**FINAL PROJECT FOR COURSE MACHINE LEARNING**

***Establishment of neural network model and analysis of Radar Traffic Data***

Students:

Linxue Lai, Weicheng He

Professor:

Christophe Cerisara

Dec. 14th ,2020

## 1 Introduction

This report is about our final project for the course Machine Learning [1]. We worked in a team of two members. Our task is to Build a deep learning model that predicts the traffic volume. The data that should be analyzed is called “Radar Traffic Data”, which can be download in Kaggle [2]. The traffic data is collected from radar sensors deployed by the city of Austin.

This report is developed by the following parts: data analysis and processing, model construction and parameter setting, experimental process and conclusions, personal experience (individual notes).

The following is our work plan:

|  |  |
| --- | --- |
| Duration | task |
| 16/11/2020 - 23/11/2020 | * Start the project: create a github project for our co-work, create a shared Google doc and a shared kaggle notebook for sharing ideas or articles referring to our project. * Conduct preliminary data analysis and read related papers |
| 24/11/2020 - 01/12/2020 | * Chose the appropriate time series analysis model: LSTM, GRU |
| 02/12/2020 - 08/12/2020 | (Weicheng HE) Construction of model LSTM and improve model.  (Linxue LAI) Construction of model GRU and improve model. |
| 09/12/2020 - 13/12/2020 | Improve codes, summarize experimental results, write report |

## 2 data analysis and processing

In order to facilitate data visualization and data analysis, we use the shareable notebook of the kaggle platform for data analysis and processing.

### 2.2 Data description:

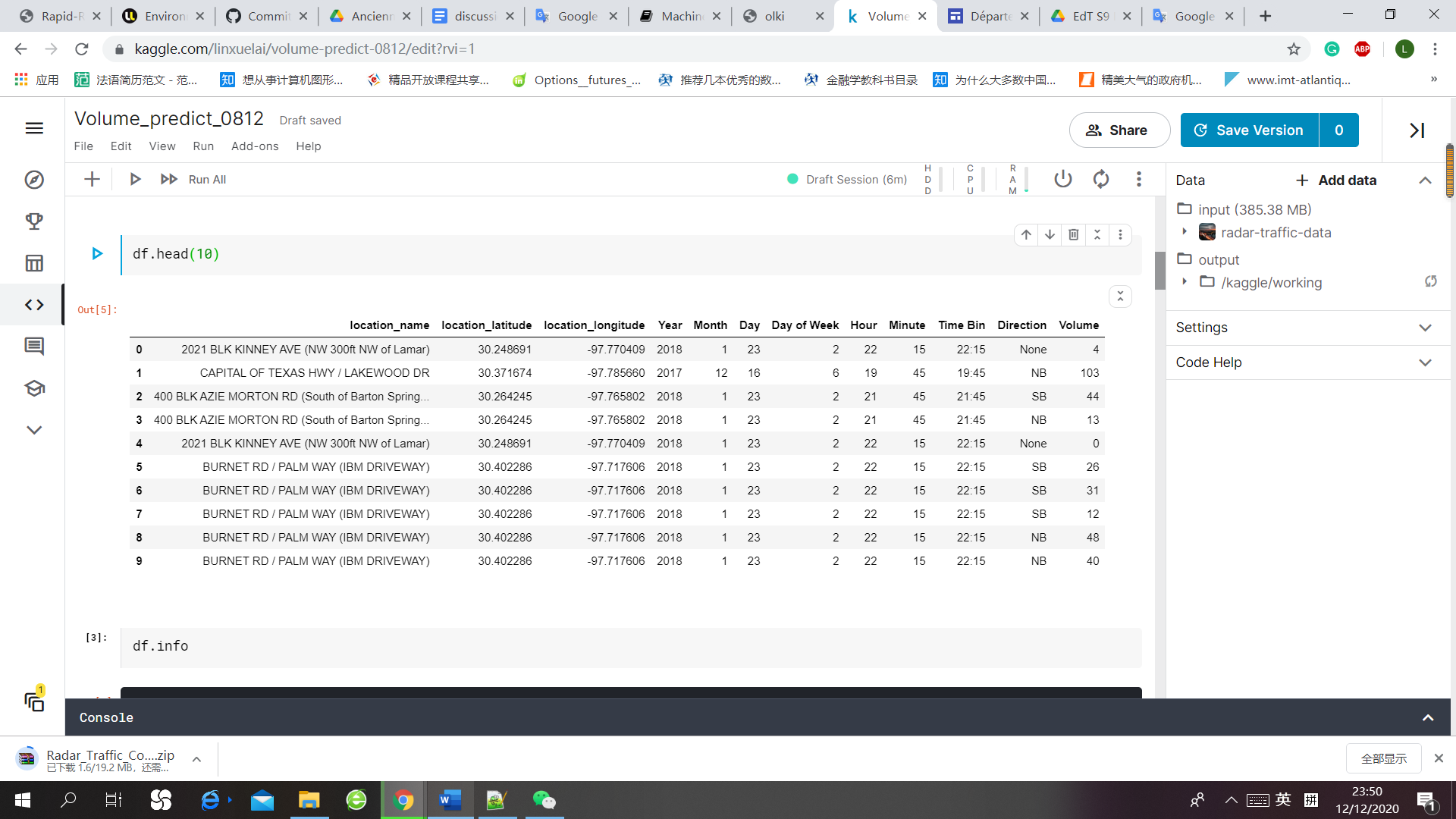
Traffic data collected from the several Wavetronix radar sensors deployed by the City of Austin. Dataset is augmented with geo coordinates from sensor location dataset.

# Load data

df = pd.read\_csv("/kaggle/input/radar-traffic-data/Radar\_Traffic\_Counts.csv")

df.head(10)

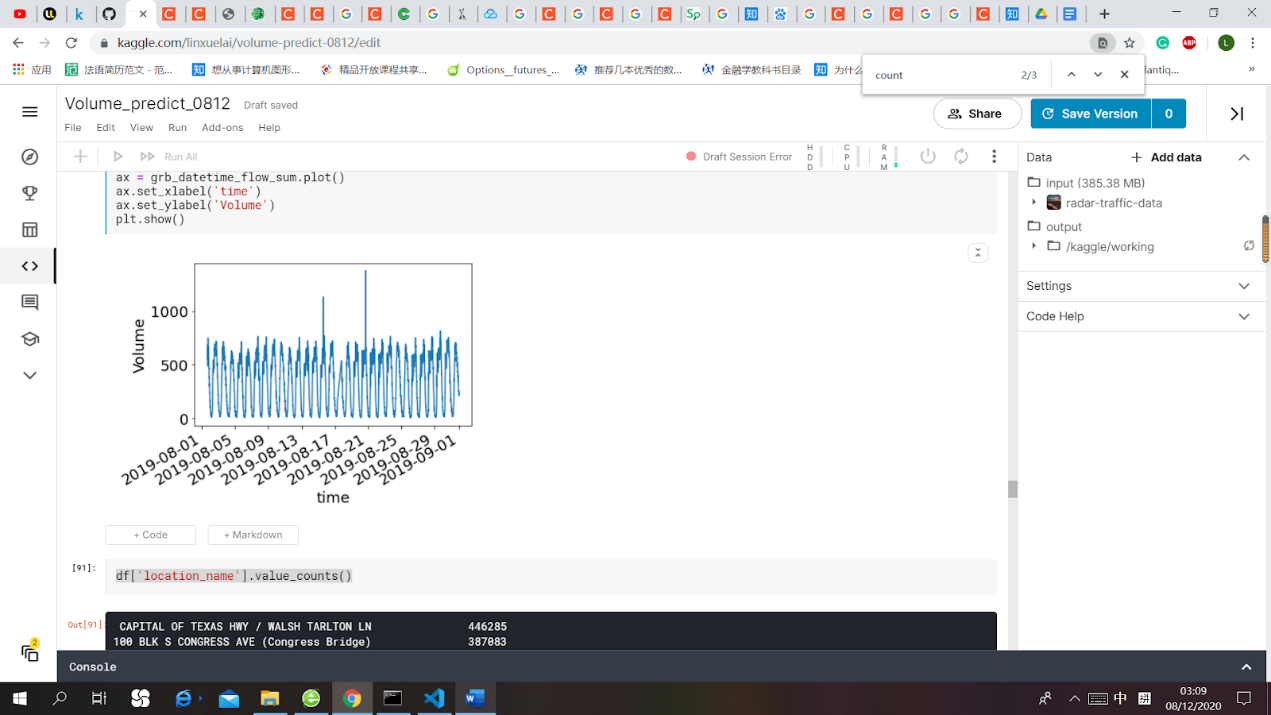
The following is an overview of some of the data:



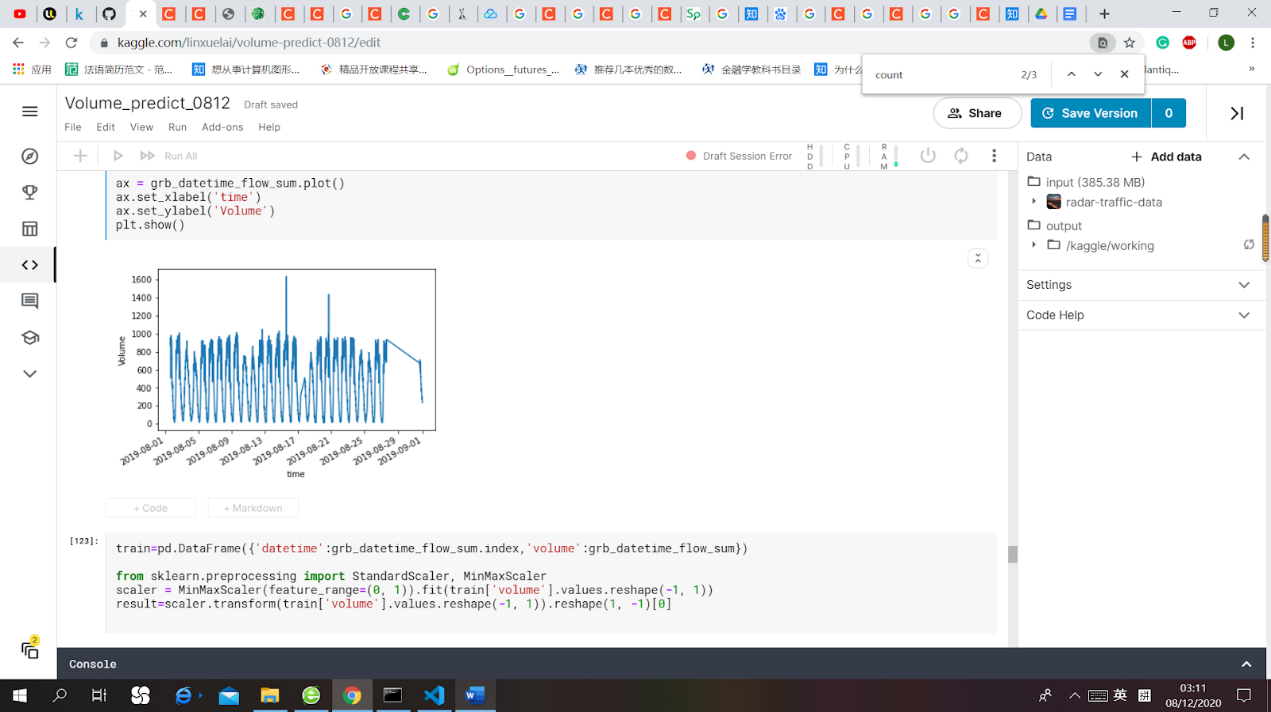
As traffic flow data has obvious periodic change characteristics, we consider using the commonly used LSTM model and GRU model for time series analysis.

There is a question to consider: Is it necessary to consider both time and space factors?

In order to simplify the problem and make it easier to understand, we first consider the changes of traffic volume at a single location over time. By looking at the data of a certain location in a certain period of time, we found that the data has an obvious cycle nature of daily changes. For example:



*Location 1： 3201 BLK S LAMAR BLVD (BROKEN SPOKE)*

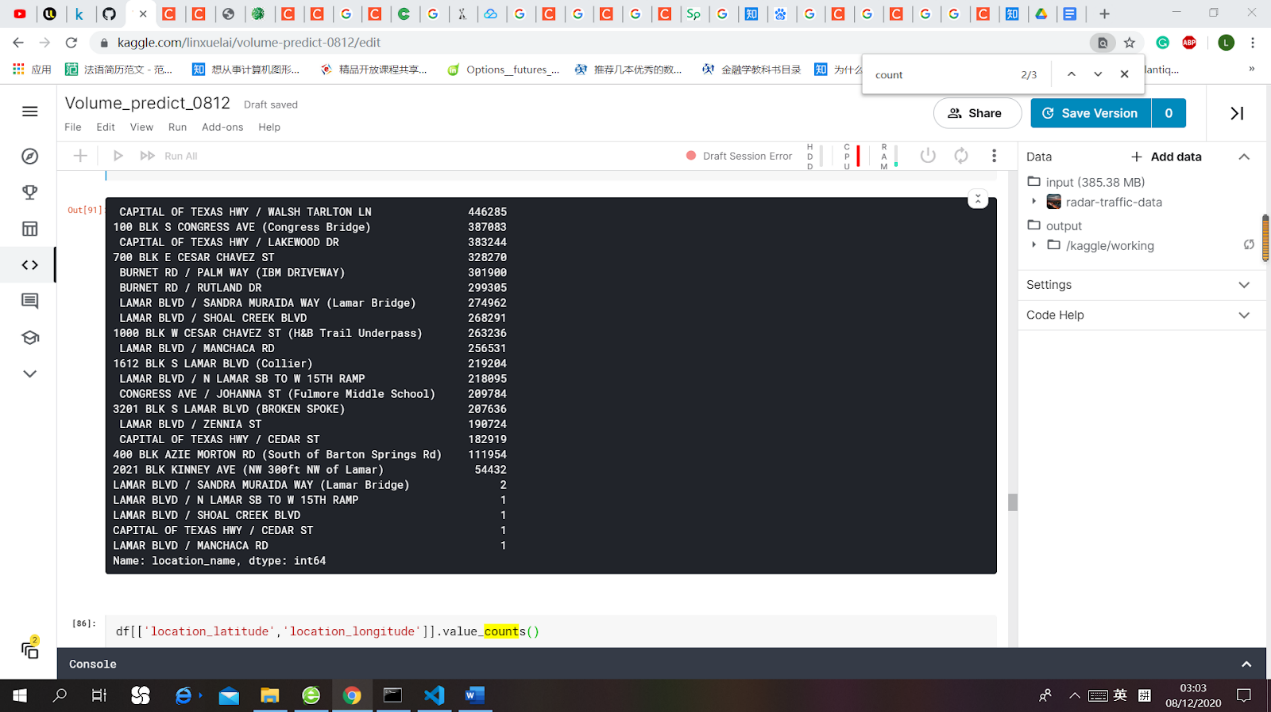
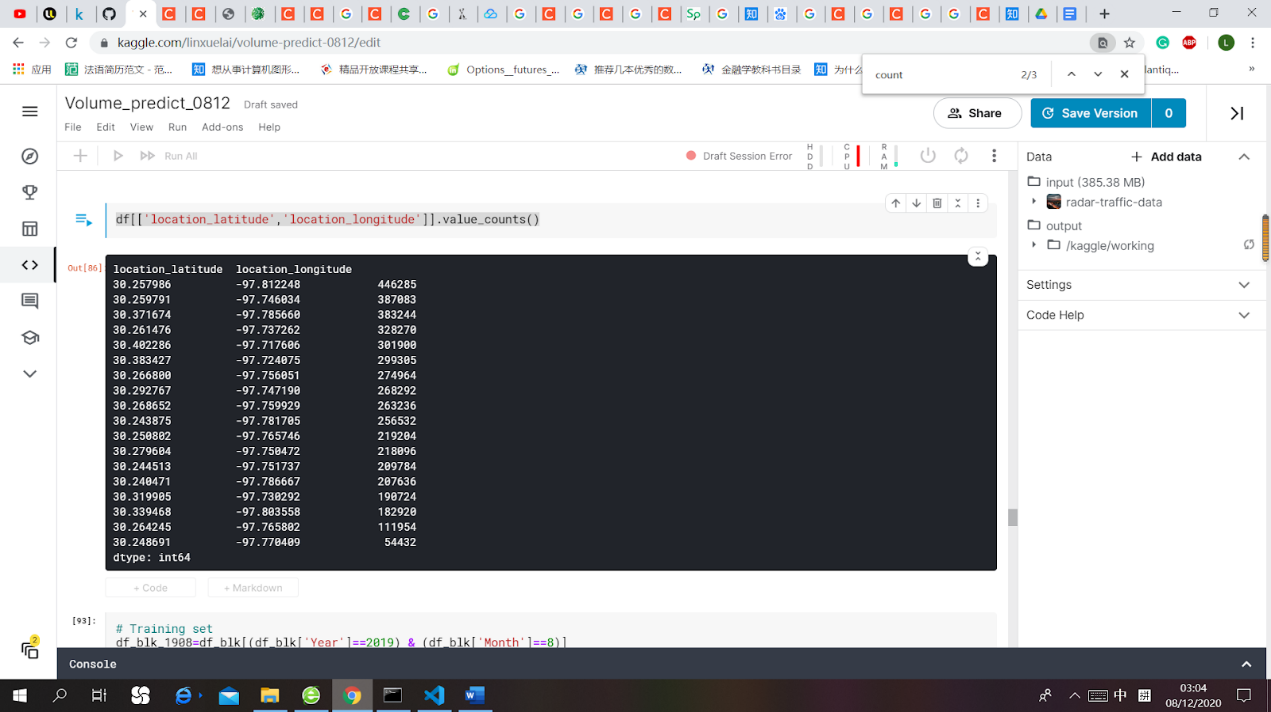


*location 2：100 BLK S CONGRESS AVE (Congress Bridge)*

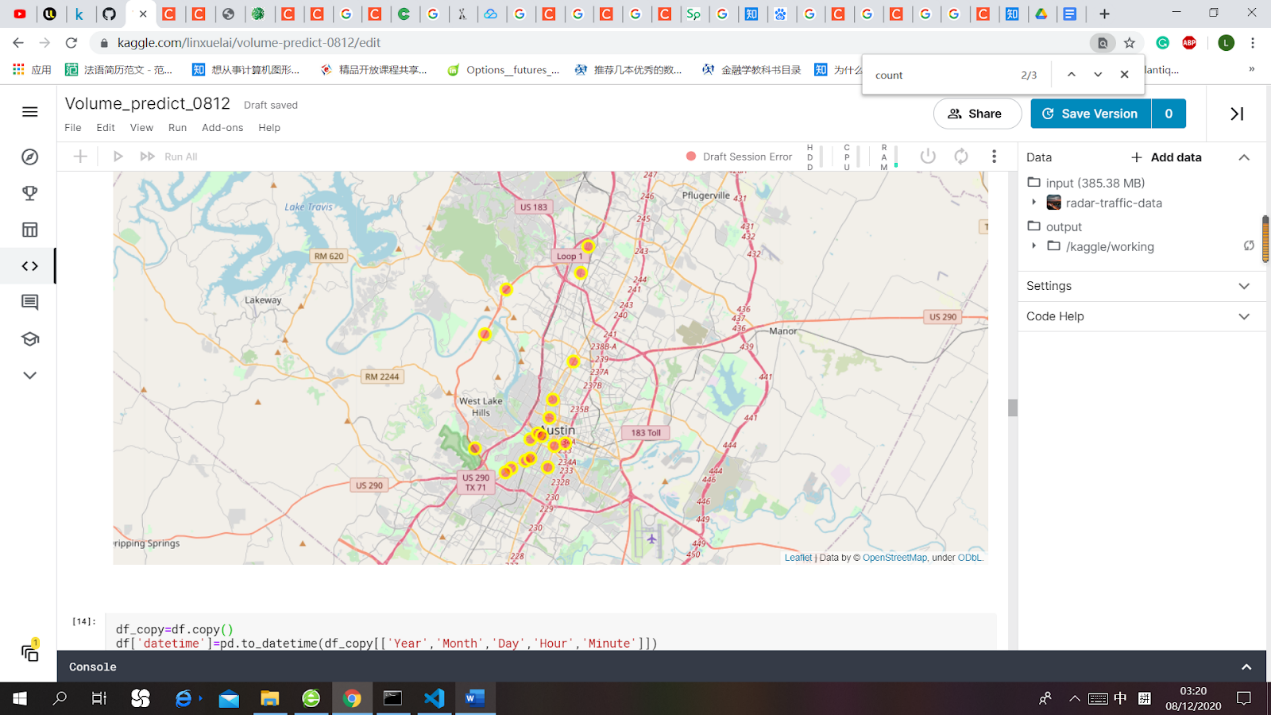
The number of locations: 23 different locations by location\_name; according to actual latitude and longitude: 18 different locations.

df['location\_name'].value\_counts()

df[['location\_latitude','location\_longitude']].value\_counts()

The following is a map-based visualization (see the codes in annexes 1):



## 3 Models

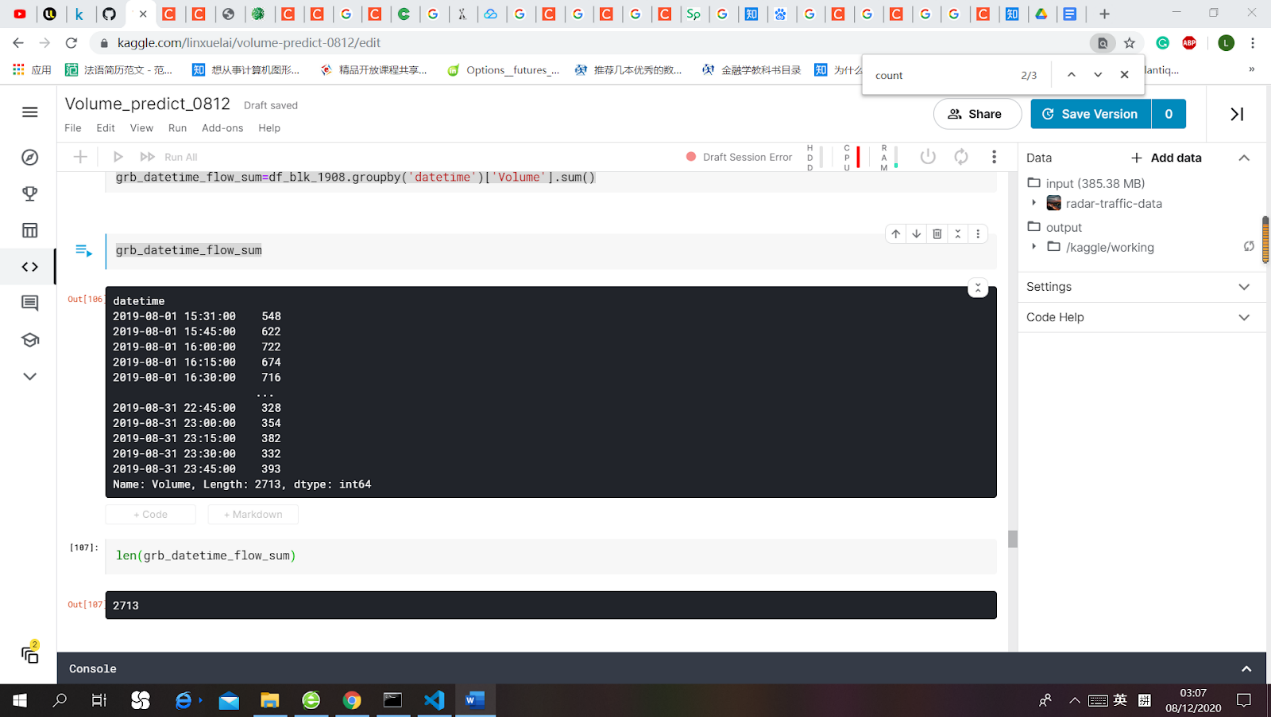
### We built an LSTM model and a GRU model, trained and tested traffic flow data at a certain location, and compared the results of the two models.

### 3.1 LSTM

### 3.2 GRU

选择了其中一个位置，选择某时间段的某地点进行GRU的训练，只考虑时序性。

相邻数据以15min为间隔



# Training set

lags = 4

config = {"batch": 256, "epochs": 600}

raw\_data = 'data/Radar\_Traffic\_Counts.csv'

location='100 BLK S CONGRESS AVE (Congress Bridge)'

units=[lags,64,64,1]

pv=predict\_volume(raw\_data,location,units,lags,config)

pv.training()

pv.model\_evaluation()

测试结果：

### 3.3 Comparison

## 4 Conclusion

## 5 Individual notes

### 5.1

### 5.2

## 6 References

[1] course link: <https://members.loria.fr/CCerisara/#courses/machine_learning/>

[2] data link: <https://www.kaggle.com/vinayshanbhag/radar-traffic-data?select=Radar_Traffic_Counts.csv>

[3]

## Annexes 1: map visualization

import folium

import pandas as pd

df = pd.read\_csv("/kaggle/input/radar-traffic-data/Radar\_Traffic\_Counts.csv")

# define the world map

world\_map = folium.Map()

# display world map

world\_map

lat\_lon = pd.DataFrame(df,columns=['location\_name','location\_latitude','location\_longitude'])

lat\_lon.drop\_duplicates(['location\_name','location\_latitude','location\_longitude'], keep='first', inplace=True)

print(lat\_lon)

lat\_lon.drop\_duplicates(['location\_latitude','location\_longitude'], keep='first', inplace=True)

print(lat\_lon)

print(len(lat\_lon))

# get the data in map

limit = len(lat\_lon)

data = lat\_lon.iloc[0:limit, :]

latitude,longitude = lat\_lon['location\_latitude'][0],lat\_lon['location\_longitude'][0]

# Instantiate a feature group in the dataframe

index = folium.map.FeatureGroup()

# Loop through the data and add each to the feature group

for lat, lng, in zip(lat\_lon.location\_latitude, lat\_lon.location\_longitude):

index.add\_child(

folium.CircleMarker(

[lat, lng],

radius=7, # define how big you want the circle markers to be

color='yellow',

fill=True,

fill\_color='red',

fill\_opacity=0.4

)

)

# Add to map

v\_map = folium.Map(location=[latitude, longitude], zoom\_start=12)

v\_map.add\_child(index)