Taxi Trip Data Analysis and Visualization using Python Libraries

Geraldi Dzakwan - 13514065
Informatics/Computer Science Undergraduate Program
School of Electrical Engineering and Informatics
Institut Teknologi Bandung, Ganesha Avenue No. 10 Bandung 40132, Indonesia
13514065@std.stei.itb.ac.id

***Absctract*—Data analysis and visualization are some parts of data science field that are currently emerging as hot topics as the demand for data scientists rises up in industry. In short, data analysis and visualization ease people to get summary and insight from unstructured data so they can furthermore take a decision based on their understandings about the data. There are currently many programming languages supported by their corresponding libraries which specialize in data analysis and visualization. Some of them which are popular amongst data scientist are R and Python. In this paper, the author would report his exploration on data analysis and visualization using Python on an accurate dataset describing a complete year (from 01/07/2013 to 30/06/2014) of the trajectories for all the 442 taxis running in the city of Porto, Portugal. The dataset is taken freely from Kaggle website. In the introduction, the author will tell about what was the works and what Python libraries are used for the works. Next, there will be problem analysis and the solution for the problem. Last, the author will describe the results of the work and how he can improve the results further.**

***Keywords*—data analysis, data visualization, python, pandas, matplotlib, scikit-learn, taxi trip dataset, kaggle**

I. INTRODUCTION

Before moving further, the author would like to describe about the detail of the dataset. Each data sample corresponds to one complete trip **[2]**. It contains a total of 9 (nine) features, described as follows :

1. TRIP_ID: (String) It contains an unique identifier for each trip.
2. CALL_TYPE: (char) It identifies the way used to demand this service. It may contain one of three possible values:
   'A' if this trip was dispatched from the central;
   'B' if this trip was demanded directly to a taxi driver on a specific stand;
   'C' otherwise (i.e. a trip demanded on a random street).
3. ORIGIN_CALL: (integer) It contains an unique identifier for each phone number which was used to demand, at least, one service. It identifies the trip's customer if CALL_TYPE='A'. Otherwise, it assumes a NULL value.
4. ORIGIN_STAND: (integer): It contains an unique identifier for the taxi stand. It identifies the starting point of the trip if CALL_TYPE='B'. Otherwise, it assumes a NULL value.
5. TAXI_ID: (integer): It contains an unique identifier for the taxi driver that performed each trip.
6. TIMESTAMP: (integer) Unix Timestamp (in seconds). It identifies the trip's start.
7. DAYTYPE: (char) It identifies the daytype of the trip's start. It assumes one of three possible values:
   'B' if this trip started on a holiday or any other special day (i.e. extending holidays, floating holidays, etc.).
   'C' if the trip started on a day before a type-B day;
   'A' otherwise (i.e. a normal day, workday or weekend).
8. MISSING_DATA: (Boolean) It is FALSE when the GPS data stream is complete and TRUE whenever one (or more) locations are missing.
9. POLYLINE: (String): It contains a list of GPS coordinates (i.e. WGS84 format) mapped as a string. The beginning and the end of the string are identified with brackets (i.e. [and], respectively). Each pair of coordinates is also identified by the same brackets as [LONGITUDE, LATITUDE]. This list contains one pair of coordinates for each 15 seconds of trip. The last list item corresponds to the trip's destination while the first one represents its start.

The dataset can be accessed at https://www.kaggle.com/c/pkdd-15-predict-taxi-service-trajectory-i/data. The csv file contains more than 1,8 million rows of record to be analyzed.

Based on those data, there are three main works the author has done, which are :

1. Find most visited places amongst others
2. Visualize the endpoint for each trip
3. Predict time taken for a particular trip

The first task is related to data analysis. The tool used for the first task is Pandas (Python Data Analysis Library). It is an open source, BSD-licensed library providing high-performance, easy-to-use data structures and data analysis tools for the Python programming language **[3]**.

The second task is related to data visualization. The tool used is matplotlib. Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms **[4]**.

The third task is related to machine learning. The tool used is scikit-learn. It is a simple and efficient tools for data mining and data analysis, freely accessible (open source), and reusable in various contexts. It is built on NumPy, SciPy, and matplotlib and it acts as a very common tool to do machine learning in Python **[5]**.

All the code written in Python that is used to solve the tasks is hosted at the author's private github. The link is https://github.com/geraldzakwan/TaxiTripExploration.

II. PROBLEM ANALYSIS AND SOLUTION

The first problem that comes across mind is how to process so many rows of data (1,8 million) in the same time. After exploring some alternatives, the author finds the best solution is to convert the data from CSV format to HDF5 format. HDF5 itself is a format designed to store large numerical arrays of homogenous type. It comes particularly handy when you need to organize your data models in a hierarchical fashion and you also need a fast way to retrieve the data **[6]**.

Because this format only allows you to store homogenous data, the author decided to only save the latitude and longitude from POLYLINE attribute. This is done by the ToH5.py script. Besides converting the data format, the author also processed the data in a way that is memory friendly. If we load all the data in the same time, we can hang the computer up as they flood the RAM. Instead, we could process the data "chunk by chunk". Each chunk contains 1000 records. Basically, the author computed the result from each chunk and combined the results from each chunk to get the final result.

```
# Mengubah format data dan melakukan pembersihan data.
# Cara penggunaan: python ToH5.py source.csv dest.h5
import sys
import pandas as pd

# Input
df = pd.read_csv(sys.argv[1], index_col=(0, 1), names=['lat', 'lon'])

# Menghilangkan index duplikat
df = df.reset_index()
df = df.drop_duplicates(subset=['level_0', 'level_1'])
df = df.set_index(['level_0', 'level_1'])

# Output
df.to_hdf(sys.argv[2], 'data')
```

Picture 2.1 : Code snippet to convert CSV format to HDF5 format
Source : Author's own resource

Next problem to solve is how do we map the coordinates to real places in Porto? Because it is very unefficient to count the occurrence of every exact points (latitude and longitude) in the dataset. The author then decided to make use of the file metaData_taxistandsID_name_GPSlocation.csv provided by Kaggle. In the file, there are 63 places with their corresponding coordinates (latitude and longitude) defined.

So, the idea is to convert the latitude and longitude found in dataset to nearest place in the meta data. Basically, the author computed the distance between the processed coordinate and all the coordinates defined in the meta data. Once the lowest distance found, then the corresponding place will be treated as a place for the processed coordinate. Each processed coordinate will be labeled by a place ID (1-63) instead of the place name to simplify the algorithm and the computing process. This is done using the CoordinateToPlace.py script.

Picture 2.1 : Code snippet to map coordinate to nearest place
Source : Author's own resource

With those steps explained, it should be self explanatory how to determine the most visited places. Next problem is the second task, which is visualizing the destination of each trip. The code which does the visualization will not be posted here, as it is a very long code and has many properties to set. You can see the author's code using matplotlib in the link given previously. Instead, the author would like to post two lines of code which show how the author get the last coordinates using Pandas library.

Picture 2.2 : Code snippet to get each last coordinates from POLYLINE features
Source : Author's own resource

For the last task, the first thing to do is to determine which attributes will be used for the machine learning model. The author started with very simple approach. It only takes two attributes, the first one is the start

point/coordinate of the trip and then the second one is the number of coordinates of the trip. So, basically, the author will predict the number of coordinates based on the start coordinate. The main reason of choosing these features is that the distance of taxi trip depends a lot on the start point. The method used is nearest neighbor and naive bayes. Those two method are found effective and those two method can run perfectly without modifying the format of the data furthermore (they suit the HDF5 format).

```
id_test = tdf.index.to_series().as_matrix()

# Initialisasi model
model = RadiusNeighborsRegressor(radius=0.0005, weights='distance')

# Training
model.fit(X_train, y_train)

# Prediksi
y_try = model.predict(X_test)

# Penulisan hasil
resdf = pd.DataFrame({'Id': id_test, 'predict': (y_try), 'actual': (y_test)}).set_index('Id')

resdf.to_csv(sys.argv[3])
```

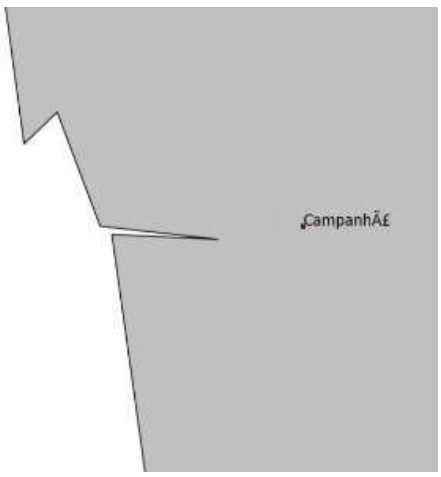
Picture 2.3 : Code snippet to predict trip time
Source : Author's own resource

Based on the author preliminary analysis on the dataset, the distance will be far if the start point is outside the centre of Porto city and will be relatively short if the start point is inside the center of Porto city. That insight makes the author decided to choose the start point as the main feature to create the learning model. Finally, if the number of coordinates can be predicted, then the time taken for each trip can be calcuated by multiplying the number of coordinates with 15 seconds.

III. IMPLEMENTATION AND RESULT

1. Most visited places

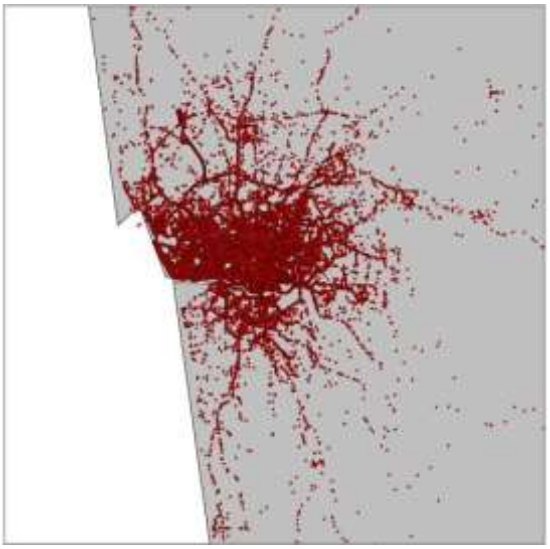
   To get the visualization, the author runs the MostVisited.py to count the occurences of all the coordinates without sampling (the author processed all of the 1,8 millions record). The author listed the top twenty coordinates. It turns out that they all mapped to a place called "Campanhã".



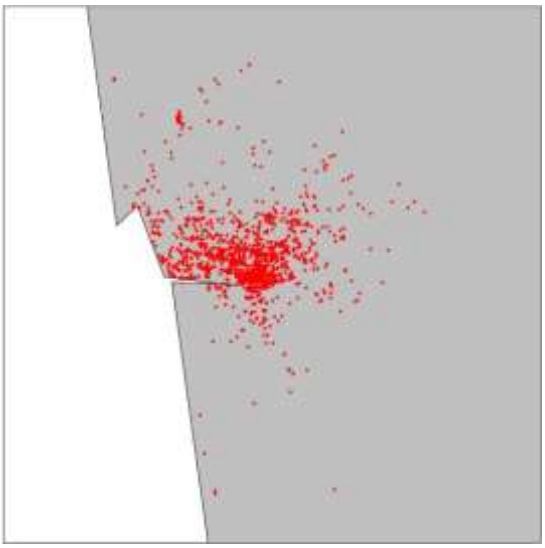
Picture 3.1 : Most visited place visualization
Source : Author's own resource

2. Visualization of trip's endpoints

   To get the visualization, the author runs the DestLocation.py script to take all the endpoints. There are two visualizations, the first one represents a hundred thousand sample data while the second represents two thousands data. The result is a map with red dots plotting all the endpoints, some of them even resemble roads in Porto. We can infer that the second one gives us clearer information about endpoints distributions.



Picture 3.2 : Trip's endpoints visualization with 100000 sample data
Source : Author's own resource



Picture 3.2 : Trip's endpoints visualization with 2000 sample data
Source : Author's own resource

3. Time prediction accuration

The author runs CreateMLData.py to convert the csv data to a format which is ready to be processed by the machine learning algorithm. After the data is converted, the Prediction.py script is run to predict the time taken for each trip using the method that is already explained in the previous chapter.

If we set the error margin between actual time and prediction time is 5 minutes, the accuration of this model is about 73% (235 corrects out of 320 test data). Of course, the accuration is still very low and it needs improvements. The prediction time details can be seen in the github link given before.

There are so many features that are not being concerned in the process of creating the model. The feature that the author thinks will create big difference if it is concerned is the DAYTYPE attribute. For example, we may find that if the DAYTYPE is B (holiday or any other special day), then the trip will take much longer than usual day as traffics usually come in holiday. If the DAYTYPE is C (a day before holiday) we may find that the trip that is using toll road or highway (crossing the border of Porto city) will take much longer as people usually travels far either to get back home before holiday or to go to the center of Porto city to find amusements.

Other than the features, change of the algorithm can also increase accuration. The algorithm that is already used is nearest neighbor and naive bayes. Next time, the author will try more advance algorithm that is provided by scikit-learn like ID3 learning or even neural network. The author of course needs to modify the format of the data so it can be used by other learning algorithms.

IV. SUMMARY AND FURTHER WORK

There are currently many programming languages that can be used to do data analysis and visualization such as R and Python. To new comers who don't know both R and Python, the author suggests to learn Python because it can be used for other purposes too (e.g. web development using Django) and it has a nice learning curve for beginners. It is also equipped with bunch of libraries to help you start on. If you pursue to be data scientist , you might want to check Kaggle website as it offers you thousand dataset with it's respective challenges.

Further work for the taxi trip dataset is to improve the accuracy of the model by adding more features to concern and find best learning algorithm which suits the context. The author may also want to work on new task which is more challenging. That would be to predict the trajectory of the taxi trip based on the start point. It would be hard since you can't see any similar pattern of the trajectory between the same two start points.

ACKNOWLEDGMENT

First of all, the author want to give his gratitudes towards Allah SWT, because of the blessing the author can finish this paper with ease.

The author also wants to thank Dr. Eng. Ayu Purwarianti, S.T., M.T. as the lecturer of this course, Socio-Informatics and Professionalism. Because of her guidance, the author is able to create a paper about his exploration on data analysis and visualization using new tools he hasn't learned before.

Lastly, the author want to thank his parents who always give support to all academics matter and to his friends who helps in the creation of this paper.

REFERENCE

[1] Budihaarto, Widodo. 2016. "Machine Learning and Computational Intelligence". Jakarta:Penerbit Andi.
[2] https://www.kaggle.com/c/pkdd-15-taxi-trip-time-prediction-ii
Accessed at May 1 2017 at 13.30
[3] http://pandas.pydata.org
Accessed at May 1 2017 at 16.00
[4] https://dzone.com/articles/quick-hdf5-pandas
Accessed at May 2 2017 at 15.00
[5] https://matplotlib.org
Accessed at May 3 2017 at 14.00
[6] http://scikit-learn.org/stable
Accessed at May 3 2017 at 19.00

AUTHENTICITY STATEMENT

The author hereby states that this paper is created based on his own ideas, his own works, and his own writings. The author also makes sure that this paper doesn't contain any form of plagiarism and the content of this paper isn't a result of translation from other papers.

Bandung, May 5 2017

Geraldi Dzakwan