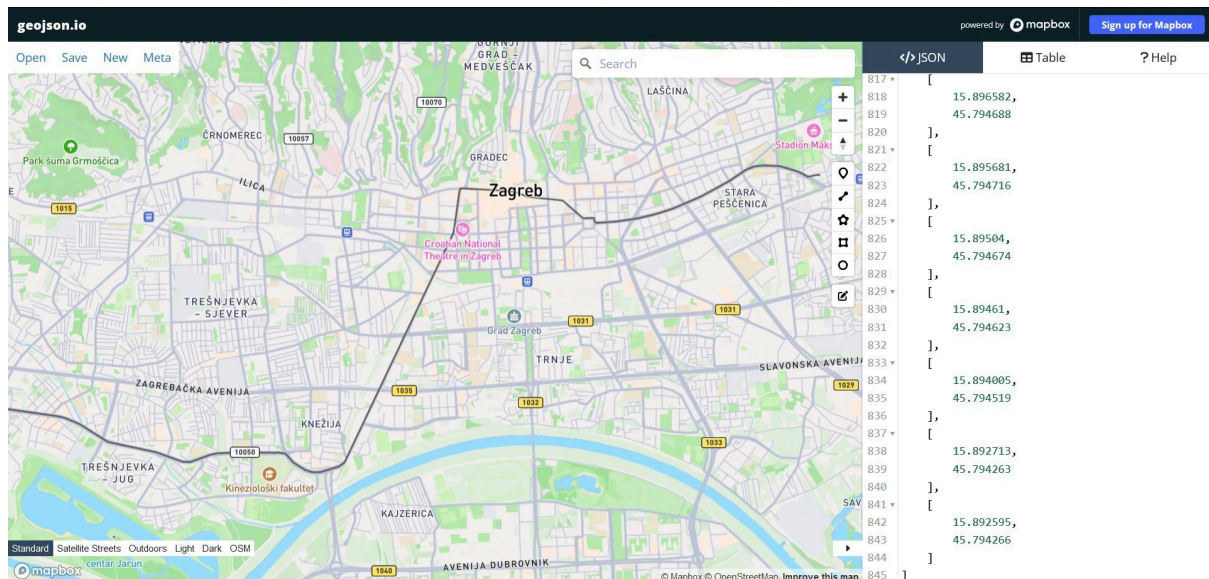


Novosti!

Zet je unaprijedio svoje podatke, sada sadrži shapes.txt (datoteka potrebna za iscrtavanje rute na karti) i sadrži podatke o pozicijama vozila.

Shapes.txt povežemo pomoću tripsID -> trips.txt -> shapesID



Sada se GTFS feed sastoji od dvije vrste poruka: one koje sadrže informacije (potvrde) da su stigli na neku stanicu i one koje sadrže točne geografske lokacije vozila.

```
new 1
1 {
2   "id": "X4OEUPS02E",
3   "tripupdate": {
4     "trip": {
5       "tripId": "0_30_804_8_10519",
6       "startDate": "20250110",
7       "scheduleRelationship": "SCHEDULED",
8       "routeId": "8"
9     },
10    "stopTimeUpdate": [
11      {
12        "stopSequence": 13,
13        "arrival": {
14          "delay": -98,
15          "time": "1736523049"
16        },
17        "departure": {
18          "delay": -98,
19          "time": "1736523049" -- Friday, January 10, 2025 4:30:49
20        },
21        "stopId": "115_4",
22        "scheduleRelationship": "SCHEDULED"
23      },
24      {
25        "stopSequence": 15,
26        "arrival": {
27          "delay": -158,
28          "time": "1736523182"
29        },
30        "departure": {
31          "delay": -158,
32          "time": "1736523182" -- Friday, January 10, 2025 4:33:02 P
33        },
34        "stopId": "1729_4",
35        "scheduleRelationship": "SCHEDULED"
36      }
37    ],
38    "timestamp": "1736523130"
39  }
40 }

new 2
1 {
2   "id": "X4OEUPS02E_102227",
3   "vehicle": {
4     "trip": {
5       "tripId": "0_30_804_8_10519",
6       "startDate": "20250110",
7       "scheduleRelationship": "SCHEDULED",
8       "routeId": "8"
9     },
10    "position": {
11      "latitude": 45.70172134399414,
12      "longitude": 16.002605438232423
13    },
14    "timestamp": "1736523130",
15    "vehicle": {
16      "id": "102227"
17    }
18  },
19 }
```

t

Te poruke su povezane preko ID, gdje poruke o dolasku na stanicu imaju ID oblika X4OEUPS02E, dok one s pozicijama vozila imaju oblik ID + vozilo ID npr. X4OEUPS02E_102227.

Koliko primjećujem nemaju sva vozila podatke o pozicijama jer je neke poruke o dolascima na stanicama nemaju odgovarajuće poruke s pozicijama vozila. (ima 280 poruka s pozicijama vozila dok 461 s trip updateom)

Sada sam odredio da se feed osvježava otprilike svakih 10-11sekundi.

Skupljanje podataka

Zet podatke sam kao i prije dohvaćao samo sam ovaj put na drukčiji način parsirao. Spremio sam podatke tripID, routeID, stopID, currentStopSequence, arrivedAt, reportedDelay te position.

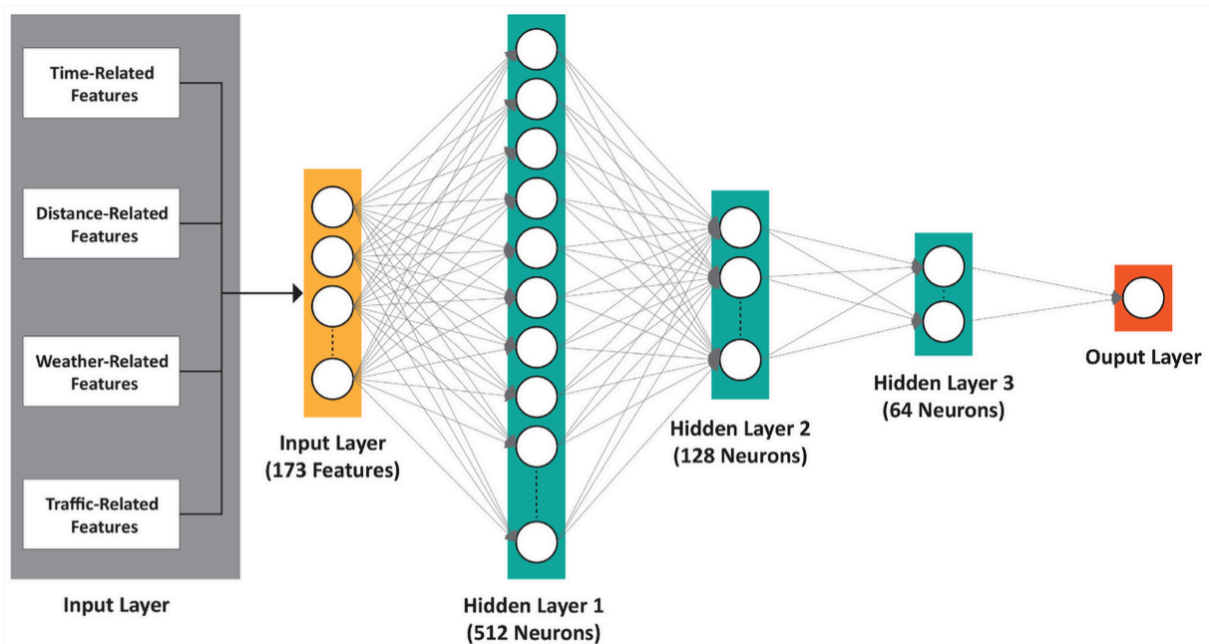
Podaci o vremenu od DHMZ, problem što su dostupne samo dvije lokacije u Zagrebu i ažuriraju ih svaki puni sat. Dohvatim podatke o temperaturi, vlazi, padalinama.

Podaci o gustoći prometa od TomTomAPI, problem je što imam 2500 API poziva dnevno. Zbog toga sam samo promatrao jednu liniju (17), koja je imala 18 vozila, te sam tih 2500 API poziva potrošio u nešto manje od sat vremena. Tu moram odrediti koliko često da ispitujem je li gužva za pojedini prijevoz. Trenutno je napravljeno da svaka nova poruka o novoj lokaciji vozila triggera jedan API poziv da saznam je li vozilo u gužvi. Podaci koje preuzmem o prometu su currentspeed i freeflowspeed te currenttraveltime i freeflowtraveltime.

Predikcije

Pročitao sam na par lokacija da koriste neuronske mreže za određivanje kašnjenja, ili neku drugu vrstu machine learninga. Tako da ću probati i ja sa neuronskom mrežom (nešto slično smo radili na umjetnoj inteligenciji prošle godine pa ću prvo to probati)

Nešto kao ovaj dijagram



Uzeti ću prikupljene podatke, spojiti s statičnim podacima od zet-a, i odrediti neke stvari poput udaljenosti vozila do sljedeće stanice, početno vrijeme, pada li kiša?, gustoća prometa na cesti kod vozila, uzeti u obzir i onaj delay podataka i onda usporediti dobiveno vrijeme (izlaz mreže) i s očekivanim vremenom dolaska (iz statičnih podataka) i sa stvarnim vremenom dolaska kojeg smo dobili od GTFS.

Nažalost još nisam ostvario neki model da Vam to pokažem, kako sam prvotno mislio.

Ali vjerujem da ću ovaj tjedan napraviti nešto poput ovog gore.