Big Data Analysis
Final Project Report

Michał Gozdera, Małgorzata Hadasz, Paweł Koźmiński

January 4th, 2023

1 Introduction

Our project aims to analyze Nextbike data, namely the information about rentals of bikes in different cities. We confront this data with weather and cities population. We use Nextbike API to gather the up-to-date data and historical Nextbike dataset from Kaggle. The short description of the files can be seen in Table 1.

2 Solution architecture

We needed to use cloud solutions to handle large volumes of data, storage, and retrieval. We utilized Google Cloud Platform (GCP). The architecture of our solution is depicted in the Figure 1. To gather the data, we use 4 data sources:

- the historical nextbike data from Kaggle (`https://www.kaggle.com/datasets/pankrzysiu/nextbike-api-history`)
- data retrieved from Nextbike API, queried every 5 minutes (`https://api.nextbike.net/maps/nextbike-live.json`)
- data about weather collected via the Python package meteostat
- data about the populations of cities collected via API (`https://countriesnow.space/api/v0.1/countries/population/cities`)

To download the data from Nextbike API every 5 minutes, we use Cloud Function `get_nextbike_data`, implemented in Python. It is a serverless solution that allows defining and running custom code. This function sends an HTTP query to Nextbike API and saves the results. For storage, we use the MongoDB database. It is well suited for our solution since it can store JSON-like objects and manage them efficiently. Cloud Scheduler is created to run `get_nextbike_data` every 5 minutes. To keep the best security practices, we use a Service Account on behalf of which the Cloud Scheduler runs the function.

File name	Group	Description
data_exploration.ipynb	Initial exploration	The preliminary analysis of the historical data
nextbike_api_exploration.ipynb		The preliminary analysis of the data from API
current_analysis.ipynb	Analytic files	Analysis of the data collected for the last few days
bikes_weather_analysis.ipynb		Comparative analysis of weather and bike usage
historical_analysis.ipynb		Analysis of the trends in historical data
population_analysis.ipynb		Comparison of number of rentals with population of the city
animations.py		The script for the dash application
cities_population.ipynb	Data manipulation	Collecting data about populations
API/get_data_from_api.py		Script used for Nextbike API data acquisition
API/transform_api_data.py		Script for API data transformation
API/transform_historical_data.py		Script for historical data transformation
API/utils.py	Technical files	Common functions
requirements.txt		Project requirements

Table 1: Files description

The initial exploration notebooks contain information about the data structure and possible inconsistencies. This process uncovered some unexpected situation that must have been addressed in the following data processing, such as possibility of non-sense coordinates values appearance. Moreover, those notebooks examine the data structure and present initial possible analysis.

One of the project challenges was to transform the data from the initial shape to one useful for the analysis. As we mostly wanted to focus on the statistics of rentals in time, both historical and live data, focused on the current station status and bike rides, respectively, must have been transformed into a convenient form. This task was realized with Python scripts: `API/transform_api_data.py` and `API/transform_historical_data.py`. Data in the final shape could be utilized in the analytic part of solution.

To efficiently analyze the weather, populations and historical Kaggle data, we need to structure it. After the initial preprocessing, they have a well-defined structure, so storing them in an SQL database is a reasonable option. We use PostgreSQL. Weather, population and Kaggle data are downloaded and processed only once, in contrary to Nextbike API data (being updated regularly). We implement this step with pandas Python library.

The analysis part of the project is performed with two main components:

- station occupancy Dash app, showing live number of available bikes on a given station,
- data analytic Jupyter notebooks including: analyzing the dependence of cities population on the number of rentals, analysis of number of collected bikes throughout the day ... .These investigations are based both on historical Kaggle data and up to date Nextbike API responses.

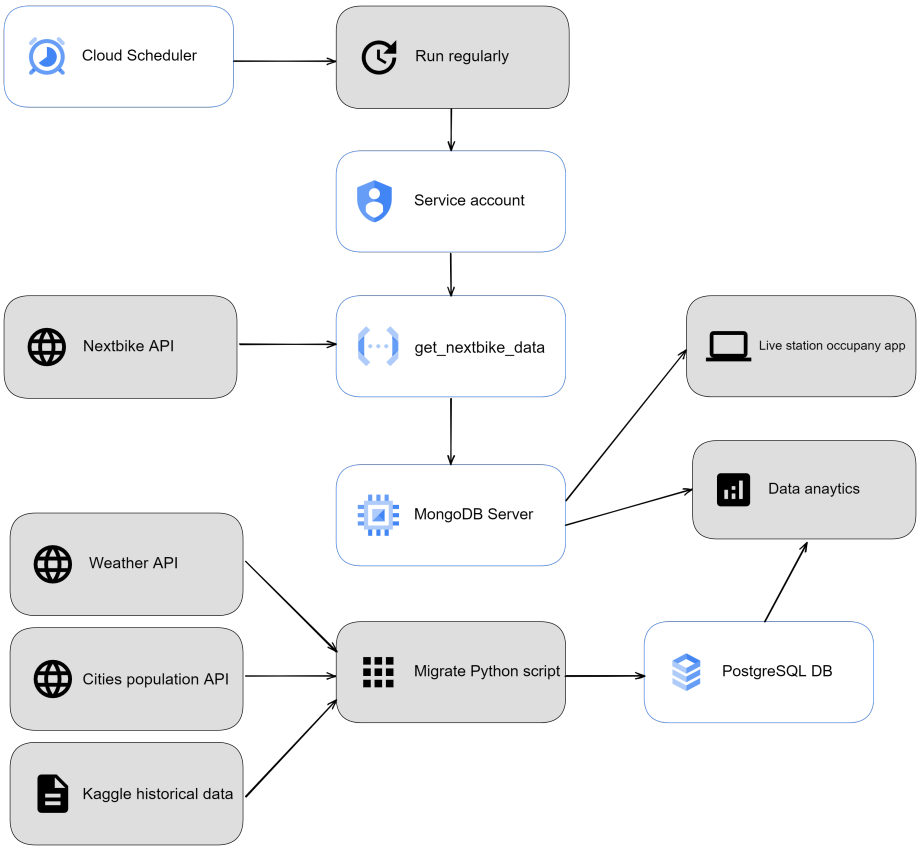

Figure 1: System schema

3 Databases

To make the verification and grading of the project possible, we made the access to our databases public. In the real world scenario, we would specify concrete IP address that can connect to the databases.

3.1 MongoDB

To connect to MongoDB use: 34.118.42.39 as the host and 27017 as port.

The data volume stored in MongoDB reached 5.4 GB, stored in one collection `bikes`.

3.2 PostgreSQL

To connect to PostgreSQL use the following properties: dbname='bda-2022-nextbike-data' user='postgres' host='34.118.39.208' password='bda-2022-postgres'

port=5432.

The data, stored in 3 PostgreSQL tables, are of size roughly about 1.5 GB.

4 Analysis

This section focuses on the introduction of the project analytic part. The main conclusions are described in the particular notebooks, here only the idea behind the problems is described.

4.1 Stations occupancy

The first thing, we decided to analyze is station occupancy. We wanted to know how many bikes are currently available. This information is useful, for example, to decide, where to go to rent a bike. Moreover, if one observes the data throughout the day, one can see how the occupation changes. To better visualize the stations, we decided to present our results on the map. The size of the bubble indicates the station occupancy. This visualization was done in the **Dash** framework. The view is updated every 30 seconds. The exemplary view can be see on Figure 2 .

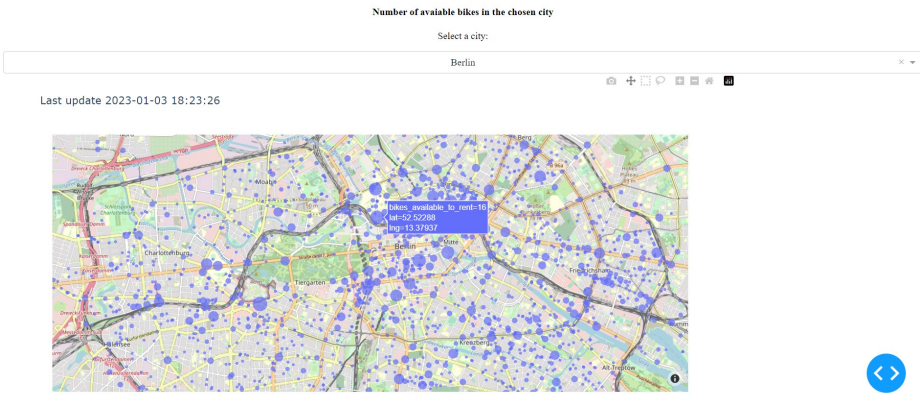

Figure 2: Dash visualization

To start the dash, one needs to go to the folder where the file `animation.py` is located and command line write `python animation.py`. The analysis is possible to see on the address displayed in the console.

The necessary libraries are included in the `requirements.txt` file.

4.2 Data analysis throughout the day

The next analysis is an investigation, of how the number of collected bikes changes throughout the day. This analysis was done for both historical and

current data, for two cities: *Bonn* and *Las Palmas de Gran Canaria*. For the historical one, the division into months was also performed. Therefore, we can see if the month influences trends. Due to analyzing data from two different cities, we can see if the trends differ between hot, tourist cities and colder and less tourist ones.

4.3 Impact of the weather on data sharing system usage

Another task was to take a look at the impact of the meteorological data on number of bikes rented. This analysis was performed comparatively for two cities: *Bonn* and *Leipzig*, once using the historical dataset, and another one for the collected data from Nextbike API to examine if the conclusions are still valid.

4.4 Analysis of data sharing system popularity considering the city population

In order to involve the data about cities inhabitants, we also compared the demand on bikes in 30 cities with the highest number of rentals. This approach leads to interesting conclusions about the cities where the system is used enormously often and where the city potential is still not fulfilled.