Način na koji dobivam početno vrijeme i vrijeme dolaska na sljedeću

Skupljam poruke od gtfs realtime i spremam ih u bazu

Kasnije napravim reflexive (samu na sebe) join tablicu kako bi odlučio kada je došla na sljedeću stanicu.

Poštujući id poruke i tripid (to bi trebale biti unique vrijednosti koje označuju jedan tramvaj?), također prije inserta u bazu radim hash svega da ne bi duplikate stavljao

Stvara mi se problem što je razlika dolaska na jednu stanicu i na drugu premala, npr. došao na stanicu 1 "20:07:12", a na sljedeću "20:07:21".

Zakoruženo na slici, možemo vidjeti da na istu stanicu u roku od jedne minute imam 5 redaka koji svaki ima drugo vrijeme dolaska. zašto?

Moguće da neshvaćam kako oni jedinstveno označuju jedan tramvaj ali trebalo bi biti (id i tripid), pa onda nekako se više tramvaja obuhvati??

Neko naivno rješenje za taj problem npr. za svaki currentstopsequence u nekom vremenskom okviru npr.

1-2 minute ograničiti da bude samo jedan unos

Još nešto u obzir nisam uzimao smjer tramvaja -> to treba pogledati (ali ne bi trebalo predstavljati problem)

Zadatak opet vratiti se na gtfs podatke ili negdje iskopati kako oni unikatno prikažu jedan tramvaj, identifikator

id text	tripid text	routeid text	stopid text	currentstopsequence integer	arrivedat timestamp without time zone	reporteddelay integer
X154YPS0F3	0_32_1701_17_32306	17	291_2	2	2025-01-12 20:07:34	115
X154YPS0F3	0_32_1701_17_32306	17	291_2	2	2025-01-12 20:07:44	125
X154YPS0F3	0_32_1701_17_32306	17	217_2	3	2025-01-12 20:08:29	84
X154YPS0F3	0_32_1701_17_32306	17	217_2	3	2025-01-12 20:08:43	98
X154YPS0F3	0_32_1701_17_32306	17	217_2	3	2025-01-12 20:08:47	102
X154YPS0F3	0_32_1701_17_32306	17	217_2	3	2025-01-12 20:08:57	112
X154YPS0F3	0_32_1701_17_32306	17	217_2	3	2025-01-12 20:09:00	115
X154YPS0F3	0_32_1701_17_32306	17	292_2	4	2025-01-12 20:09:07	46
X154YPS0F3	0_32_1701_17_32306	17	292_2	4	2025-01-12 20:09:20	59
X154YPS0F3	0_32_1701_17_32306	17	292_2	4	2025-01-12 20:09:33	72
X154YPS0F3	0_32_1701_17_32306	17	292_2	4	2025-01-12 20:09:34	73
X154YPS0F3	0_32_1701_17_32306	17	220_2	5	2025-01-12 20:10:19	17
X154YPS0F3	0_32_1701_17_32306	17	220_2	5	2025-01-12 20:10:29	27
X154YPS0F3	0_32_1701_17_32306	17	220_2	5	2025-01-12 20:10:30	28
X154YPS0F3	0_32_1701_17_32306	17	220_2	5	2025-01-12 20:10:40	38
X154YPS0F3	0_32_1701_17_32306	17	220_2	5	2025-01-12 20:10:41	39
X154YPS0F3	0_32_1701_17_32306	17	220_2	5	2025-01-12 20:10:48	46
X154YPS0F3	0_32_1701_17_32306	17	220_2	5	2025-01-12 20:10:52	50

Zasad se ta obrada podatka vrši preko postgresql, ja msm da to nije najbolji način (za nekakvu demonstraciju vjerojatno dobro), čuo sam za bolje stvari na predmetu napredne baze podataka bilo je predavanje o tokovima podataka i tamo se spominje obrada stream i batch podataka. To bi bio nekakav must kada bi se skaliralo radilo nešto ozbiljno, ali o tome nemam pojma.

Moram promjeniti, malo urediti poboljšati svoju sql naredbu, jer preformanse nisu baš velike, spaja par tablica da bi izvukao neke informacije, većina tablica ima do 2000redaka i jedna ima pola milijuna redaka (nju ne koristim baš) i obrada traje 3.6 sekunde da bi na izlazu dobio oko 1400 redaka koje mogu koristiti za treniranje.

To bi trebalo biti puno brže s obzirom da trebam puno više podataka za treniranje i potrebno je napomenuti da sam za ovaj demo pokušaj, pratio samo liniju broj 17, od njih puno više!!

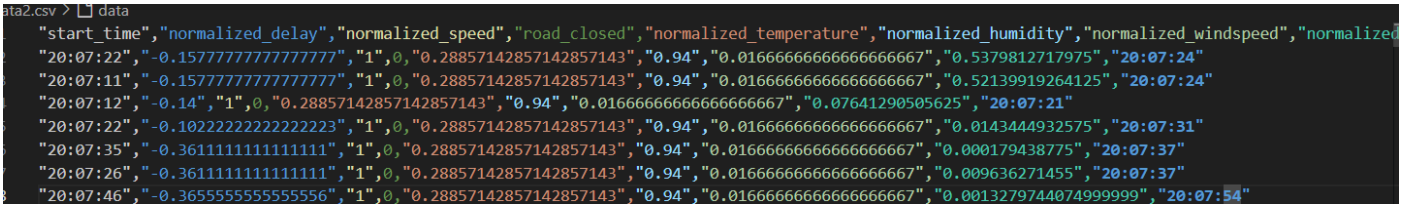
Poznato TomTom API limit -> potrebno smanjiti broj api poziva, jer trenutno za svaku novu poruku o lokaciji vozila izvodi upit (not ideal)

Treniranje modela

Sad sam napravio da učitam te obrađene podatke ubacim u jednu jednostavnu mrežu i tensorflow odradi svo treniranje i testiranje (vrlo jednostavno s moje strane)

Problem tu je odrediti ulazne podatke i normalizirati ih.

Trenutno izgleda ovako



Ali moram još neke podatke dodati u ovo poput vremena koji kada preuzmem od dhmz bude u tekstu oblačno, kiša, vedro .... to treba pronaći sve moguće slučajeve i dodati kao ulaz u mrežu jer po meni stvari poput kiše i snijega jako utječu na promet. (hot encoding)

Općenito promjeniti taj python file da prima argumente dataset i model, istrenira i testira model, spremi ga i tako dalje.

Kasnije se pozabaviti s nekakvim optimizacijskim problemima, promjeniti parametre poput strukture mreže, aktivacijske funkcije...

Treba i odrediti koliko ću modela imati, za svaku liniju jedan model? Za sve linije jedan model?

```
35/35 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 212654240.0000 - val_loss: 161491808.0000
Epoch 50/50
35/35 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 209658864.0000 - val_loss: 157571072.0000
9/9 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 182757328.0000
Test Loss: 157571072.0
9/9 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step

Predictions (Input -> Predicted Output):
Start Time: 20:07:22 -> Input: 0.48, 0.40, 1.00, 0.00, 1.00, 1.00, 1.00, 0.05 -> Predicted Time: 20:48:52
Start Time: 20:07:11 -> Input: 0.45, 0.25, 0.00, 0.00, 1.00, 1.00, 1.00, 0.06 -> Predicted Time: 16:34:51
Start Time: 20:07:12 -> Input: 0.08, 0.36, 0.63, 0.00, 1.00, 1.00, 1.00, 0.01 -> Predicted Time: 17:48:36
Start Time: 20:07:22 -> Input: 0.36, 0.28, 1.00, 0.00, 1.00, 1.00, 1.00, 0.06 -> Predicted Time: 20:00:55
Start Time: 20:07:35 -> Input: 0.14, 0.41, 1.00, 0.00, 1.00, 1.00, 1.00, 0.01 -> Predicted Time: 19:31:04
Start Time: 20:07:26 -> Input: 0.51, 0.37, 1.00, 0.00, 1.00, 1.00, 1.00, 0.01 -> Predicted Time: 20:41:48
```