the existing colors of the platform to create a consistency between these two software's.

It was decided that Agile software development model is going to be used, because it enables more flexible approach to the development process.

10.2 First steps, Inspecting the data

The first step was to install all necessary programs and libraries to get started. At the same time, it was checked if free of charge assets could be used in an environment, which is going to be later in commercial use. As search went on, the result was, many of the communal programs free of charge can't be used when creating an app to commercial use and that must be considered when creating the final app. In the development at its early stages these free of charge programs are good to test and to see if they are worth of buying or not. One the tools that was briefly considered was PyCharm IDE (integrated development environment) for python data-science and web development but at this point it wasn't necessary and benefits of using something like that in this project seemed too low to just try it for fun.

The first steps included a small test CSV -file, exported from the e-commerce platform database, and Jupyter Lab using Anaconda Navigator and Pandas. Anaconda Navigator was chosen in the beginning of the development phase, because it was easy to launch and use, and it allows many applications to be launched from the same place if the need would rise. Jupiter Lab was used mainly in this stage for data analytics to discover and test how existing data could be used, what kind of data frames created and visualised for the project.

10.3 Creating the Django app

After installing Django and all other necessary programs and libraries, creating a Django app was easy.

The project was created in virtual environment and Visual Studio Code, was connected straight to Git and GitHub to ensure all the code could be used from multiple computers and to keep the code safe in its own repo.

10.4 Testing visualisations

Testing visualisation started with cufflinks. One commonly used visualisation tool is an interactive filled Line Chart, and it was also tested out in this software as visualisation method as seen in figure 5.

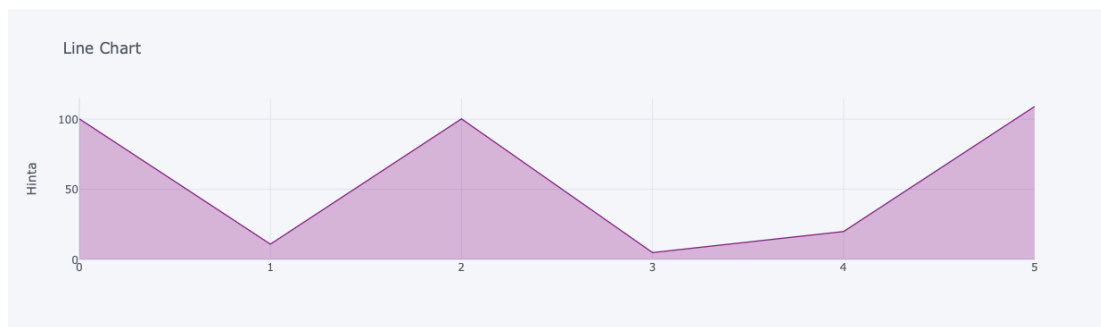


Figure 5. Line chart.

Some of the Cufflink's visualisations couldn't be used due to errors in the actual cufflinks package. Errors seemed to be there already three years ago, and hadn't been fixed so few questions rose then about how well will it perform in the future. It seemed cufflinks could be used in some of the visualisations but as the tests went on with Jupyter lab and especially later, when transferring codes to Visual Studio Code, it was obvious cufflinks would not be the best fit. There were problems with Plotly versions and Cufflinks would need to have quite old version of Plotly to run. It was then decided not to be used due to these uncertainties.

Visualisation library Seaborn was not as appealing and Matplotlib was checked which seemed ok. Finally, Plotly was checked on its own and discovered.

We can see Plotly line chart with small alterations to colour scheme in figure 6.

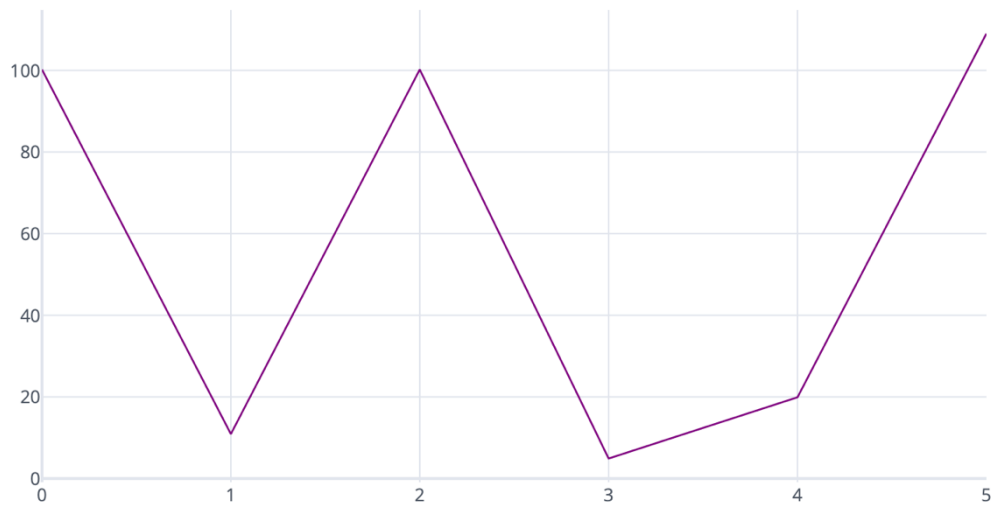


Figure 6. Plotly Line chart.

Bar charts are very common and can be shown as interactive chart. Point on a wanted bar with a mouse and more specific information can be seen on a chart.

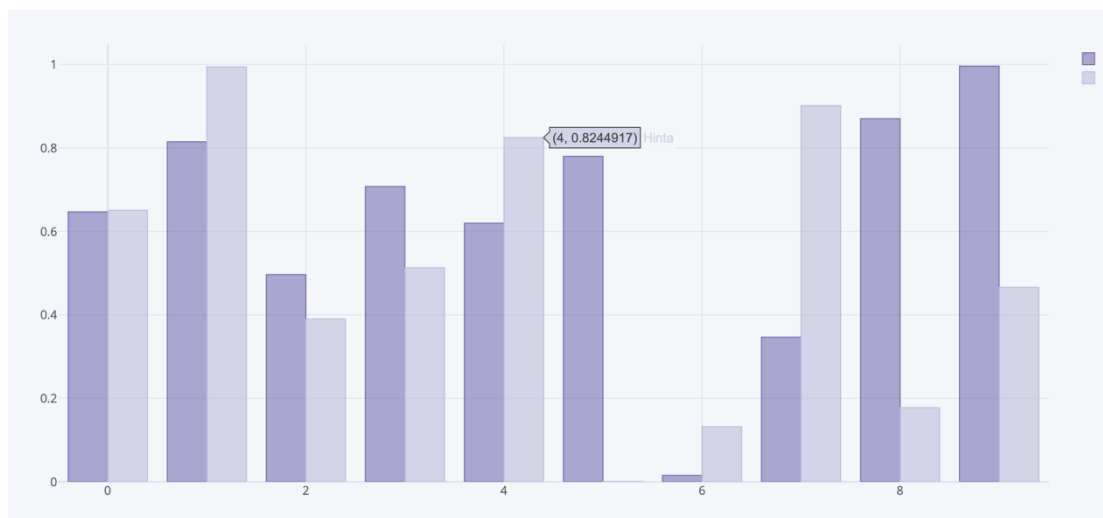


Figure 8. Interactive Bar chart

Pie chart was one of the requirements for the application and in figure 9. is an interactive Pie presented.

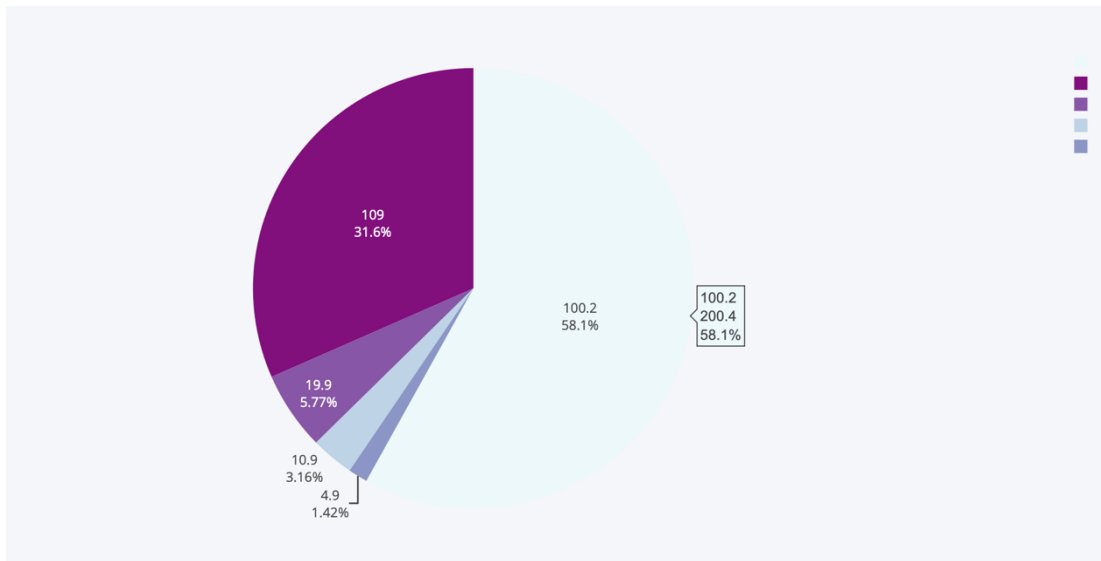


Figure 9. Interactive Pie Chart.

While searching through documentation for web applications called Dash was found. When studying about Dash, Django started to seem like a bit over kill at this point. Django app was already at the point where it was saving pictures, but Dash seemed lighter and suitable way to go.

After installing dash, it was faster to create visualisations. Here in figure 10. is an example code from Dash communal site, a quick start tutorial for creating a minimal app.

```

from dash import Dash, html, dcc, callback, Output, Input
import plotly.express as px
import pandas as pd

df = pd.read_csv('https://raw.githubusercontent.com/plotly/datasets/master/gapminder_unfiltered.csv')

app = Dash()

app.layout = [
    html.H1(children='Title of Dash App', style={'textAlign':'center'}),
    dcc.Dropdown(df.country.unique(), 'Canada', id='dropdown-selection'),
    dcc.Graph(id='graph-content')
]

@callback(
    Output('graph-content', 'figure'),
    Input('dropdown-selection', 'value')
)
def update_graph(value):
    dff = df[df.country==value]
    return px.line(dff, x='year', y='pop')

if __name__ == '__main__':
    app.run(debug=True)

```

Figure 10. A Minimal Dash App (plotly 2024)

10.5 Creating the app with Dash

Next step was to develop the app with Dash and it was created using besides Python and Dash as a framework, with CSS and HTML. Pandas was used to manipulate the raw data and to form data for the visualisations.

In figures 11. and 12. you can see few examples made with test data of interactive visualisations using Dash.

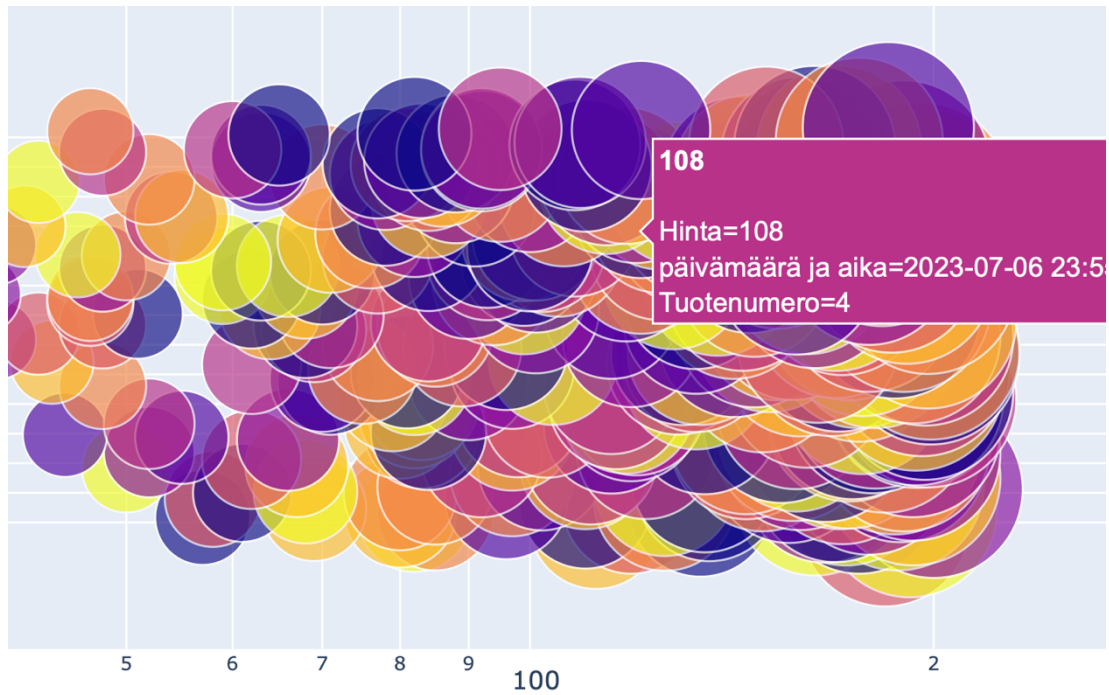


Figure 11. Interactive scatterplot with test data.

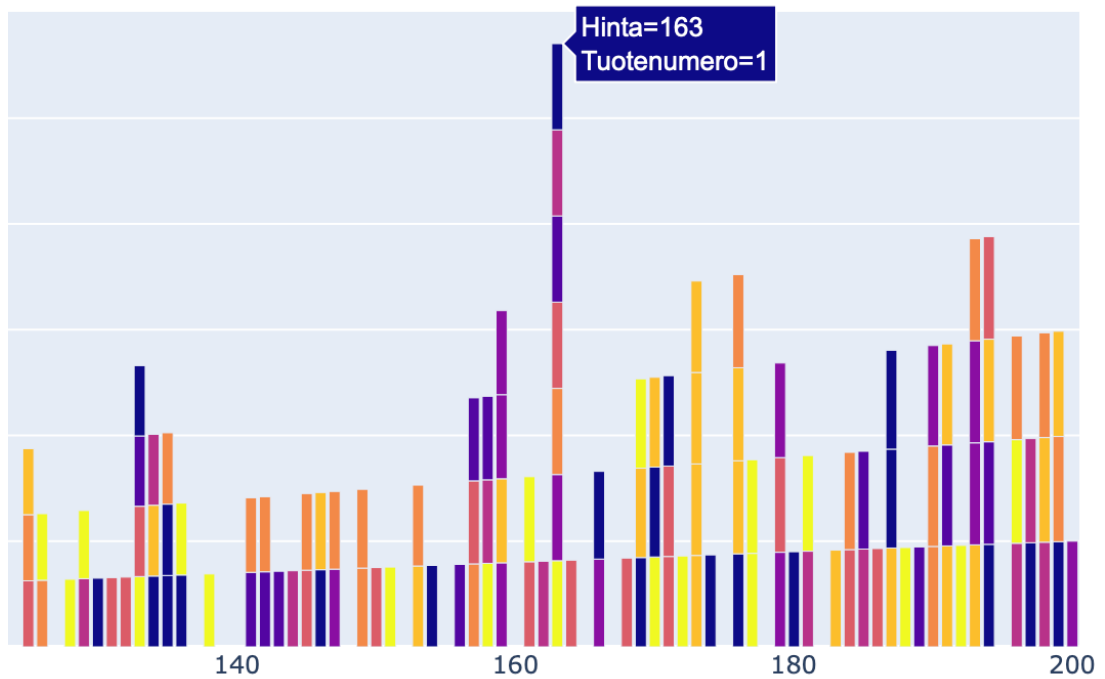


Figure 12. Interactive bar chart with test data.

It was easy to follow alterations while coding because Dash includes something called "hot-reloading" and it means that Dash will automatically reload your browser when you make changes to your code. This is of course voluntary to use and can be turned off. This feature is activated by default when you run your app with `app.run(debug=True)` (Plotly 2024).

10.5.1 Dash HTML

Components (`dash.html`) contains a component class for every HTML tag as well as keyword arguments for all the HTML arguments.

10.5.2 Dash Core Components module

The Dash Core Components module (`dash.dcc`) includes a component called Graph. Graph renders interactive data visualizations using the open source `plotly.js` JavaScript graphing library. `Plotly.js` supports over 35 chart types and renders charts in both vector-quality SVG and high-performance WebGL.

With dash, as mentioned above, it would be possible to add needed CSS and HTML without using separate files, but it was chosen to separate CSS to its own file for clarity and for future development.

10.5.3 React

React is JavaScript library for building user interfaces. It has been very popular over the last few years because it brings the power of reactive, declarative programming to the world of front-end development (Plotly 2024).

10.6 Streamlit framework, yet another possibility

Another framework that caught the eye was Streamlit framework. While looking into it, it seemed also very easy and fast to use. Because it was entirely

free, it caught the attention, but it was found out, it cannot scale as well, because of its restrictions which is understandable for free software. When moving from Django to Dash, Dash would still offer more possibilities that can be purchased when using it in commercial environment than Streamlit.

10.7 Connecting e-commerce platform and app via Api

APIs are mechanisms that enable two software components to communicate with each other and this communication is using a set of definitions and protocols. These define how the two communicate with each other using requests and responses. Probably the most familiar way to describe API architecture is in term of client and server. The one who is sending the request is called the client and one sending the response is called the server. There are different ways that APIs can work depending on why they were created (AWS 2024).

With this created visualization software being the client, it makes a request to server to fetch certain information. The server responds by delivering the requested information. For this to happen, there were also changes made to the main software, where the data is being fetched from.

Sales predictability was left out at this point due to the workload it would cause as the software was already behind from the schedule.

10.8 Cron

To fetch data automatically with certain sequence, the Linux based cron is needed to handle the automation on the server side.

Cron is a daemon that runs continuously in the background looking for events to be executed. Cron can wake up once a minute, examine configuration files and execute the commands if the time matches the listed time in the files. One critical point to remember about cron jobs is, there are two types of jobs:

System jobs and Users jobs and they run unsupervised so, if a program requires user input, you shouldn't call program in a cron job (Smith 2024, p. 398).

11 TESTING THE SOFTWARE

The process of software testing is a series of processes. Its purpose is to make sure, the computer code does what it is supposed to do and it doesn't do anything it is not supposed to. The software should be consistent as well as predictable so that it doesn't bring surprises to users (Myers 2011, p 2).

The testing of the software went on simultaneously when creating the software. While there isn't much functionality yet the user could use, the risks of the software behaving as it shouldn't isn't present yet but when in the future the development goes on, this must be considered carefully.

At the end testing it was realized this software needs yet another component. So far, the software worked as planned when it was tested manually. This software should also fetch information automatically without needing to start the program manually. This will be handled from the server side with Cron.

12 CONCLUSION

When starting with this thesis, it was not all clear to what kind of journey this thesis would lead to, even if there were planned steps to be taken from the start. It really was a process of learning more about software development and data-analytics as a whole and learning new things along the way. It served its purpose excellently.

The idea of this thesis was discovered in the summer of 2023 while in practical training and it rose from the working life need. Familiarizing with the subject began in the autumn 2023. Going through materials, software development models, learning about the software that could be used and about data-analysis. Looking into Pandas, Anaconda, Anaconda Navigator and Python, searching through different frameworks and tools gave a solid understanding to go from.

Answers to many practical questions while developing the software were found from discussion boards from Stack overflow and Geeks for Geeks site. While doing the research about Dash, their customer service representative kindly pointed out good places to find information while making inquiries about prices of non-commercial use. This software was developed with community edition Dash, so no money was needed during the development for this.

This project took about a year, and it was delayed from the schedule by several months. One of the reasons for the delay was to be too optimistic how fast the author of this thesis could develop the software and to find right components and tools to be used while working at the same time. Also, one of the planned stages of predicting the sales was left out, because it became clear this project was a big enough even without the prediction part and it would deserve a thesis of its own. It would also cause even more delays.

Testing the software was the moment it became clear the goals of this thesis were met, and it was enjoyable to see how the work paid off. The last

modifications were made after testing the data and realizing the need to improve it a bit further.

For the future development there are lots of possibilities: to add more functionality, to add prediction models, to alter or add more visualizations and to enhance appearances.

The software did what it was supposed to do – fetching data via API, altering it and visualizing the data.

REFERENCES

AmbleWebSol 2018. What is SDLC (software development cycle). Accessed and retrieved 9.6.2024.

<https://www.amplewebsol.com/what-is-sdlc-software-development-life-cycle>

AWS (Amazon Web Services) 2024. What is an API (Application Programming Interface)? Accessed and retrieved 21.7.2024.

<https://aws.amazon.com/>

Anaconda. Anaconda Distribution. Retrieved 24.3.2024.

<https://www.anaconda.com>

Anaconda Inc. Anaconda Navigator. The Desktop Portal To Data Science. Accessed and retrieved 9.6.2024.

<https://docs.anaconda.com/free/navigator/>

Anaconda. Our History. Retrieved at 24.3.2024.

<https://www.anaconda.com>

Arnowitz, J., Arent, M. & Berger, N. (2007). Effective prototyping for software makers. San Francisco, Calif: London: Morgan Kaufmann; Elsevier Science(distributor)

GeegsForGeeks 2024. What is Spiral Model in Software Engineering? Accessed and retrieved 9.6.2024

<https://www.geeksforgeeks.org/software-engineering-spiral-model/?ref=lbp>

git. –fast-version-control. Retrieved at 24.3.2024.

<https://git-scm.com>

GitHub. About GitHub and Git. Retrieved 24.3.2024.

<https://docs.github.com/en/get-started/start-your-journey/about-github-and-git>

Djangofoundation: Why Django? retrieved 24.3.2024.

<https://www.djangoproject.com/start/overview/>

Hummert, C., & Pawlaszczyk, D. (2022). Mobile forensics -- the file format handbook: Common file formats and file systems used in mobile devices. Springer International Publishing AG.

Jupyter. JupyterLab: A Next-Generation Notebook Interface. Accessed and retrieved 9.6.2024.

<https://jupyter.org>

Kate Brush, Valerie Silverthorne. TechTarget. Agile software development. Accessed and retrieved 9.6.2024

<https://www.techtarget.com/searchsoftwarequality/definition/agile-software-development>

Kemp Peter 2010. Waterfall model. Accessed 9.6.2024 and retrieved from https://en.wikipedia.org/wiki/Winston_W._Royce

Larsen, Rob. Beginning HTML and CSS, John Wiley & Sons, Incorporated, 2013. Accessed and retrieved 6.7.2024.

<http://ebookcentral.proquest.com/lib/samk/detail.action?docID=1138437>.

Liebowitz, J. (Ed.). (2020). *Data analytics and ai*. Auerbach Publishers, Incorporated.

Myers, G. J., Sandler, C., & Badgett, T. (2011). *The art of software testing*. John Wiley & Sons, Incorporated.

Narayan Praneal 2023. What is the Snowflake Data Platform? Snowflake Inc 2024. Accessed and retrieved 14.7.2024

<https://www.snaplogic.com/blog/snowflake-data-platform>

Plotly: Getting started with Plotly. Retrieved 7.7.24.

<https://plotly.com/python/getting-started/>

Plotly 2024. Accessed and retrieved 14.7.2024

<https://dash.plotly.com>

Plotly 2024. Dash Layout Accessed and retrieved 16.7.2024.

<https://dash.plotly.com/layout>

Plotly 2024. A Minimal Dash App. Accessed and retrieved 14.7.2024.

<https://dash.plotly.com/minimal-app>

Plotly 2024. React for Python Developers: a primer. Accessed and retrieves 16.7.2024.

<https://dash.plotly.com/react-for-python-developers>

PyPi 2024. pandas: powerful Python data analysis toolkit. Accessed and retrieved 9.6.2024.

<https://pypi.org/project/pandas/>

Python Institute. Python® – the language of today and tomorrow. Retrieved at 24.3.2024.

<https://pythoninstitute.org>

Runeson, P., Host, M., Rainer, A., & Regnell, B. (2012). *Case study research in software engineering : Guidelines and examples*. John Wiley & Sons, Incorporated.

Shaw, B., Badhwar, S., Bird, A., Bharath, C. K. S., & Guest, C. (2021). *Web development with Django: Learn to build modern web applications with a Python-based framework* (First published 2021.). Packt Publishing Ltd.

Shirowzhan, S. (2022). *Data Science, Data Visualization, and Digital Twins*. IntechOpen. Johanna Schmidt, Chapter 6, Visual Data science.

Smith, R. W. (2004). *Linux+ study guide : Exam: xk0-001*. John Wiley & Sons, Incorporated.

Snowflake Inc 2024. Accessed and retrieved 14.7.2024

<https://streamlit.io>

Tonder, M. (2013). *Ideasta kaupalliseksi palveluksi: Matkailupalvelujen tuot-
teistami- Teknologia ja etiikka sosiaali- ja terveysalan hoidossa ja hoivassa*.
Helsinki: Restamark.

Visual Studio Code. Getting Started. Retrieved at 24.3.2024.

<https://code.visualstudio.com/docs>

w3schools: Django introduction. retrieved 24.3.2024.

https://www.w3schools.com/django/django_intro.php

Wikipedia 2024. Comma-separated values. Accessed and retrieved 9.6.2024.

https://en.wikipedia.org/wiki/Comma-separated_values