

# 02\_report

October 22, 2025

## 0.0.1 GitHub Repository:

<https://github.com/fheflo1/IND320>

## 0.0.2 Streamlit App:

<https://ind320-fheflo1.streamlit.app/>

## 0.1 AI Usage and Reflections

AI is used for all parts of the project. Setting up the resources correctly and guiding me through my issues. I had around 37% code completions from Copilot. I believe most of this code completions were because I requested modifications to enhance the plots and general code it suggested. Translating my Norwegian comments to English and adding more was also something useful it helped with. Two things AI really help me with was accelerating my coding efficiency and optimizing the code. After coding the essentials for my app, it was slow, and I could then reason with AI to improve and optimize, which was very helpful.

## 0.2 Compulsory Work

When presented with the task of connecting to both Cassandra and MongoDB, I began thinking of structures I could use and figured I would use something I'm familiar with. I have used a structure of bronze, silver and gold notebooks. Instead of doing everything in one notebook, connecting to Cassandra, retrieving data from Elhub, then process it and place it in a MongoDB server. The structure is like this: - **Bronze**: Connects to the Cassandra cluster, then retrieves the data from Elhub and places it in the cluster as a 'bronze table', in this case called 'production\_raw'. - **Silver**: Retrieves the bronze table and does the preprocessing steps and saves it as a new 'silver table'. In this case only selecting the wanted columns. - **Gold**: Used to silver table to analyze with plots and visualizations.

The first notebook, 01\_cassandra\_setup, was initially used to test the connection and is not important in this assignment, since everything necessary regarding this connection is also done in the bronze notebook.

The fifth notebook, 05\_db\_connections, inserts the silver table into MongoDB, and the data is after this stage ready to be used in the Streamlit app.

Connecting the Streamlit app to MongoDB was easy. Setting up the page with all the requirements was not too much trouble either. The problems began to show itself when using the app, as it was laggy and slow every time I changed the graph between months. I could not specifically remember what was said about cache from the lectures, but with some help of Google and AI, I added

cache\_resources for the server connections, and cache\_data for loading the data from the server every tenth minute. The app performance was increased drastically. I did not cache how effecting this was in part 1 on the projects since we only worked with a small csv file. I changed therefore also old code from the first assignment to include this cache\_data.

Even though I'm very fond of Streamlit, am I missing some options for the esthetics which's bugs me. On the app page, have I added a widget button for every page so that the user can navigate to an interesting page from the opening of the app. I had AI help me customize these buttons using CSS, which was a hassle to get right. When it was finally looking good, I understood that if the window is not in full screen, the buttons overlayed. Streamlit is good, but I can't see how to make it esthetically pleasing and bug-free.