A02 – REPORT ON ADDITIONAL TRANSPORTATION DATASET.

Rajbir Bhattacharjee

R00195734

Table of Contents

Exercise 6

Choice of Dataset

The dataset used in this case is the DublinBikes DCC dataset.[1] In addition, this will be combined with the met-Eirann dataset[2] because biking demand varies with weather, and people prefer not to bike when the weather is bad.

The DublinBikes dataset has the following structure:

[1] URL for Dublin Bike dataset: https://data.gov.ie/dataset/dublinbikes-api?package_type=dataset

[2] URL for Met Eirann weather data: https://www.met.ie//climate/available-data/historical-data

Column	Type	Label	Description
STATION ID	numeric	STATION ID	Globally unique identifier of station.
TIME	timestamp	TIME	Time of fetching the data.
LAST UPDATED	timestamp	LAST UPDATED	Time of last updated information.
NAME	text	NAME	Station name.
BIKE STANDS	numeric	BIKE STANDS	Station total number of bike stands.
AVAILABLE BIKE STANDS	numeric	AVAILABLE BIKE STANDS	Station available bike stands.
AVAILABLE BIKES	numeric	AVAILABLE BIKES	Station available bikes.
STATUS	text	STATUS	Station status (Open/Close).
ADDRESS	text	ADDRESS	Station address.
LATITUDE	numeric	LATITUDE	Station latitude.
LONGITUDE	numeric	LONGITUDE	Station longitude.

Figure 1 screenshot from https://data.gov.ie/dataset/dublinbikes-api/resource/76fdda3d-d8be-441b-92dd-0ee36d9c5316?inner_span=True

This dataset is mostly in the form of csv files, one for each quarter. The files are very large, each around 300MB in size and each having about 3 million rows. The rows are arranged as such:

1. First they are ordered by the day when the readings are gathered
2. Then they are ordered by the station from which they are gathered

There is also a REST API that can be used to query the data instead of downloading large CSV files.[3] The API can fetch the last snapshot or the historical data. The details of querying these are pasted below from the website. For example, the historical data for 2021-04-13 17:00:00 can be fetched by the following URL:

https://data.smartdublin.ie/dublinbikes-api/historical/?init=2021-04-13%2017%3A00%3A00

or with the following Curl command:

```
curl -X 'GET' \
  'https://data.smartdublin.ie/dublinbikes-api/historical/?init=2021-04-13%2017%3A00%3A00' \
  -H 'accept: application/json'
```

[3] Dublin Bike DCC REST API: https://data.smartdublin.ie/dublinbikes-api

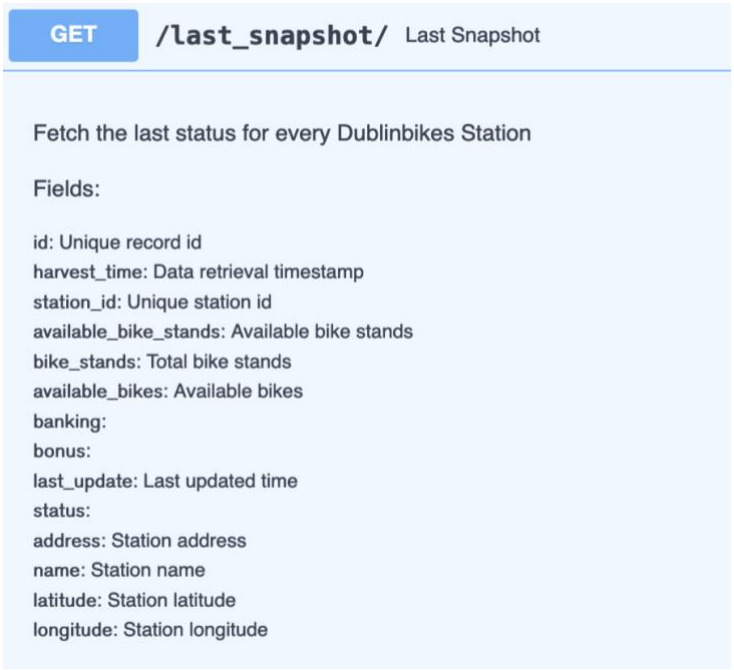

Figure 2 Screenshot of API from https://data.smartdublin.ie/dublinbikes-api

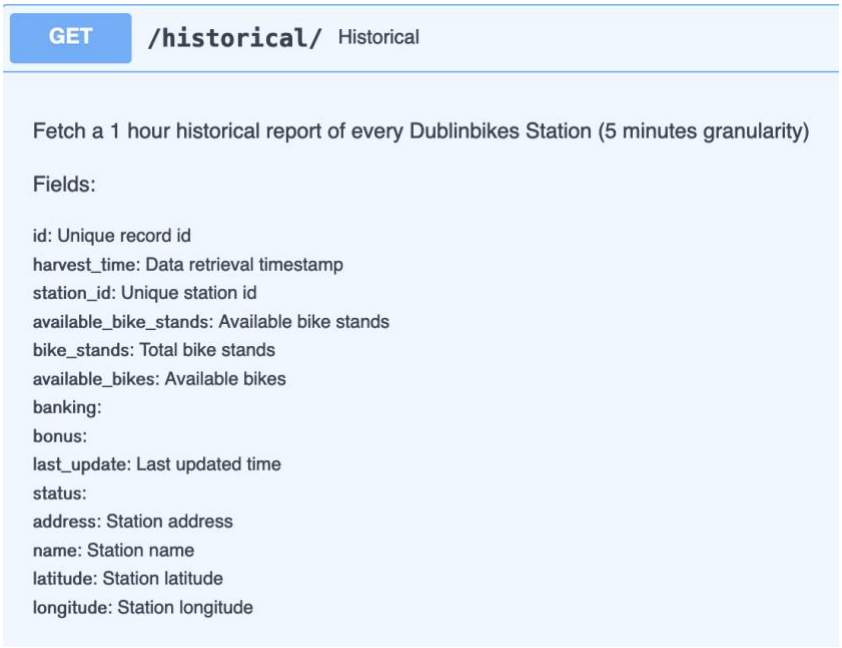

Figure 3 Screenshot of API from https://data.smartdublin.ie/dublinbikes-api

The Met Eirann dataset has the following fields:

```
date:  -  Date and Time (utc)
rain:  -  Precipitation Amount (mm)
temp:  -  Air Temperature (C)
wetb:  -  Wet Bulb Temperature (C)
dewpt: -  Dew Point Temperature (C)
vappr: -  Vapour Pressure (hPa)
rhum:  -  Relative Humidity (%)
msl:   -  Mean Sea Level Pressure (hPa)
ind:   -  Indicator
```

Task

The task here would be to predict the number of available bikes at any station.

Usefulness

The dataset here does not give any data about how many bikes were actually rented out at any station in a given hour. Hence, what we will use is the number of bikes at any station at a given hour. The idea is that if a station has no bikes, then there more demand at that station than there is supply, and bikes from other stations must be transported to that station.[4]

Being able to predict that some bicycles will be out of bikes soon will help make better decisions and streamline logistics for bicycle transport between stations.

Planning the transport once predictions are available

[4] One improvement to the dataset in further collections could be to also have a cyclist-in and cyclist-out data associated with the same dataset. This data is available in another dataset, the Cycle Counter Dublin Dataset [4], however, the stations in these two datasets are different, so the data cannot be joined. This could be a suggestion to the transport bodies that collect this data. https://data.gov.ie/dataset/dublin-city-centre-cycle-counts/resource/ce0b8fa0-58ad-4b75-941c-93be56b68b96?inner_span=True

We will first discuss how predictions can be used, before describing how we will go about predicting. Once a prediction about the number of bikes available at each station is available, for a future hour in the day, this can be used to optimally transfer bikes.

If there are three stations, A, B and C, and A and B have 5 and 6 bikes each, and C has 0 bikes, then bikes could be transferred from A to C or B to C, or a both from A and B to C. Each of them would fulfil C's demand, but will have a different cost based on based on fuel required and time taken to transport.

Linear programming, or other optimization methods can be used to find an optimal bike transfer strategy to reduce time, costs and fuel.

Preparing the Data

To prepare the data Spark SQL can be used. The following steps must be performed

1. Join the weather data with the DublinBikes data
    a. The weather data is per weather station. We only consider weather stations that are within a 10 kilometres of Dublin city centre.
    b. The weather data is on an hourly basis, so nothing more needs to be done there.
    c. Next we select all bike stations so that we can assign each bike station to its closest weather. This can be done by discarding other fields and only having the latitude and longitude of the bike stations.
    d. The assignment will be done by a full join, followed by a grouping to and aggregated by the minimum haversine distance.
    e. Now that we have all the bike stations assigned to a weather station in a separate table, we perform a 3-way join between the full bike data, the weather reading and the new table. This will prevent the memory from exploding, as a full join will not be necessary and we can only do an inner join.
    f. This step is optional, what might help is getting weather readings for an hour before and an hour earlier. To achieve this the following modifications to the above steps must be done:
        i. The inner join should be modified so that readings within an hour's difference will also be included. This would mean that for every bike reading, there will be multiple rows with weather readings from the closest weather station.
        ii. This can be aggregated by the timestamp of the bike reading, and the bike station id where the weather readings can be converted to a list.
        iii. Finally the list can be converted to columns, so that each bike reading has three columns for weather – one for one hour earlier, one for the current weather, and one for one hour later.
    g. The bike station data is gathered every 5 minutes. We can either take this same window and predict for it, or we can predict for every 1 hour window. If the latter is chosen, then we will have to combine the entries for every bike station by the hour.

This can be done by aggregating by the hour and bike station, and then taking the minimum number of bikes present in the station at any point of time.

2. Prepare the data
    a. The rainfall should be converted to categorical numbers
    b. Same must be done for wind
    c. The same must be done for temperature
    d. Demands are periodic in nature, for example demand on weekdays may follow a certain pattern and on weekdays a different pattern, and similarly for different hours of the day. To capture this the following must be done:
        i. Timestamp must be converted to day of year, and the sine or cosine of the value taken
        ii. Time stamp must be converted to day of week and the sine or cosine of the value taken
        iii. Timestamp must be converted to hour of day and the sine and cosine must be taken
        iv. The three above columns will be added to the dataset
3. Now using the above enriched data we have the following columns added:
    a. Weather data for one hour prior
    b. Weather data for one hour post
    c. Current weather data
    d. Timestamps expressed as cosines or sines of day of year, day of week, hour of day
4. Following this Spark MLlib will be used to train a model that predicts the Available Bikes at each station. A variety of classifiers should be used and initial performance estimated. The best performing models must be fine-tuned for the best hyperparameters.
5. Once we have the trained model, at serving time it needs to predict the number of bikes. To do this, it needs the current state of the bike station, the number of bikes etc. It also needs the weather readings for the current time and the last hour, as well as the prediction for the next hour. The same conversions must be done on this data as was done to the training data, and then this can be used for prediction. Spark Streaming can be used at serving time.

Difficulty

Again, the it is not necessarily more difficult than other problems we have done before, but there are more steps involved, and also there is an additional task of combining two different datasets. All of this can be effectively performed in Spark, but may be time consuming because there are more steps involved.