User's Manual & Installation Guide

Streaming Uber Taxi Demand Prediction

Liz Aharonian & Ori Ben-Zaken
supervisors: Barak Bar-Orion, Yohana Khoury, Yoav Einav, Tal Doron

June 2019

USER'S MANUAL & INSTALLATION GUIDE

TABLE OF CONTENTS

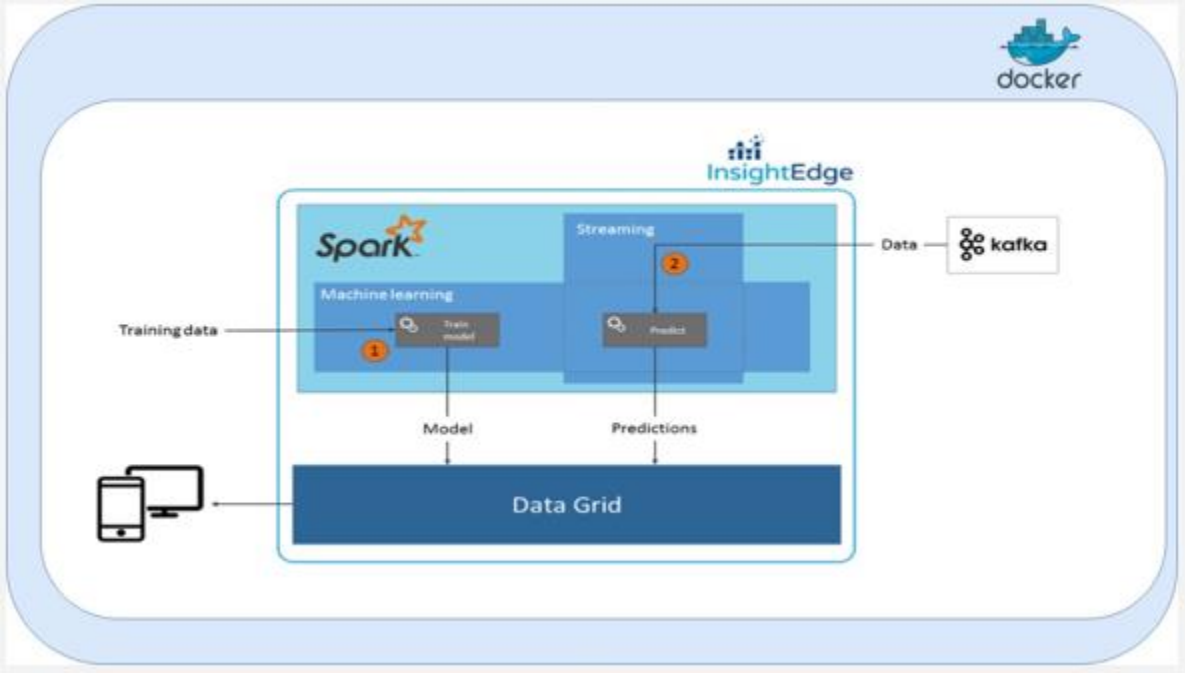
1.0 GENERAL INFORMATION

1.1 System Overview

Our goal is to predict demand per time interval for Uber Taxi Ride, using ML and big data processing tools.
The demand parameter is calculated according to time intervals and area, this gives two major advantages to our predicting model:

1. Help the model's users to know which area they should give service.
2. Using a simple function based on the demand to predict and calculate the fare amount.

Our architecture is presented in the next schema, details about installation and the way to run the different components are detailed in the next pages.



2.0 INSTALLATION & CONFIGURATION

Important remark: most of the project runs at Linux environment. In order to make it run in any desired environment and OS, we wrapped it using docker.
However, in order to run Tableau BI tool, it is required to have windows environment.
Follow the install and running instructions taking into consideration the running environment.

Prerequisites:

1. **Java Environment**
2. **Python 3**

2.1 Install InsightEdge [windows]

Download I9E[1] installation zip using the next link:

https://gigaspaces-releases-eu.s3.amazonaws.com/insightedge/14.5.0/gigaspaces-insightedge-enterprise-14.5.0-m7.zip

Get into <XAP_HOME>/ xap-license.txt, change the text inside it to: "tryme".

2.2 Install Docker

Install Docker and Docker-Compose from Docker official website:

Install Docker:

https://docs.docker.com/v17.12/install/

Install Docker Compose:
https://docs.docker.com/compose/install/

2.3 Install Tableaue [windows]

Install Tableaue BI application fron Tableaue's website.

Get into <XAP_HOME>/insightedge\tools\jdbc\tableau

Follow the next instructions detailed in the README.txt file:

1. Run install maven rep of XAP and InsightEdge:
<XAP_HOME>/tools/maven/installmavenrep.cmd and
<XAP_HOME>/insightedge/tools/maven/insightedge-maven.cmd
2. Run build-jdbc-client.cmd command: **(**
<XAP_HOME>/insightedge/tools/jdbc/build-jdbc-client.cmd
3. Copy the created insightedge-jdbc-client.jar to C:\Program Files\Tableau\Drivers

[1] I9E - InsightEdge

3.0 USING THE APPLICATION

3.1 Running the project

NOTE: as mentioned, docker can be installed on Windows\Mac\Linux.
The next demonstration relevant to Linux which requires "sudo" before any docker command if the user is not root.

Go into our docker folder at:
https://github.com/OriBenZaken/Final_Project_Uber/

Go to: Final_Project_Uber/docker
Our Docker contains 6 images:

- Three zookeeper servers – runs the zookeeper servers which the Kafka server depends on.
- Kafka server.
- Kafka producer – responsible to run the Kafka producer.
- InsightEdge consumer – responsible to run I9E first, and then the Kafka consumer.

3.2 Download Model File

This step is required in order to build docker images of our project later on, and supply them our serialized trained best-model of taxi demand prediction per area and time.

Please go to this link:
https://drive.google.com/file/d/1bLlpWQCQwM2HInZKLQE46pdXOYwCqRe4/view?usp=sharing

Download the model file and locate it in Final_Project_Uber/docker/src directory.

3.3 Build and Run Docker

Run the next command to build the local images:
./build-local-images.sh

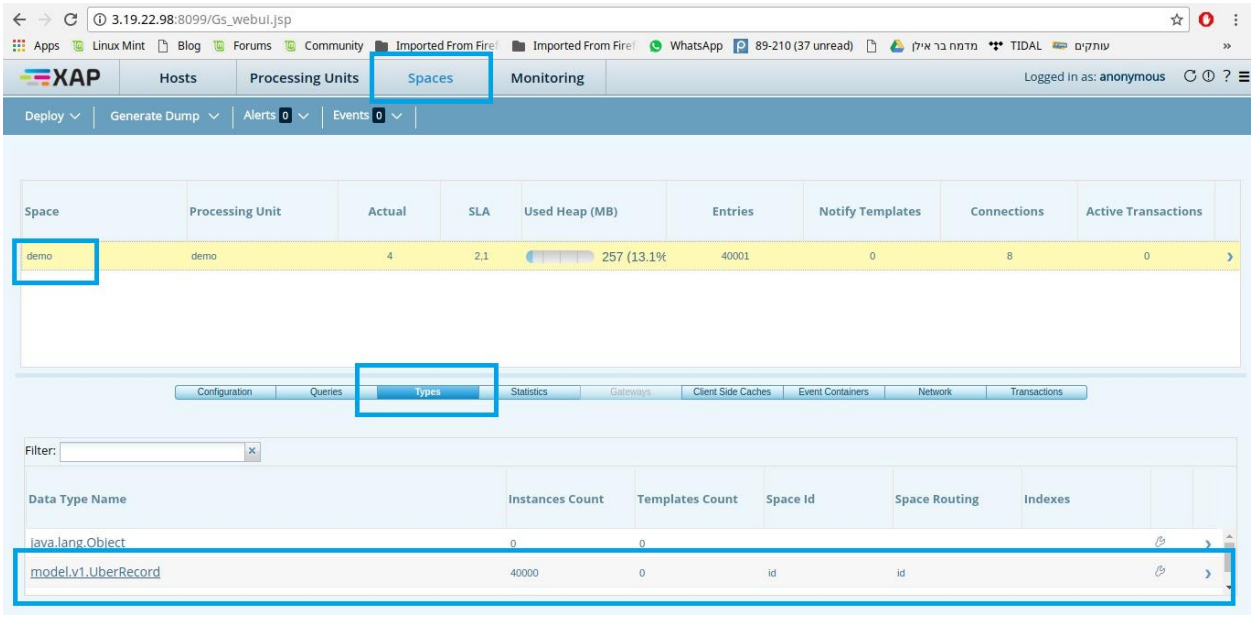
Then, run the docker compose, using the command:
sudo docker-compose up -d && sudo docker attach producer

The docker compose is responsible to run and create containers of all the existing images we have created, in addition to images which available remotely.
The docker compose responsible to run each of them in a certain order, to ensure each image runs only when the images that it is depends on have already been activated.

The above command starts all our needed services in detached mode and right after that attaches to the Kafka producer service, which is waiting for input – new ride information.

Attach to the Kafka consumer in order to see its output:
sudo docker attach consumer

Next, go to InsightEdge grid (Xap), at the address:
http://localhost:8099/
Go to spaces > demo (DB name) > Types, see the table: "model.v1.UberRecord"
The table contains 40,000 records of past rides:



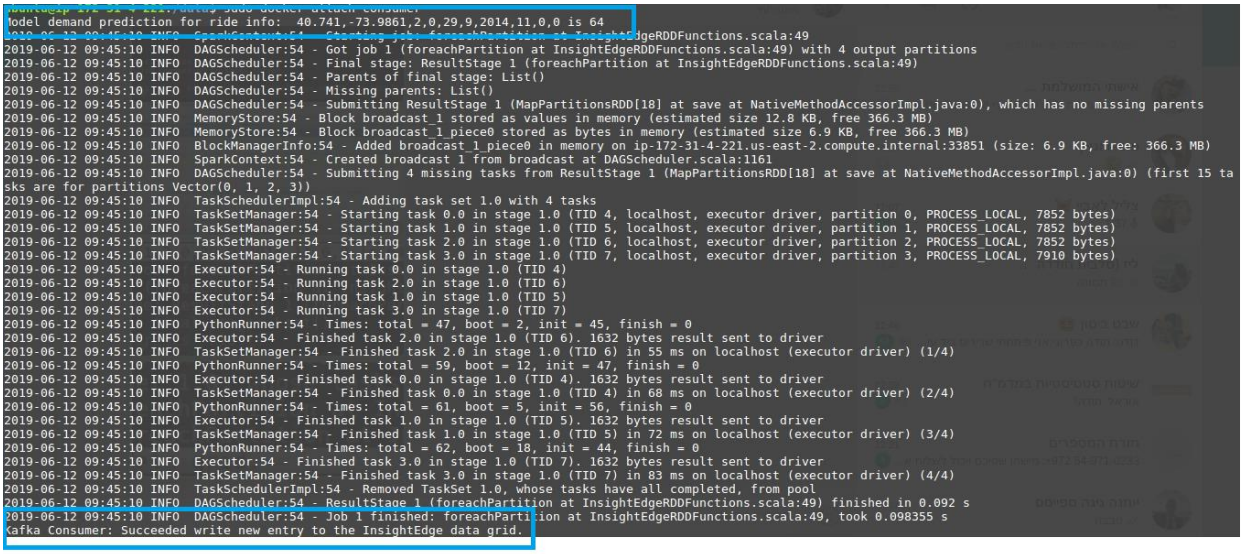
Enter new ride information in the specified format[2]:

```
ubuntu@ip-172-31-4-221:/data/final_proj/Final_Project_Uber/docker$ sudo docker-compose up -d && sudo docker attach producer
Creating network "docker_default" with the default driver
Creating docker_zoo1_1 ... done
Creating docker_zoo2_1 ... done
Creating docker_zoo3_1 ... done
Creating docker_kafka1_1 ... done
Creating producer        ... done
Creating consumer        ... done
Kafka Producer: Trying to establish connection to kafka server...
Kafka Producer: Succeeded to establish connection to kafka server.
Enter a new ride info:
<longitude, base, weekday, day, month, year, hour, isWeekend, isHoliday> 40.741,-73.9861,2,0,29,9,2014,11,0,0
Producer sent messages!
```

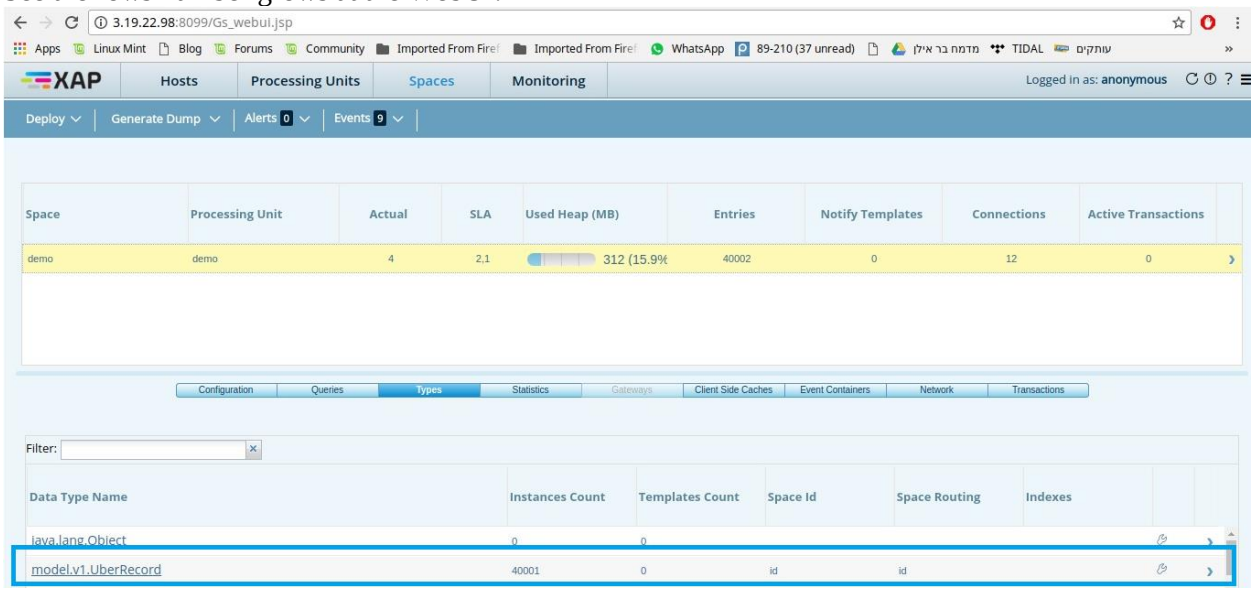
Now, the new entered record is passed to the consumer via Kafka's queue.

You can see the output in the consumer window:

[2] See the appendix for more details about the input format.



See the rows number grows at the WebUI:



To stop the docker, and close the network created by it press:
CTRL C
And then enter:
sudo docker-compose down

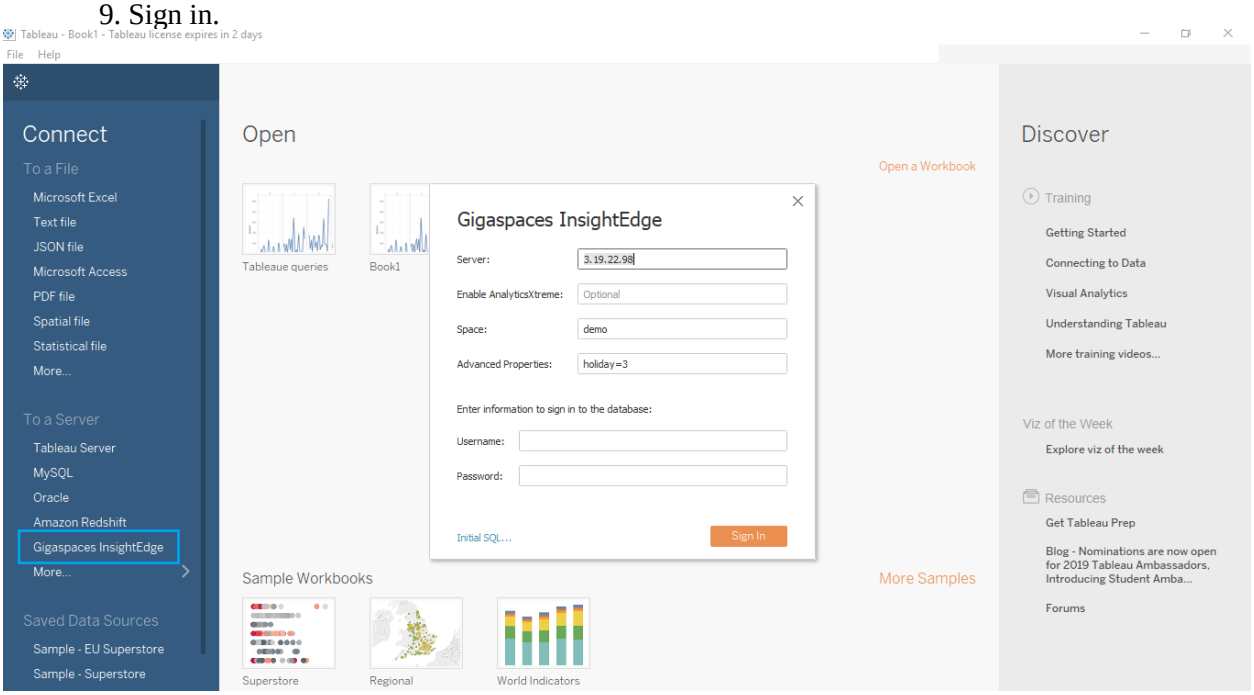
3.4 Running Tableaue [windows]

Precondition to run Tableaue: run I9E using the docker compose detailed in the previous stage.

Then, follow the next instructions:

1. Run <TABLEAU_HOME>\bin\tableau.exe -DConnectPluginsPath=<XAP_HOME>\insightedge\tools\jdbc
5. In Tableau's "To a Server" menu, choose "Gigaspaces InsightEdge".

6. In Server field, fill the locator – insert the ip of the linux computer in which you operated the previous stage (stage 3.1)
7. In Space field, write the space name – "demo"
8. All other fields are optional
9. Sign in.



Create a new sheet or open the existing sheet we have already created, to do so select:
File > Open:
Take the Tableaue queries.twb file from:
https://github.com/OriBenZaken/Final_Project_Uber/tree/master/tableau/

4.0 APPENDIX

4.1 Producer Input Format:

As mentioned in the previous section, new ride information need to be inserted into the producer.
Here is information about the data types and correct format:

Feature:	**Data type:**	**Notes:**
latitude	DoubleType	Latitude where the meter was disengaged.
longitude	DoubleType	Longitude where the meter was disengaged
base	IntegerType	Indicates the order's area: 0 - 'B02512', 1 - 'B02598', 2 - 'B02617', 3 - 'B02682', 4 - 'B02764', 5 - 'B02765', 6 - 'B02835', 7 - 'B02836'
weekday	IntegerType	0 – Sunday 1 – Monday 2 – Tuesday 3 – Wednesday 4 – Thursday 5 – Friday 6 – Saturday
day	IntegerType	Day of date
month	IntegerType	
year	IntegerType	Four digits format
hour	IntegerType	
isWeekend	IntegerType	0 – not weekend 1 – weekend
isHoliday	IntegerType	0 – holiday 1 – not holiday Determined according US holidays.