

Budapesti Műszaki és Gazdaságtudományi Egyetem

Villamosmérnöki és Informatikai Kar

Távközlési és Médiainformatikai Tanszék (TMIT)

László Gábor

**Közösségi média aktivitásfigyelő szolgáltatás**

Konzulens

Nagy-Rácz István

BUDAPEST, 2018

Hallgatói nyilatkozat

Alulírott **László Gábor**, szigorló hallgató kijelentem, hogy ezt a diplomatervet meg nem engedett segítség nélkül, saját magam készítettem, csak a megadott forrásokat (szakirodalom, eszközök stb.) használtam fel. Minden olyan részt, melyet szó szerint, vagy azonos értelemben, de átfogalmazva más forrásból átvettem, egyértelműen, a forrás megadásával megjelöltem.

Hozzájárulok, hogy a jelen munkám alapadatait (szerző(k), cím, angol és magyar nyelvű tartalmi kivonat, készítés éve, konzulens(ek) neve) a BME VIK nyilvánosan hozzáférhető elektronikus formában, a munka teljes szövegét pedig az egyetem belső hálózatán keresztül (vagy hitelesített felhasználók számára) közzétegye. Kijelentem, hogy a benyújtott munka és annak elektronikus verziója megegyezik. Dékáni engedéllyel titkosított diplomatervek esetén a dolgozat szövege csak 3 év eltelte után válik hozzáférhetővé.

Kelt: Budapest, 2018. 04. 15.

...…………………………………………….

László Gábor

Tartalomjegyzék

[1 Absztrakt 5](#_Toc500496245)

[2 Abstract 6](#_Toc500496246)

[3 Bevezetés 7](#_Toc500496247)

[4 Adatgyűjtés 9](#_Toc500496248)

[4.1 A gyűjtőmodulok közös jellemzői 9](#_Toc500496249)

[4.1.1 A bemeneten kapott keresőkifejezésen elvégzett műveletek 11](#_Toc500496250)

[4.2 Facebook crawler modul 12](#_Toc500496251)

[4.2.1 Facebook API 12](#_Toc500496252)

[4.2.2 A crawler 12](#_Toc500496253)

[4.3 Twitter crawler modul 14](#_Toc500496254)

[4.3.1 Twitter API 15](#_Toc500496255)

[4.3.2 A crawler 15](#_Toc500496256)

[4.4 Instagram crawler modul 17](#_Toc500496257)

[4.4.1 Instagram API 17](#_Toc500496258)

[4.4.2 A crawler 17](#_Toc500496259)

[4.5 Az adatok tárolása 18](#_Toc500496260)

[4.5.1 A tárolást igénylő adatok kiválasztása 18](#_Toc500496261)

[4.5.2 Az SQLite rendszer 19](#_Toc500496262)

[4.6 Az adatgyűjtő modul értékelése 19](#_Toc500496263)

[4.6.1 Statisztikák 19](#_Toc500496264)

[5 A kapott adatok validálása, a keresett profilok pontos beazonosítása 21](#_Toc500496265)

[5.1 Ötletek, fejlesztési irányok 21](#_Toc500496266)

[5.1.1 Legpontosabb egyezés 21](#_Toc500496267)

[5.1.2 Legtöbb követővel rendelkező 21](#_Toc500496268)

[5.1.3 Cache-ben tárolt, kézzel validált halmaz alapján 22](#_Toc500496269)

[5.2 Validáláskor felmerült problémák 22](#_Toc500496270)

[5.3 A validálás értékelése 23](#_Toc500496271)

[6 Megjelenítés 24](#_Toc500496272)

[6.1 Megjelenítési tervek 24](#_Toc500496273)

[6.1.1 Összehasonlító felület 25](#_Toc500496274)

[6.1.2 Időtartományban kereső felület 26](#_Toc500496275)

[6.2 Plotly 27](#_Toc500496276)

[6.2.1 Dash 27](#_Toc500496277)

[6.3 Más megjelenítési módok 28](#_Toc500496278)

[7 A gyűjtőmodul tesztelése valós adatokon 29](#_Toc500496279)

[8 Továbbfejlesztési lehetőségek 30](#_Toc500496280)

[9 Összegzés 31](#_Toc500496281)

[10 Függelék 32](#_Toc500496282)

[10.1 cache-ben tárolt csapatok és főbb mutatóik 32](#_Toc500496283)

[Irodalomjegyzék 37](#_Toc500496284)

# Absztrakt

A XXI. században egyre nagyobb teret nyernek azok az elemzési megoldások, amelyek a publikusan elérhető közösségi oldalak adatait dolgozzák fel. Az elérhető információk birtokában képesek lehetünk jól leírni egy közösségi profil működését és megbecsülhetjük a benne rejlő potenciált is.

Diplomamunkám célja adatgyűjtés közösségi oldalakról abból a célból, hogy a gyűjtött adatokkal biztosítsam közösségi oldalakon lévő profilok aktivitásának elemzését. Az adatok gyűjtését és feldolgozását angol nyelvű, futballcsapatokhoz kapcsolható profilokra specializálom. A feladatom része az adatok legyűjtése, feldolgozása, valamint vizualizálása, illetve a lehetséges továbbfejlesztési irányok ismertetése.

Az adatgyűjtéshez három közösségi oldal jelentett forrást. Munkám során megismertem a Facebook, a Twitter, az Instagram közösségi oldalakat, azok felhasználói profiljainak felépítését, továbbá az adatok hozzáférésének módját. Megterveztem és implementáltam egy olyan szoftvert, amely képes erről a három oldalról adatokat gyűjteni és rendezetten tárolni. Ezután a gyűjtött adatok minőségi javulása érdekében adatelkészítő lépéseket hajtottam végre.

Vizualizálási megoldásokat dolgoztam ki annak érdekében, hogy a kapott adatok jól megjeleníthetők legyenek. Áttekintettem a megjelenítésben rejlő lehetőségeket. TODO

Ezt követően valós adatokkal teszteltem a szoftverem profilokat felismerő moduljának hatékonyságát. Feltártam az esetleges gyenge pontokat és szűk keresztmetszeteket.

Végül áttekintettem az elkészített szoftver továbbfejlesztési lehetőségeit a gyűjtő modultól az adatelőkészítésen át a tényleges elemzésig. Javaslatot tettem az így feltárt hiányosságok és a prototípus helyenkénti egyszerűségéből fakadó pontatlanságok kijavítására, valamint a hatékonyság növelésére.

# Abstract

TODO AKTUALIZÁLNI!!!

In the 21st century, those data mining solutions which are based on getting information from social networks, have gained more ground recently. With the information available, we can describe the operation of a community profile and estimate its potential.

The goal of my thesis is gathering data from social networks in order to provide the collected data to analyze the activity of profiles on social networking sites.

I specialize in collecting and processing data in English-language profiles linked to football teams. My task was to gather, and analyse and visualize information, and furthermore to describe potential development directions.

During the data mining progress, three social network sites were used. The working project was a good chance to get acquainted with Facebook, Twitter and Instagram and structure of their user’s profiles. I also managed to know the method of accessing information. A so-called data mining software was architected and implemented, which is able to gather and sort information from these three social network sites. Then, in order to improve the quality of gathered information, I made some data preparing.

I have developed visualization solutions so that the data can be displayed well. I reviewed the potential of the display.

Next, I tested the efficiency of my software with real information. Incidental weak points and cross-sections were located. I succeeded in finding the most suitable solution to finding the best match for the given query.

Finally, I reviewed the possible development directions of the finalized software, including the gathering module, data preparing and the actual analysis. I made a proposal for repairing the uncovered deficiencies, inaccuracies of the prototype and proposal for increasing effectiveness.

# Bevezetés

Egy magára valamit is adó futballklubnak nem elég csupán a pályán helytállnia. A marketing, az imázs éppen olyan fontos, mint a rúgott gólok száma, vagy a nézőszámok alakulása. Ahhoz, hogy egy csapat ne csak a tabellán foglaljon el jó helyezést, hanem a közösségi médiák népszerűségi listáin is, aktív részvétel kell az egyes platformokon.

Az elmúlt évtized a közösségi oldalak térnyeréséről szólt. Kezdetben a magánszemélyek kapcsolattartását segítették elő, később azonban a cégek is felismerték a benne rejlő lehetőségeket. Egy felhasználó átlagosan közel két órát tölt el valamelyik közösségi oldal böngészésével [1].

Nem kivételek ez alól a sportegyesületek sem, amelyek jelentős rajongói bázist tudtak kiépíteni ezeken az oldalakon, egyfajta mérhető népszerűségi versenyt generálva. Ezen adatok egy része, például a kedvelők vagy követők száma publikusan is elérhető, látható az egyszeri látogató számára. Vannak azonban olyan metrikák, amik nem ennyire triviálisak, illetve nem feltétlenül érhetők el akárkinek, ugyanakkor hasznosak lehetnek egy-egy klub közösségi aktivitásának leírásához. A meglévő beépített statisztikai mutatók mellett jó volna egy olyan lehetőség is, amelyik több közösségi portál adatait is képes összefésülni és ezzel átfogó képet adni az aktivitásról. Ezeken felül vizualizációval is lehetne támogatni az elemzéseket.

Feladatomban három fő közösségi oldalon, a Facebookon, a Twitteren és az Instagramon vizsgálom a nagyobb európai futballcsapatok aktivitását. A feladatnak részét képezi a tényleges hivatalos oldalak automatikus felismerése és validálása, valamint a lehető legtöbb releváns jellemző paraméter legyűjtése. Az elkészült fő modul alkalmas lesz az egy vagy több kulcsszó szerinti keresésre, emellett pedig a keresés gyorsítása érdekében a nevesebb csapatok statikus attribútumait, például azonosíit fixen tárolja. A naprakészen gyűjtött adatokból historikus adathalmaz állítható össze, ami elősegíti az elemzéseket, predikciókat, valamint a vizualizálást is. Az egyes modulok képesek önállóan is működni.

Az adatgyűjtéshez alaposan áttanulmányozom a Facebook, a Twitter és az Instagram közösségi oldalak szerkezetét, felhasználói profiljainak felépítését. Az egyes oldalak publikus API-jait felhasználva olyan szoftvert fejlesztek, amely képes egy névre keresve profiladatokat gyűjteni a három forrásból.

A találatok gyűjtésekor jól behatárolom a ténylegesen szükséges attribútumok és metaadatok halmazát és különböző adatelőkészítő lépésekkel hozom őket könnyen kezelhető formára. A gyűjtés során különösen nagy hangsúlyt kap a kedvelők és követők számának vizsgálata, emellett pedig más fontos attribútumok is legyűjtésre kerülnek. Ezután kidolgozok egy olyan eljárást, amellyel kiválaszhatók a pontos találatok, biztosítva, hogy ténylegesen a keresett csapat adatait kapjuk vissza.

Áttekintem a megjelenítési lehetőségeket, hogy a kapott adatokat érthető, elemzésre alkalmas formára tudjam hozni. Megjelenítési terveket dolgozok ki és implementálok is egyet közülük. TODO

Végül áttekintem a szoftver lehetséges továbbfejlesztési irányait és javaslatot teszek az egyes részek későbbi javítására.

# Adatgyűjtés

Az közösségi média aktivitás megfigyelését végző rendszer egyik legfontosabb része az adatok gyűjtéséért felelős részegység. Ennek feladata, hogy egy megadott inputra minél több potenciális kapcsolódó adatot gyűjtsön össze a megfelelő forrásokból.

Az interneten fellelhető közösségi oldalak kimeríthetetlen publikus adatforrást jelentenek. Az ilyen oldalak jellemzően elérhetőek a fejlesztők számára valamilyen API-n keresztül, így bányászatuk alapvetően nem ütközik különösebb nehézségbe. Az egyetlen valódi gondot az okozza, hogy az utóbbi időszakban különböző szigorítások léptek életbe az egyes oldalakon elérhető adatok láthatóságával kapcsolatban, így azok olykor csak korlátozottan érhetők el. PL Cambridge Analytica

A feladatomhoz három olyan közösségi oldalt választottam ki (Facebook, Twitter, Instagram), amelyek a regisztrált felhasználóik számában nagyjából azonos nagyságrendet képviselnek, stílusuk és funkciójuk viszont különböző. Fontos szempont volt még, hogy jelentős felhasználói bázissal rendelkezzenek, illetve figyelembe vettem azt is, hogy kellő mennyiségű gyűjthető adatot biztosítsanak.

A következő fejezetben ismertetem az adatgyűjtő modul működését, részletesen kitérve az egyes források sajátosságaira, az API-k használatára és az adatok tárolásának módjára.

## A gyűjtőmodulok közös jellemzői

Az általam elkészített gyűjtőmodulok moduláris szerkezetűek. Önállóan is alkalmasak az adott forrás tematikus gyűjtésére, együtt pedig megadják a kulcsszóhoz tartozó összes elért paramétert. Mindhárom modult Python nyelven implementáltam.

Minden blokk visszaadja az összes kiválasztott attribútumot egy kulcs-érték párokat tartalmazó adatszerkezetben. Modulonként 6-6 jellemző paraméter érhető el, de ez a lista igény szerint könnyen tovább bővíthető. Az összesített szótár típusú eredményen felül minden attribútumra készítettem tagfüggvényeket is arra az esetre, ha csak egy-egy érték frissítésére van szükség, például a követők számának naprakész lekérdezésére. Ez a megfontolás a skálázhatóságot segíti elő.

A lekérdezések eredményeit lehetőség van egy általam definiált függvénnyel CSV formátumba is kimenteni a további elemzések és statisztikák készítésének elősegítésére. Ehhez segítségül hívtam Python Pandas [2] nevű, adatelemzést segítő modulját. Ezen felül létrehoztam egy adatbázis alapú tárolási megoldást is azokra a paraméterekre, amelyeket tartósan is érdemes tárolni. PÉLDÁK

Az egyes modulokban legyűjtött paramétereket a 1. ábrán lévő Venn-diagram jeleníti meg. EZ KUKA

1. ábra - A gyűjtött paraméterek Venn-diagramja

Látható, hogy vannak olyan értékek, amik minden forrásnál megjelentek, valamint vannak olyanok is, amik csak az adott forrást jellemezték. Előbbire példa a név, egyedi azonosító (ID), like-ok/követők száma, valamint egy weboldal linkje. Utóbbinál előjött például a fiók létrehozásának ideje, kategória, illetve a közzétett képek vagy bejegyzések száma.

Az egyes adatgyűjtő modulokban közös még, hogy a szükséges kulcsokat és tokeneket külön config fájlokba rendezve adtam meg, így függetlenítve az alkalmazást a személyre szabott azonosítóktól való függéstől. Az egyes feladatrészeket az alábbiakban fejtem ki részletesen.

### A bemeneten kapott keresőkifejezésen elvégzett műveletek

Miután a cél egy olyan komplex gyűjtőmodul elkészítése, amely összefogja mindhárom adatforrást, fontos a bemenetet egységes formára hozni, amit mindegyik API fel tud dolgozni. Erre több lépést is implementáltam.

Az egyes adatforrásoknál a keresés során már egy karakter eltérés is különbségként jelenik meg. Különösen zavaró, ha az eltérés csupán abból fakad, hogy az egyik helyen ugyanaz nagy kezdőbetűvel szerepel, míg a másikon kicsivel. Ezt elkerülendő, az összes modulnál a bemeneten kapott keresőkifejezést kisbetűs formára hoztam.

Egyes csapatok nevei tartalmazhatnak speciális karaktereket, amelyeket nem minden API tud feldolgozni. A hibaüzenetek elkerülése végett készítettem egy karaktereket vizsgáló és cserélő modult, amely minden gyűjtőmodulban felhasználásra került. A futballcsapatok nevében előforduló speciális karakterek vizsgálatához generáltam egy listát az európai első osztályú csapatok neveiből, és ebből a listából kivettem az angol ABC-nek megfelelőket. Ez alapján a hagyományosabb, ékezetes kis-és nagybetűk mellett az alábbi listában szereplő karaktereket cseréltem ki az alakban hozzájuk legközelebb álló valid, angol ABC-nek megfelelő karakterre:

|  |
| --- |
| ř, ş, â, ä , ø , ñ, å, æ, č, ğ, à, Ł, ł, ń, ž, Š, Ž, š, ć, î |

A több forrásból való gyűjtés miatt időnként belefutottam karakterkódolási anomáliákba. Ezt feloldandó a Django modul[3] smart\_str függvényét hívtam segítségül, amely hatékonyan megszünteti a lehetséges hibákat.

Mivel minden forrás tartalmazhat webes linket, ezért fontos ezek elérhetőségének és helyességének ellenőrzése. Ehhez egy Python-os megoldást választottam, amely ellenőrzi, hogy az adott link elérhető-e és onnan értelmes adat visszaérkezik-e. Ha igen, akkor tárolja a linket, ellenkező esetben nincs értelme a tárolásnak.

A CSV formátum használata miatt indokolttá vált, hogy minden gyűjtött attribútumból eltávolítsam a vesszőket. A vesszők használata nem befolyásolja az adatok értelmezését, nincs megkülönböztető erejük.Helyettük pontosvesszőket szúrtam be.

## Facebook crawler modul

A Facebook korunk egyik legelterjedtebb közösségi oldala. Csaknem minden negyedik ember használja világszerte, míg a hazai viszonyra igaz, hogy már jóval ötmillió fölött van a Facebook felhasználók száma. A későbbiekben tárgyalt két portállal ellentétben ez megkülönbözteti a természetes személyeket és az egyes publikus oldalakat. Előbbinek az automatizált vizsgálata meglehetősen komplikált a biztonsági beállítások miatt, míg utóbbinak az elemzésére számos jól konfigurálható lehetőség áll az informatikusok rendelkezésére. Az alábbiakban a Facebook crawler modulját ismeretem részletesen.

### Facebook API

A Facebook esetében a Graph Search [4] biztosít lehetőséget a keresésre, elemzésre. Használatához egy ’access token’-re van szükség, amit az admin felületen generálhatunk, ugyanakkor nincs szükség külön Python modul alkalmazására a Facebook eredményes bányászatához. Elég a generált tokent megadni minden lekérdezés végző URL végén.

Az így elérhető attribútumok bővebb listája elérhető a szolgáltatás online dokumentációjának oldalán[5]. Az alábbiakban csak az általam használt mérőszámokra térek ki részletesebben.

### A crawler

A keresés ebben a modulban három fő részből áll össze. A három lépésre azért van szükség, mert előbb meg kell tudni egy oldal ID értékét ahhoz, hogy a bővebb attribútumhalmaz hozzáférhetővég váljon.

Az első lépés a szabadszavas keresés, melynek bemenete egy csapat teljes neve (szóközök használata megengedett). Ennek lényege, hogy a beírt kifejezésre visszaadja a keresés nyers eredményét egy listában (’id’, kedvelők száma). A megadott kereső kifejezést a modul mentesíti a szóközöktől. A speciális karakterek használata nem okoz problémát a kereséskor.

Szabadszavas keresésre az alábbi url mutat példát.

|  |
| --- |
| [https://graph.facebook.com/search?q="<team\_name>"&type=page&access\_token= <access\_token>](https://graph.facebook.com/search?q=%22%3cteam_name%3e%22&type=page&access_token=%20%20%3caccess_token%3e) |

A második lépés a validálás a kedvelések számának maximuma alapján. A validálás során a kedvelők számát tekintettem perdöntőnek, azaz a legnagyobb követői bázissal rendelkező oldalt választja ki az alkalmazás a keresési eredményeket tartalmazó listából.

A validálás eredményeként kapott oldal részletes attribútum listájának lekérdezése ID alapján a harmadik lépésbe történik meg. Egy ID alapú, érvényes Graph Search lekérdezésre az alábbi kifejezés ad példát. Az ID helyére a lekérdezni kívánt oldal azonosítója kerül, míg az ’access token’ értelemszerűen a fentebb említett értékkel azonos.

|  |
| --- |
| [https://graph.facebook.com/v2.4/<id>?fields=id,name,likes,link,talking\_about\_count,category&access\_token=<access\_token>](https://graph.facebook.com/v2.4/%3cid%3e?fields=id,name,likes,link,talking_about_count,category&access_token=%3caccess_token%3e) |

Jóval kevesebb lekérdezést (és ezáltal időt) igényel, ha csak a validált fiókról kérjük le a részletes attribútum listát. Erre később a validálással foglalkozó fejezetben részben térek ki bővebben. A Facebook API-n keresztül lekérdezett attribútumokat az 1. táblázat tartalmazza. NEM AKTUÁLIS!!!

|  |  |  |
| --- | --- | --- |
| Attribútum neve | Magyarázat | Típus |
| name | oldal látható neve | String |
| id | oldal egyedi azonosítója | Int |
| likes | oldal kedveléseinek száma | Int |
| talking about count | megemlítések száma | Int |
| category | oldal kategóriája | String |
| url | oldalra mutató link | String |

1. táblázat - A Facebook-ról gyűjtött attribútumok

Hibakezelésre akkor volt szükség a modulban, amikor a keresés első fázisa üres listát adott vissza. Ebben az esetben értelmezhetetlen a validálás maga, modul által reprezentált eredmény minden értéke a ’NaN’ (not a number) kifejezést kapta, jelezve, hogy a kért keresés nem járt sikerrel.

A Facebook egyik lekérdezési típusa lehetőséget nyújt a követők számának alakulását visszamenőlegesen is megtekinteni. Erre egy három hónapos (93 napot lefedő), offsettel további idősávokra kiterjeszthető időablakot biztosít a múltban. Sajnos ez a lehetőség nem adaptálható tökéletesen a projektemben, ugyanis a tényleges követőszámot csak az általunk birtokolt oldal esetén van mód lekérdezni. Publikus oldalaknál egy trükk szükséges ahhoz, hogy a kért adathoz hozzájussunk. A lekérdezés ugyanis megadja a követők számát a múltban egy-egy adott napon, de mindezt országok szerinti bontásban teszi, így a kapott eredményt még össze kell fésülni. Az így kapott adat több visszamérés után sok helyen drasztikusan eltért a valóságtól, így a használatát végül elvetettem.

Összeségében kevés nehézséggel kellett szembenéznem a Facebook modul implementálása során. MIÉRT NEHÉZ? SOK ÉRZÉKENY ADAT, EZ A LEGSZEMÉLYESEBB. A lefedettség, tehát a sikeres találatok aránya az előre meghatározott csapatnév listán körülbelül 96% volt.

## Twitter crawler modul

A Twitter[[1]](#footnote-1) nevű közösségi oldal számos, közösségi médián alapuló szövegbányászati és kutatási feladat alapanyagául szolgál. Világszerte több mint félmilliárd regisztrált felhasználója van, melyek közül körülbelül 300 millió tekinthető aktívnak. Működésének alapjai a felhasználók által írt, maximum 280 karakter hosszúságú formázatlan üzenetek, az úgynevezett tweet-ek. Ezek a bejegyzések 2017 szeptemberéig 140 karakterben voltak limitálva [5], így a megnövelés lehetőséget biztosít több hasznos információ közlésére, ami az egyes adatgyűjtéseknél is hasznos lehet.

Ugyan a profilok létrehozásakor nem szükséges sok személyes adatot megadni, mégis egy-egy precízen kitöltött adatlap megfelelő forrásként tud szolgálni az adott felhasználó vagy oldal rövid, felületes feltérképezéséhez. Emellett pedig a tweet-ek hosszkorlátozása is segítségünkre lehet, ugyanis ezek a státuszüzenetek garantáltan csupán néhány szóból állnak, amelyek jó eséllyel a közlendő üzenet kulcsszavai. Az alábbiakban a Twitter crawler modulját ismeretem részletesen.

### Twitter API

A Twitteren történő adatgyűjtéshez legelőször egy érvényes fiókra van szükség. Ezután nyílik lehetőség saját alkalmazást regisztrálni. A Twitter sok más, hasonló közösségi oldalhoz hasonlóan OAuth[7] azonosítást használ. Az OAuth egy nyílt azonosítási szabvány, amely tokenek segítségével biztosítja a biztonságos hozzáférést az egyes szolgáltatásokhoz. Az adatgyűjtőhöz a Python nyelvhez készített Twitter modult[8] használtam fel.

Az azonosításhoz szükséges kulcsok az alábbiak:

* consumer\_key
* consumer\_secret
* access\_token
* access\_token\_secret

Az első kettő állandó, míg a tokeneket értelemszerűen időnként újra kell generálni. Mivel az összes felhasznált script esetén szükséges az azonosítás, ezért érdemes ezt a műveletet a külön fájlban (config) megtenni.

### A crawler

Összesen több mint ötvenféle kéréstípus és attribútum érhető el a fejlesztők számára, melyek között vannak keresésre, kapcsolati hálók felkutatására és önálló tweetek kezelésére vonatkozó kérések egyaránt. Ezek közül én az User entitást [9] és a hozzá tartozó kéréseket választottam ki.

Amint azt a Twitter dokumentációja is írja, User bárki és bármi lehet az oldalon. Ennek nyomán az általam vizsgált futballklubok is User-ként jelennek meg a Twitter világában, tehát a hozzá kapcsolódó attribútumok is így érhetők el.

A Twitter fejlesztői dokumentációja részletezi, hogy melyik felhasználói adathoz milyen kérésekkel lehet hozzáférni. Első közelítésben egy adott kulcsszóhoz, kulcsszavakhoz tartozó profilokat kellett összegyűjtenem. Ehhez megvizsgáltam a lehetőségeket a Twitter dokumentációjában.

A számomra ideális kérés pont ezt a funkciót látja el, ugyanis egy vagy több kulcsszóra adja vissza a kapcsolódó profilokat. Ha több kulcsszót adunk meg, akkor azokat a profilokat adja vissza, amelyekben a keresett szavak mindegyike megtalálható.

A keresés során sem időarányos, sem darabszám szerinti limitet nem léptem át. Ez a dokumentáció szerint elvileg száz találatot jelentene. A korlátoknak főleg a tweetek keresésénél van jelentősége, ám ez nem képezte részét a feladatomnak. A választott attribútumokat a 2. táblázat tartalmazza. AKTUALIZÁLNI!

|  |  |  |
| --- | --- | --- |
| Attribútum neve | Magyarázat | Típus |
| name | oldal látható neve | String |
| id | oldal egyedi azonosítója | Int |
| followers\_count | követők száma | Int |
| created at | oldal létrehozásának ideje | Date |
| statuses | tweetek száma | Int |
| url | oldalhoz tartozó link | String |

2. táblázat - A Twitter-ről gyűjtött attribútumok

Az ’expanded url’ meglétét külön kellene ellenőrizni, ugyanis nem kapcsolódik minden tweethez, hiánya pedig hibaüzenetet generál. A sima url paraméter csak a Twitter által rövidített linkeket adja meg, ezért az url esetén elkészítettem egy segédfüggvényt, amely a Twitter által előírt, rövidített (t.co) linkeket[10] kibontja a teljes címmé.

A dátum formátumát szintén átalakítottam egy könnyebben kezelhető, kényelmesebb alakra. Az eredeti dátumformátumra az alábbi sor mutat egy példát.

|  |
| --- |
| Wed Okt 05 21:18:15 +0000 2016 |

Ebből az alábbi időbélyeget generáltam le.

|  |
| --- |
| 2016-10-05 |

A lekérdezések eredményét JSON formátumban kaptam meg, amely már nem tartalmazza az óra és perc adatait.

Miután a Twitter keresője követőszám szerinti csökkenő sorrendben adja vissza a találatokat, a Facebooknál alkalmazott validálási eljárás egyszerűbben alkalmazható, ugyanis elegendő a lista első elemét venni és a későbbiekben ezt kézzel ellenőrizni.

A tesztelés során hasonló problémákba és akadályokba futottam bele, mint az előbbiekben tárgyalt esetben. A lefedettség valamivel kisebb, mint a Facebook esetén, de még itt is 90% fölötti.

## Instagram crawler modul

Az Instagram[[2]](#footnote-2) egy fényképek és rövid videók megosztásán alapuló közösségi hálózat, mely kifejezetten okostelefonról történő használatra készült. A szolgáltatás iOS-ről, Androidról és Windows Phone-ról egyaránt elérhető.

Felhasználóinak száma 2017 szeptemberében lépte át a 800 milliót[11]. Az alábbiakban részletesen ismertetem az Instragram crawler modulját.

### Instagram API

Az Instagram API-ja hasonló elven működik, mint a korábban már bemutatott API-k. Egy bejegyzett fiók birtokában lehetőség nyílik alkalmazást regisztrálni, amihez egy kliensazonosító tartozik. Ezt az azonosítót felhasználva kérhetünk tokent az alkalmazásunkhoz. Az alkalmazás kódjában már csak a kapott tokent kell megadni, ami a Twitter-nél látott, négy különböző API key-t és egyéb tokeneket kérő megoldáshoz képest egy lényegesen egyszerűbb forma.

### A crawler

Az Instagram bányászatakor az InstagramAPI[12] nevet viselő Python modult használtam. Ez lehetőséged biztosít felhasználónévre keresni. A kereső a speciális karakterek jelenlétére érzékenyen reagált ezért ebben az esetben kiemelten szükség van a szűrésre. A keresés eredményét követők száma szerinti csökkenő sorrendben adja vissza, ez pedig megfelel az eddig alkalmazott alap validálási mechanizmusnak. Az így kiválasztott profil bővebb adatait egy következő lekérdezés adja meg, amely a profil egyedi azonosítóját kapja paraméterként.

Az eredményt JSON formátumban adja vissza a program. Az ezzel az eljárással gyűjthető adatok számossága messze nem olyan nagy, mint a korábbiakban ismertetett két modulé, de ez főképp abból fakad, hogy az Instagram lényegesen lazább, kötetlenebb, mint az eddig ismertetett oldalak, tehát jóval kevesebb profiladatot kér. Az Instagramos aktivitást nem csupán a követők vagy követettek száma írja le jól, hanem a közzétett képek és videók mennyisége is, ezért ezt is beválogattam a gyűjtendők közé. Az Instagram modullal gyűjtött attribútumokat a 3. táblázat foglalja össze.

|  |  |  |
| --- | --- | --- |
| attribútum neve | Magyarázat | Típus |
| username | felhasználónév | String |
| id | egyértelmű numerikus azonosító | Int |
| media | közzétett képek/videók száma | Int |
| followers | követők száma | Int |
| followed by | követettek száma | Int |
| url | profilhoz kapcsolható link | String |

3. táblázat - Az Instagram-ról gyűjtött attribútumok

Az említett hat alapadaton felül lehetőség van gyűjteni további értékeket is. Ezeket későbbi bővítési lehetőségként kezeltem.

A három vizsgált közösségi oldal közül az Instagram mutatta a legkisebb lefedettséget. Ez hozzávetőlegesen 80%-ot jelent. Ennek oka szintén a laza formátumban és elnevezési szokásokban keresendő.

## Az adatok tárolása

### A tárolást igénylő adatok kiválasztása

Egy ilyen, adatokkal foglalkozó feladat során, mint amilyen a közösségi média aktivitás megfigyelése, szükség van valamilyen jól konfigurálható adattárolási megoldásra. Ahhoz, hogy ezt a számunkra legmegfelelőbb módon tudjuk eszközölni, szükség van arra, hogy átgondoljuk, pontosan mik azok az adatok, amelyeket érdemes tárolni, és mik azok, amiknek a tárolása irreleváns.

A későbbiekben tárgyalt validálási megoldás hibái és hiányosságai miatt fontosnak tartottam kiépíteni egy cache-t, amiben kezdetben százas nagyságrendben tároltam el népszerű futballcsapatok neveit aszerint, hogy milyen formában szerepelnek az egyes platformokon. Emellett szintén eltároltam az ID értékeket, amiket a három közösségi oldalon használnak.

Ezen kívül a historikus adatok tárolása is hasznos lehet, mivel egyik oldal sem támogatja a korábbi dátumokhoz tartozó támogatószámok visszakeresését. Itt szintén a népszerű csapatoknál van értelme rögzíteni naponta egyszer az aktuális követőszámot.

### Az SQLite rendszer

Az adatok tárolását az SQLite3 nevet viselő relációs adatbáziskezelő rendszerben valósítottam meg. Az SQLite számos programozási nyelvből és fejlesztői környezetből használható kényelmesen, köztük Pythonból is, amely nyelvet a gyűjtőmodulok (és később a megjelenítő modul) is használnak.

Ez a megvalósítás platformfüggetlen és egyszerű felépítésű, ezért könnyen hordozható. Mivel a tárolásra szánt adataim között nincsenek bonyolult kapcsolati rendszerek és maguk a tárolandó adatok sem képviselnek jelentős méretet, ez a pehelysúlyú tárolási megfontolás éppen elegendő a feladatok elvégzésre.

## Az adatgyűjtő modul értékelése

A dolgozatomban bemutatott szolgáltatás első és egyik legfontosabb része az adatgyűjtés. Ennek során közösségi oldalakról gyűjtöttem le a megadott input névre illeszkedő oldalak adatait. A három közösségi oldal vizsgálata során kiválogattam a számomra lényeges attribútumokat.

Ahhoz, hogy ezt el tudjam végezni, megismertem három közösségi oldal API-ját. A gyűjtött adatok egy részét egy jól kezelhető és könnyen feldolgozható formátumban eltároltam. Az adatgyűjtő modult alkalmassá tettem a későbbi bővíthetőségre mind gyűjtött attribútumok, mind további közösségi oldalak tekintetében.

### Statisztikák

Ahhoz, hogy előzetesen visszamérjem az adatgyűjtő modult, készítettem néhány statisztikát a vizsgált csapatokról. Ebből most egy konkrét értéket ismertetek.

A legáltalánosabb és legmeghatározóbb jellemző a közösségi oldalak publikus profiljainál a kedvelők és a követők számának alakulása. Ennek alapján a teszteléshez generált futballklub lista minden elemére lekérdeztem ezeket az értékeket. A 4. táblázat tartalmazza a konkrét számokat a legnépszerűbb csapatokra.

|  |  |  |  |
| --- | --- | --- | --- |
| Név | Facebook kedvelők | Twitter követők | Instagram követők |
| Real Madrid | 101 860 235 | 23 178 122 | 48 795 270 |
| FC Barcelona | 101 772 122 | 21 054 020 | 49 688 397 |
| Manchester United | 73 294 499 | 10 997 183 | 17 456 085 |
| Christiano Ronaldo | 120 810 791 | 51 740 957 | 99 688 397 |

4. táblázat - A legnépszerűbb csapatok követőinek száma

A táblázatban az így kapott toplista első három eleme, a Real Madrid, az FC Barcelona és a Manchester United látható. Christiano Ronaldo összehasonlításként szerepel, ugyanis ő a legnépszerűbb futball témájú entitás mindhárom forrásnál. Az első három helyen minden forrásnál ugyanaz a három klub emelkedik ki a mezőnyből, de még így is messze elmaradnak a legnépszerűbb játékos oldalaitól. További generált statisztikákat a függelék tartalmaz.

TODO SOKKAL TÖBB DOLOG KELL!

VALUE COUNTS-OK, ELDOBANDÓ ATTRIBÚTUMOK!

### Az adatgyűjtés során felmerült nehézségek

# A kapott adatok validálása, a keresett profilok pontos beazonosítása

Ahhoz, hogy az alkalmazás megbízhatóan és pontosan működjön, fontos biztosítani azt a tulajdonságot, hogy a visszaadott profiladatok pontosak és relevánsak legyenek. Ennek megoldására több lehetőséget is véggondoltam, míg több iteráció után kialakítottam egy kellően pontos formulát. A lehetséges alternatívákat a következő rész ismerteti.

## Ötletek, fejlesztési irányok

### Legpontosabb egyezés

Kézenfekvő megoldás az, ha a megadott kulcsszóra, kulcsszavakra a legpontosabban illeszkedő találatot tekintjük a megfelelőnek. Sok esetben valóban működik ez a szisztéma, de a tesztelés során egyre többször futottam bele abba a jelenségbe, hogy a tökéletes egyezés sem elégséges. Például, ha egy csapatnak a rövidített alakja szerepel a hivatalos oldalnál és nem a kézenfekvő, ám hosszú forma, akkor a gyűjtőmodul rendszerint pontatlanul működött.

Official szó jelenléte egy oldal esetén árulkodó lehet, de nem elég általános jelenség. Különösen ritkán fordul elő a kis bázissal rendelkező klubok esetén.

### Legtöbb követővel rendelkező

A másik, valamivel eredményesebb működi mód volt az, amikor a találati listát követők száma szerint csökkenő sorrendbe rendeztem és a lista első elemét tekintettem a keresett oldalnak. Ezt mindhárom forrásra el tudtam végezni, ugyanis a kedvelők és követők száma egy olyan érték, amit az összes forrásnál el tudtam érni.

Általában elégséges is lehetne ez a gondolkodásmód, de viszonylag hamar találkoztam is egy súlyos tévedéssel. A Panathinaikos keresőkifejezés ugyanis a görög kosárlabdacsapatot adja vissza, ugyanis hiába ismert az azonos nevű futballegyesület, a kosár szakosztály európai, sőt, világszinten is lényegesen magasabb szinten jegyzett. Ez a tulajdonság pedig a követők számában is megmutatkozik.

### Cache-ben tárolt, kézzel validált halmaz alapján

A végleges megoldásnak a cache alapú, validált halmazból történő keresést választottam. Ez ugyan időigényes, de csak egyszer kell elvégezni. Ennek oka abban keresendő, hogy egy jól felfuttatott, milliós támogatói bázissal rendelkező oldal kis eséllyel fog nevet változtatni, az ID érték pedig szinte teljesen biztosan megmarad.

A cache-t a legbiztosabban kézi ellenőrzéssel lehet pontossá tenni. A befektetett munka eredménye egy 100%-ban pontos és ellenőrzött halmaz, ami lefedi az összes szükséges attribútumot. Ezt érdemes minden népszerű csapatra elvégezni.

A cache használatával akár az input field is kiiktatható, legördülő vagy ikonos, esetleg hierarchikus menü elég. Emellett fenntartható egy szabadszavas keresési lehetőség is, amiből a gyakorta előforduló keresőkifejezéseket naplózni lehet és idővel beépíteni a cache-be.

Tárolni elegendő a pontos neveket és a hozzájuk kapcsolódó ID értékeket. Az elkészített crawler-ek ezek alapján már el tudják végezni a keresési feladatokat.

## Validáláskor felmerült problémák

A megtalált oldalak validálása az egyik kulcsfeladata az elkészült alkalmazásnak, ezért kiemelten fontos, hogy ez a modul a legjobban működjön. A vizsgálat során tapasztalt érdekes hibaforrásokról az alábbiakban írok.

A szabadszavas keresés elgépelés ellen nem védett. Ezt külön kellene beleépíteni, ami megoldható, de a korábban említett, pontatlanságot okozó nehézségeket nem oldaná meg.

Nehézséget okozhatnak még az egyes névalternatívák, becenevek. Elég csak a Ferencváros-Fradi, esetleg a Tottenham-Spurs név-becenév párokra gondolni, de például a Manchester Unitedre használt „Red Devils” is elég jellegzetes hivatkozás. Ezeket külön kezelni alapos ismereteket igényelne és jelentős többletmunkát okozna. Ugyancsak problémás, hogy szinte az összes csapat nevének van mozaikszavas megfelelője, vagy más rövidítésre. A legkirívóbb ilyen példa az Olympique Lyon nevű francia csapatra használt „OL” profilnév, amely egy hagyományos kereső számára akár túl rövid is lehet.

Vannak olyan kifejezések, amelyek számos egyesület nevében szerepelnek. A legáltalánosabb „Club” szó mellett ilyen még angolszász területeken például a „United”. Konkrét példákat az 5. táblázat tartalmaz.

|  |
| --- |
| Gyakori névelemek |
| FC, SC, FK |
| City |
| United |
| Town |
| Atletico |
| Real |
| Dinamo, Torpedo |
| Lokomotiv |

5. táblázat - Gyakori névelemek futballcsapatok neveiben

Ezeken felül nem jelentenek jelentős megkülönböztető erőt a városok, kiváltképp a fővárosok nevei, mivel azokban jellemzően több csapat is működik. Fontos kezelni ezeket a névelemeket, ugyanis jelenlétük, vagy éppen hiányuk érzékenyen érintheti a modulok működését.

## A validálás értékelése

Összességében megállapítható, hogy bár sokféle fejlesztési irány elképzelhető a gyűjtött adatok validálása során, a fix cache használata tűnik a legpontosabb és legmegbízhatóbb választásnak. Előnye, hogy gyors, pontos, illetve mivel a futballcsapatok száma véges, előbb-utóbb teljes lefedettséget is biztosíthat. Cserébe viszont korlátozott, csak az adott témakörre illik rá, ezért a szabadszavas keresés lehetősége sem elvetendő. Az alkalmazás felajánlható alternatívákkal félig automatizálható marad, ezért egy félig cache-re épülő, félig szabadszavas keresőmodul nyújthatja a legoptimálisabb megoldást.

# Megjelenítés

Az eddig tárgyalt modulokban ismertettem, hogyan működik az általam implementált közösségi média aktivitást figyelő alkalmazásom adatgyűjtő és profilokat ellenőrző modulja. Az így kapott adathalmaz ugyan már pontos és releváns, de egy átlagos felhasználó számára nem több néhány számnál. Fontos lépés a gyűjtött adatok érthető és látványos reprezentálása.

A megjelenítő modul során több alapfeltételt is kitűztem célul. Az egyik, hogy az ábrák legyenek érthetőek. Mivel döntően numerikus adatokat kell vizuálisan ábrázolni, ezért különféle grafikon- és diagramtípusok használatában gondolkodtam.

A másik szempont, amit választottam, az interaktivitás biztosítása. Ne csupán statikus ábrákat tárjak a felhasználók elé, hanem módjuk legyen szűkíteni-bővíteni a látni kívánt attribútumokat, oszlopokat és egyéb elemeket. Az egyes lekérdezések személyre szabhatósága is elengedhetetlen, ezt viszonylag könnyen biztosítani lehet egy bizonyos szabadsági fokig.

A megjelenítő modul tervezése és implementálása során végiggondoltam, hogy melyek lehetnek azok a metrikák, amelyek a legjobban jellemezhetik a közösségi oldalon mutatott aktivitást. Számításba vettem, hogy milyen kiinduló halmazból dolgozhatok. Ezen felül áttekintettem a további bevonható forrásokat, például más adathalmazokat, amelyek hatással lehetnek a csapatok közösségi média oldalainak sikeréhez vagy bukásához. Az alábbiakban a megjelenítő modult ismertetem részletesen.

## Megjelenítési tervek

Az implementálás megkezdése előtt szükség van olyan felületi tervekre, amelyeket előzetesen össze lehet állítani a szükséges elrendezést. Ezeknek az elkészítésére a Moqups[[3]](#footnote-3) nevű webes alkalmazást használtam fel. A Moqups ingyenes hozzáférhető bizonyos korlátozásokkal, például csak 100 objektumot lehet elhelyezni, de ez a célnak éppen megfelelt. TODO

### Összehasonlító felület

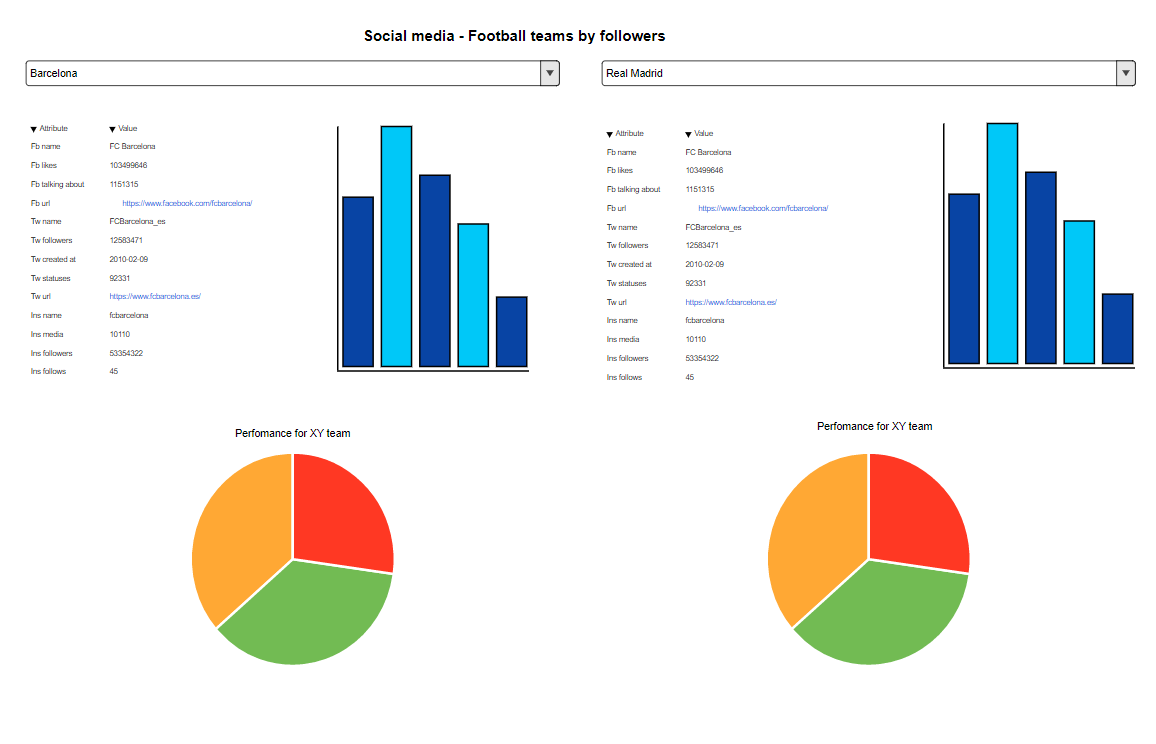
Ezen a felületen két csapat ugyanazon metrikáit lehet megjeleníteni. A csapatok kiválasztásáról egy-egy dropdown menü gondoskodik. Ez a megvalósítás a korábban ismertetett cache alapú tárolást preferálja. Itt szabadszavas keresésre nincs mód, viszont a dropdown menübe szöveget gépelve szűkíthető a kiajánlott lista.

Az első megjelenített elem egy táblázat, amely tartalmazza az adatgyűjtő modul által visszaadott értékeknek az a halmazát, ami elemzésre alkalmas. Tehát például a követőszámokat, közétett képek számát, illetve a profilokhoz kapcsolódó URL-eket. Nem tartalmazzák viszont az egyes oldalak ID értékeit, mivel ezek önmagukban csak a kódszinten történő információ lekérdezésekhez szükségesek és értelmezettek.

Az egyes táblázatok jobboldalán a három platformhoz tartozó követői számok jelennek meg oszlopdiagram formájában.

A táblázat és az oszlopdiagram alatt egy kördiagram reprezentálja az adott csapat aktuális szezonban nyújtott teljesítményét. Az ehhez szükséges adatokat külső adatforrásokból lehet begyűjteni.

A felülethez készült tervet az 2. ábra jeleníti meg.



2. ábra - Összehasonlító felület

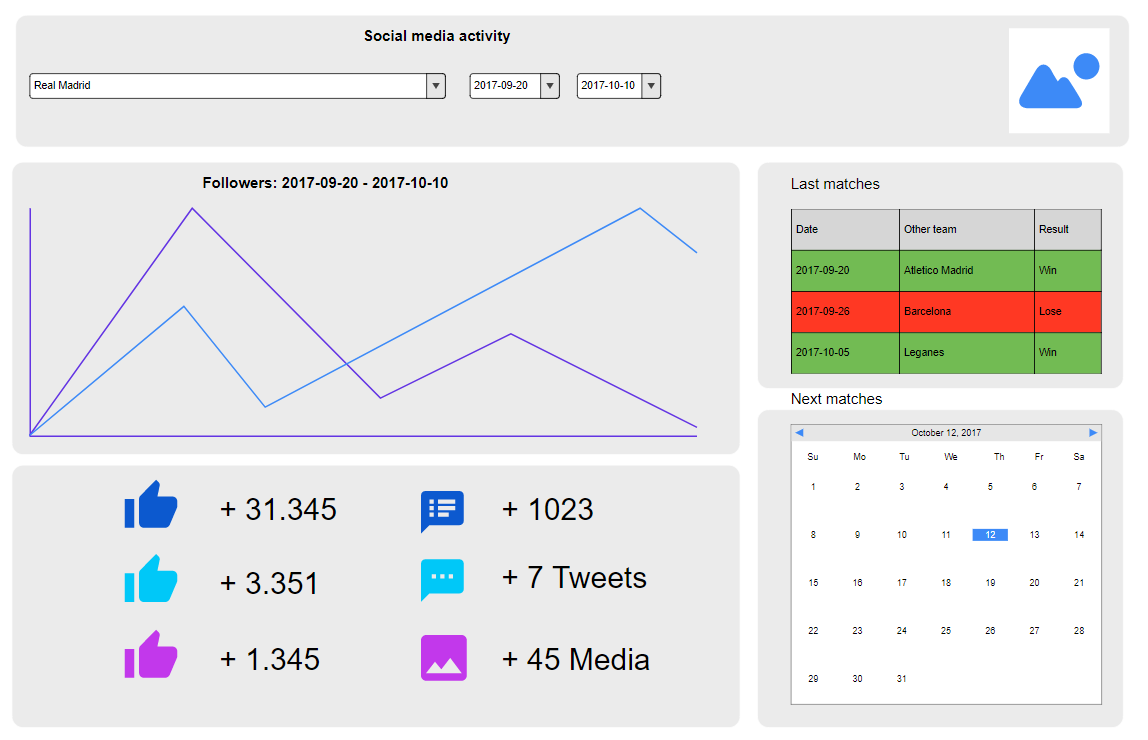
### Időtartományban kereső felület

Ezen a felületen egy csapathoz kaphatók meg időtartományra szűkített adatok.

Az input formon beállítható a kért csapat neve és egy kezdő-, valamint egy végdátum. A vonaldiagram megjeleníti a választott időszakra az egyes platformokon mutatott követői számokat, illetve azok tendenciáját. A diagramtól jobbra az időszakban lejátszott mérkőzések eredményei szerepelnek, az előzőekben említett külső forrásokból összeszedve. Ez alatt egy naptár található, amin a soron következő mérkőzések szerepelnek.

A vonaldiagram alatt a választott időintervallumban végbement relatív követő szám-változások, illetve médiatartalomra vonatkozó értékek kaptak helyet a hozzájuk tartozó ikonokkal együtt. A felhasznált színek utalnak a konkrét közösségi oldalakra.

A felülethez készült tervet az 3. ábra jeleníti meg.



3. ábra - Időtartományban kereső felület

## Plotly

A plotly[13] egy interaktív, böngésző alapú adatelemző és vizualizálást támogató eszköz. Kezdésnek azért esett rá a választásom, mert több programozási nyelvet támogat, köztük a Pythont is. Valamivel többet tud, mint a hagyományos Python-os megjelenítő eszköz, a mathplotlib funkciói, ezért kiindulási alapnak megfelelt.

Noha a plotly alapvetően egy webes felület, létezik offline verziója, mellyel az ábrák és adatok nem egy központi szerveren tárolódnak, hanem a lokális gépen, illetve notebookban.

A plotly másképp viselkedik, mint amit a matplotlib-nél tapasztalhatunk. Itt egy ábrát alapvetően különféle objektumokkal és egymásba ágyazott dict-ekkel definiálhatunk. Az ábra minden egyes jellemzőjéhez tartozik egy kulcs-érték páros.

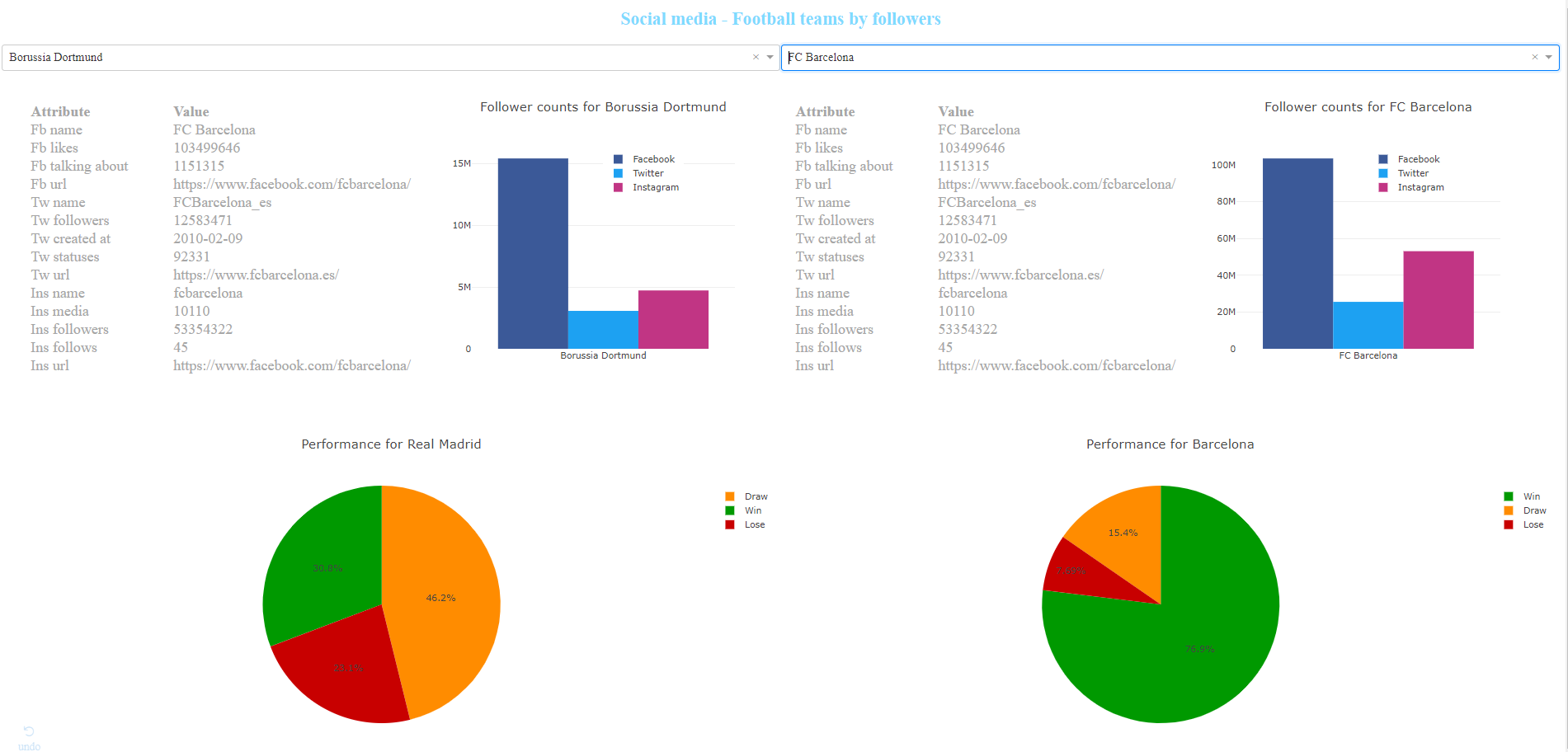
Ezeket a jellemzőket a plotly két kategóriába sorolja, a trace-ekbe és a layout-ba. A trace-ek olyan objektumok, melyek egy adatsort írnak le az ábrán, például egy Scatter objektum. Egy ábrán több trace is lehet, ha például kétféle adatsort is ábrázolni szeretnénk. A trace-ek emellett ábrákon belül is kombinálhatók, egyetlen ábrán megjeleníthetünk többféle diagramot is. A layout jellemzők pedig az ábra egészére vonatkozó formázási utasítások, például az ábra címe, a háttér színe, a tengelyfeliratok, illetve további szövegek.

A Plotly használatával lehetőségünk van a korábban említett Pandas Python modulban használt DataFrame típusú objektumok megjelenítésére is. Ezeket a táblázatos formákat akár egy az egyben is át lehet adni a megfelelő függvényeknek, tehát nem kell például HTML kompatibilis alakra hozni.

### Dash

A Dash[14] egy Python alapú Open Source keretrendszer adatelemzéssel foglalkozó webes alkalmazások készítésére. Modern felhasználói felület elemeket biztosít a grafikonok mellett, ezért egy alapszíntű interaktív megjelenítésre megfelelő. A felületet többségében HTML alapú elemekből lehet összeállítani, így külön más tudást nem igényel. Ezekkel a komponensekkel lehetőség van kellő komplexitású felületeket létrehozni. Emellett a CSS teljes tárházát fel lehet vonultatni a design során.

Az alábbi ábrán látható az összeshasonlító nézet Dash-ban elkészített implementációja.



4. ábra - Az összehasonlító felület Dash-ban elkészítve

## Más megjelenítési módok

HIÁNYOS[[4]](#footnote-4)

# A gyűjtőmodul tesztelése valós adatokon

HIÁNYOS

# Továbbfejlesztési lehetőségek

Dolgozatom utolsó részfeladataként számításba vettem az esetleges továbbfejlesztési irányokat. Az alábbi lehetőségeket érzem reális fejlesztési célnak a jövőre nézve.

* Több attribútumtípus gyűjtése
* Több adatforrás bevonása
* Historikus adathalmaz generálása és felhasználása predikciós célokra
* Más megjelenítési formák és eszközök bevonása

# Összegzés

A közösségi oldalakon, szabadon hozzáférhető publikus tartalom kimeríthetetlen forrást jelent különböző adatbányászati feladatok megoldásához. Gyűjtésükre nehéz egységes és általános megoldást találni, ugyanakkor forrás-specifikus megvalósítások elkészítése különösebb nehézség nélkül megvalósítható.

A közösségi oldalak sosem látott népszerűségnek örvendenek. Minden valamire való cég igyekszik megjelenni a lehető legtöbb platformon. Nincs ez másképp sportcsapatok esetében sem. Az oldalaikról kinyerhető népszerűségi információk hasznosak lehetnek például potenciális befektetők számára.

Ahhoz, hogy felhasználható adataim legyenek, adatgyűjtést végeztem. Ennek elvégzésére adatgyűjtő modult készítettem, amely három különböző közösségi oldalról képes adott input névre illeszkedő profiladatokat gyűjteni. Megismertem a Facebook, a Twitter és az Instagram publikus API-ját. Elkészítettem egy-egy gyűjtőmodult mind a három közösségi oldalhoz. Elkészítettem egy fő modult, ami összefésüli az egyes gyűjtőmodulok eredményeit.

A releváns találatok biztosítása érdekében végiggondoltam a profilok validálás lehetőségeit. Több megoldást is számításba véve kialakítottam a legpontosabb eredményt biztosító megoldást. Az így összegyűjtött adatok tárolására olyan megoldást javasoltam, amely elősegíti a későbbi feldolgozást. A modulok segítségével lehetővé tettem statisztikák generálását.

A megjelenítő modulhoz látványterveket készítettem, amelyek alapján implementálhatók az egyes felületek. Megismertem több Python-os megoldást a vizualizálásra. Implementáltam egy dashboard-ot a kidolgozott látványterv alapján.

Végül megoldásokat javasoltam a szoftver továbbfejlesztésére.

# Függelék

## cache-ben tárolt csapatok és főbb mutatóik

(Gyűjtés időpontja: 2017. 12.05.)

|  |  |  |  |
| --- | --- | --- | --- |
| team | facebook\_likes | twitter\_followers | instagram\_followers |
| 1. FC Köln | 754034 | 590424 | 152611 |
| 1. FSV Mainz 05 | 441916 | 171919 | 44867 |
| 1899 Hoffenheim | 274598 | 176032 | 68820 |
| AC Milan | 24827964 | 6160093 | 3854248 |
| ACF Fiorentina | 2045951 | 587158 | 227757 |
| AFC Bournemouth | 354723 | 332210 | 170530 |
| AS Monaco | 5004969 | 1928366 | 760077 |
| AS Nancy | 143797 | 80138 | 20667 |
| AS Roma | 8821951 | 1506869 | 1223651 |
| AS Saint-Étienne | 818632 | 651933 | 101138 |
| Angers SCO | 171420 | 85065 | 28166 |
| Arsenal FC | 37947647 | 12279656 | 226296 |
| Atalanta | 200054 | 250742 | 75315 |
| Athletic Bilbao | 53273 | 13100 | 166179 |
| Atlético Madrid | 13746278 | 45547 | 3865185 |
| Bayer Leverkusen | 2700359 | 398553 | 309958 |
| Bayern München | 43597794 | 4146627 | 11259013 |
| Bologna FC | 546748 | 139608 | 55488 |
| Bor. Mönchengladbach | 980938 | 451140 | 216438 |
| Borussia Dortmund | 15409612 | 3069987 | 4723298 |
| Burnley FC | 11602 | 298029 | 122449 |
| CA Osasuna | 249374 | 207467 | 31445 |
| CD Alavés | 110939 | 167100 | 44765 |
| CD Leganés | 52375 | 204076 | 30268 |
| Cagliari Calcio | 337003 | 312354 | 117049 |
| Celta Vigo | 474104 | 363307 | 104299 |
| Chelsea FC | 47864196 | 10841090 | 10474711 |
| Chievo Verona | 121099 | 239155 | 32695 |
| Crystal Palace | 1060249 | 640507 | 221745 |
| Deportivo La Coruña | 399589 | 309797 | 73641 |
| Dijon FCO | 128352 | 74328 | 18772 |
| EA Guingamp | 226021 | 236980 | 22206 |
| Eintracht Frankfurt | 619026 | 412687 | 122597 |
| Empoli FC | 104395 | 127623 | 34187 |
| Espanyol Barcelona | 399678 | 334849 | 78459 |
| Everton FC | 3042846 | 1381271 | 510329 |
| FC Augsburg | 247432 | 184989 | 60247 |
| FC Barcelona | 103521788 | 25527780 | 53079727 |
| FC Crotone | 121867 | 18396 | 40691 |
| FC Ingolstadt 04 | 86026 | 72160 | 27278 |
| FC Lorient | 241772 | 246674 | 38502 |
| FC Metz | 201665 | 133142 | 28968 |
| FC Nantes | 348541 | 442596 | 75153 |
| FC Schalke 04 | 2897519 | 659242 | 406423 |
| Genoa CFC | 283516 | 287724 | 79835 |
| Girondins Bordeaux | 791754 | 316017 | 53075 |
| Granada CF | 356038 | 322071 | 42576 |
| Hamburger SV | 53701 | 658923 | 168524 |
| Hertha BSC | 359482 | 314663 | 91540 |
| Hull City | 1014946 | 392137 | 105652 |
| Inter | 16879743 | 1481759 | 1669263 |
| Juventus | 30333370 | 810341 | 8368040 |
| Lazio Roma | 787633 | 509451 | 217296 |
| Leicester City | 6618060 | 1079759 | 1865348 |
| Lille OSC | 748551 | 613193 | 84041 |
| Liverpool FC | 5058874 | 9066224 | 5038804 |
| Manchester City | 27856572 | 5279780 | 5674329 |
| Manchester United | 73779232 | 15852334 | 19729490 |
| Middlesbrough FC | 443069 | 261664 | 65881 |
| Montpellier HSC | 295575 | 304104 | 43711 |
| Málaga CF | 1057873 | 617882 | 73224 |
| OGC Nice | 441887 | 359447 | 122225 |
| Olympique Lyon | 2723691 | 1401101 | 490632 |
| Olympique Marseille | 4816164 | 2662901 | 62104 |
| Paris Saint-Germain | 32207564 | 5772182 | 10488852 |
| Pescara Calcio | 130641 | 49975 | 30252 |
| RB Leipzig | 344037 | 139145 | 112064 |
| Real Betis | 833292 | 518127 | 223907 |
| Real Madrid | 105975730 | 27568120 | 53804831 |
| Real Sociedad | 1197989 | 16891 | 82258 |
| SC Bastia | 289236 | 295024 | 28242 |
| SC Freiburg | 238481 | 240198 | 32915 |
| SD Eibar | 70531 | 169743 | 33298 |
| SM Caen | 162562 | 176034 | 28292 |
| SSC Napoli | 4004165 | 1261523 | 722957 |
| SV Darmstadt 98 | 158874 | 90215 | 42030 |
| Sampdoria | 296429 | 310218 | 85848 |
| Sassuolo Calcio | 255992 | 239937 | 78193 |
| Sevilla FC | 1652075 | 894158 | 336239 |
| Southampton FC | 1728262 | 888212 | 313374 |
| Sporting Gijón | 286333 | 43536 | 38722 |
| Stade Rennais | 434213 | 301415 | 56994 |
| Stoke City | 1304226 | 873809 | 284775 |
| Sunderland AFC | 1698777 | 800637 | 176227 |
| Swansea City | 1348861 | 850533 | 204508 |
| Torino FC | 71798 | 21804 | 50186 |
| Tottenham Hotspur | 8618244 | 2504752 | 1525344 |
| Toulouse FC | 356198 | 443214 | 45385 |
| UD Las Palmas | 229500 | 240227 | 93150 |
| US Palermo | 324196 | 181747 | 60742 |
| Udinese Calcio | 459724 | 264258 | 63625 |
| Valencia CF | 3334130 | 1053559 | 366087 |
| VfL Wolfsburg | 1118994 | 266195 | 251145 |
| Villarreal CF | 880413 | 377320 | 105750 |
| Watford FC | 392107 | 394577 | 173197 |
| Werder Bremen | 1002100 | 20864 | 143871 |
| West Bromwich Albion | 796628 | 764095 | 165393 |
| West Ham United | 2297916 | 1330110 | 474809 |
| Santos | 3556643 | 2598717 | 522496 |
| Fenerbahce | 9717237 | 6301733 | 2760795 |
| Galatasaray | 12978703 | 273306 | 3936375 |
| AL Ahly SC | 11434090 | 2432527 | 2884457 |
| Besiktas | 6053268 | 3302602 | 1710888 |
| New York Red Bulls | 1016787 | 193904 | 111516 |
| Los Angeles Galaxy | 1848454 | 423337 | 453795 |
| Boca Juniors | 8300064 | 3097118 | 1971370 |
| River Plate | 8554728 | 2589838 | 1542978 |

# Irodalomjegyzék

1. <http://www.digitalhungary.hu/kozossegi-media/Mennyi-idot-toltunk-el-a-kozossegi-mediaval/1722/>, [Hozzáférés: 2017. október]

1. <http://pandas.pydata.org/pandas-docs/version/0.16.2/>, [Hozzáférés: 2017. szeptember]

1. <https://docs.djangoproject.com/en/1.11/ref/utils/> [Hozzáférés: 2017. október]

1. <https://developers.facebook.com/docs/graph-api> [Hozzáférés: 2017. november]

1. <https://developers.facebook.com/docs/graph-api/using-graph-api/> [Hozzáférés: 2017. október]
2. <https://www.theguardian.com/technology/2017/nov/08/twitter-to-roll-out-280-character-tweets-to-everyone> [Hozzáférés: 2017. november]

1. <https://dev.twitter.com/oauth> [Hozzáférés: 2017. szeptember]

1. <https://github.com/bear/python-twitter> [Hozzáférés: 2017. szeptember]

1. <https://dev.twitter.com/overview/api/users> [Hozzáférés: 2017. szeptember]

1. <https://help.twitter.com/en/using-twitter/url-shortener> [Hozzáférés: 2017. szeptember]

1. <http://mashable.com/2017/09/25/instagram-800-million> [Hozzáférés: 2017. szeptember]

1. <https://github.com/facebookarchive/python-instagram> [Hozzáférés: 2017. szeptember]

1. <https://plot.ly/python/> [Hozzáférés: 2017. október]

1. <https://plot.ly/products/dash/> [Hozzáférés: 2017. október]

1. <https://twitter.com/> [↑](#footnote-ref-1)
2. <https://www.instagram.com/> [↑](#footnote-ref-2)
3. <https://app.moqups.com/> [↑](#footnote-ref-3)
4. A következő félév anyagát képezi [↑](#footnote-ref-4)