Twitter használata a térinformatikában

Készítették: Bodnár Ákos
Építőmérnök BSC hallgató
Horváth Viktor Győző
Építőmérnök BSC hallgató
Kiss Ambrus
Építőmérnök BSC hallgató
Papp Viktor
Építőmérnök BSC hallgató

Konzulens: Dr. Barsi Árpád
BME Fotogrammetria és Térinformatika Tanszék

Tartalom

Bevezetés

Dolgozatunk célja egy olyan módszer kidolgozása, melynek segítségével térinformatikai elemzést hajthatunk végre a Twitteren megosztott tartalmak alapján. A problémát kétféle módszerrel akartuk megoldani.

Első megközelítésben algoritmusokat írtunk Python környezetben, amelyek a segítségével egyrészt rácsatlakoztunk az élő adatfolyamra és a kód indításától a leállításáig gyűjti az adatokat. Ez az adatgyűjtés hasznos, ha egy eseményről akarunk adatot gyűjteni az adott esemény ideje alatt, viszont nagyobb elemzési lehetőséget biztosít az, ha archív adatokat elemzünk. Ezután feldolgoztuk az adatsorokat és csak a számunkra releváns adatokat tartottuk meg (pl. maga a bejegyzés, időbélyeg és a legfontosabb, a földrajzi hely). Miután az adatbányászat megtörtént, OpenRefine-ban meghatároztuk földrajzi koordinátájukat az adatoknak. Erre azért volt szükség, mert a legtöbb adat, melyet letöltöttünk csak utalást tartalmazott a helyre, pontos koordinátát nem.

Miután az adatokat számunkra elfogadható formára hoztuk, feltöltöttük azokat a tanszék által rendelkezésünkre bocsájtott adatbázisba. Itt tovább finomítottuk az adatokat, a hosszúsági és szélességi koordinátákat átalakítottuk, majd betöltöttük QGIS-be, ahol térinformatikai elemzéseket hajtottunk végre és térképen ábrázoltuk azokat. A térinformatikai elemzés mellett néhány statisztikai elemzést is végeztünk a tweetek (Twitteren közzétett bejegyzések) száma alapján.

Másik megoldásunk a problémára egy webalapú megközelítés, melynek alapja egy leegyszerűsített honlap, ami egy keresőből és egy térképből áll. Ennek segítségével vizsgáljuk az események időbeli lefolyását, a trendeket és naprakész adatokat vizsgálatát. A Twitter-en keresztül lekérdezve megjeleníthetjük a keresett címszavú találatokat Google térképen. A jelekre (markerekre) kattintva megtekinthetjük az üzenet tartalmát is.

Mindezek után összevetjük eredményeinket a két megoldásból, leírjuk a két megoldás előnyeit és hátrányait, valamint, hogy mikor melyiket célszerű alkalmazni. Végül kitérünk a Twitter jelentőségére a mai világban marketing és politikai szempontból is, valamint gazdasági szempontból is megvizsgáltuk a dolgozatot. Azt is megvizsgáltuk, hogy érdemes lenne-e felhő alapú szolgáltatást igénybe venni, milyen költségekkel járna az. Dolgozatunk végén pedig összegezzük eredményeinket és következtetéseinket.

1. Adatok beszerzése

Feladatunkban a Twitter oldaláról gyűjtöttünk adatokat. A Twitter egy olyan közösségi oldal, ahol a felhasználók leírhatják rövid gondolataikat (ún. tweeteket), kinyilváníthatják véleményüket, amolyan mikroblogként funkcionál. Ez az oldal viszonylag népszerű a világban, bár hazánkat nem érte el olyan szinten, mint más közösségi médiák (pl. Facebook, Instagram). Népszerűsége abból ered, hogy a felhasználók követhetnek híresebb embereket, intézményeket (színészeket, írókat, híroldalakat, stb.) és olvashatják azok bejegyzéseit, akár hozzá is szólhatnak.

Azért választottuk kutatásunkhoz ezt a médiát, mert egyrészt könnyen hozzáférhető volt a hivatalos fejlesztői (developer) API (Application Programming Interface, ezen keresztül tudunk csatlakozni a Twitter szerveréhez és onnan letölteni az adatokat), másrészt annak ellenére, hogy hazánkban nem örvend nagy népszerűségnek, a világ többi részén igen elterjedt és napi szinten használják.

1.1. Szükséges kulcsok beszerzése a szolgáltatótól

Ahhoz, hogy a közösségi oldalaktól adatokat tudjunk beszerezni, szükségünk van ún. API kulcsokra. Ezeket a kulcsokat úgy tudjuk beszerezni, ha az adott oldalra regisztrálunk, mint fejlesztő (természetesen itt nyilatkozni kellett, hogy ezt kutatásra használjuk, viszont vannak fizetős csomagok, melyeket cégek vehetnek igénybe és azzal üzleti célokra is használhatják az adatokat).

A Twitter esetében négy kulcsot kapunk, egy fogyasztói kulcsot (és ennek titkos verzióját), mely azt adja meg, hogy milyen csomagunk van, ugyanis az ingyenes csomaggal korlátozott az adatmennyiség, melyet le tudunk tölteni, valamint két belépési tokent.

Az általunk használt (ingyenes) kulcs nagy hátránya, hogy csak 7 napra visszamenőleg lehet letölteni tweeteket, viszont számbeli határt nem szabnak.

Miután a kulcsok megvannak, már kezdhetjük is letölteni a kívánt adatokat. Az alapos dokumentáció rengeteg segítséget nyújt számos felhasználási területhez.

1.2. Adatok letöltése Twitterről

Az adatok letöltését egy Python környezetben megírt kód segítségével végeztük. A kód megírásához roppant sok segítséget találtunk az interneten. Kétféleképpen lehet adatot letölteni: egyrészt úgy, hogy az élő adatfolyamba csatlakozunk és attól az időpillanattól kezdve, hogy elindítjuk a programot, egészen addig míg le nem állítjuk tölti le (az általunk

megadott feltételeknek eleget tevő) adatokat. Másik megközelítés, hogy visszamenőleg töltjük le az archív adatokat. A következőkben mindkettőre kitérünk részletesen.

1.2.1. Közvetlen adatnyerés – Stream listener

Ez a megoldás úgy működik, hogy a program becsatlakozik („hallgatózik”) az élő adatfolyamba és úgy tölti le a releváns tweeteket. Ez a művelet addig tart, amíg mi le nem állítjuk a programot. Ezzel kikerülhető az a probléma, hogy archív adatokat csak hét napra visszanyúlva lehet letölteni (az általános kulcsokkal), ám hátránya, hogy a programnak folyamatosan futnia kell. A kódhoz Mikael Brunila munkáját használtuk fel. [1]

A kód első része importálja a szükséges modulokat. A Pythonhoz már előttünk írtak egy olyan modult, mellyel végre lehet hajtani az adatok letöltését, ez a modul a Tweepy.

```
import tweepy
from tweepy.streaming import StreamListener
from tweepy import OAuthHandler
from tweepy import Stream
import json
```

A következő részben megadjuk az előző fejezetben leírt kulcsokat.

```
consumer_key = "consumer key"
consumer_secret = "consumer secret key"
access_token = "acces token"
access_secret = "acces secret token"

auth = OAuthHandler(consumer_key, consumer_secret)
auth.set_access_token(access_token, access_secret)

api = tweepy.API(auth)
```

Ezután megadjuk a fájlstruktúrát és kiterjesztést. Mi JSON (JavaScript Object Notation) kiterjesztést használtunk. A JSON egy kisméretű, szöveg alapú szabvány, ember által olvasható adatcserére. A JavaScript szkriptnyelvből alakult ki egyszerű adatstruktúrák és asszociatív tömbök reprezentálására (a JSON-ban objektum a nevük). A JavaScripttel való kapcsolata ellenére nyelvfüggetlen, több nyelvhez is van értelmezője. [2]

```
@classmethod
def parse(cls, api, raw):
    status = cls.first_parse(api, raw)
    setattr(status, 'json', json.dumps(raw))
    return status

tweepy.models.Status.first_parse = tweepy.models.Status.parse
tweepy.models.Status.parse = parse
```

Ebben a részben megadjuk, hogy a „StreamListener” modult használja a Tweepy-ből, valamint, hogy milyen néven mentse el.

```
class MyListener(StreamListener):

    def on_data(self, data):
        try:
            with open('twitter_nhl_rawdata.json', 'a') as f:
                f.write(data)
                return True
        except BaseException as e:
            print("Error on_data: %s" % str(e))
        return True

    def on_error(self, status):
        print(status)
        return True
```

Végül megadjuk azokat a kulcsszavakat, melyekre rá akarunk keresni. Példánkban jégkorongmérkőzésekre kerestünk rá.

```
twitter_stream = Stream(auth, MyListener())
twitter_stream.filter(track=['nhl', 'hockey', 'NY Rangers', 'Philadelphia
Flyers', 'Florida Panthers', 'Winnipeg Jets', 'Ottawa Senators',
                             'Chicago Blackhawks', 'Arizona Coyotes',
'Edmonton Oilers', 'Calgary Flames', 'San Hose Sharks', '#NHL'])
```

A kódot kb. 15 órán át futtattuk. A fenti lekérdezésben jégkorongmérkőzésekre kerestünk rá, és aznap futtattuk, amikor mérkőzés volt. Arra voltunk legfőképp kíváncsiak, hogy meg tudjuk-e határozni a tweetek alapján a stadionok helyzetét, valamint egyéb elemzéseket akartunk levezetni az adatokból. A leállítás után több mint 45 000 tweetet töltött le az alkalmazás. Ezeknek az adatoknak a további feldolgozását az 1.3. fejezetben írjuk le.

Az egyes nyers tweetek egyetlen sorban vannak, amik számos információkat tartalmaznak az adott bejegyzésről (dátum, felhasználó, mikor csatlakozott a Twitterhez, hány éves, helyzete (ha megadta), stb.). Egy sor (részlete) a következőképpen néz ki:


```
{"created_at":"Thu Sep 27 13:41:58 +0000
2018","id":1045307562973908993,"id_str":"1045307562973908993","text":"RT
@myrondueck: Well we may as well get this out now...I got body checked into the
boards the other night at my ice hockey game and fractur\u2026","source":"\u003ca
href=\"http:\/\/twitter.com\/download\/iphone\" rel=\"nofollow\"\u003eTwitter for
iPhone\u003c\/a\u003e","truncated":false,"in_reply_to_status_id":null,"in_reply_to_
status_id_str":null,"in_reply_to_user_id":null,"in_reply_to_user_id_str":null,"in_r
eply_to_screen_name":null,"user":{"id":823002078,"id_str":"823002078","name":...
```

1.2.2. Archív adatok letöltése

Az archív adatok letöltéséhez is Pythont alkalmaztunk és annak Tweepy modulját. Ehhez a kódhoz Bhaskar Karambelkar munkáját használtuk fel. [3]

```
import tweepy
auth = tweepy.AppAuthHandler("consumer key","secret consumer key")

api = tweepy.API(auth, wait_on_rate_limit=True,
                 wait_on_rate_limit_notify=True)

if (not api):
    print ("Can't Authenticate")
    sys.exit(-1)

import sys
import jsonpickle
import os
```

Ebben a részben adhatjuk meg a keresendő kulcsszót, a maximális tweetek számát, és azt, hogy mennyit töltsön le lekérdezésenként, valamint a fájl nevét és kiterjesztését.

```
searchQuery = '#championsleague'
maxTweets = 100000
tweetsPerQry = 500
fName = 'fname.json'
```

Itt lehetőségünk van megadni azt, hogy mikortól keressen meddig visszamenőleg (természetesen ennek gátat szabhat a felhasználói kulcsunk). Mi itt nem adtunk meg semmit, mivel az összes adatot akartuk az elmúlt hétből.

```
sinceId = None
max_id = -1L
tweetCount = 0
print("Downloading max {0} tweets".format(maxTweets))
with open(fName, 'w') as f:
    while tweetCount < maxTweets:
        try:
            if (max_id <= 0):
                if (not sinceId):
                    new_tweets = api.search(q=searchQuery,
count=tweetsPerQry)
                else:
                    new_tweets = api.search(q=searchQuery,
count=tweetsPerQry,
                                            since_id=sinceId)
            else:
                if (not sinceId):
                    new_tweets = api.search(q=searchQuery,
count=tweetsPerQry,
                                            max_id=str(max_id - 1))
                else:
                    new_tweets = api.search(q=searchQuery,
count=tweetsPerQry,
                                            max_id=str(max_id - 1),
                                            since_id=sinceId)
```

```
            if not new_tweets:
                print("No more tweets found")
                break
            for tweet in new_tweets:
                f.write(jsonpickle.encode(tweet._json, unpicklable=False) +
                        '\n')
            tweetCount += len(new_tweets)
            print("Downloaded {0} tweets".format(tweetCount))
            max_id = new_tweets[-1].id
        except tweepy.TweepError as e:
            # Just exit if any error
            print("some error : " + str(e))
            break
print ("Downloaded {0} tweets, Saved to {1}".format(tweetCount, fName))
```

Ebben a részben pedig a Bajnokok Ligájára kerestünk rá. Itt is az volt a célunk, hogy meghatározzuk a stadionok helyzetét, valamint egyéb térinformatikai és statisztikai elemzéseket készítsünk, melyeket a 2. fejezetben írunk le.

1.3. Adatok feldolgozása

Miután letöltöttük az adatokat, azokat fel kellett bontani, ugyanis a számunkra fontos információk mellett sok metaadat is letöltésre került. Ezek olyan információkat tartalmaztak, mint például, hogy mikor készítette a felhasználó a profilját, különböző engedélyezési adatok (pl. engedélyezt-e a GPS helyzetét, más felhasználók számára a láthatóságot, stb.).

Az első és legfontosabb feladat az volt a feldolgozásnál, hogy kihámozzuk azokat az adatokat, melyekre nekünk szükségünk van. Mivel rengeteg adatról van szó, elkerülhetetlen volt, hogy egy újabb kódot írjunk, mely kigyűjti ezeket. A legfontosabb az volt, hogy helyadatunk legyen valamilyen formában. A felhasználók többsége nem engedélyezte azt, hogy GPS koordinátákat adjon az eszköze a Twitter felé. Ez nem azt jelenti, hogy egyáltalán nincs adat a helyzetéről, mert voltak olyanok, akik megadták, hogy melyik városban tartózkodnak. Akik ezt sem adták meg, annak úgy adtunk helyadatot, hogy a megadott szülővárosa (vagy éppen ahol lakik), vagy országa helyzetét szedi ki, azzal az egyszerűsítéssel élve, hogy akkor onnan tweetel. Ha a fentiek közül egyik sem áll rendelkezésre, akkor eldobja a tweetet az algoritmus.

Természetesen ez az algoritmus nem tökéletes, ugyanis születtek olyan fals pozitívok, melyeknél ugyan a felhasználó adott meg helyzetet, az nem valós volt, hanem valami kitalált.

Egy feldolgozott adat a következőképpen néz ki:


```
{ "data": [
        {„user_tweet": "Calgary Flames @ San Jose Sharks 2018-09-27:\n\nSan
Jose Sharks: 60.3%\nCalgary Flames: 39.7% https://t.co/3Zcr06R10V",
            "post_time": "Thu Sep 27 13:41:58 +0000 2018",
            "user_id": 942828958541864960,
            "features": {
                "screen_name": "barloweanalytic",
                "location": "Virginia, USA",
                "tweets": 1,
                "geo_type": "User location",
                "primary_geo": "Virginia, USA",
                "id": 942828958541864960,
                "name": "Barlowe Analytics"
            }
        },
```


Az algoritmus megtartotta a JSON formátumot a könnyebb kezelhetőség érdekében. Ezután a további feldolgozást, a helyadatok meghatározását egy szabadon hozzáférhető szoftverben, az OpenRefine-ban végeztük.

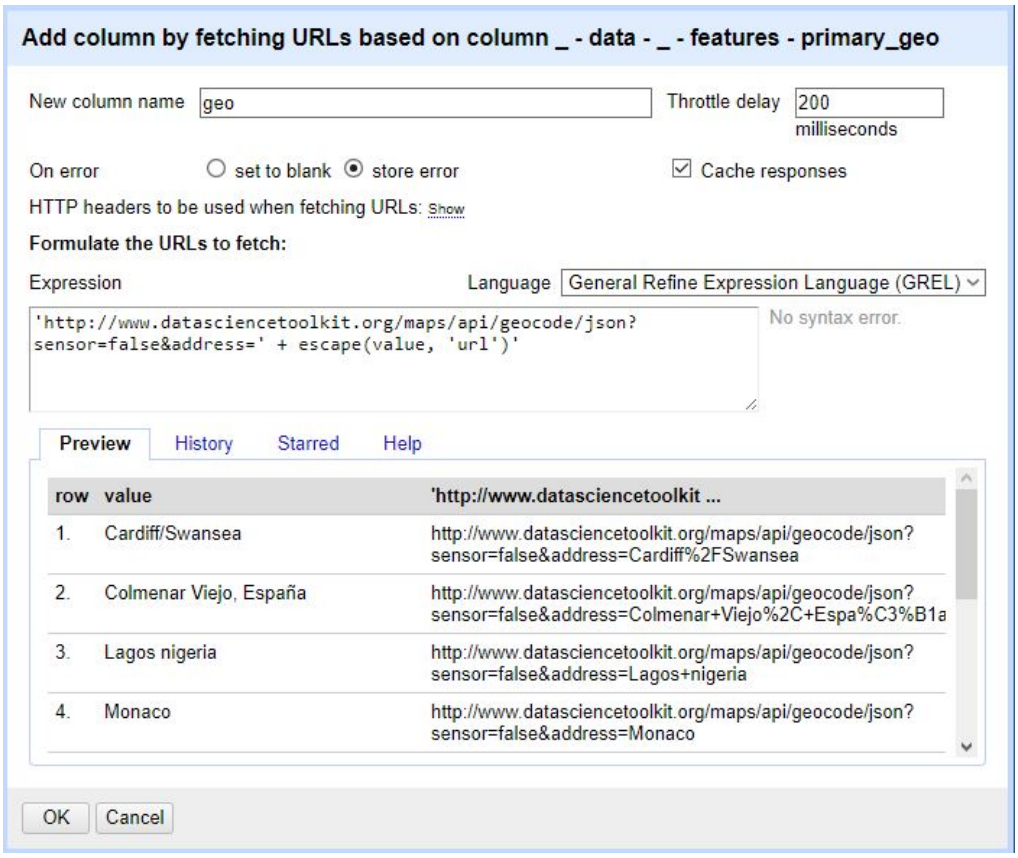
1.3.1. Helyadatok meghatározása, geokódolás

A geokódolás lényegében olyan helymeghatározás – például koordinátapárok, cím vagy egy helységnév – átalakításának folyamata, mely földfelszínen található helyhez köthető. [4] A mi esetünkben, ha a felhasználók nem adtak meg koordinátákat, akkor helyiségneveink vannak (utcák, városok, országok), amikből koordinátákat kell készítenünk.

Számos lehetőség van erre, a Google, Bing, Yahoo, valamint az OpenStreetMap (OSM) is kínál lehetőséget geokódolásra. Természetesen a Google, Bing, Yahoo azoknak egy része ingyenes, de más részük fizetős. Itt főleg az egyidejű lekérdezések számát szabják meg (pl. a Google napi 2 000 lekérdezést engedélyez), az OSM pedig, bár ingyenes, teljesítmény okokból korlátozza a lekérdezés sebességét, így nagy adat esetén ez viszonylag lassú.

Szerencsére találtunk egy olyan megoldást, mellyel hozzájuthatunk a koordinátákhoz relatíve gyorsan és ingyenesen, ez pedig a DataScienceToolKit. Ez egy ingyenesen hozzáférhető, nyílt forráskódú (open-source, OS) eszközökkel rendelkező oldal, mely segítséget nyújt, többek között a geokódoláshoz is. [5]

Miután betöltöttük az adatokat OpenRefine-ba és azt számunkra elfogadható formátumra rendeztük, létrehoztunk egy új oszlopot, mely a helyadatot tartalmazó oszlopra hivatkozik és egy URL segítségével geokódolja azt (1. ábra).



1. ábra: Geokódolás OpenRefine-ban

Miután a geokódolás befejeződött, sok olyan metaadatot kaptunk, melyekre nincs szükségünk a továbbiakban, így eldobtunk mindent, kivéve a hosszúsági és szélességi koordinátákat (2. ábra).

_ - data - _ - user_i	_ - data - _ - pos	_ - data - _ - user_tweet	_ - data - _ - featur	_ - data - _ - feat	_ - data - _ - feat	_ - data - _ - feat	lat	lng
1132690471	Mon Oct 08 15:28:11 +0000 2018	RT @Scoredrawretro: Shirt Showcase: Newcastle United Home 1998 Worn by The Entertainers including @alanshearer , Faustino Asprilla, Nobby…	1132690471	FootballHistoryBoys	TFHBs	Cardiff/Swansea	51.62079	-3.94323
1002093444	Mon Oct 08 15:28:01 +0000 2018	RT @podium_EE: ⚽ Tottenham Hotspur - FC Barcelona, siga en directo el partido de la Champions https://t.co/6c8DbBTI4I #ChampionsLeague http…	1002093444	Рафики	DavidHormigos	Colmenar Viejo, España	40.65909	-3.76762
254140618	Mon Oct 08 15:26:25 +0000 2018	RT @Heineken_NG: Blockbuster moves, heart stopping action and over the top gaffes. This week's matches had it all. Here are the world's mos…	254140618	Auwalu_4four	auwal44	Lagos nigeria	6.45306	3.39583
2193500617	Mon Oct 08 15:25:48 +0000 2018	#Barcellona, #Suarez potrebbe saltare l'#Inter in #ChampionsLeague per infortunio. le ultime https://t.co/v6LsWIFiVG	2193500617	Sportnotizie24	sportnotizie24	Monaco	48.13743	11.57549
228490863	Mon Oct 08 15:22:39 +0000 2018	RT @ultrassnl: Het uitvak bij Tottenham is uitverkocht. Er reizen 4200 PSV supporters af naar London! Tottenham Hotspur - PSV Eindhoven…	228490863	PSV Inside	psvinside	Noord-Brabant, Nederland	51.66667	5
957349752261234700	Mon Oct 08 15:19:24 +0000 2018	RT @NCN_it: EURORIVALI - VINCONO #PSG E STELLA ROSSA, PAREGGIA IL #LIVERPOOL CLICCA QUI - https://t.co/giT8wjtJtf #SscNapoli #Uci #Stella…	957349752261234700	PSG en Direct 🌐	PSG24hours	Parc des Princes	48.84145	2.25307
217980441	Mon Oct 08 15:18:40 +0000 2018	RT @CaraotaDeportes: Hoy #8Oct arrancamos 🎷 #DeportesAlGrano con: 📍Grand Slam histórico de @ronaldacunajr24 en postemporada #MLB 📍En #Champi…	217980441	María Isabel Moya	MarisabelMoya	La Guaira-Venezuela	10.59901	-66.9346
2533560792	Mon Oct 08 15:18:26 +0000 2018	EURORIVALI - VINCONO #PSG E STELLA ROSSA, PAREGGIA IL #LIVERPOOL CLICCA QUI - https://t.co/giT8wjtJtf #SscNapoli… https://t.co/J2IDyM5XSK	2533560792	Napolicalcionews.it	NCN_it	Napoli	40.83333	14.25
1344706963	Mon Oct 08 15:13:15 +0000 2018	Champions League: momento di crisi per Real e Bayern. Ce ne parla il nostro @LapoDeCarlo1 https://t.co/pQQTOSP0aN… https://t.co/dSZQn9wEHT	1344706963	SNAI	SNAI_Official	Italia	42.83333	12.83333

2. ábra: Geokódolt adatok

Sajnos sokszor azokban az esetekben is, mikor a felhasználó megadta a koordinátáit, a geokódolás hibás eredményhez vezetett. Ezt a problémát végül az adatbázisban oldottuk meg, ugyanis ott egy szelekcióval ki tudtuk szűrni azokat az adatsorokat, melyek eleve rendelkeztek koordinátákkal és a későbbiekben azokat használtuk a hibásak helyett.

1.4. Adatok feltöltése adatbázisba

Miután az adatokat számunkra megfelelő formára hoztuk és geokódoltuk azokat, már tölthettük is fel adatbázisba, melyre több lehetőségünk is volt. Egyrészt kimenthettük CSV-formátumba (Comma Separated Value; vesszővel elválaszott érték), vagy létrehozhattunk egy template-et (sablont) az adatok txt-be történő kimentésére. Mi az utóbbit választottuk, és egy olyan sablont hoztunk létre, mely SQL-ben fogalmazza meg azt, hogy hozzon létre egy új táblát, melyeknek megadjuk az oszlopait, majd azokat feltöltjük az adatokkal. Alapvetően két táblát hoztunk létre az adatokból, egyet a helyadatoknak (3. a.) ábra) és egyet a leíró adatoknak (3. b.) ábra).

	uid character (50)	lng character varying	lat character varying
1	942828958541864960	-77.44675	37.54812
2	257225771	-79.4163	43.70011
3	851490258470539264	10.43595	59.83734
4	96340068	-71.21454	46.81228
5	73299652	-114.08529	51.05011
6	707936378596827137	-75.69812	45.41117
7	316140178	-74.82877	40.15511
8	463352208	[null]	[null]
9	3549116836	-83.013402	40.100924
10	726729864	[null]	[null]
11	875091524383461376	-113.46871	53.55014
12	131876515	-73.58781	45.50884
13	38755662	-112.00693	57.66443
14	97504929	-87.658544	41.902042
15	110831311	100	60
16	28250031	-98.50063	38.50029
17	352770578	-79.4163	43.70011
18	175793258	-98.5	39.76
19	1031360900501004289	-96.790329	32.781179
20	1864400232	-75.69812	45.41117
21	2464773175	-57.55754	-38.00228
22	16190409	-86.556439	34.718428

a.) Helyadatok táblája

	misc_uid character (50)	time text	tweet text	loc text
1	9428289585418...	"Thu Sep 27 13...	"Calgary Flames @ San Jose Sharks 2...	"Virginia, USA"
2	257225771 ...	"Thu Sep 27 13...	"@NHL @NHLBruins @NHLFlyers Pla...	"Toronto, Ontario"
3	8514902584705...	"Thu Sep 27 13...	"Vanlig garderobe prat i hockey!"	"Asker, Norge"
4	96340068 ...	"Thu Sep 27 13...	"RT @yb2908: @thehockeyexpert @d...	"Québec, Canada"
5	73299652 ...	"Thu Sep 27 13...	"RT @AllALittleCrazy: If anyone hasn\...	"Calgary"
6	7079363785968...	"Thu Sep 27 13...	" It was a great experience, I played w...	"Ottawa, Ontario"
7	316140178 ...	"Thu Sep 27 13...	"@steffuhkneee I was talking about t...	"Levittown"
8	463352208 ...	"Thu Sep 27 13...	"RT @NHL: This is going to be a PP un...	"Twitterverse"
9	3549116836 ...	"Thu Sep 27 13...	"RT @FromTheFaceoff: 26 years ago, ...	"Columbus, OH"
10	726729864 ...	"Thu Sep 27 13...	"Holcombe face Buckingham test: We...	"Were Global ;-)"
11	8750915243834...	"Thu Sep 27 13...	"RT @THW_Oilers: Today marks the a...	"Edmonton, Alberta"
12	131876515 ...	"Thu Sep 27 13...	"RT @randersonahl: Its no secret, Joe ...	"Montréal, Québec"
13	38755662 ...	"Thu Sep 27 13...	"RT @jon_bois: why watch overtime p...	"Wood Buffalo, Alberta"
14	97504929 ...	"Thu Sep 27 13...	"HIJOLE D I O S M I O"	"chicago, il"
15	110831311 ...	"Thu Sep 27 13...	"Россию уважают! ЧМ-2023 пройдё...	"Россия"
16	28250031 ...	"Thu Sep 27 13...	"RT @jeannakadlec: THE CAPTAINS O...	"Kansas"
17	352770578 ...	"Thu Sep 27 13...	"Hockey twitter! I need your help. My ...	"toronto, ontario"
18	175793258 ...	"Thu Sep 27 13...	"@MarkDParrish torn rotator cuff is t...	"United States"
19	1031360900501...	"Thu Sep 27 13...	"@rjchoppy Heres my NFL* power ra...	"Dallas, TX"
20	1864400232 ...	"Thu Sep 27 13...	"RT @MikeBoucher86: Just scored Sto...	"Ottawa, Ontario"
21	2464773175 ...	"Thu Sep 27 13...	"Últimamente me están siguiendo cu...	"Mar del Plata, Argentina"
22	16190409 ...	"Thu Sep 27 13...	"Classic! @weloveuahhockey @UAHC...	"Huntsville, AL"

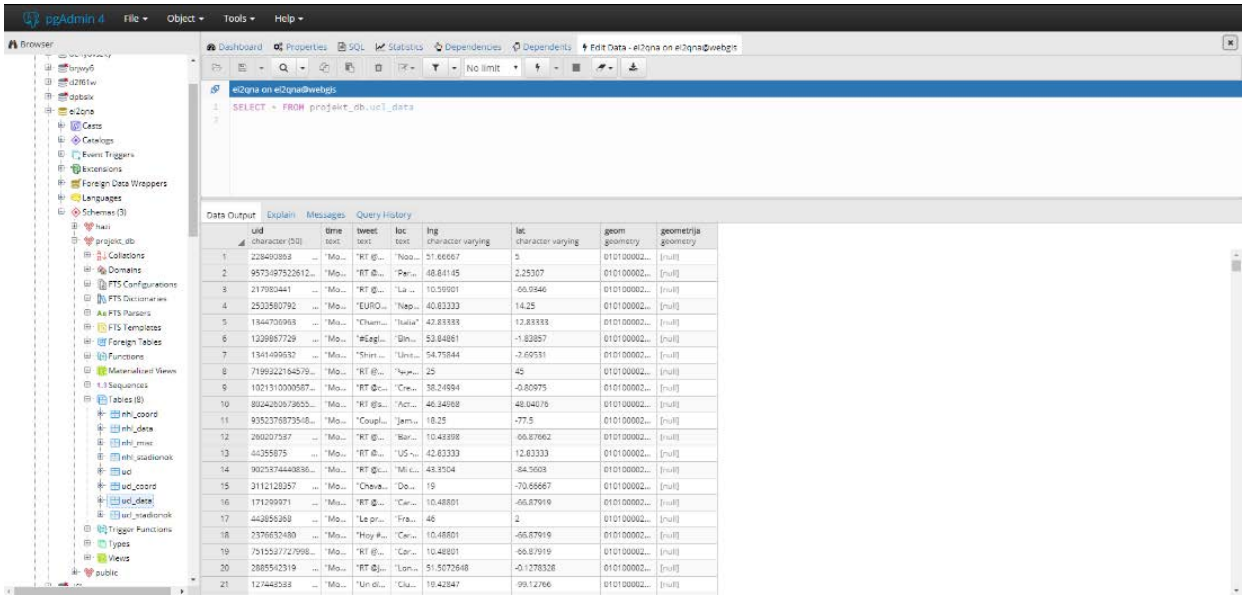
b.) Leíróadatok táblája

3. ábra: Adatbázisba feltöltött adatok

2. Adatbázisba feltöltött adatok kezelése, elemzése

2.1. Adatok térinformatikai szoftver által érthető formára hozása

Az adatok feldolgozásánál a legnagyobb problémát az adatok sokfélesége okozta. Ezt PostgreSQL függvényekkel SQL nyelven írt lekérdezésekkel/szkriptekkel igyekeztünk megoldani. A PostgreSQL egy open source relációsadatbázis-kezelő rendszer, ennek egy webes kezelőfelülete a pgAdmin. Első lépésben előkészítettük a már feltöltött adatbázist új oszlopokkal az újfajta adatokhoz. Az adatbázis kezeléséhez pgAdmin 3.0-át használtunk, az adatokat pedig a tanszéki webgis szerveren tároltuk. (4. ábra)



4. ábra: pgAdmin kezelői felület

Ehhez írtuk a következő SQL kódot:

```
ALTER TABLE projekt_db.ucl_data
ADD COLUMN geom geometry(Point,4326)
```

A geometriánál a point a geometria pont típusára utal, a 4326 pedig a pontok vetületi rendszerének EPSG kódja (WGS 84)

Ezután problémát okozott, hogy az adatbázisba importált adatok közt a koordináták nem "szám" típusú adatként lettek feltöltve, hanem "character varying" típussal, ami egy korlátlan hosszúságú szövegtípus. Ezt egyszerűen feloldottuk egy kasztolással (adat más típusra konvertálása), ahol ezt a stringet átkonvertáltuk tizedestört típussá.

Az előzőekben létrehozott új oszlopot a következő kóddal töltöttük fel:

```
UPDATE
  projekt_db.ucl_data
SET
  geom = (ST_Transform(ST_SetSRID(ST_MakePoint(CAST(ucl_data.lng AS DECIMAL), CAST(ucl_data.lat AS DECIMAL)), 4326), 4326))
```

Ennél a függvénynél is meg kellett adni a WGS84 vetületi rendszer EPSG kódját, ezután a PostgreSQL `ST_MakePoint()` függvényével alakítottuk át egy számítógép által jobban érthető formátumra. Ez a formátum egy ember számára olvashatatlan bináris kódot jelent, a geometria típusú adatokat a számítógép általában gyorsabban és pontosabban tudja kezelni. Könnyebben használja fel egyéb térinformatikai lekérdezéseknél. (5. ábra)

lng character varying	lat character varying	geom geometry
51.66667	5	0101000020E6100000000000000001440B79C4B7155D54940
48.84145	2.25307	0101000020E61000002ECA6C90490602400F9C33A2B46B4840
10.59901	-66.9346	0101000020E6100000DDB5847CD0BB50C034F44F70B1322540
40.83333	14.25	0101000020E6100000000000000802C404963B48EAA6A4440
42.83333	12.83333	0101000020E6100000268DD13AAAAA29404963B48EAA6A4540

5. ábra Példa a geometria típusú adatra

A másik jelentkező probléma, hogy azokon a helyeken, ahol a felhasználók a készülék GPS koordinátáit használták a tweethez, azt az OpenRefine geokódolása nem tudta rendesen feldolgozni. Ezt a problémát is egyszerűen egy SQL lekérdezéssel tudtuk feloldani:

```
SELECT
  uid,
 SUBSTRING(loccoord, 1, STRPOS(loccoord, ',')-1) as lat,
  SUBSTRING(
    loccoord,
    STRPOS(loccoord, ',')+1,
    LENGHT(loccoord)
  ) as lon
FROM
  projekt_db.ucl_coord
WHERE
  loctype = '"Tweet coordinates"',
  ucl_data.uid = ucl_coord.uid
```

Mivel a nyers adatok között a pontos koordináták vesszővel voltak elválasztva, könnyedén szét lehet őket válogatni a vessző pozíciója alapján. A vessző pozíciójának számszerű megtalálásához az `STRPOS()` függvényre volt szükség. Ez egy stringen belül megmondja az általunk keresett karakter helyét. Ehhez képest már egyszerű volt két részre bontani az egészet, ha tudtuk a teljes hosszát (`LENGHT()` függvénnyel ez is lehetséges). (6. ábra) A két táblát egyszerűen össze tudtuk kapcsolni a felhasználói azonosítók alapján (uid), illetve a pontos koordinátákat is egyszerűen ki tudtuk szűrni a nyers adatokból a geometria típusa alapján. (‘Tweet coordinates’ érték a loctype oszlopban)

	loccoord text	lat text	lon text
1	"40.45297246, -3.68935891"	"40.45297246	-3.68935891"
2	"41.38061977, 2.12152231"	"41.38061977	2.12152231"
3	"51.52074461, -0.07335546"	"51.52074461	-0.07335546"
4	"38.4518635, 27.20301495"	"38.4518635	27.20301495"
5	"51.55696202, -0.28002518"	"51.55696202	-0.28002518"
6	"45.0702, 7.6777"	"45.0702	7.6777"
7	"36.175, -115.136"	"36.175	-115.136"
8	"40.82805501, 14.19310466"	"40.82805501	14.19310466"
9	"6.9179, 79.863"	"6.9179	79.863"

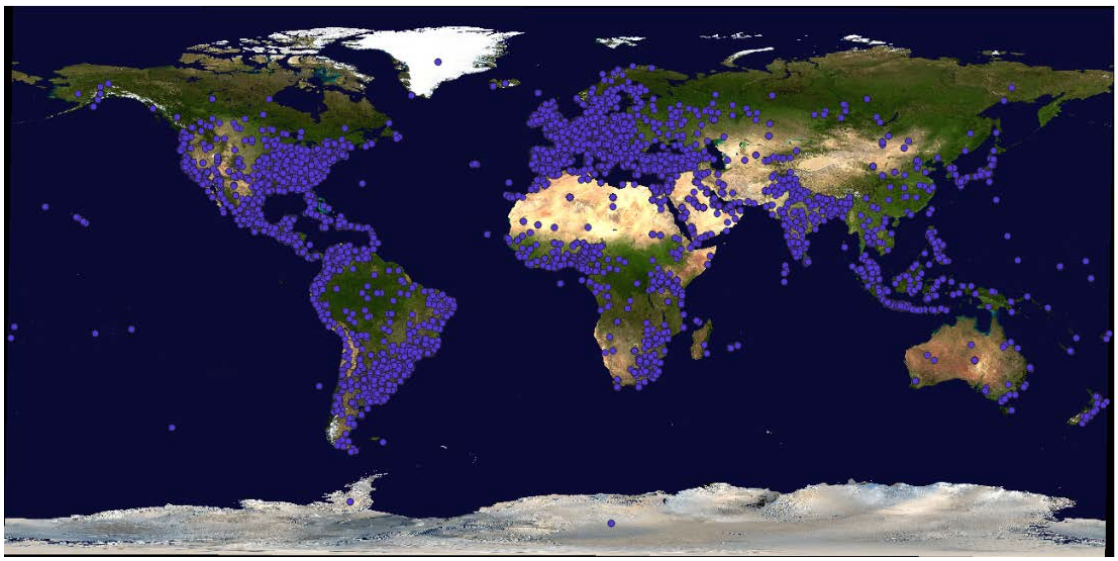
6. ábra Szétválasztott adatok

Ezek után már csak a pontok geometriává alakításánál kellett megváltoztatni a kódot, hogy a geokódolt vagy a nyersanyagból használt pontos koordinátát használja, amennyiben létezik ilyen. (7. ábra)

```
CREATE TABLE projekt_db.ucl
AS SELECT *, CASE
WHEN projekt_db.ucl_coord.loctype='"Tweet coordinates"'
```

```
THEN
ST_SetSRID(ST_MakePoint(SUBSTRING(loc FROM 2 FOR POSITION(',' IN
loc)-2)::decimal, SUBSTRING(loc FROM POSITION(',' IN loc)+2 FOR
LENGTH(loc)-POSITION(',' IN loc)-2)::decimal), 4326)
ELSE
geom
END vegleges
FROM projekt_db.ucl_coord
RIGHT JOIN projekt_db.ucl_data
USING(uid)
```

Az utóbbi megoldásra csak a labdarúgásos adatbázisnál volt lehetőségünk, mivel a jégkorongos adatbázisban a Tweet coordinates értékek nem mindig számot tartalmaztak. Ezért annál csak az eredeti geometriává alakító szkriptet futtatuk le.

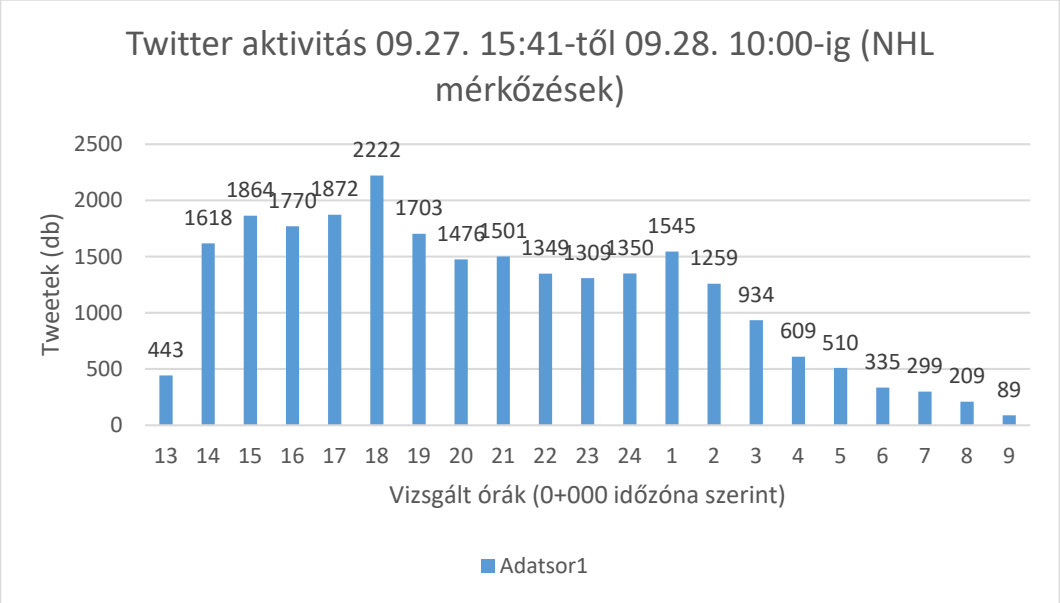


7. ábra A pontok ábrázolva QGIS-ben

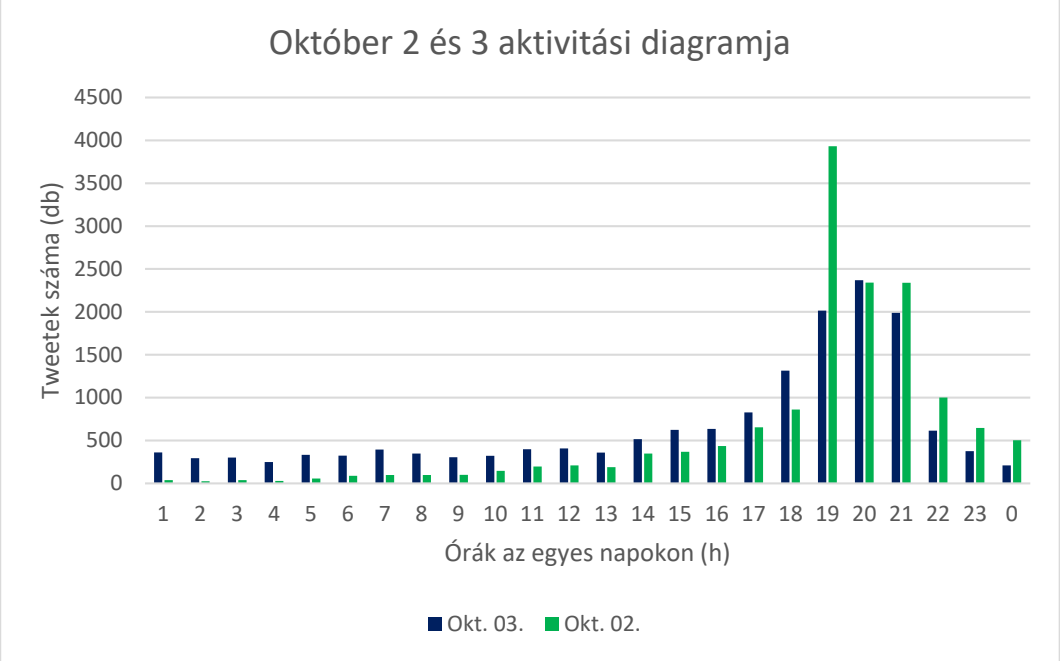
2.2. Adatok térinformatikai és statisztikai elemzése

A térinformatikai elemzéseket QGIS környezetben végeztük, mely egy szabadon hozzáférhető, nyílt forráskódú szoftver. Az adatokkal, amiket adatbázisba töltöttünk fel, sok elemzést végre lehet hajtani, ezekből párat mutatunk be a következőkben. A térinformatikai elemzések mellett néhány statisztikai elemzést is végeztünk, amiket Excellel készítettünk.

Az egyik elemzés, amit elkészítettünk, arra irányult, hogy meghatározzuk, mikor aktívak a felhasználók az adott témában, pusztán az adatok számszerűségéből meghatározható-e az esemény ideje, időtartama. Először Excelben megvizsgáltuk a tweetek száma alapján, hogy mely időben hányan írtak az adott témában (8. ábra).



8. a. ábra: Twitter aktivitás a vizsgált időtartamban



8.b. ábra: Bajnokok ligája mérkőzések (okt. 2-3.) Twitter aktivitási diagramja

A 8.b. ábrán látszik, hogy október 2-án többen tweeteltek a mérkőzés alatt, így kíváncsiak voltunk arra, hogy melyik csapat vonzotta az embereket ennyire, minek köszönhető ez a kiugró aktivitás. Ezt nehéz kiszűrni pusztán az adatok alapján, egyik lehetőségünk, hogy rákeresünk a tweetekben az egyes csapatok neveire és az alapján próbálunk következtetni. Ebben viszont benne van az a hiba lehetőség, hogy az emberek máshogy nevezik meg a csapatot, mint ahogy mi rákeresünk (viszont szerencsére sokan több megnevezést is leírnak a tweetjükben). Egy ilyen lekérdezésre mutatunk példát a következőkben:

```
SELECT
tweet,
time
FROM projekt_db.ucl
WHERE ucl.tweet LIKE '"%Barcelona%"' AND ucl.time LIKE '"%Oct 02 19%"'
```

A legtöbben a lekérdezések alapján a Real Madridot említették tweetjükben (kb. 400-an), ezután a Barcelonát és a Manchester City-t. A Real Madrid esetében lehetséges, hogy azért tweeteltek sokan arról a meccsről, mert kiosztottak egy piroslapot a CSZK Moszkva egyik játékosának.

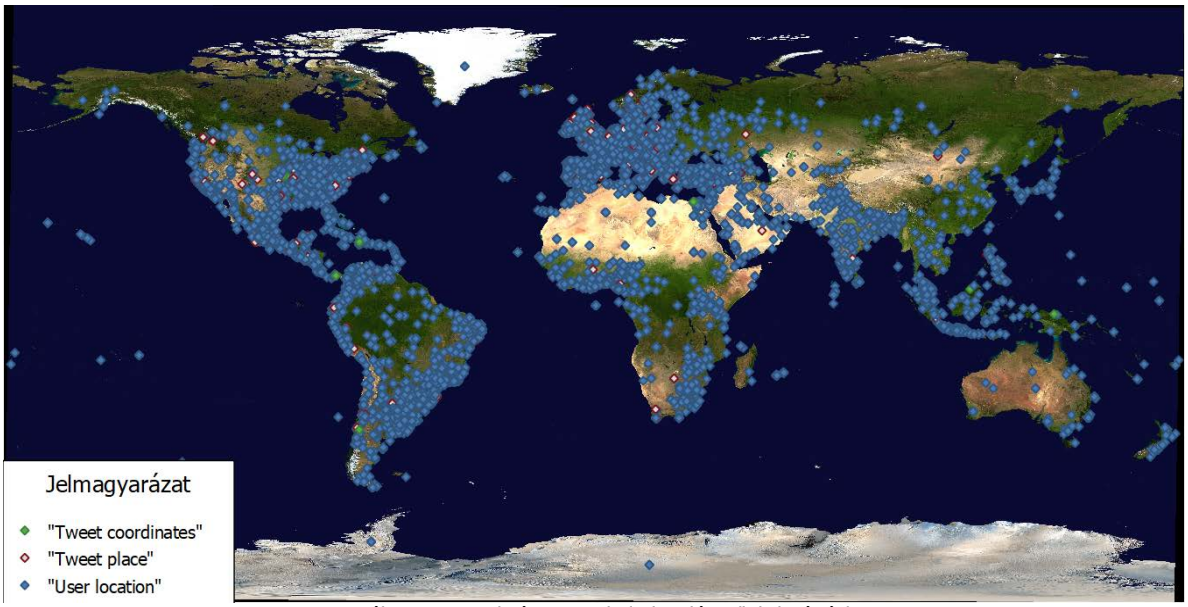
A következőben a geometriai adatok típusait vizsgáltuk, melyik fajtából mennyi került az adatbázisba. Számszerűen ezt is egy egyszerű SQL lekérdezéssel kaptuk meg, de mellé készítettünk egy térképi megjelenítést is.

```
SELECT COUNT(*), loctype
FROM projekt_db.ucl
GROUP BY loctype
```

Ezen lekérdezés eredménye:

Tweetek (db)	Adat típusa
171	"Tweet coordinates"
1105	"Tweet place"
36880	"User location"

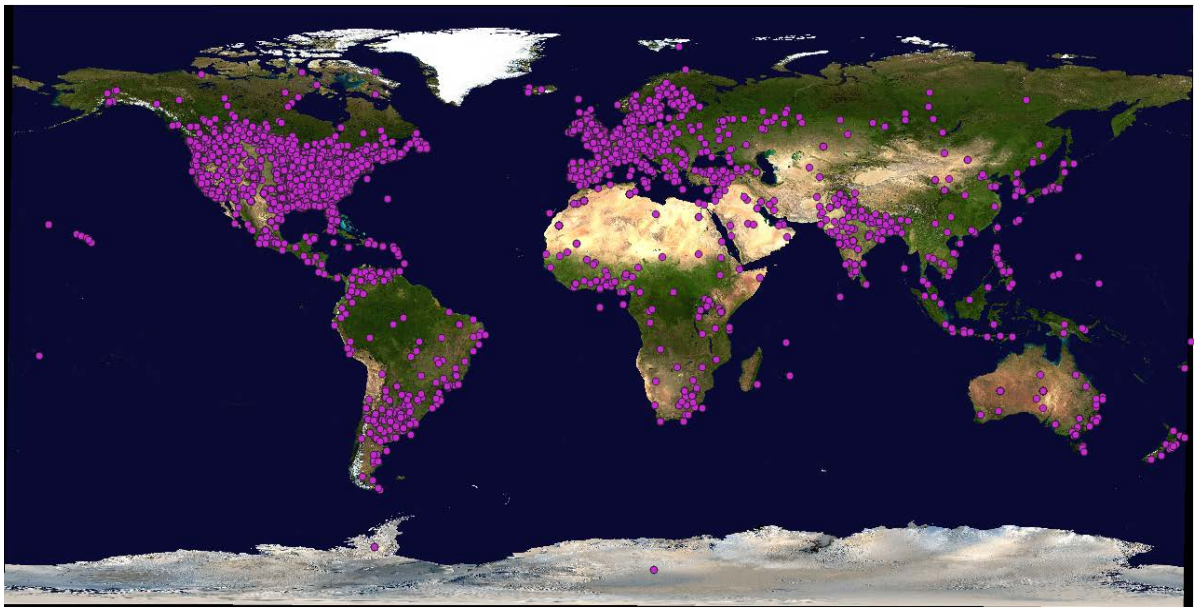
Az adatok típusánál három lehetséges opció van. Egyrészt a „Tweet Coordinates”, ami lényegében az, hogy a felhasználó engedélyezte az eszközén azt, hogy a Twitter lekérdezhesse GPS helyzetét. A másik kettő, a „Tweet place” és a „User location” pedig a felhasználó által megadott helyzetet jelöli, ami sok minden lehet, gyakran fiktív hely is, mivel ebben az esetben a felhasználó maga írja le, hogy hol van. A kettő között az a különbség, hogy a „Tweet place” magára a tweet-re vonatkozik, hogy az hol készült, a „User location”, pedig arra, hogy a felhasználó hol tartózkodik éppen. Számunkra a legfontosabbak azok, melyek a valós koordinátát tartalmazza. (10 ábra)



10. ábra: A pontok típus szerinti eloszlása (labdarúgás)

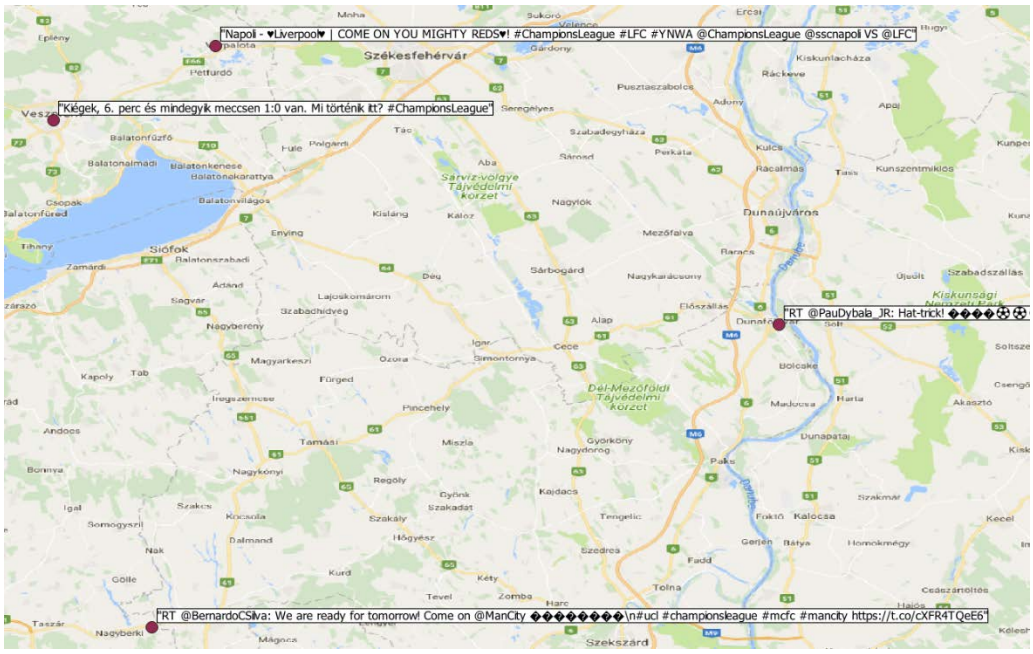
Ebből a térképből azt is megmondhatjuk mely helyeken népszerű a twitter használata.

Ugyanilyen megjelenítést készíthetünk a jégkorongos adatbázisról is. (11. ábra)



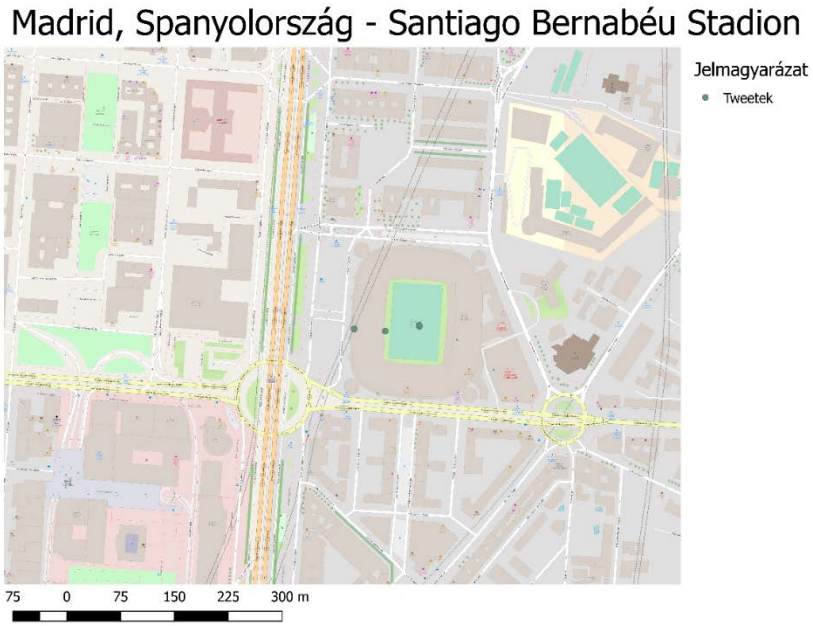
11. ábra: A pontok típus szerinti eloszlása (jégkorong)

De készíthetünk akár olyan térképet is, amin az adott helyről küldött tweeteket olvashatjuk. Az alábbi térképen egy Google Maps alapú térképet láthatunk, rajta a küldött tweetekkel. (12. ábra)



12. ábra: Tweetek a térképen

Megvizsgáltuk azt is, hogy Twitter alapján megtudjuk-e azt állapítani, hogy hol helyezkednek el az egyes sportlétesítmények, melyekben a mérkőzések zajlottak. Ebben a kérdésben eredményt a jégkorong esetében nem kaptunk, ami annak köszönhető, hogy csak nagyon kevesen engedélyezték GPS helyzetük elérését a Twitteren. Ellenben a Bajnokok Ligája esetében látványos eredményeket sikerült elérnünk, ugyanis többen is engedélyezték az elérést, ráadásul olyanok is, akik éppen a stadionban tartózkodtak (13. ábra).



13. ábra: A Santiago Bernabéu Stadion és az onnan beérkezett tweetek helyzete

3. Elemzés végrehajtása webes felület létrehozásával

A feladat megoldását egy másik oldalról is megközelítettük. Ehhez egy internetes oldalt hoztunk létre, amely közvetlen módon a Twitter hivatalos API-ján keresztül lekérdezi az adatokat, majd megjeleníti nekünk egy Google térképen a keresett címszavainkat. A végeredmény lényegében azonos akart lenni az előző fejezetekben tárgyaltakkal, azonban mire a végére jutottunk, a technikai okok és megoldási lehetőségek miatt enyhén más eredmények keletkeztek.

Az eredeti terv szerint Wordpress-ben szerettük volna megalkotni a honlapot, amit azonban végül megváltoztattunk és más fejlesztői környezetben oldottunk meg. Ennek fő oka az volt, hogy a Wordpress-ben bizonyos programozási nyelveket nem vagyunk képesek használni, mint például a C#, azonban mivel ebben van a legnagyobb tapasztalatunk, így egyszerűségi okok miatt ezt választottuk.

3.1. Visual Studio

Több programozási nyelvet kellett használni a feladatok megoldhatósága és komplexitása miatt, így C#, HTML és Javascript került a kódba. Választásunk azért esett ezekre az eszközökre, mivel ezekben van a legtöbb tudásunk és gyakorlatunk, nyilván más nyelvekben egyaránt megírható lenne a honlapunk. Ahhoz, hogy a programozási nyelvek közötti átjárhatóságát biztosítsuk, a Microsoft Visual Studio-t, mint fejlesztői környzet és az ASP .NET Core 2.1 framework-öt alkalmaztuk.

3.1.1 HTML

Bizonyos részeit a honlapnak HTML-ben kellett megírni, mivel csak ez a nyelv teszi lehetővé a webfelület létrehozását. Valójában ez csak a honlap kinézetéért felelős. Az egyes részek, amelyek ebbe belekerültek, a kereső és térkép elhelyezése és a letöltött adatoknak a mennyiségét jelző funkciója.

3.1.2 Javascript

Ez maga a webes felületeknek a programozási nyelve, ami mindenféle objektumot tud kezelni. Így a legtöbb függvény ebben lett megírva vagy Javascript-té lett átalakítva, hogy a honlapon megjeleníthető és értelmezhető legyen.

3.1.3 C#

A C# egy objektum orientált programozási nyelv, amelyet azért használtunk, mivel legtöbb gyakorlatunk ebben van. Ebben a nyelvben a back-end-ért felelős részek lettek megírva. Azokat a részeket, melyeket meg kellett jelenítenünk a honlapon, kénytelenek voltunk a

későbbiekben összekötni Javascripttel is értelmezhető kódokkal, s ezek stringek által voltak megoldhatók.

3.2. A honlap

A honlap létrehozásánál az egyszerűségre törekedtünk. Lényegében egyetlenegy kereső és egy térkép található rajta. A keresőbe beírva a címszavunkat, kidobja nekünk piros markerekkel, hogy honnan lettek az adott tweetek elküldve. Ezekre a markerekre kattintva átdob minket a hivatalos Twitter oldalára, ahol meg is nézhetjük az adott tweetet.

A honlap létrehozásánál HTML-t alkalmaztunk, amely jelenlegi állapota szerint egy roppant egyszerű kód. A térkép és a kereső elrendezését a következőkben adtuk meg.

```
<!DOCTYPE html>
<html>
<head>
    <title>Geolocation</title>
    <meta name="viewport" content="initial-scale=1.0, user-scalable=no">
    <meta charset="utf-8">
    <style>
        #map { height: 100%}
        html, body {
            height: 100%;
            margin: 0;
            padding: 5%; }
    </style>
</head>
```

Ehhez a megoldáshoz a Google Maps hivatalos API-jának leírását használtuk fel. [6]

3.2.1 Google Maps API

A honlapunk elkészítéséhez a Google által létesített online térképet használjuk. Ennek a csatlakoztatását ugyanúgy oldottuk meg, ahogyan a Twitter esetén, azaz a hivatalos API-n keresztül. Ehhez a Cloud Google platformon belül kell regisztrálni, majd azonnal kérhető egy felhasználói kulcs. A mi feladatunk megoldásához a Maps JavaScript felhasználói kulcsot használtuk.

3.2.2 Twitter szerveréhez való csatlakozás és adatok lekérdezése

Az életünk megkönnyebbítése miatt egy Twitterizer2 lib-et használtunk. Ezek a lib-ek magányszemélyek által írt kódcsomagok. Ez abban segített, hogy az egyes URL-eket és egyebeket ne nekünk kelljen beírni a kódunkba, majd mindenre hivatkozni, hanem előre megírt függvényekként tudjuk őket használni.

Magát a Twitterhez való csatlakozást hasonlóan a 1.2 fejezetben tárgyaltak alapján lehet elképzelni. Ugyanazok a belépési kódok lettek felhasználva. Ebben a fejezetben azonban C#-ban lettek megírva az autentikációs belépések és a csatlakozások, bár máshogy néz ki a kód, ennek ellenére ugyanazt a célt szolgálja, mint a pythonos verzió. Míg maga a hitelesítés

azonosan néz ki, utána eltérő lesz maga a válaszként kapott adatok felhasználása és továbbítása. Lépésekre bontva az alábbi folyamatok történnek:

1. A keresőre nyomva elküldjük a szerver felé azt az információt, hogy csatlakozni akarunk és adatokat lekérni

```
[HttpGet]
        public IActionResult Search(string searchterm)
        {
            return View("Add", GetGeoTweets(searchterm));
        }
```

Ez a lépés lényegében az alap-csatlakozás és válaszra várás a Twitter szerveréről. Ha az adatokat egyből egy adatbázisba szeretnénk menteni, akkor egy HttpPost funkcióval ezt is megtehetnénk. Ebben az esetben a funkció bele lett írva a kódunkba, ha későbbiekben alkalmazni szeretnénk, viszont nem lett használva, mivel ezeket az előző fejezetekben tárgyaltuk.

2. Ezek után hitelesítjük magunkat az auth. kódokkal, hogy megfelelő hozzáférésünk van az információk lekérdezéséhez.

```
private string GetGeoTweets(string searchterm)
        {
            OAuthTokens tokens = new OAuthTokens();

            tokens.ConsumerKey = "4sOHHUCiWK2CTSYj4VA17nxBQ";
            tokens.ConsumerSecret =
"w5xxizs4qB0PLDzuQSSt6a0Dc3m83hsgcncYu9booXKnKUpumh";
            tokens.AccessToken = "4643374648-
V4QZGrTYjyfaXVpG9lLF8ezUJo56YdeXBPHWwxj";
            tokens.AccessTokenSecret                                    =
"tL8hcO1QpL28LEL1A8cWr6lzTQtXoG3KmXOm2UrsFoCYT";
```

3. Megadjuk a szerver felé, az egyes specifikációkat, hogy milyen válaszokat akarunk kapni. (pl. válaszok száma,…)

```
TwitterResponse<TwitterSearchResultCollection> result = TwitterSearch.Search(tokens,
searchterm, new SearchOptions()
            {
            Count = 1000
         });
```

4. A kapott válaszok értelmezése és rendezése a későbbi felhasználásra.

```
List<TwitterStatus> geos = result.ResponseObject.Where(x => x.Geo != null).ToList();

List<TwitterStatus> nogeos = result.ResponseObject.Where(x => x.Geo == null).ToList();
```

Itt a 4. lépésben külön szedtük két listára a válaszokat. Lényegében az egyik lista, amelyet létrehoztunk, megadja az összes választ, amelyet kapunk a szervertől, de nem rendelkezik geokódolt adatokkal és egy másik listára, amelyben csak a geokódolt válaszokat listázzuk ki.

Ezek lesznek a későbbiekben azok az eredmények, amelyek megjeleníthetők lesznek a térképen.

3.2.3. A markerek helyének lekérdezése majd megjelenítése

A továbbiakban a következő nehézséget kellett még feloldani, mielőtt megjeleníthetnénk a már kész koordinátákkal rendelkező pontjainkat. Ez pedig az, hogy a C#-os nyelvünkből egy Javascript által is értelmezhető bemenetet adjunk. Egy ilyen formátum, amit mind a két nyelv egységesítve tud értelmezni, az a string. Az átalakítás során egy JSON struktúra jött lére, amit már a honlapunkon megjeleníthetünk. Ez a következő képen néz ki:

```
string markers = "[";

            foreach(TwitterStatus ts in geos)
            {
                double lng = ts.Geo.Coordinates.FirstOrDefault().Longitude;
                double ltd = ts.Geo.Coordinates.FirstOrDefault().Latitude;
                   string url = string.Format(@"http://twitter.com/{0}/status/{1}",
   ts.User, ts.Id);

                markers += "{";
                   markers += string.Format(CultureInfo.InvariantCulture, "'title':
   '{0}',", ts.Source);
                   markers += string.Format(CultureInfo.InvariantCulture, "'lat':
   '{0}',", ltd);
                   markers += string.Format(CultureInfo.InvariantCulture, "'lng':
   '{0}',", lng);
                   markers += string.Format(CultureInfo.InvariantCulture,
   "'description': '{0}'", url);
                markers += "},";
            }
```

A következő részben az átadott JSON struktúrát megjelenítjük a térképen markerekkel és hivatkozást társítunk hozzájuk, ami kattintásra jön elő:

```
for (i = 0; i < markers.length; i++) {
                var data = markers[i]
                var myLatlng = new google.maps.LatLng(data.lat, data.lng);
                var marker = new google.maps.Marker({
                    map: map,
                    draggable: true,
                    title: data.title,
                    animation: google.maps.Animation.DROP,
                    position: myLatlng
                });
                marker.setMap(map);
                (function (marker, data) {
                    google.maps.event.addListener(marker, "click", function (e) {
```

```
                var win = window.open(data.description);
                win.focus();
            });
        })(marker, data);
```

Fontos tudnivalók: A használt Twitter API-kódunk sajnálatosan egyszerre csak 100 darab tweetnek a lekérdezését engedélyezi és csak 7 napra visszamenőleg. A szervertől gyakran hasonló válaszokat fogunk kapni, ha napi intervallumban akarjuk megnézni a változásokat. A száz darab lekérdezésből mindig 90% feletti arányban kapunk nem geokódolt adatokat, amelyek ideiglenesen megtelítik az adott lekérdezésünk limitjét, ezért kevés megjelenített pontunk lesz.

Ehhez a fejezethez az asp snippets weboldal fórumán található segítséget használtuk fel. [7]

3.2.4 Adatforgalom számláló

Beépítettünk a rendszerbe érdekességnek egy adatszámlálót, mellyel követhetjük, hogy az egyes lekérdezésünk során mekkora mennyiségű adatforgalmunk volt kilobyte-ban. Ezt a következőképpen oldottuk meg:

```
float length = (result.Content.ToString().Length * 2)/1024;
string ret = "this response's size is: " + length + "KB";
return ret;
```

Valójában a lekérdezett adatunk stringjének hosszát vizsgáljuk meg. Ez magába foglalja az összes általunk befogadott adatot, amit 2-vel megszorzunk, mivel egy karakternek 2 byte a nagysága, majd elosztottuk 1024-el, hogy kilobyte-ban kapjuk meg.

3.3 A késztermék és lekérdezett adatok megjelenítése

Végeredményben létrejött a honlapunk, amely a keresőből és a térképünkből áll. Egy lekérdezés folyamán megmutatjuk, hogyan néz ki az eredményünk. Az alábbi keresés a 'scubadive' címszóra lett keresve. Így megnézhetjük, hogy az elmúlt napokban, hol búvárkodtak az emberek és milyen helyekről osztották ezt meg. (14. Ábra)

14. ábra. Web alapú keresésnek a megjelenítése

3.4 A jövőbeli fejlesztések

Későbbiekben tovább fejleszthető az oldalunk és további érdekes megoldásokkal állhatunk elő. A webes felületünkhöz akár hozzácsatolhatnánk az eddig általunk létrehozott adatbázist is, hogy több adathoz juthassunk; nyilván, ha nem szeretnénk egy szervert üzemeltetni, akkor egy prémium Twitter előfizetéssel hozzáférhetnénk az összes kívánt adathoz.

Mik lehetnének ezek a kívánt adatok? Trendek követése, hashteg-ek (kettőskeresztek), bizonyos dátumok alapján való keresés.

Ezekhez a jövőben egy sajátos kezelőfelületet (User Interface-t) is lehetne adni, ahol az ember egyszerű konzolokon például kiválaszthatná, hogy milyen időintervallumban akar keresni, netán térségen belüli változásokat akar lekövetni és bármi egyéb ésszerű opciókat felsorolni. Ezentúl fontos lenne más közösségi oldalak bevonása is. Az egyik legnagyobb nehézséget a kevés geokódolt adat jelentette, így az Instagram bevonása - ahol az emberek többsége pontos koordinátákat a Twitter-nél gyakrabban megadja - rendkívül javítaná az oldal működését.

4. Módszerek összehasonlítása, eredmények összegzése

Az előzőekben leírt módszereket közvetlenül összehasonlítani nem szerencsés, ugyanis a kettő funkciójában eltér. A következőkben összegezni fogjuk előnyeiket és hátrányaikat, majd levonjuk a következtetéseket, hogy mikor melyik megoldást jobb alkalmazni.

4.1. Adatbázis segítségével történő térinformatikai kiértékelés

Ennek a módszernek a lényege, hogy letöltünk minden olyan releváns adatot, mely megfelel az általunk támasztott kritériumoknak. A letöltés után az adatokat (miután kívánt formátumra hoztuk) adatbázisba rendszerezzük és utána kezdődik a kiértékelés különböző térinformatikai rendszerek segítségével.

A módszer nagy előnye, hogy túlzott programozási képességeket nem igényel, ugyanis az interneten számos dokumentációt találtunk, melyeket mozaikszerűen összeillesztve különösebb nehézségek nélkül megkaptuk a végeredményt. További előnye, hogy időben régebbi adatokat is szerezhetünk (annak függvényében, hogy milyen kulcsunk van), valamint az adatok megjelenítésében és elemzési módjában is nagyobb szabadságunk van, valamint mivel ezek egy harmadik szoftverben történtek (Qgis), így programozási tudást nem igényelt, egyedül a szoftver ismerete volt szükséges hozzá.

A módszer egyik legnagyobb hátránya, hogy szükség van egy aktív szerverre, melyen az adatbázis futhat. Ennek a megvalósítására több megoldás is létezik, pl. bérelhetünk SQL szervert online felhőben vagy vehetünk egyet (ezt részletesebben a 6.2. fejezetben mutatjuk be). Másik hátránya a módszernek, hogy tárolnunk kell az adatokat.

Ezt a módszert akkor javasoljuk, ha egyrészt van lehetőségünk adatbázis bérlésére vagy kiépítésére, másrészt akkor, hogy ha komolyabb elemzési műveleteket akarunk végrehajtani valamilyen térinformatikai szoftverben.

4.2. Lekérdezés és megjelenítés webes felületen keresztül

Lényegében ez a módszer próbálja kihagyni azt a lépést az előzőhöz képest, hogy egy adatbázist létrehozzunk, amiben az adatokat tároljuk és direkt módon a Twitter adatbázisához csatlakozva lekérdezzük a kívánt információkat, majd megjelenítsük őket.

A módszer előnye, hogy direkt módon megkapjuk az adatokat a Twitter adatbázisából. Nem kell külön üzemeltetnünk egy szervert. Maga a végtermék használata egyszerű és értelmezhető. Bármilyen felhasználó képes használni, nem kell hozzá semmilyen szakmabeli tudás. Könnyen készíthetők tematikus térképek, valós időben nézhetjük az emberek

gondolatait. Tovább fejleszthető a honlap bármilyen egyedi igényre szabva, további keresési lehetőségekkel.

Hátránya, hogy az autentikációs kulcstól nagyban függ az adatok beszerzésének lehetősége. Így például az ingyenes licensszel kevés találat van keresésekként és időben is órák telhetnek el, mire új, számunkra releváns adathoz jutunk.

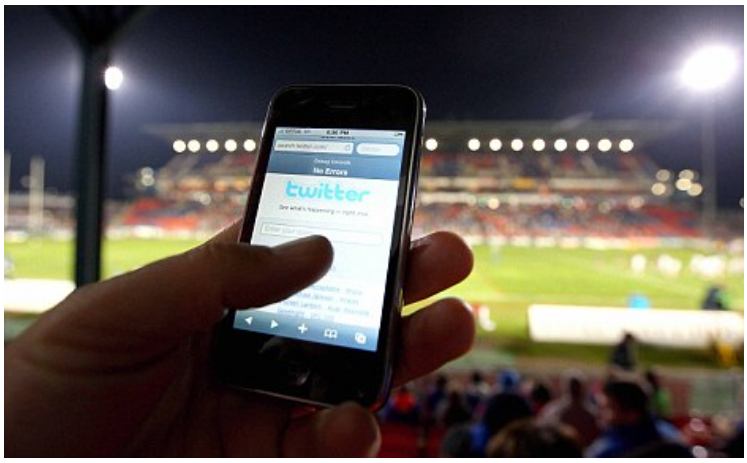
Ezt a módszert akkor javasoljuk, ha rendelkezünk a kellő programozási háttérrel, valamint akkor, ha nincs keretünk egy adatbázis bérlésére, fenntartására. Ez egy látványos reprezentatív módot biztosít, melyet jól ki lehet használni, de térinformatikai elemzéseket nehéz benne végezni.

5. Kinyert adatok felhasználása

A különböző oldalak, mint a Google, Facebook, Twitter, stb. egyik legfőbb bevételi forrása a reklámok mellett a felhasználók adataival való kereskedelem. Itt nem arra kell gondolni feltétlenül, hogy személyes adatokat tesznek közzé, hanem arra, hogy pl. milyen hirdetéseket nézünk meg, egyes képek, videók, bejegyzések mennyire érdekelnek, online vásárlás esetén milyen termékek után érdeklődünk, egyszóval a szokásainkat elemzik. Ezek felhasználásával már kiderülhet, hogy egy adott személynek mi az érdeklődési köre, ez alapján célzott hirdetéseket helyezhetnek el nekünk a közösségi oldalak vagy különböző felméréseket, tanulmányokat készíthetnek.

Feladatunkban a Twitteren sporteseményekre kerestünk rá. Egyik célunk ezzel az volt, hogy megvizsgáljuk, hogy meg tudjuk-e találni az események helyét (stadionokat), másrészt, hogy megállapítsuk, hogy például melyik csapat szurkolótábora használja aktívabban vagy nagyobb számmal van-e jelen bizonyos közösségi média platformokon. Ezeknek a vizsgálatoknak az eredményét a 2. fejezetben mutattuk be.

Amik elsőre csak egyszerű adatoknak tűnnek, felhasználók egy-egy tweettel, fotóval jelentkeznek az egyes felületeken, az nagyobb létszámú felhasználó esetén már mérhetetlen lehetőséget rejt magában. Gazdasági, marketing, statisztikai szempontból is elemezhetjük az adatokat, melyek konkrétan revolúciót jelenthetnek a teljes sport, vagy bármilyen más kultúra számára. Egy brit médiatanulmány szerint, melyet a Kantar Media a Sportscope-ban mutat be, azt jelenti ki, hogy a 2018-as évben a stadionok környékére érkező szurkolók már több mint 25%-a van jelen aktívan a közösségi média felületein, köztük a Twitter vezető szerepet tölt be (15. ábra), de a passzív szám jóval magasabb lehet, emellett évről évre növekednek az értékek. [8]



15. ábra. Statisztikák is jól mutatják, hogy egyre többen használják a Twittert az élet minden területén

Régen a futball csapatoknak általában a fő bevételi forrása a jegyek eladásából származott. A világ fejlődött, a csapatok hírneve egyre jobban bejárta a világ különböző pontjait, megnőtt az érdeklődés és a technikai fejlődéssel együtt jöttek újabb bevételi források. A mai napig az angol első osztályban, a Premier League-ben az egyik legnagyobb bevétel a televíziós közvetítési díjakból származik. Ez a következő 10 évben gyökeresen megváltozhat a közösségi médiáknak köszönhetően, melyben vezető szerepet vállal a Twitter.

A változás, megújulás legelső látható jele, az amerikai NFL-ben volt tapasztalható, hiszen ott 2018. júliusában, a liga hivatalos vezetősége és a Twitter bejelentette, hogy 10 csütörtök esti mérkőzést élőben fognak a felületen közvetíteni, ezzel is mozgósítva az „online" szurkoló tábort. A történet itt nem fejeződik be, hiszen egyre inkább látható változásokat tapasztalhatunk már az európai közösségekben is, jól mutatva azt, hogy az európai top klubcsapatok közül szinte kivétel nélkül nyomatékosan elkezdték fejleszteni, használni a Twittert. Az Arsenal, a Barcelona és a Bayern München is hivatalos megegyezésekkel ásta be magát a projektbe (16. ábra), és olyan posztokat szolgáltatnak, melyekkel a rajongók még közelebb érezhetik magukat a szeretett csapatukhoz (17. ábra). Híres játékosok mikroüzenetekkel üdvözlik az embereket, vlogokat készítenek, közösségi történéseket dokumentálnak (karácsonyi show-k, iskolák meglátogatása, jegyajándékozás az utcán véletlenszerűen, nyereményjátékok) és sok esetben még személyes történeteket is megosztanak a közönséggel a sportolók, akik már-már sztár státuszba lépnek elő a mai modern világban, hiszen olyan szintű a kereslet, az érdeklődés az általuk közölt tartalom iránt.

16. ábra: A Bayern München Twitter oldala a tweetek és követők számával

17. ábra: A Bayern München - AEK Athén mérkőzés összefoglalóját 192-en retweetelték és 1100-an kedvelték

Mivel a sportok egyre jobban térnek át az online felületekre, azt nem szabad elfelejteni, hogy még azért mindennek a központja mégis a stadionok. A helyzet pedig úgy alakul át, amint telefonokból lettek okostelefonok, úgy lesznek stadionokból is okos stadionok. (A tendencia ezt mutatja.) Az egyik élő példa rá a Kaliforniában található Levi's Stadium, amelyet a legmagasabb szinten felszerelt „high-tech"-stadionnak tartanak a Földön. Az egész létesítményen belül ingyen Wi-Fi hotspot van, applikációt (18. ábra) készítettek a szurkolóknak arra, hogy melyik mellékhelység található a székükhöz legközelebb, rendelhetnek rajta ételt és italt a büféből. Az angol első osztályú futball ligában a Tottenham új stadionja is hasonló szintekre szeretne lépni, Angliában első módon, melyhez valószínűleg a többi csapat is fel fog zárkózni modernizáló fejlesztésekkel. Tényszerűen kijelenthető, hogy ilyen egyszerű, életet megreformáló dolgokat az esetek 90%-ában a Twitterről és egyéb közösségi médiumokból nyert adatokkal indítottak el.

18. ábra: A Levi's Stadium applikációja iOs rendszeren, melynek megalkotásában nagy szerepet játszott a csapat szurkolóinak tweetjeinek vizsgálata, elemzése

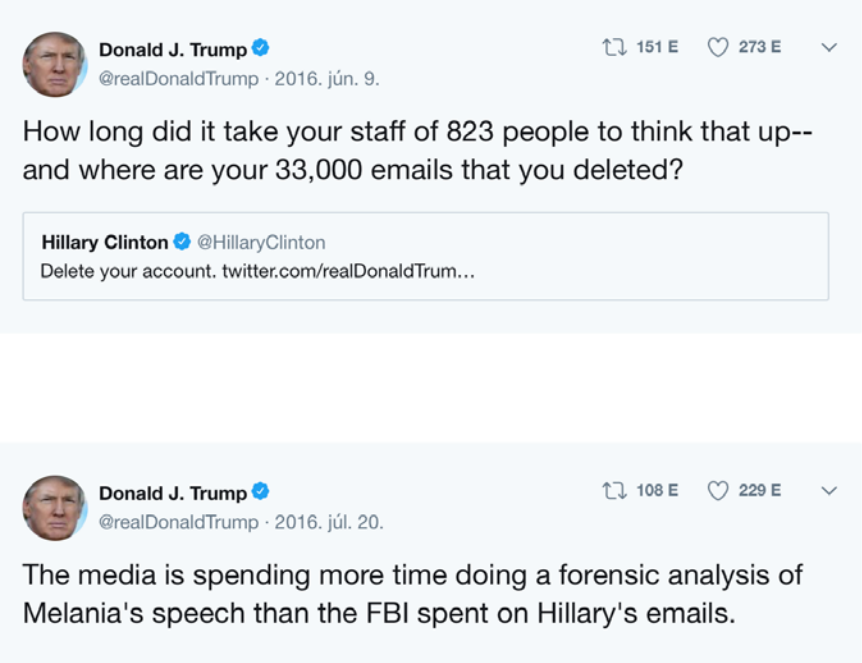
A sport rész a folyamatban lévő változások csak kis százalékát fedik le még mindig, hiszen kivétel nélkül a Twitter részese lett, vagy lesz idővel mindennek. Nemcsak gazdasági, tudományos, pénzügyi, hanem szociológiai elemzéseket is tudunk végrehajtani általa. Az emberek vallási, politikai hovatartozására vonatkozóan kereshetünk rá, kutathatunk a témákban.

Kétségkívül a Twitter szerepe a világon akkor alapozódott meg megkérdőjelezhetetlenül, amikor a 2016-os amerikai elnökválasztásban hatalmas szerep jutott rá. A mai napig vitatott a téma, de néhány álláspontón keresztül könnyen érthetővé lehet tenni. A mikroüzenetek hatása a felhasználói közegre hatalmas volt a 2015-ös és a 2016-os években. Rövid, agresszív és lényegre törő, leginkább így lehetne megnevezni ezeket az üzeneteket, melyek a Twitternek köszönhetően nagymértékben mozgósították az adott politikai pártok híveit, és olyan közeghez is eljutottak az üzenetek a közösségi média felületének köszönhetően, akik alapesetben nem foglalnak állást politikai témákban, esetleg nem foglalkoznak ilyen ügyekkel. A hírek, bejegyzések a lehető legegyszerűbb módon, a hírfolyamot betöltve, azt görgetve bukkannak fel előttünk, az alapján, hogy kiket követtünk be, kiknek kedveltük a bejegyzéseit. Szakértők szerint Donald Trump körülbelül 2 milliárd dollárnak megfelelő médiaértékkel bíró kampányt folytatott a tweetjeivel, ingyen. Míg a televízió és a nyomtatott hírközlés szerepe csökkenőben van, az emberek nagytöbbségének a kezében lévő okostelefon, laptop internetelérésével lett a

legegyszerűbb módja a nagyközönség tájékoztatásának. Donald Trump ehhez egy olyan stratégiát is kitalált, ami mindenképpen hatékonynak bizonyult. (19. ábra)

19. ábra: Donald Trump kampányának legfontosabb szlogenje tweetként

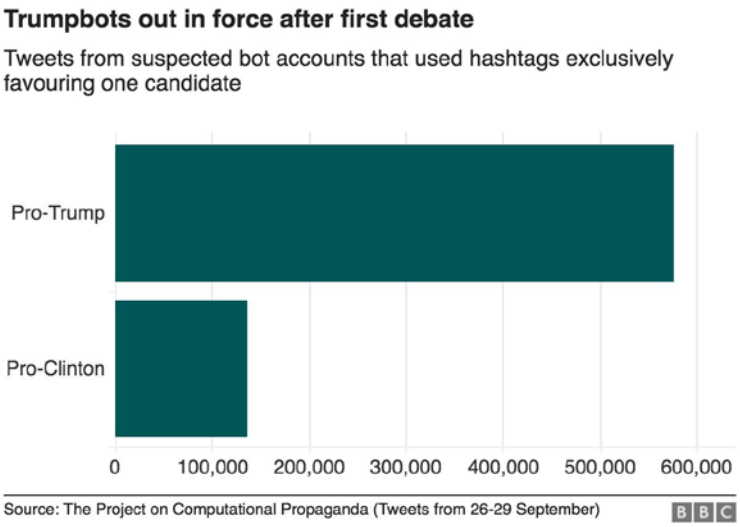
Trump, a kampánya alatt általában éjszakai, hajnali órákban osztotta meg a gondolatait a követőivel, akik a reggeli órákban miután felébredtek és kezükben lévő okos készülékekkel kapcsolódtak a világhálóhoz, azonnal szembesültek Trump tweetjeivel. A siker alapja abban rejtőzhet, hogy míg más médiumok által kapott hírekben az ember nem minden esetben tudja eldönteni, hogy a hír igaz-e vagy sem, vagy hogy melyik pártot szolgálja támogatásával, itt az ember elsőkézből kapja meg, szinte privát üzenetként az adott híreket és véleményeket. Legtöbbet megosztott és like-olt tweetjeiben általában a szlogenjét hangoztatta: „Make America Great Again” (19. ábra), erősen bírálta az ellenzéket, Hillary Clintont, főképp az e-mail botránya miatt (külügyminiszterként nem a hivatalos titkosított e-mail csatornát használta, hanem a könnyen feltörhető saját fiókját – ezzel is segítve a külföldi cyber kémek munkáját), és az FBI-t az általa hiteltelennek minősített munkavégzésükkel kapcsolatban (20. ábra).



20. ábra: Donald Trump két legismertebb tweetje, melyet a választások alatt tett közzé

Habár az ellenzék, Hillary Clintonék is jelen voltak a közösségi platformokon, közel sem olyan aktivitással, mint Trump. Statisztikák szerint az oldalán 2009 óta több mint 36 000 tweet lett posztolva. A Pew Research Center felmérése szerint 2005-ben az amerikaiak alig 7 százaléka használta a közösségi médiát, az elnökválasztás ideje alatt ez a szám már 65 volt. A jelenlegi elnök győzelméhez, viszont nemcsak az ő elsöprő ereje kellett, hanem más, már negatívnak mondható tényezők jelenléte is a médiában. Hatalmas feltűnést keltett a BuzzFeed portál elemzése, amikor vizsgálták, hogy az elnökválasztási kampány idején a tekintélyes amerikai lapok, portálok oldalain megjelent 20 legolvasottabb hír, illetve ugyanabban az időszakban az álhíreket generáló portálok 20 legolvasottabb története mekkora forgalmat generált a platformokon. Az eredmény lehangoló volt, és megmutatta, milyen könnyen lehet befolyásolni e felületek felhasználóit is. A hamis híreket 8,7 millióan osztották meg, kommentálták a Facebookon, a megbízható híreket ugyanakkor majd másfél millióval kevesebben, 7,3 millióan. [9] Emellett a Twitter sem úszta meg kisebb negatív megbélyegzés nélkül, hiszen hivatalos közleményt adtak ki arról, hogy a választások vége felé haladva több mint 50 000 oroszországi székhelyű Twitter felhasználó (bot, amiket automatikusan algoritmusok generáltak bizonyos oldalak befolyásolása érdekében, és kulcsszavakra generáltak posztokat, retweeteltek (megosztották, újra közzétették, nem feltétlenül az eredeti szerző által) az adott párt üzeneteit) használta aktívan a felületet. Statisztikák szerint több mint 670 000 amerikaihoz

jutottak el azok az üzenetek, tweetek, melyeket ezek a botok generáltak, és általában Trump malmára hajtva ezzel a vizet (21. ábra), hiszen számára nézve pozitív hatásúak voltak ezek a bejegyzések. A generált felhasználók üzeneteikben több fronton támadták Clintont, és előre, nem igazoltan hirdettek eredményt Trump győzelmével kapcsolatban, miközben a szavazások még folytak. A szavazás alatt számszerűen 175 993 tweetet tettek közzé az automatizált felhasználók, melyekből 8,4 százalék tényszerű eredményt közölt, ezzel jelentős hatást gyakorolva a szavazó polgárokra, emellett Trump tweetjeit 470 000-szer osztották meg újra, vagyis retweetelték. [10] Ámbár a sok pozitívum mellett elmondható, e gyors hírközlési formátum negatív hatással is lehet az emberekre, és ezek után több helyen is megkezdték a közösségi háló felületeinek jogi szabályozását, tervezetek készítését ezzel kapcsolatban, hiszen óriási hatalom van a közösséget befolyásoló hálózatok kezében.



21. ábra: A szavazás folyamán a botok száma oszlopdiagramként megjelenítve, attól függően melyik bot melyik elnökjelöltet pártfogolta (Trump mellett: kb. 580 000 tweet, Clinton mellett: kb. 140 000 tweet)

6. Gazdasági elemzés

Dolgozatunk elkészítésekor egyik szempont volt, hogy lehetőleg ingyenesen oldjuk meg a feladatot. Szerencsére minden szolgáltatásnak és szoftvernek volt szabadon hozzáférhető változata. A következőkben ismertetjük a dolgozat azon részeit, ahol választani lehet fizetős kereskedelmi és ingyenes megoldás között.

6.1. Adatszolgáltatások díjai

A Twitter esetében három különböző ajánlat van. Egyrészt van az ingyenes szolgáltatás, mellyel használhatjuk a Twitter API-ját, de a letöltött adatokat haszonszerzési célokra nem használhatjuk, valamint visszamenőleg csak hét napra tudunk adatot letölteni. A másik két verzió (premium és enterprise (vállalati)) fizetős, melyeknek előnye, hogy egyrészt időben távolabb tudunk visszamenni, gyorsabb az adatforgalom. Ezeknek a díja sok mindentől függ, a mi esetünkben a premium $149/hónapra (41 936.392 Ft/hó) jön ki, a másiknak a díja sajnos nem elérhető, a Twitter kalkulál egy árat igényünktől függően. A premium és az enterprise között a különbség az, hogy a premiummal maximum 30 napra visszamenőleg tudunk keresni, míg az enterprise csomaggal bármennyi időre visszamenőleg, gyorsabb a lekérdezés (kb. kétszer gyorsabb).

További költség lehet a különböző helyadatok geokódolása, amelyről az 1.3.1. fejezetben írtunk. Erre számos lehetőség áll rendelkezésünkre. A legkézenfekvőbb megoldás a Google vagy az OSM (OpenStreetMap) lehet, mindkettőnek megvan az előnye, hátránya. A Google esetében, bár van ingyenes adatnyerésre lehetőség, ez limitált és napi 2000 lekérdezésnél nem enged többet, csak ha fizetünk. Az OSM esetében pedig, bár ez egy ingyenes megoldás, korlátozott az adatforgalom, így jelentősen lelassítja a munkafolyamatot, mivel esetünkben hatalmas mennyiségű adatról van szó. A díjak a Google esetében $5.00/1000 lekérdezés/hónap (1 407.261 Ft), ami a példánkban kb. $500/hónap (140 726.147 Ft).

6.2. Felhőalapú számítástechnika – Cloud Computing

A felhőalapú számítástechnika (Cloud Computing) ma már széleskörben elterjedt, sokan használják. Számos felhőalapú szolgáltatás vált elérhetővé, ezek közös vonása, hogy a szolgáltatásokat nem egy dedikált hardvereszközön üzemeltetik, hanem a szolgáltató eszközein elosztva, a szolgáltatás üzemeltetési részleteit a felhasználótól elrejtve. Az alapján, hogy a szolgáltatáshoz milyen módon és a felhasználók milyen köre férhet hozzá, négy típus különíthető el: privát, hibrid, közösségi, illetve publikus felhő. Általánosságban a cloud computingnak számos előnyös tulajdonsága van magánfelhasználóknak és cégeknek egyaránt [12]:

1. Költséghatékony
    a. A hardvereszközök megvásárlásának költségét a szolgáltatás használatának díja váltja fel – ez például lehet a bérelt számítási kapacitás, hálózati forgalom, vagy a felhasználók száma alapján kiszámolt összeg.
    b. Az üzemeltetési, sokszor nehezen tervezhető feladatok nem a felhasználókat terhelik.
2. Bárhonnan elérhető
    a. Egy felhő-alapú megoldás (főleg publikus felhő) szolgáltatás esetében a szolgáltatás bárhonnan könnyen elérhető lehet.
    b. A szolgáltatások jelentős része platformtól független.
3. Rugalmas, méretezhető
    a. A számítási kapacitás a felmerülő igényeknek megfelelően skálázható.
    b. Amennyiben nincs rá szükség, használata felfüggeszthető.
4. Megbízható, biztonságos
    a. A szolgáltatásokért felelős szerverparkok védett helyeken találhatók.
    b. A szolgáltatások nem egy dedikált helyről, hanem több osztott szerverfarmról származnak.

A cloud computing típusai [13]:

- Szoftver-alapú számítási felhő (Software as a Service, SAAS): a szoftvert nyújtja szolgáltatásként jellemzően egy vékony kliensen keresztül; egy böngészővel lehet elérni, pl. Google docs. A felhasználó nem fér hozzá a mögöttes tartalomhoz, pl. az operációs rendszerhez és hardver infrastruktúrához,
- Platform-alapú számítási felhő (Platform as a service, PAAS): esetenként skálázható operációs rendszerszolgáltatás (Pl. Microsoft Azure, Google App Engine),
- Infrastrukturális számítási felhő (Infrastructure as a Service, IAAS): Virtuális hardverszolgáltatás (pl. Amazon EC2).

A mi esetünkben olyan szolgáltatás lenne hasznos, amivel pl. adatbázist kapunk, így nem kell nekünk megvenni és üzemeltetni a szervert, ezt megteszi helyettünk a szolgáltató. Ilyen szolgáltatást nyújt pl. a Microsoft Azure vagy a Google. Mindkét esetben van lehetőségünk ingyenes próbaidőre, aminek letelte után dönthetünk, fizetünk-e a továbbiakban (mi ezzel a lehetőséggel nem éltünk, mivel a tanszék biztosította a szervert). Mindkét esetben lehetőség van akár használati óra alapján számlázni. A Google-nél egy 1.7 GB RAM-mal, 3GB tárhellyel rendelkező gép esetén $0.1/óra (28.145 Ft/óra). [14] Az Azure esetében 7GB RAM,

választható tárhellyel (első 32 GB ingyenes, utána $0.115 GB-onként (32.367 Ft/GB)) $0.2522/óra (70.982 Ft/óra). [15]

Ha a feladatunkat egy Google szerveren végeztük volna, akkor az előadás napjáig a bérleti díj $108 (30 396.848 Ft) lenne, egy Azure szerveren $273 (76 836.476 Ft), amennyiben úgy számoljuk, hogy az adatbázisra október 1-től lenne szükségünk). Ha magunk állítanánk össze egy szervert, akkor annak megvan az az előnye, hogy csak egyszer kell kifizetni és utána kedvünk szerint használjuk, de benne van az a hátrány, hogy ha valami gond van vele, akkor nekünk kell szervizelni, valamint ha adatvesztés lép fel pl. áramkimaradás vagy bármilyen okból nekünk kell vállalni a felelősséget, míg a felhőalapú szolgáltatások esetén a szolgáltató garantálja azt, hogy ilyen nem fordul el (vagy ha mégis ő vállalja a felelősséget). Egy számunkra megfelelő szervergép mai áron $390-500 között mozognak (110 000 – 140 000 Ft). [16]

7. Összefoglalás

Dolgozatunk célja egy olyan módszer kidolgozása volt, aminek a segítségével térinformatikai elemzést hajthatunk végre a Twitteren megosztott tartalmak alapján.

Első megközelítésben egy olyan algoritmust írtunk Python környezetben, aminek a segítségével rácsatlakoztunk az élő adatfolyamra és a kód indításától a leállításáig gyűjti az adatokat, ez volt az ún. Stream Listener. Ez egy igen hasznos adatgyűjtési mód, viszont nagy hátránya, hogy a kódnak sokáig kell futnia ahhoz, hogy használható mennyiségű adatunk legyen. Nagyobb elemzési lehetőséget biztosít az, ha archív adatokat elemzünk. Ekkor attól függően, hogy milyen API áll rendelkezésünkre, több napra (vagy akár hónapokra) visszamenőleg is tudunk adatot szerezni. Erre egy második Python kódot használtunk fel, majd a letöltött adatokat, egy harmadik segítségével feldolgoztuk és csak a számunkra releváns információkat tartottuk meg (pl. maga a bejegyzés, időbélyeg és a földrajzi hely vagy koordináták).

Miután az adatgyűjtés megtörtént, OpenRefine-ban geokódoltuk az adatokat, majd feltöltöttük az adatbázisunkba. A geokódolásra azért volt szükség, mert a legtöbb adat, melyet letöltöttünk, csak utalást tartalmazott a helyre, pontos koordinátát nem, de még így is tartalmazott ún. fals pozitívokat az adatsor. Az adatbázis kezelésekor számos akadályba ütköztünk, mivel az adatok nagyon sokfélék voltak. Ahhoz, hogy megtudjuk jeleníteni a pontokat egy térinformatikai rendszerben át kellett alakítani azokat ún. geometriává, amit a gép értelmezni tud. Ezek után QGIS környezetben megtudtuk jeleníteni a pontokat és azok alapján térinformatikai elemzéseket tudtunk készíteni. Emellett statisztikailag is elemeztük a letöltött adatokat.

Másik megoldásunk a problémára egy webalapú megközelítés, melynek alapja egy leegyszerűsített honlap, ami egy keresőből és egy térképből áll. Ezek segítségével vizsgáljuk az események időbeli lefolyását, a trendeket és naprakész adatok vizsgálatát. A Twitter-en keresztül lekérdezve megjeleníthetjük a keresett találatokat Google térképen. A jelekre (markerekre) kattintva megtekinthetjük az üzenet tartalmát is.

Az eredeti terv szerint Wordpress-ben szerettük volna megalkotni az oldalt, amit azonban végül megváltoztattunk és más környezetben oldottuk meg. Ennek oka az volt, hogy a Wordpress-ben bizonyos programozási nyelveket nem vagyunk képesek használni, így egyszerűségi okok miatt ezt választottuk.

Az oldal fejlesztési lehetőségeit is részleteztük, mint például a kezelő felület fejlesztését, az adatbázissal való összekapcsolást vagy azt, hogy ha a Twitter premium API-ját használnánk, akkor sokkal több adat állna rendelkezésre, többet tudnánk egyszerre megjeleníteni.

Mindezek után összevetettük eredményeinket. Mind a két megoldásnak vannak előnyei és hátrányai, az, hogy melyiket alkalmazzuk a körülményektől, gazdasági háttértől és az igényektől függ. Kitérünk a Twitter jelentőségére a mai világban marketing és politikai szempontjából is, ugyanis tagadhatatlan a mai világra gyakorolt hatása ennek a közösségi médiának.

Gazdasági szempontból is megvizsgáltuk a feladatot. Megvizsgáltuk, hogy megérné-e felhő alapú szolgáltatást igénybe venni. Ez a kérdés is sok tényezőtől függ, az egyes megoldásoknak a költségeit a 6.2. fejezetben leírtuk. Lényegében a nagy pozitívuma a felhőalapú szolgáltatásoknak, hogy skálázható, magyarán csak arra az időtartamra kell kifizetni, amíg azt használjuk, probléma, adatvesztés esetén a szolgáltató vállalja a felelősséget és a karbantartást sem nekünk kell vállalni, míg ha magunk veszünk vagy építünk egy szervergépet, azt csak egyszer kell kifizetni és addig használjuk, amíg akarjuk, viszont adatvesztés esetén nem tudjuk a felelősséget hárítani, a karbantartásról is nekünk kell gondoskodni.

Köszönetnyilvánítás

Szeretnénk megköszönni a dolgozat elkészítéséhez nyújtott hatalmas segítségét és munkáját konzulensünknek Dr. Barsi Árpádnak, valamint az adatbáziskezelés terén nyújtott segítségét Dr. Molnár Bencének; nélkülük ez a dolgozat nem jöhetett volna létre.

Irodalomjegyzék

[1]: Mikael Brunila - Scraping, extracting and mapping geodata from Twitter (2017. Március 27. - http://www.mikaelbrunila.fi/2017/03/27/scraping-extracting-mapping-geodata-twitter/ - utoljára elérhető: 2018. 10. 12.)

[2]: A JSON formátum weboldala (http://www.json.org – utoljára elérhető: 2018. 10. 03.)

[3]: Bhaskar Karambelkar - How to use Twitter's Search REST API most effectively. (2015. Január 5. - https://www.karambelkar.info/2015/01/how-to-use-twitters-search-rest-api-most-effectively./ - utoljára elérhető: 2018. 10. 12.)

[4]: ArcGIS hivatalos oldala, a geokódolás leírása (http://pro.arcgis.com/en/pro-app/help/data/geocoding/what-is-geocoding-.htm - utoljára elérhető: 2018. 10. 03.)

[5]: DataScienceToolKit hivatalos oldala (http://www.datasciencetoolkit.org/about - utoljára elérhető: 2018. 10. 03.)

[6]: Google Maps platform Webes java script api leírásának hivatalos oldala - https://developers.google.com/maps/documentation/javascript/geolocation/ - utoljára elérhető: 2018. 10. 03.

[7]: ASP Snippets honlapnak fóruma, a markerek megjelenítése JavaScript kódolással

https://www.aspsnippets.com/Articles/Integrate-Google-Maps-in-ASPNet-MVC.aspx?fbclid=IwAR2SoGwLmbGDIn4LQO9y1CHNjE1TZYk0V8C1c6oOoTDMt6Wr7cIby89wkeA - utoljára elérhető: 2018. 10. 25.

[8]: A Twitter befolyása a sportlétesítményekre https://www.kantarmedia.com/uk/thinking-resources/blog/the-future-of-football - utoljára elérhető: 2018. 10. 25.

[9]: A Twitter hatása az amerikai elnökválasztásra https://nepszava.hu/1112580_donald-trump-twitter-forradalma - utoljára elérhető: 2018. 10. 25.

[10]: A Twitter-botok hatása az amerikai elnökválasztás alatt https://www.theguardian.com/technology/2018/jan/19/twitter-admits-far-more-russian-bots-posted-on-election-than-it-had-disclosed - utoljára elérhető: 2018. 10. 25.

[11]: A Page rang – Wikipédia cikk (https://hu.wikipedia.org/wiki/PageRank - utoljára elérhető: 2018. 10. 25.)

[12]: Cloud Computing - http://searchcloudcomputing.techtarget.com/definition/cloud-computing/ - utoljára elérhető: 2018. 10. 25.

[13]: Felhő alapú számítástechnika - https://hu.wikipedia.org/wiki/Felh%C5%91_alap%C3%BA_sz%C3%A1m%C3%ADt%C3%A1stechnika - utoljára elérhető: 2018. 10. 25.

[14]: Google SQL szerver ára – Google hivatalos oldala https://cloud.google.com/sql/pricing - utoljára elérhető: 2018. 10. 25.

[15]: Microsoft Azure SQL szerver ára – Microsoft Azure hivatalos oldal https://azure.microsoft.com/hu-hu/pricing/details/sql-database/managed/ - utoljára elérhető: 2018. 10. 25.

[16]: Árukereső – szerver gépek árai https://www.arukereso.hu/szerver-c3641/ - utoljára elérhető: 2018. 10. 25.