Above the Surface

Organizing the digitally processed archaeological finds of the north/southline

Abhilash M. Abhilash   Camilla Santos Andersen   Chris Atherton   Radu David   Danny de Vries
12345678   12345678   12345678   12345678   14495643
University of Amsterdam
Amsterdam, The Netherlands

ABSTRACT

1 INTRODUCTION

From 2003 to 2012 excavations took place for the creation of the North/South metro line in Amsterdam. At Damrak and Rokin, which are unlikely archaeological sites due to being in the city center, archaeologists had a chance to physically access the riverbed. During these excavations in the Amstel over 700,000 objects were preserved which resulted in the archaeological collection called 'Below the Surface' [1] commissioned by the Municipality of Amsterdam.

The collection has a great variety of objects, from tools over centuries old to credit cards recently lost which makes the collection a rare source of urban history. All objects are digitally processed (e.g. photographed, labeled, metadata added) and displayed on a front-end website at *belowthesurface.amsterdam*. This website shows an overview of all the objects and a detail page with metadata of a particular object but no further categorization or classifications. This research aims to further organize this collection of objects with a focus on grouping the items by *functional properties*, determining *cultural relevance*, and researching *object relationships*.

2 RELATED WORK - CHRIS

2.1 Below the Surface - Danny

2.2 Museum research - Desk Research - Chris

2.3 Antique collectors - Chris

2.4 Archaeologists - Chris

2.5 Academic Research - Chris

2.6 Machine Learning - Radu

3 METHODOLOGY - ABHILASH

3.1 Glushko 6 Questions - Abhilash and Chris

3.2 Data Dictionary and Metadata - Chris

3.3 Ontology - Chris

3.4 Data Model - Abhilash

3.5 Technical Implementation of the website

The website is a custom front-end mostly using web standards and open-source software and libraries. The assumption is that it will mainly be used in a desktop environment by the users to explore on larger screens so the website is not fully responsive and thus not mobile-optimized.

3.5.1 Front-end frameworks. The web application is created with the open-source front-end framework Svelte [2] and UI framework SvelteKit which allows the application to be built-in interface components, each chart is rendered separately making it more efficient to add functionality (e.g. add datasets, render different chart types) in the future but also makes the website performant when more data and charts are added. Svelte can be downloaded as a module (package) from NPM [3] and uses the JavaScript back-end run-time Node.js [4]. For the charts the JavaScript charting library Chart.js [5] is integrated into the components which allows charts to be rendered in HTML5 Canvas without much configuration.

3.5.2 Dataloading. The processed and transformed dataset was used as a primary data source of which subsets of the dataset in *.csv* format, roughly one per year chart and section, are converted to *.json*. These data files are loaded on page load of the browser. For this prototype, no back-end was set up and no database queries are being made. Any filter options and updating of the charts are more custom, it uses JavaScript utility functions to allow the data to be pre-processed and have only the data change. For the map page, an additional Chart.js Geo Plugin is used to render the Chloropleth map. It uses a TopoJSON [6] file to render the regions of the World.

3.6 Machine Learning Model

3.6.1 Data-set presentation. Bellow the surface provides a data-set[?] of all the objects resulting from the excavations. The data is provided in the form of a .csv file, with 139190 rows and 163 columns. Each row corresponds to an object. Describing each object is well outside of the scope of the purposes of this section, however, an explanation of the relevant columns is necessary.
The following columns are relevant for the purposes of the ML model:

- vondstnummer - represents a unique inventory number, in the form of a string. Every object has a vondstnummer. Example: "NZC1.00001MTL001".
- object - a description of the contents of the object. Example: "sieve residue"
- subcategorie - a categorisation of the object material. Example: "metal: copper alloy"
- objectdeel - describes the object type morphologically (if it is part of a bigger object, a set, etc). Example: "fragment"
- vlak_min - Describes the minimum depth at which the object might have been found. Example: "-22.0"

---

[1] https://belowthesurface.amsterdam/en

[2] https://svelte.dev
[3] https://www.npmjs.com
[4] https://nodejs.org/
[5] https://www.chartjs.org
[6] https://github.com/topojson/topojson

- vlak_max - Describes the maximum depth at which the object might have been found. Example: "-22.01"
- begin_dat - The beginning of the interval of the estimated year of the object. Example: "1675.0"
- eind_dat - End of the interval of the estimated year of the object. Example: "1725.0"
- niveau1 - The category in which the object is placed. Example: "Communication & Exchange"

For the columns *object, subcategorie, objectdeel, vlak_min, vlak_max, begin_dat, eind_dat, niveau1* there are rows in the dataset in which one or more of these columns are blank.
The column *niveau1* can take the value of one of 12 pre-determined categories, as well as the value "Not classified". As previously mentioned, there are rows where this column is blank.

3.6.2 Objectives. Our objective is to create a machine-learning model that will complete the missing data for the "niveau1" column. This means that our model will predict a value in the niveau1 column, for the rows where currently that column is blank or has the value "Not classified". The prediction will be based on the values in the *object, subcategorie, objectdeel, vlak_min, vlak_max, begin_dat, eind_dat, niveau1* columns, which will act like input to the machine-learning model.

3.6.3 Deliverables. In order to achieve our objectives, the following files are delivered:

- process_dataset.py - a simple python script that takes the original 163 column .csv files and consolidates it into another .csv files that only contains the columns of interest. The name of this .csv file is "selected_dataset.csv"
- machine_learning.py - this python script is the backbone of the machine-learning process. It is a more-complex script that does the following steps:
  – Loads the "selected_dataset.csv" dataset
  – Preprocess the data (completes the values with 0 or placeholders here they are blank", etc)
  – Converts text stings to vectors
  – Splits the data into unlabeled and labelled data based on the values in the "niveau1" column
  – Splits the labelled data using a training, testing, and validation split
  – Builds the ML model
  – Compiles the model
  – Trains the model
  – Tests the model
  – Predicts the values of niveau1 for the unlabelled data
  – Saves the updated dataset into a file named predicted_dataset.csv
- predicted_dataset.csv - a file containing the dataset in selected_dataset.csv but with the column selected_dataset.csv fully completed.

3.6.4 Model description.

4 RESULTS - ABHILASH
4.1 Website - Camilla and Danny
The original website has a lot of individual objects and detailed metadata about properties. The aim of the website was mainly to

summarise the collection to allow the before-mentioned users to explore the broad dataset and find interesting patterns.

The design is based on the branding of the original below the surface projects and follows standard information and visual design principles.

4.1.1 Overview summary page. Create a storytelling kind of page with 5 sections. HIER ENUMERATE ALLE SECTIES MET CHART TYPES

4.1.2 Collection detail subset. Based on the categories of the summary page a user can create a subset of the collection.

4.2 Machine Learning Model - Radu
4.3 Dataset - Abhilash
4.4 Ontology and Data Dictionary - Chris
5 CONCLUSION - CAMILLA
6 DISCUSSION - CAMILLA
6.1 Reflection
7 FUTURE WORK - CAMILLA
8 CONTRIBUTION OF AUTHORS
- *Abhilash:* did stuff
- *Camilla:* did stuff
- *Chris:* did stuff
- *Radu:* did stuff
- *Danny:* programmed the website and visualizations (front-end development) as well as conceptualized the interface interactions (interaction design) and look and feel (visual design).

9 ACKNOWLEDGEMENTS

10 APPENDIX
10.1 Hosted source code
The source code of the web-based visualization, Python notebooks of the machine learning model and datasets used are hosted on GitHub using the MIT License. Under the *uvaio* username we have several code repositories:

(1) Notebooks: Source Code for the Jupyter Notebooks for data processing and machine learning. https://github.com/uvaio/notebooks
(2) Website: Source Code for the custom front-end website and interface. https://github.com/uvaio/website
(3) Datasets: The processed and modeled datasets and the ontology. https://github.com/uvaio/datasets

10.2 Live version

A live demo version of the front-end website and visualization (desktop only) is hosted on Netlify and can be viewed using the following link https://uvaio.netlify.app

10.3 Website screenshots

Shown in Figures 1 and 2.

REFERENCES

[1]]DatasetBelow [n. d.]. Bellow the surface dataset. https://statics.belowthesurface.amsterdam/downloadbare-datasets/Downloadtabel_EN.csv. Accessed: 2023-10-08.
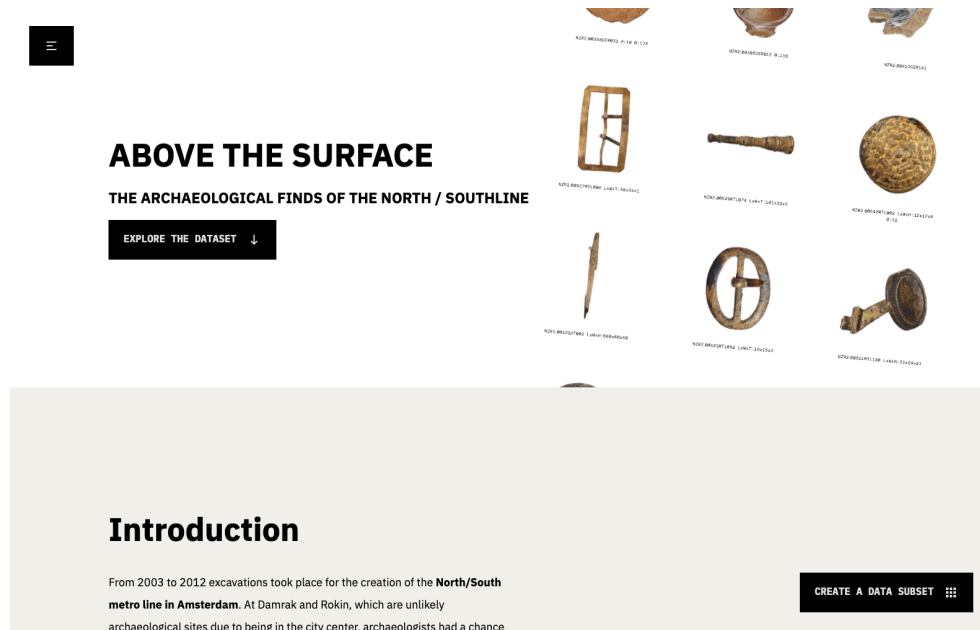
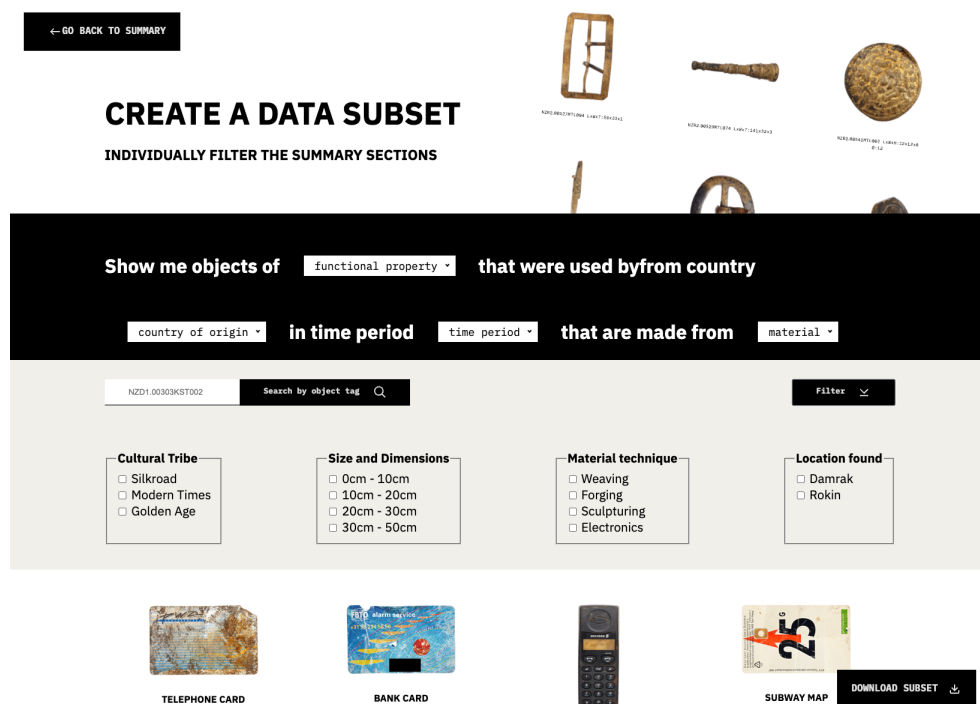**Figure 1: Screenshot of the 'overview landing page' of the website**

Figure 2: Screenshot of the detail 'create a subset' page of the website