



ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

**NEUMANN JÁNOS
INFORMATIKAI KAR**



SZAKDOLGOZAT

**OE-NIK
2021**

Hallgató neve:
Hallgató törzskönyvi száma:

**Munkácsi Zoltán
T/006318/FI12904/N**

Óbudai Egyetem
Neumann János Informatikai Kar
Kiberfizikai Rendszerek Intézet

SZAKDOLGOZAT FELADATLAP

Hallgató neve:	Munkácsi Zoltán
Törzskönyvi száma:	T/006318/FI12904/N
Neptun kódja:	DSFH2V

A dolgozat címe:

A hiányzó láncszem megtalálása Football Manager segítségével
Find the missing link via Football Manager

Intézményi konzulens:	Légrádi Gábor
Külső konzulens:	

Beadási határidő:	2022. május 15.
-------------------	-----------------

A záróvizsga tárgyai:	Számítógép architektúrák Big Data és üzleti intelligencia specializáció
-----------------------	---

A feladat

Vizsgálja meg a jelen trendeket az adatalapú játékosmegfigyelésben! Ismertesse az adatelemzés elterjedésének kihatását a futball világra! Mutassa be, hogyan használják a futball klubok a Football Manager statisztikáit a való életben! Tárolja a Football Manager játékos adatait relációs adatbázisban! Határozzon meg módszereket a csapat számára hiányzó kvalitású játékosok megtalálására! Fejlesszen ki és teszteljen le egy, az adatbázist felhasználó, és ezen módszereket megvalósító rendszert! A rendszerben legyen elérhető a játékosok listázása, szűrők és megadható paraméterek alapján! A rendszer legyen képes játékosokat ajánlani megadott paraméterek alapján, akiket érdemes lehet megfigyelni a rendszert használó csapatnak!

A dolgozatnak tartalmaznia kell:

- a jelen trendek ismertetését az adatalapú játékosmegfigyelés terén,
- a játékosok pontozásának részletes leírását, magyarázatát,
- a rendszertervet, mely tartalmazza a funkciókat és az adatmodellt,
- a választott fejlesztő eszközök ismertetését és a választás indoklását,
- a tesztelés terveit és az eredményeket,
- az elért eredményeket,
- a felhasználási lehetőségeket.



Fleiner Rita
.....
Dr. Fleiner Rita
intézetigazgató

A szakdolgozat elévülésének határideje: **2024. május 15.**
(OE TVSz 55.§ szerint)

A dolgozatot beadásra alkalmasnak tartom:

.....
külső konzulens

Légrádi György
.....
intézményi konzulens



HALLGATÓI NYILATKOZAT

Alulírott hallgató kijelentem, hogy a szakdolgozat / diplomamunka saját munkám eredménye, a felhasznált szakirodalmat és eszközöket azonosíthatóan közöltem. Az elkészült szakdolgozatomban / diplomamunkámban található eredményeket az egyetem és a feladatot kiíró intézmény saját céljára térítés nélkül felhasználhatja.

Budapest, 2022.05.13


hallgató aláírása



KONZULTÁCIÓS NAPLÓ

Hallgató neve: MUNKA'CSI ZOLTÁN Neptun kód: DSTH2V Tagozat: NAPPALI

Telefón: +3630 193 08 13 Levelezési cím (pl: lakcím): 5008, SZOLNOK, POLLACK MIHÁLY UTCA 9

Szakdolgozat / Diplomamunka¹ címe magyarul: A HIÁNYZÓ LÁNCZEM MEGTALÁLÁSA FOOTBALL MANAGER SEGÍTSÉGÉVEL

Szakdolgozat / Diplomamunka² címe angolul: FIND THE MISSING LINK VIA FOOTBALL MANAGER

Intézményi konzulens: LEGRADI GÁBOR Külső konzulens:

Kérjük, hogy az adatokat nyomtatott nagybetűkkel írja!

Alk.	Dátum	Tartalom	Aláírás
1.	2021.10.21	TEMA MEGBESZÉLÉSE	Legradi Gábor
2.	2021.10.30	IRODALOMKUTATÁS ELKEZDÉSE	Legradi Gábor
3.	2021.11.20	IRODALOMKUTATÁS BEFEJEZÉSE	Legradi Gábor
4.	2021.12.02	PROJEKT ÉRTÉKELÉSE	Legradi Gábor

A Konzultációs naplót összesen 4 alkalommal, az egyes konzultációk alkalmával kell láttamoztatni bármelyik konzulenssel.

A hallgató a Szakdolgozat I. / Szakdolgozat II. (BSc) vagy Diplomamunka 1 / Diplomamunka 2 / Diplomamunka 3 / Diplomamunka 4³ tantárgy követelményét teljesítette, beszámolóra / védésre⁴ bocsátható.

Budapest, 20.12.12.08...

Legradi Gábor
intézményi konzulens

¹ Megfelelő aláhúzendő!

² Megfelelő aláhúzendő!

³ Megfelelő aláhúzendő!

⁴ Megfelelő aláhúzendő!



KONZULTÁCIÓS NAPLÓ

Hallgató neve: MUNKÁCSI ZOLTÁN Neptun kód: D5TH2V Tagozat: IVAPPALI

Telefon: +36 30 153 0813 Levelezési cím (pl: lakcím): 5008, SZOLNOK, POLLACK MIHÁLY UTCA 9

Szakkolgozat / Diplomamunka¹ címe magyarul: A HIÁNYZÓ LÁNCZÉK MEGTALÁLÁSA FOOTBALL MANAGER SEGÍTSÉGÉVEL

Szakkolgozat / Diplomamunka² címe angolul: FIND THE MISSING LINK VIA FOOTBALL MANAGER

Intézményi konzulens: LEGRADI GÁBOR Külső konzulens:

Kérjük, hogy az adatokat nyomtatott nagybetűkkel írja!

Alk.	Dátum	Tartalom	Aláírás
1.	04.15	SPECIFIKÁCIÓ PONTOSÍTÁSA	Legradi Gábor
2.	04.22	FELSZÉRTÉS ÁLLAPOTÁNAK ÉRTÉKELÉSE	Legradi Gábor
3.	04.29	TOVÁBBI KÉRDÉSEK ÁTBESZÉLÉSE	Legradi Gábor
4.	05.06	EREDMÉNYEK ÉRTÉKELÉSE	Legradi Gábor

A Konzultációs naplót összesen 4 alkalommal, az egyes konzultációk alkalmával kell láttamoztatni bármelyik konzulenssel.

A hallgató a Szakkolgozat I. / Szakkolgozat II. (BSc) vagy Diplomamunka 1 / Diplo-mamunka 2 / Diplomamunka 3 / Diplomamunka 4³ tantárgy követelményét teljesítette, beszámolóra / védésre⁴ bocsátható.

Budapest, 20 22.05.10

Legradi Gábor
intézményi konzulens

¹ Megfelelő aláhúzendő!

² Megfelelő aláhúzendő!

³ Megfelelő aláhúzendő!

⁴ Megfelelő aláhúzendő!

Tartalomjegyzék

1. RÖVID TARTALMI ÖSSZEFOGLALÓ A TÉMA TERÜLETÉRŐL, A FELADATRÓL	10
1.1. Bevezetés.....	10
1.2. Terület bemutatása	11
2. A MEGOLDANDÓ PROBLÉMA MEGFOGALMAZÁSA	13
2.1. A hatékony játékosmegfigyelés fontossága	13
2.2. Adataalapú döntéstámogatás megjelenése a futballban	13
2.2.1. Adataalapú játékosértékelés	14
2.2.2. Előnye a hagyományos módszerekkel szemben	15
2.2.3. A játékosok piaci értékének megbecsülése adatok alapján	16
2.3. Adatok forrásai.....	16
2.3.1. Football Manager.....	17
2.3.2. Adatok elérése.....	18
3. A SZAKDOLGOZAT CÉLJA	19
3.1. Célkitűzés.....	19
3.2. Elérendő állapot	20
3.3. Célként kitűzött felhasználási lehetőségek.....	20
4. AZ IRODALOM ALAPJÁN A LEHETSÉGES MEGKÖZELÍTÉSI MÓDOK ÁTTEKINTÉSE ÉS KIVÁLASZTÁSA	21
4.1. Megközelítési módok	21
4.2. Piacon elérhető hasonló szolgáltatások	21
4.3. Eltérés az eddigi megoldásokhoz képest.....	22
5. A RÉSZLETES SPECIFIKÁCIÓ LEÍRÁSA.....	23
5.1. Vázlatos rendszerterv	23
5.2. Adatabázis.....	25
5.2.1. Lehetőségek áttekintése	25
5.2.2. Választás	27
5.3. Webalkalmazás	28
5.3.1. Lehetőségek áttekintése	28
5.3.2 Választás	30
5.4. Gépi Tanulás Komponens	31

5.4.1. Választás	31
5.5 Adatbetöltő script	31
6. A TERVEZÉS SORÁN VÉGZETT MUNKAFÁZISOK ÉS TAPASZTALATAIK LEÍRÁSA.....	32
6.1 Adatbázis tervezése	32
6.2 Django webalkalmazás tervezése.....	33
6.3 Gépi tanuló komponens tervezése.....	33
6.4 Adatbetöltő script tervezése	34
7. MEGVALÓSÍTÁS LEÍRÁSA	35
7.1 Adatbázis kialakítása.....	35
7.2 Webalkalmazás kialakítása	36
7.2.1 Környezet kialakítása.....	36
7.2.2 Django projekt felépítése	37
7.2.3 Felhasználói felület	38
7.2.4 Logika	42
7.3 Gépi tanuló komponens.....	42
7.3.1 Játékos ajánlás.....	43
7.3.2 Alulértékelt játékosok keresése	44
7.4 Komponensek közötti kapcsolatok kialakítása	45
7.4.1 Adatbázis – Webalkalmazás	45
7.4.2 Adatbázis – Gépi tanuló komponens	46
7.4.3 Webalkalmazás – Gépi tanuló komponens.....	46
7.5 Adatbetöltő script.....	47
8. TESZTELÉS.....	48
8.1 Tervezés	48
8.2 Adatbázis.....	48
8.3 Webalkalmazás és Gépi Tanuló komponens tesztelése	48
8.4 Adatbetöltő tesztelése.....	48
9. AZ EREDMÉNYEK BEMUTATÁSA, ÉRTÉKELÉSE	50
9.1 Eredmények.....	50
9.1.1 Komponensek	50
9.2 Értékelés.....	50
9.2.1 Felhasználási lehetőségek	51

10. MEGVALÓSÍTÁS ELEMZÉSE, TOVÁBBFEJLESZTÉSI LEHETŐSÉGEK.....	52
Továbbfejlesztési lehetőségek.....	52
11. A SZAKDOLGOZAT TARTALMI ÖSSZEFOGLALÓJA MAGYAR ÉS ANGOL NYELVEN.....	53
Magyar nyelven.....	53
In english.....	53
12. IRODALOMJEGYZÉK	54
13. MELLÉKLETEK.....	56
1. Melléklet – Játékosok pontszámai	56
2. Melléklet – Egységtesztek leírása, eredménye	58
Webalkalmazás üzleti logikájának egységtesztjei	58
Gépi Tanuló komponens üzleti logikájának egységtesztjei.....	59

1. RÖVID TARTALMI ÖSSZEFOGLALÓ A TÉMA TERÜLETÉRŐL, A FELADATRÓL

1.1. Bevezetés

A nagy számú statisztikai adatok gyökeresen megváltoztatták az utóbbi években a futball világát. Csapat szinten, játékos szinten és a globális folyamatokat tekintve is megváltozott a sikerhez vezető út.

Az első mérkőzés, amelyen mérték a játékosok pozíciójának adatait játék közben a 2006-os Világbajnoki döntő volt. Ezeknek az adatoknak a gyűjtésével és feldolgozásával az edzők sokkal jobban ki tudták használni a csapatuk képességeit. Azáltal, hogy képesek voltak minden játékos pontos mozgását követni, sokkal hatásosabb és részletesebb taktikát tudtak kidolgozni. [5]

A pozíciós mérések felhasználása túlmutat a játékosok és a labda helyzetének ismeretén. A mesterséges intelligenciáknak köszönhetően képesek vagyunk előrejelzést adni a meccsek kimenetére, játékos sérülésekre, hogy mennyire térül meg egy játékos megvásárlása befektetési szempontból és a taktikai döntések hatékonyságára is. Ezeket az adatokat a csapatok edzői és vezetőség szakemberei fel tudják használni az edzéseken, a csapatösszeállításban, taktika kiválasztásában és a pályán kívüli tevékenységekben is (rehabilitációs program kialakítása, utánpótlásnevelés). [6]

A Grand View Research kutatása alapján a globális sportanalitikai piac értéke 2020-ban 1 milliárd dollár körül mozog, az előrejelzések szerint 2025-re akár a 4,5 milliárd dollárt is elérheti, ebből az európai labdarúgó elemzési piac értéke 100-200 millió dollár is lehet. Ebből látható, hogy a verseny egyre intenzívebb, és azon csapatok, melyek időben lépnek, komoly versenyelőnyre tehetnek szert: alulértékelt játékosokat vásárolnak, előre jelezhetik az ellenfél gyengepontjait, követhetik a játékosok napi fizikai állapotát, megelőzhetik a sérüléseket, ezáltal felzárkózhatnak a nagyobb költségvetésű klubok mellé. [9]

Később kifejtésre fog kerülni, de a szakdolgozatom célja egy olyan döntéstámogató rendszer létrehozása, ahol az erre megfelelő pénzügyi eszközöket ráfordítani képtelen csapatok is lehetőséget kapnak számukra értékes, de a piac által alulértékelt játékosok megtalálására és ezáltal szerződtetésére.

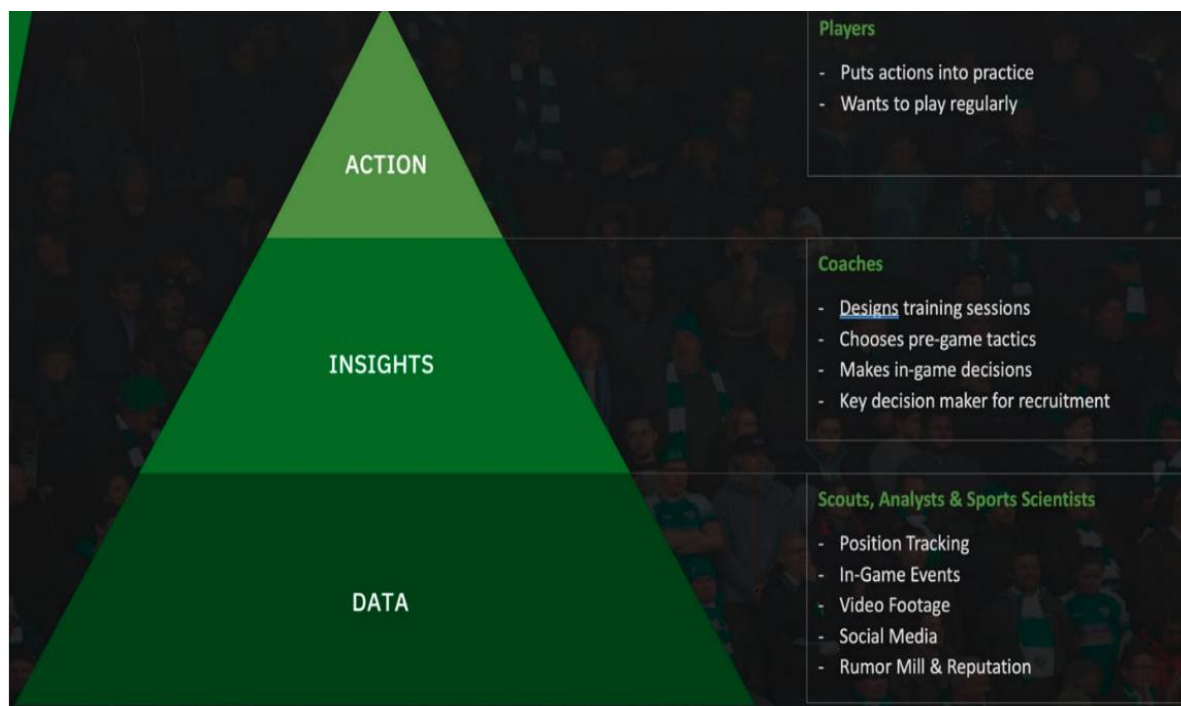
1.2. Terület bemutatása

Egy sportcsapat játékosokkal foglalkozó személyzete az alábbi szerepkörökre bontható (az ellátószemélyzetten kívül):

1. Edzők és a Menedzsment
2. Sport Science¹ és Orvosi stáb
3. Hatékonyság elemzés
4. Utánpótlás biztosítás

Az 1.1 ábrán látható egyszerűsített adatfeldolgozási modell alapján látható, hogy a 2, 3, 4 szerepkörök tagjai alakítják a kutatások eredményét és az összegyűjtött adatokat meglátásokká, tanácsokká. Ezeket a meglátásokat próbálják az Edzői és Menedzsment (1. szerepkör) csoport tagjainak használható formában átadni. Az 1. szerepkör szakemberei ezeket a meglátásokat figyelembe veszik az edzések tervezésekor, taktikai döntéseknél. Konkrét utasítás formájában jut el a játékosokhoz, akiknek gyakorlaskor és meccsen már csak az edzői instrukciókra kell figyelniük. [6]

Ha valamilyen releváns adat összegyűjtésre, majd feldolgozásra kerül, az mindegyik szerepkör szakértőinek felelőssége, hogy eljuthasson az Edzők és Menedzsment csapathoz (1. szerepkör), akik alkalmazhatják az edzések során, a csapatösszeállításakor és a meccstaktikák megválasztásakor. [6]



1.1 Ábra - Szerepkörök és fázisok az adatfeldolgozásban [6]

¹ Sport tudomány

A fenti ábrán látszódik, hogy a begyűjtött és feldolgozott adatok mekkora része és milyen formában jut el az egyes szerepkörökhöz.

Big Data és Üzleti Intelligencia szakirányos hallgatóként a háromszög aljára szeretnék koncentrálni, ami az adatokkal való munkát jelenti. Ezen belül megvizsgálom, hogy a játékosmegfigyelők hogyan tudják felhasználni a rendelkezésükre álló nagy mennyiségű strukturált és strukturálatlan adatot munkájuk során

2. A MEGOLDANDÓ PROBLÉMA MEGFOGALMAZÁSA

2.1. A hatékony játékosmegfigyelés fontossága

Egy csapat számára hatalmas helyzeti előnyt jelent, ha sikerül egy magas kvalitású játékos áron alul szerződtetniük, tehát kevesebb pénzért, mint amennyit a képességei vagy a benne rejlő lehetőségek miatt ér. Ez hosszú távon kifizetődő és kisebb csapatok esetében a klub pénzügyi fenntartásának egyik eszköze is lehet a megfelelő játékosok megszerzése/kinevelése, majd értékesítése.

Egy példa, amely mutatja ennek fontosságát:

A pénzügyi gondokkal küzdő angol Birmingham City FC 2020 júliusában a fiatalok között szerződtetett és saját akadémiáján nevelkedett, 17 éves játékosát (Jude Bellingham), a német Borussia Dortmund-hoz engedte el 25 Milliós angol font ellenében. Az átigazolásért kapott összeg megmentette a csapatot a csődtől és a sportágban szokatlan módon visszavonultatták a játékos mezsámát hálájuk jeléül.[10]

2.2. Adataalapú döntéstámogatás megjelenése a futballban

Az utóbbi időben a legjobb sportcsapatok egyre több pénzt fektetnek az adatelemzésbe, ennek megfelelően egyre jobban fejlődnek sport elemzési technikák.

Egy nagy hiány ezen a területen jelenleg, egy olyan platform vagy alkalmazás megléte, ahol a sportszakemberek együtt alkalmazhatják az elemzési technikákat és a fejlett vizualizációkat. [6]

A mesterséges intelligencia bevonása egy újabb mérföldkövet jelent. Az ideális felhasználási módja főként az analitikai hatékonyság javítása és a nagyszámú lehetőségek szűrése. Ezenkívül az adatok pusztán elemzése alapján olyan meglátások felé terelhetik a szakértőket, amelyeket lehet, hogy figyelmen kívül hagytak volna. Egy rugalmas és könnyen konfigurálható eszközt kell a sportszakemberek kezébe adni, amely a mesterséges intelligencia és az adatelemzés erejét felhasználva adataalapú döntések meghozásában segíti őket. [6]

Ezzel a megközelítéssel kockázatok is járnak: különösen a nem megfelelő analitikai módszerek lehetséges alkalmazása vagy a torz adatok használata. Ezek mérsékelhetők képzett elemző szakemberekkel bevonásával, vagy hosszú távon a legjobb módszer kitapasztalásával. [6]

A felhasználóknak képesnek kell lenniük arra, hogy a folyamat részeként más szakértőkkel együttműködve felülvizsgálják vagy megerősítsék a következtetéseket. [6]

Sok futballklub felhasználja az adatokat a döntéshozatali folyamatai során, de ennek módja és mértéke klubonként nagyon eltérő. A legtöbb klub azonban inkább titokban tartja, hogy mit csinál az adatokkal kapcsolatban.

Az elitklubok előnyben részesítik a személyre szabott eszközöket, amelyek bizonyos területeket vizsgálnak, a kisebb kluboknál pedig hajlamosak olyan kész szoftvert választani, amely videofelvételeken alapul. [13]

2.2.1. Adatalapú játékosértékelés

A focisták rangsorolása és a teljesítményük adatalapú értékelése egyre központibb szerepet kap a futball világában. A játékosokat mérőszámok alapján pontozzák és ez alapján összehasonlíthatóvá válik a teljesítményük, rangsorolhatóvá válnak. Meg kell említeni, hogy ezeket a számokat időközönként frissíteni kell, különben gyorsan elavulttá válnak. [14]

Egy ilyen pontozási rendszerben a játékosok teljesítményét el lehet helyezni a hasonló szerepkörben játszó játékosokhoz képest. Ennek több lehetséges felhasználási területe is van:

- A sporttal foglalkozó televíziós csatornák, weboldalak, videojátékok gyártói használják ezeket az adatokat a játékos teljesítmények összehasonlítására, a szurkolók magasabb minőségű szórakoztatása céljából.
- Az edzők és a csapat menedzsment is érdekeltek az adat-vezérelt eszközökben, mivel ezek képesek támogatni a taktikai elemzéseket és követhetővé válik a játékosok pályán nyújtott teljesítménye.
- A játékosmegfigyelők és teljesítmény elemzők folyton keresik a lehető legjobb adat-vezérelt eszközöket, amelyekkel növelhetik a keresett képességekkel rendelkező tehetséges játékosok felfedezésének hatékonyságát. [14]

A játékosmegfigyelők munkáját nagyon nagy mértékben könnyíti meg, ha számadatok alapján tudják összehasonlítani a játékosokat, akár visszamenőleg is. Ez alapján szűkíthetik a megfigyelt játékosok listáját, ezzel erőforrást spórolva. Az adatok másik fontos felhasználási módja, hogy alátámaszthatják a megérzéseiket vagy felülvizsgálatra készíthetik őket. [15]

A szakdolgozatom keretein belül ezeket a számadatokat és az ez alapján elérhetővé váló összehasonlítás lehetőségét szeretném felhasználni, egy olyan rendszert létrehozva, ahol a játékosmegfigyelők képesek szűrni az általuk fontosnak tartott paraméterek alapján, illetve a gépi tanulás segítségével profilra szóló játékosajánlásokat is kaphatnak.

2.2.2. Előnye a hagyományos módszerekkel szemben

Ahhoz, hogy jobban megjósolhassuk, hogy egy játékos valóban rendelkezik-e azokkal a készségekkel, amelyekre a csapatnak szüksége van, az adatelemzés nélkülözhetetlen. A következő három állítás alátámasztja ezt:

- A „hagyományos”, nem adat alapú játékosmegfigyelésben sok a torzítás és az elfogultság. Egy példa erre: Ha egy játékosmegfigyelő, aki éppen a csapat új szabadrúgás rúgóját keresi kilátogatott volna Cristiano Ronaldo első bajnokok ligája meccsére a Real Madrid-ban, akkor minden bizonnyal őt tartaná a lehető legjobb választásnak, mivel két gólt is szerzett szabadrúgásból azon a meccsen. A számok ezzel szemben azt mutatják, hogy „átlagos” arányban képes értékesíteni a szabadrúgásokat, de erről az őt látó játékosmegfigyelőt már nehezen lehetne meggyőzni. [4]
- Az adatelemzés segítségével növelhető a játékosmegfigyelők hatékonysága. Erre jó példa az, ha a szakemberek minden tehetséges játékost meg szeretnék vizsgálni, az közel lehetetlen lenne pusztán azok sokasága miatt. Az adat-vezérelt technikák segítenek leszűkíteni a halmazt a megadott paraméterek szerint (Például: életkor, poszt). Már csak ez a művelet is sok időt és anyagi forrást szabadít fel, amit a még hatékonyabb elemzések tudnak fordítani a szakemberek. [14]
- Az adatelemzés nem növeli a klub költségeit jelentős módon, sőt csökkentheti azokat. A mai, magas átigazolási díjak és játékos bérek miatt a rossz igazolások anyagi kockázatot is jelentenek. A nem az elvárásoknak megfelelően teljesítő játékos posztja továbbra is hiány lesz, a bére terheli a csapat erőforrásait. Lehet, hogy tehetséges játékosról van szó, viszont, ha egy nem megfelelő csapatban hosszú távon rosszul teljesít, az a piaci árának az esését is jelenti, ami már pénzben kifejezhető veszteség a csapat számára.

Az játékosról gyűjtött adatok kulcsfontosságúak abban is, hogy a játékosmegfigyelő, aki kiválasztotta a megfelelő játékost, alá tudja támasztani az érvelését a döntéshozók számára. Ez azt jelenti, hogy a játékos vásárlásában érdekelt tulajdonosoknak, igazgatóknak is betekintést kell kapniuk a felderítés eredményeibe és ez elképzelhetetlen a megfelelő adatok nélkül. [15]

2.2.3. A játékosok piaci értékének megbecsülése adatok alapján

A menedzserek egyik legfontosabb döntései az átigazolásokhoz kapcsolódnak. A játékosok szerződötésekor, eladásakor az átigazolási díjak és a piaci értékek meghatározása mindig nehéz feladat.

A piaci értékek az átigazolási díjak² becsléseiként foghatók fel, így ezek fontos szerepet játszanak a tárgyalásokban.

Korábban ezeket az értékeket futball szakértők becsülték meg, de a közönség egyre inkább beleszól az árak alakulásába. Nem elhanyagolható szempont a játékos reklámértéke, nemzetisége, vallása, ezek a tényezők új szurkolókat vonzhatnak be a klubhoz. Az új szurkolók egyenesen arányosak az új nézőkkel, az új vásárlókkal, ami a beáramló tőke növekedését jelenti. [16]

De a szakértők és tömegek általi értékelések vegyítése nem átlátható folyamat, ritkán frissíthető és nem reprodukálható. Az adatelemzés tehát itt is egy megalapozott alternatívát vagy kiegészítő megközelítést jelenthet.

Az eredmények azt sugallják, hogy a piaci érték adat-vezérelt becslése képes felülkerekedni a tömegek számos gyakorlati korlátján, miközben összehasonlíthatóan pontos számokat állítanak elő. Eredményeink fontos hatással vannak a futballmenedzserekre és a játékosmegfigyelőkre, mivel az adatelemzés lehetővé teszi a piaci érték pontos, objektív és megbízható becslését, amely bármikor frissíthető. [16]

2.3. Adatok forrásai

A játékosmegfigyeléshez nagy mennyiségű strukturálatlan adat áll rendelkezésre: videók az egyes játékosok teljesítményéről, írott riportok, különböző statisztikák elemző weboldalakon. Ezek közül a videók figyelése mára a felderítés utolsó szakaszába tolódott, ahol már az adatok igazolták a játékosban rejlő lehetőségeket és a szakember megnézi a saját szemével is. A statisztikákat és írott riportokat pedig különböző módszerek szerint már számítógépes programok elemzik. [14]

Érdekesség, hogy a mérkőzések statisztikáinak készítésekor sokszor emberi erővel jegyzik fel a történéseket. Ez a feladat még nincs teljesen automatizálva, ám már dolgoznak a megfelelő videó elemző szoftvereken, amelyek megbízható pontossággal rögzítik az eseményeket. [14]

A statisztikák elemzése egy adat-vezérelt teljesítményértékelési mutató kiszámítását jelenti, amely számszerűsíti a játékos teljesítményének minőségét egy adott mérkőzésen. Ez egy összetett feladat, mivel a teljesítmény minőségének nincs objektív és közös

² Ez az a díj, amit a játékos szerződötésekor meg kell fizetni az ő játékjogát tulajdonló csapatnak

definíciója. A szakirodalomban eddig számos adat-vezérelt rangsoroló és kiértékelő algoritmussal találkozhatunk. [14]

A projektem során ilyen számszerűsített teljesítményekhez hasonló adatokkal fogok dolgozni. A Football Manager nevű videójáték adatait fogom felhasználni.

2.3.1. Football Manager

A Football Manager egy foci menedzser szimulációs játék, ahol a játékosok közel 60 szellemi és fizikai képessége számszerűsítve szerepel, amit több mint 1300 játékosmegfigyelő munkája segít. Ennyi ember sokkal átfogóbb információt tud összegyűjteni a játékosokról, mint amilyet képes lenne bármelyik klub egyedül kezelni. Figyelik a játékosokat edzéseken és mérkőzéseken, összehasonlítják és pontozzák őket. Ha egy játékos kiemelkedik a többiek közül magasabb pontszámot kap, amivel felfigyelhet rá egy nagyobb klub. Visszajelzések alapján folyamatosan javítják a pontozási hibákat a játék aktuális kiadása előtt, ezért az adatbázis valós képet ad a játékosok tudásáról. [18]

Míg a legjobb ligákban játszó csapatoknak megvan az a lehetősége, hogy jelentős felderítési költségvetéssel dolgozzanak, a Football Manager értékes erőforrás lehet a kisebb klubok számára. [18]

A futballfelderítők megfigyeléseket végeznek a pályán, aztán elküldik ezeket a jelentéseket a vezetőségnek és ők ellenőrzik őket, vagy lekérnek adatokat egy adatbázisból, ami visszaad nekik játékosokat, akik megfelelnek bizonyos kritériumoknak. Aztán felkérlik a felderítőket ezeknek a játékosoknak a megfigyelésére.

Mindkét módszer működik, felderítés legjobb módja keverni őket. [18]

A felderítés első szakasza nagy merítéssel történik, ahol sok potenciális név kerül a kalapba. Ezen a ponton nem igazán fontos, hogy honnan származik az információ, érkezhetsz ügynököktől is, vagy egy számítógépes játék adatbázisából. Ezután az érdekes játékosokat online elemzik. Akik átmennek ezen a szűrőn, azért kiküldenek egy játékosmegfigyelőt, hogy megnézzék élőben is. [18]

2.3.2. Adatok elérése

A Football Manager nevű játék 2017-es kiadásának játékos-pontszámokat tartalmazó adathalmaza elérhető a Kaggle.com oldalon. Ez fogja képezni a rendszerem alapját játékosok teljesítményére vonatkozó pontszámokat. [20]

A fenti adathalmaz frissített verziója (2020-as) szintén a Kaggle.com oldalon érhető el. Felhasználok a benne szereplő személyes adatokat a játékosokról, hogy színesebb képet lehessen kapni róluk, illetve több szempont szerint lehessen szűrni őket. [21]

Ez a két .csv fájl, külön táblákra bontva fogja adni a rendszerem adatbázisát. Mivel mindkettő relációs adatokat tartalmaz, ezért egy relációs adatbáziskezelő rendszert fogok választani a tárolásához.

3. A SZAKDOLGOZAT CÉLJA

3.1. Célkitűzés

Ahogy az ismertett szakirodalomból kiderül: az, hogy a játékosok mennyire képesek jól együtt játszani, meghatározza a csapat sikerességét. Másképp mondva, a csapat sikeressége azon múlik, hogy egymáshoz és az edző taktikájához mennyire jól illő játékosokat tudnak megszerezni.

Ez azt jelenti, hogy a jó igazolások és a rossz igazolások arányát a lehető legmagasabbra kell emelni. Az, hogy egy igazolás jó-e vagy rossz-e, bevált vagy nem vált be, nem lehet egyértelműen kijelenteni, ha nincs számokkal alátámasztva.

Nem könnyű feladat kiválasztani, hogy egy csapat melyik játékost szeretné szerződtetni, bármilyen posztról legyen is szó. Ha van egy nagyszerű játékos, akinek a világon mindenki tisztában van a kiemelkedő képességeivel, valószínűleg több csapat is szeretné megszerezni, ami extrém magas átigazolási díjat is generálhat. [12]

A játékosmegfigyelők célja az, hogy még a többiek előtt megtalálják a nagy tehetségeket, illetve az ígéretes fiatal játékosokat, ezzel elérve, hogy a piaci értékénél olcsóbban tudják szerződtetni. Az utóbbi időben kezdett elterjedni, hogy ezek a szakemberek adatelemzőkkel együttműködve nagy adathalmazokból kinyert értékes információkkal tudják javítani a munkájuk hatékonyságát. Elképesztő eredményeket lehet elérni, pusztán azzal, hogy az adatelemzést figyelembe veszik. [5]

Ezzel szemben nem teljeskörű az adatalapú játékosmegfigyelés térhódítása. Azok a csapatok, akik nem tudnak megfelelő mennyiségű erőforrást allokálni a játékosmegfigyelők döntéseinek adatalapú támogatására, nem tudják kihasználni ennek az igazi előnyeit. Úgy gondolom, hogy amíg el nem jut oda minden csapat, hogy kellő hatékonysággal tudják használni az interneten fellelhető strukturált és strukturálatlan adatokat, addig ezek a csapatok hátrányban lesznek a többiekhez képest.

A szakdolgozatom keretein belül az ő munkájuk segítésére szeretnék létrehozni egy felületet, amit ingyenesen használhatnak a megfelelő játékosok keresésekor, illetve ajánlásokat is kaphatnak a rendszertől.

Egy olyan adatbázist fogok használni ehhez, ahol a játékosok képességei diszkrét pontszámokkal vannak jellemezve, ezzel elérve, hogy egymáshoz hasonlíthatók lesznek különböző szempontok alapján. A szakemberek számára számok formájában is látszódnia fog, hogy melyik játékos miben erős, illetve gyenge.

3.2. Elérendő állapot

Szeretném elérni a szakdolgozatom késznek tekintéséhez, hogy

- feldolgozzam és bemutassam a rendelkezésre álló forrás-adatokat,
- eltároljam őket egy relációs adatbázisban,
- a lekérdezések optimalizálása érdekében megfelelő indexeket hozzak létre,
- létrehozok egy webes felületet, ahol a felhasználók elérhetik ezeket az adatokat,
- a felületen lehetőségük legyen a paraméterek szerint szűrni, rendezni a játékosokat,
- gépi tanulás segítségével lehetőséget kapjanak játékos ajánlásokat kapni.

A felhasználó által elérhető pontos funkciók a 7.2.3 fejezetben kerültek kifejtésre.

3.3. Célként kitűzött felhasználási lehetőségek

A felhasználók képesek lesznek az adatbázisban található összes játékos pontszámait megtekinteni, és ezeket összehasonlítani más játékosok pontszámaival, ezáltal egyfajta rangsort alkotva közöttük.

A további eltárolt adatok arra is alkalmasak, hogy rendezzük, illetve szűrjük a játékosokat a piaci áruk, nemzetiségük, magasságuk vagy bármely más tulajdonságuk alapján.

Ezek a lehetőségek nagy mértékben le fogják csökkenteni azoknak a játékosoknak a számát, akiket továbbra is érdemes lehet megfigyelni. Ha a szakemberek fel tudnak állítani olyan szempontokat, amelyek alapján kizárhatók a játékosok a további megfigyelésből, akkor ezeket alkalmazni tudják majd. Ez egyértelműen sok megtakarított időt és másra fordítható energiát jelent a számukra.

Ezek mellett lesz lehetőség arra, hogy a rendszer, beállítható paraméterek alapján, képes legyen játékosokat ajánlani, azok profilja alapján és alulértékelt játékosokat keresni. Ezekkel a módszerekkel olyan játékosokkal találkozhatnak a felhasználók, akiket nem biztos, hogy figyelembe vettek volna a keresés során.

4. AZ IRODALOM ALAPJÁN A LEHETSÉGES MEGKÖZELÍTÉSI MÓDOK ÁTTEKINTÉSE ÉS KIVÁLASZTÁSA

4.1. Megközelítési módok

Többféle megközelítés jöhet szóba:

- Meccsek statisztikájának felhasználása, elemzése

Sok csapat használja ezt a módszert az adatelemző csapatuk segítségével. Biztos vagyok benne, hogy ezzel lehet a legnagyobb eredményeket elérni jelenleg, de nekem nem áll rendelkezésemre az a sportszakmai tudás, amivel megfelelő szempontok alapján tudnám elemezni a statisztikákat.

- Videós adatok, írott jegyzetek elemzés

A videóanyagok képfeldolgozás segítségével történő elemzése még a jövőbe mutató irányzat, de mindenképpen nagyon hatásosnak gondolom. A folyószöveg feldolgozására az adott nyelvet értő mesterséges intelligenciák képesek, de egy ilyen elkészítése túlmutatna a szakdolgozatom keretein.

- Kiszámított számértékek használata

Ez a legkevésbé komplex megoldás és ha megfelelő minőségűek az adatok, akkor ezzel a módszerrel is jelentős eredményeket lehet elérni. A felhasználó dolgát is megkönnyíti, hogy számszerűsítve látja, hogy melyik játékos miben erős és miben gyenge.

A három közül az utolsó megoldást fogom választani, mivel ehhez van meg a megfelelő tudásom és ilyen típusú rendszert nem találtam még.

A megoldás lényege, hogy a játékosokat a képességeik alapján kiszámított számértékek alapján lehet rangsorolni és összehasonlítani. Ha nem is pontos statisztikákról van szó, arra alkalmas a megoldás, hogy megmutassa a játékosok közti különbségeket és alapot biztosítson a szűréseknek.

4.2. Piacon elérhető hasonló szolgáltatások

- **WyScout:** Egy olasz cég, amelynek a fő profilja a futball játékosok felderítésének, a meccselemzésnek és az átigazolási dinamikák megértésének segítése. Videóelemző eszközöket és digitális adatbázisokat biztosítanak futball edzőknek, csapatoknak és játékosoknak. A céljuk, hogy a felhasználó részletes képet kapjon nagy számú játékos teljesítményéről a játékmintákról és a taktikákról.

Leginkább videóelemzések tekintetében kimagaslóak, nagyon részletes videó-adatbázisuk van a meccsekről, amely nagy segítséget jelent az elemzések során.

Ez egy fizetős alkalmazás, a legolcsóbb elérhető csomag 250 euró évente, ez havonta 50 percnyi videó elérését biztosítja.

- **InStat:** Egy orosz cég, amelynek a profiljába több sportág is tartozik. Az InStat Scout egy webes platform csapatok, játékosok és játékvezetők teljesítményelemzésére. Hozzáférést biztosít az egyéni statisztikák széles skálájához. A platform innovációja, hogy minden statisztika videóepizódokhoz kapcsolódik. Ez azt jelenti, ha a felhasználó kiválasztja egy játékos utolsó öt gólját, akkor egyből nézheti is a róluk készült képanyagot. A statisztikák nagy adatbázisának és a videóknek köszönhetően, játékosfelderítési lehetőséget is biztosít. Az InStat Scout költsége 2000 euró évente.
- **Transfermarkt:** Egy német cég által készített weboldal, amely egy aktuális általános futball tudásbázis. Nagyon sok információ megtalálható rajta a gólokkal, eredményekkel, statisztikákkal és játékosokkal kapcsolatban. Nagy hangsúlyt fektetnek az átigazolásokra és az átigazolási hírekre is. Kutatások alapján az ott megjelenő átigazolási „pletykák” nagy arányban helytállónak bizonyulnak. A játékosok piaci árát havonta frissítik különböző mutatók alapján és többnyire meg is egyezik a tényleges értékükkel. A Transfermarkt ingyenesen használható.

4.3. Eltérés az eddigi megoldásokhoz képest

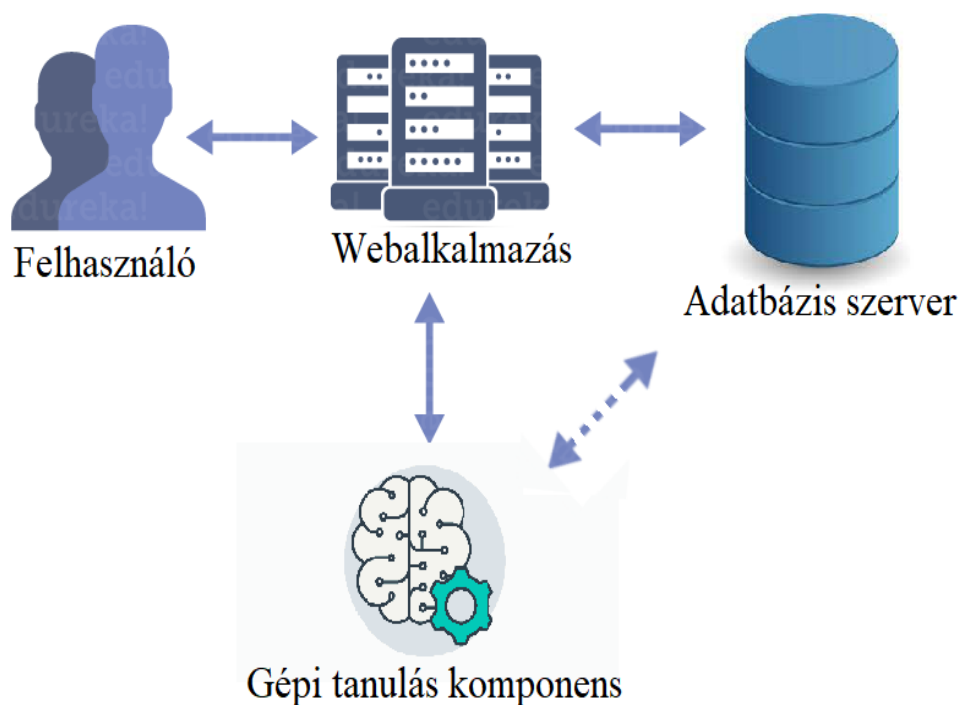
Az eddigi tapasztalataim alapján a hasonló feladatot betöltő weboldalak és alkalmazások mind szoftver által feldolgozott statisztikákon alapulnak, ez alapján teszik lehetővé a játékosok teljesítményének megismerését. A videók és statisztikák elérhetősége is fontos eleme a piacon jelenlévő megoldásoknak.

Az én rendszerem ezzel szemben más megközelítést alkalmaz és más a betölteni kívánt szerepe is. Én szakértők és a közösség által meghatározott pontszámokat fogom alkalmazni, amely nem felel meg a meccs-statisztikát felhasználó megoldások szerepének, hanem a megfelelő potenciállal rendelkező játékosokat lehet vele azonosítani. Ezek után érdemes a kiválasztott játékos statisztikáit is megvizsgálni, de arra vannak elérhető eszközök.

Ezenkívül a gépi tanulás alapú játékosajánlást tekintem a megoldásom nagy erősségének, amire nem láttam még eddig példát, viszont fontos szerepe lehet a megfelelő profilú játékosok azonosításában.

5. A RÉSZLETES SPECIFIKÁCIÓ LEÍRÁSA

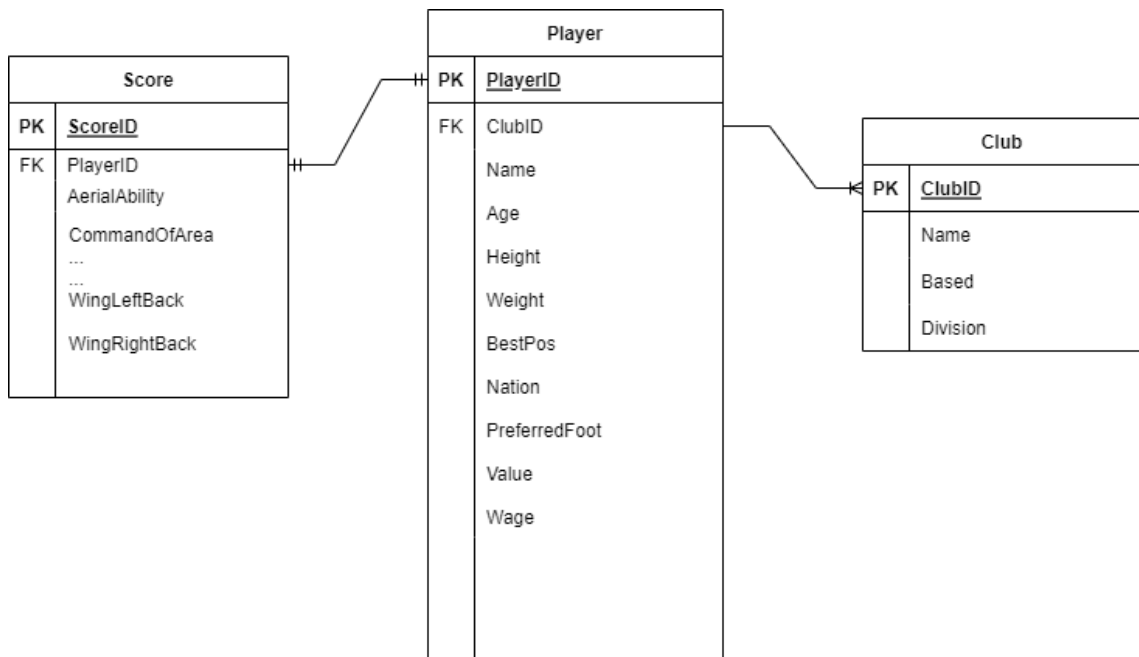
5.1. Vázlatos rendszerterv



5.1 ábra – Vázlatos rendszerterv

A felhasználók a saját böngészőjükből érhetik el a Webalkalmazást, ami kiszolgálja a kéréseket. A hozzákapsolat adatbázis szerverről kéri le az adatokat, amelyeket továbbít a kliensek felé. Ha a felhasználó szűrő értékeket állít be, vagy rendezni akarja a játékosokat, akkor ezek az adatbázis lekérdezésekben fognak érvényesülni és webalkalmazásnak a lekérdezések összeállításán kívül csak továbbítania kell az adatokat.

A Gépi tanuló komponens több különálló programot, amelyek a belső számítások elvégzése után játékosajánlásokat képes végezni. A Webalkalmazástól csak paramétereket fog kapni, az adatforrása szintén az adatbázis. Használat során a Gépi tanuló komponens a Webalkalmazástól kapott paraméter és a belső számítások eredményeként játékos azonosítókat küld vissza a Webalkalmazásnak, amelyeket az egy SQL lekérésbe ágyazva továbbít az adatbázisszervernek és az onnan megkapott adatokat továbbítja a felhasználónak.



5.2 ábra - Adatmodell

5 táblát fog tartalmazni az adatbázis:

- Score:
 - A játékosok pontszámait tartalmazza. Ez alapján lehet őket rangsorolni a felhasználó vagy a Gépi Tanuló Komponens által. Az összes mező nem fért volna rá az ábrára, ezeket az 1. Melléklet-ben tüntettem fel.
- Player:
 - A játékosok személyes adatait tartalmazó tábla. Ezek szerint is lehet szűrni, de főként informáló funkciójuk van.
- Club:
 - A Csapatok találhatók meg ebben a táblában. A nevük, az ország és a liga, amelyben szerepelnek
- Score_hist:
 - Új adat-betöltések során az elavult játékos pontszámok itt tárolódnak egy megfelelő *ValidTo* mezővel kiegészülve, a későbbi historikus riportok lehetőségét biztosítva.
- Player_hist:
 - Új adat-betöltések során az elavult játékos tulajdonságok (például: életkor, testsúly, fizetés) itt tárolódnak egy megfelelő *ValidTo* mezővel kiegészülve, a későbbi historikus riportok lehetőségét biztosítva.

A táblák elsődleges kulcsára indexeket is definiáltam a lekérdezések gyorsítása érdekében.

5.2. Adatbázis

Ahogy 5.2 ábrán is látható a fő funkciókat tekintve 3 tábla lesz az adatbázisban. Mivel az adatbázist csak az adatok elemzésére és megjelenítésre fogom használni, ezért csak olvasni fogok az adatbázisból, írni csak nagyon ritkán, az adatbetöltő script³ segítségével.

5.2.1. Lehetőségek áttekintése

Strukturált adatokat fogok feldolgozni, ezért egy relációs adatbáziskezelőt kell választanom, az alábbi lehetőségeket gondoltam végig:

- **MSSQL**
 - Előnyei:
 - Sok fajta kiadás, különböző árazással, különböző célokra.
 - Teljeskörű üzleti megoldások. Számos üzleti kiegészítőt tartalmaz, mint például az ETL megoldások vagy adat tisztító folyamatok. Ezenkívül biztosít eszközöket teljeskörű adminisztrációhoz, online analitikák számításához, adatbányászathoz és adatvizualizációhoz.
 - Gazdag dokumentáció áll rendelkezésre.
 - Felhőalapú adatbázis támogatás. Az elérhető megoldások: Microsoft Cloud, Azure SQL Database, SQL Server on Azure Virtual Machines. [3]
 - Hátrányai:
 - Az ingyenes verziója, az Express, csak belépő szintű eszközöket kínál. Ha komolyabb projektet kell támogatni vele, akkor másik kiadást kell választanunk, ami magas költségekkel jár.
 - Gyakran változó és nem tiszta licenszfeltételek.
 - Aki nem régóta foglalkozik ezzel a környezettel, nehézkes lehet számára a lekérdezés optimalizálás és a hatékonyság növelése. [3]
- **Oracle**
 - Előnyei:
 - Folyamatos innováció a napi munkafolyamatok megkönnyítésére.

³ A 6.4 fejezetben van kifejtve.

- Erős technikai támogatás és dokumentáció, közösségi támogatás. Ha valami probléma adódik, majdnem biztosan lehet rá találni megoldást.
 - Az Oracle multimodell megoldása hatalmas mennyiségű adat befogadását és feldolgozását teszi lehetővé.
 - Erős szinkron adatfeldolgozás. [3]
- Hátrányai:
 - Az ingyenes kiadások funkciói korlátozottak. Az ennél magasabb szintű kiadásai pedig nagyon magas költségekkel járnak.
 - Erőforrás-igényes technológia. Sok lemezterületet igényel, hatékony infrastruktúrára van szüksége.
 - Megfelelő szakértelem kell a kezeléséhez. [3]
- **PostgreSQL**
 - Előnyei:
 - Nagy vertikális skálázhatóság.
 - Egyéni adattípusok támogatása. Alapértelmezés szerint nagyszámú adattípust támogat, de lehetővé teszi a felhasználók számára, hogy saját adattípusokat hozzanak létre. Ez a NoSQL megközelítés támogatását jelenti.
 - Könnyen integrálható külső eszközök.
 - Nyílt forráskódú a projekt.
 - Erős fejlesztői közösség, ahol segítséget nyújtanak a problémák megoldásában.
 - A bonyolult lekérdezések és az előre definiált funkciókkal megvalósított egyedi interfészek széles választéka miatt a PostgreSQL tökéletesen illeszkedik az adatelemzéshez és az adattárházakhoz.
 - Jelentősen felgyorsítja a hatalmas mennyiségű adat feldolgozását. [3]
 - Hátrányai:
 - Inkonzisztens dokumentáció. Mivel a PostgreSQL közösség meglehetősen elosztott, ezért nem mindenhol egységes a dokumentáció.
 - Nincsenek az adatbázis állapotára vonatkozó ellenőrzési eszközök. [3]
- **MySQL**
 - Előnyei:
 - Ingyenesen használható a személyi használatra, de vannak más lehetőségei is vállalati célokra.

- Egyszerű a felépítése és a kezelése.
- Támogatja a legnagyobb felhőszolgáltatókat, olyan platformokon érhető el, mint az Amazon Web Services, Microsoft Azure. Ez teret adhat a növekedésnek. [3]

○ Hátrányai:

- Sok mérnöki megoldást kíván a skálázhatóság. Ha előre látható, hogy az adatbázis jelentősen fog növekedni, nem ez a megfelelő adatbázis megoldás.
- Mivel az Oracle tulajdonában áll, nem teljesen nyílt forráskódú, ami gátolja a közösségi fejlesztéseket. Ha bármilyen módosítást szeretnénk a környezetben, az komoly költségekkel jár az Oracle felé.
- Nem követi teljesen az SQL szabványokat, nem tartalmaz néhány szabványos szolgáltatást. [3]

5.2.2. Választás

A fenti megoldások közül a PostgreSQL-t találtam a legalkalmasabbnak, szerintem az illik legjobban a projekthez.

- A skálázhatóság fontos, mivel folyamatosan fejlődik ez a terület és egyre növekednek a játékos adatait nyilvántartó adatbankok. Elképzelhető, hogy a jövőben nagyságrendekkel több játékos adatai is rögzítve lesznek.
- Szeretném kihasználni az egyéni adattípus lehetőségét és számomra kézenfekvő adatstruktúrákat létrehozni.
- A bonyolult lekérdezések és az adatelemzés támogatása nagyon fontos lesz a munka folyamán.
- A munkám során szeretnék törekedni a nyílt forráskódú megoldások használatára.

5.3. Webalkalmazás

A Webalkalmazás feladatai:

- Meg kell valósítani egy egyszerű kezelőfelületet a felhasználók számára.
- Kezeleni a Klienstől bejövő kéréseket, SQL lekérdezések formájában továbbítani az Adatbázis felé és az onnan visszajött adatokat, visszaküldeni a Kliensnek.
- Szükség esetén kommunikálnia kell a Gépi Tanuló komponenssel, ami adatot fog neki szolgáltatni. Ezeket fel kell dolgoznia az adatbázis segítségével és továbbítani a kliensnek.

5.3.1. Lehetőségek áttekintése

Az alábbi elterjedt szerveroldali keretrendszerek használatát gondoltam végig:

- **C# ASP.NET**
 - Előnyei:
 - MVC architektúrát követ, ami lehetővé teszi az elválasztását az adatok beolvasásának, feldolgozásának és továbbításának.
 - A Visual Studio (a keretrendszer fejlesztőkörnyezete) nagy segítséget jelent a kódolási idő csökkentésében. Segíti a kód áttekintését, ami a minőségét is javíthatja.
 - A fejlesztőkörnyezetnek köszönhetően nagyon gazdag eszköztárral rendelkezik, amik nagyban könnyítik a fejlesztési munkákat.
 - Könnyen kezelhető alkalmazást lehet vele létrehozni, mivel a konfigurációs állományok változtatásával a szerver újraindítása nélkül is tudjuk változtatni az alkalmazást. [2]
 - Hátrányai:
 - A karbantartásakor figyelni kell, hogy több webszerver-erőforrást igényel, mint a hasonló megoldások.
 - Nincs minden átfogóan dokumentálva.
 - Magas hozzáértés kell a használatához, illetve többféle technológiának az ismerete szükséges. [2]
- **Python Django Framework**
 - Előnyei:
 - ORM (Objektum-Relációs leképezés) támogatás, ami megkönnyíti az adatbázis kezelését.

- Nagyon jó biztonsági megoldásokat biztosít, mint például megakadályozni a kód futtatást a megjelenített weblapon.
 - Nagyon sok kész megoldás van, amit hozzá lehet csatlakoztatni és nem a fejlesztőnek kell megírni. Néhány példa ezekre:
 - Bejelentkezés kezelése
 - Adminisztrátor kezelőfelület
 - Munkamenet kezelése
 - Külön PostgreSQL csomag, plusz funkciókkal
 - Nyílt forráskódú rendszer nagy aktív felhasználói közösséggel.
 - Megkönnyíti az alaptól komplex, adatbázisra támaszkodó weboldalak írását. [7]
- Hátrányai
 - Kis projektekhez, ahol nincsenek kihasználva az előnyei, feleslegesen nagy szoftver.
 - Ha túl sok adattáblát használunk, bonyolult lehet a kezelése, mert a keretrendszer által kezelt adatmodellek egyetlen fájlban tárolódnak.
 - Bonyolult problémákba lehet belefutni, ha eltérünk a keretrendszer által javasolt megoldásoktól. [7]
- **Java Spring Framework**
 - Előnyei:
 - Ez egy könnyű konténer megoldás, ami azt jelenti, hogy nem kell a fejlesztőnek osztályokat örököltetni és interfészeket implementálni a kódolás során. Nagy teljesítményű, könnyen skálázható alkalmazásokat tudunk vele létrehozni.
 - Előre megírt megoldásokat tartalmaz, amelyeket a fejlesztők egyértelműen megbízhatónak tartanak. Ezeket saját paraméterek szerint lehet konfigurálni.
 - A keretrendszer fejlesztői nagy hangsúlyt fektetnek a gyorsaságra.
 - Több biztonsági megoldást is tartalmaz, mint például, hogy figyel a külső forrásból származó függőségeket. A Java Spring alkalmazásával teljesen biztonságos webalkalmazást tudunk létrehozni. [11]
 - Hátrányai:
 - Komplex a fejlesztése, nagy szakértelmet igényel. Aki még nem használta ezt a keretrendszert, először tapasztalatot kell benne szereznie, mielőtt ténylegesen tudná használni.

- Nem sok útmutató és irányelv tartozik a fejlesztéséhez. Sokszor magukra vannak utalva a fejlesztők. [11]

- **Ruby on Rails**

- Előnyei:

- Könnyű megtanulni ennek a keretrendszernek az alapjait, de időbe fog telni amíg valaki megtanulja kihasználni a tényleges előnyeit.
 - Sok előre megírt eszköz található benne, amit használhatnak a fejlesztők. Ezek nagy segítséget adnak kódolási hibák megtalálásához, az optimalizáláshoz és a teszteléshez.
 - Nagy és aktív fejlesztői közösség, akik segítséget tudnak nyújtani a problémák megoldásakor.
 - A bevett szoftvertervezési minták követésével nagyon hatékony és jól működő megoldásokat lehet létrehozni. [19]

- Hátrányai:

- A rugalmasság hiánya. Ami az alapvető feladatokat és funkciókat illeti, a Ruby on Rails szinte kifogástalan. Mivel azonban sok az alapértelmezett, előre beállított objektum, nincs sok választási lehetőség egy probléma megoldása során. Tehát a keretrendszer választás előtt át kell gondolni az igényeket és hogy ennek megfelelnek-e a beépített modulok.
 - Folyamatosan fejlődik és alakul át a nyelv, ezért naprakész tudás szükséges hozzá.
 - Ha a fejlesztés során nem megfelelő megoldásokat alkalmazunk, az erősen kihatással lehet a teljesítményidőre, illetve problémákat okozhat a rendszer különböző elemei közötti kapcsolatban. [19]

5.3.2 Választás

A fenti megoldások közül a Python Django keretrendszer használatát gondolom a legalkalmasabbnak.

- Jó adatbázis feldolgozás támogatása van (ORM⁴), ami fontos lesz a projektem során.
- A beépített modulok felhasználása mellett rugalmas alkalmazást tudok írni.
- A munkám során szeretnék törekedni a nyílt forráskódú megoldások használatára.
- Van már tapasztalatom a Python nyelv használatában és mivel a Gépi Tanuló komponenst is ezen a nyelven fogom elkészíteni, jól fog illeszkedni hozzá.

⁴ 7.4.1 fejezetben kifejtésre került.

5.4. Gépi Tanulás Komponens

A Gépi Tanulás Komponens képes lesz az adatbázis elérésére, onnan az adatok lekérésére. Ezek alapján gépi tanulási eljárások segítségével modelleket hoz létre, amelyek segítségével, a Webalkalmazás kérésére játékos ajánlásokat tud majd nyújtani.

Két fajta Gépi Tanuló modellt fogok használni:

- Klaszteranalízis: Egy felügyeletlen tanításon alapuló eljárás. A lényege, hogy az adatokat halmazokba (Klaszterekbe) sorolja az eljárás.
- Lineáris regresszió: Egy felügyelt tanításon alapuló eljárás. A lényege, hogy megtalálja a kapcsolatot a magyarázó változók (jelen esetben a játékosok tulajdonságai) és a célváltozó között.

5.4.1. Választás

Az adatelemzés és a gépi tanulás során két meghatározó programozási nyelv terjedt el, az R és a Python. Mivel eddig nem használtam még egyáltalán az R nyelvet, viszont a Pythonban már van tapasztalatom a gépi tanulás algoritmusok terén, ezért az utóbbit fogom használni.

5.5 Adatbetöltő script ⁵

Azért, hogy ne csak statikusan, az előre beégetett adatokon lehessen dolgozni és lekérdezéseket futtatni, a rendszer kiegészítésre kerül egy python adatbetöltő scripttel.

Ez a script SQL utasítások segítségével tölti be az adatokat az adatbázisba.

⁵ Rövid, egy részfeladat automatizálására szolgáló program

6. A TERVEZÉS SORÁN VÉGZETT MUNKAFÁZISOK ÉS TAPASZTALATAIK LEÍRÁSA

6.1 Adatbázis tervezése

Először fel kellett mérnem, hogy milyen igényeket kell a rendszernek kiszolgálni és milyen lehetőségek vannak:

- Az játék korábbi verziójából származó adattábla a játékosok személyét tekintve csak a nevüket, életkorukat és születési dátumokat tartalmazza.
- Mivel szerettem volna ennél több információt megjeleníteni róluk, keresni kezdtem és megtaláltam ugyanennek az adatbázisnak egy frissebb változatát. Ez már több adatot tartalmazott.
- Az új adatokkal kiegészülve frissítettem az adatmodellt, így a korábbi 1 tábla helyett, már 3 táblából áll az adatbázis. (Eszerint frissítettem a dokumentációban a Vázlatos rendszertervet)

Meg kellett vizsgálnom még az adattisztítás szükségességét. Ez alapján a következő két feladat várt rám:

- A rendelkezésre álló új adattáblában több játékosnak a neve (a különleges karaktereket tartalmazók) helytelen karakterekkel jelent meg⁶, mivel a fájl rossz karakterkódolással lett mentve.
- Sok (~10.000) olyan játékos van, akik a korábbi adatbázisban szerepeltek, de a frissítettben nem. Ennek az oka, hogy ezek a játékosok felhagytak időközben a profi pályafutásukkal (a két adathalmaz aktualitása között 3 év van).

Az adatok szempontjából ez azt jelenti, hogy vannak olyan játékosok, akiknek vannak pontszámaik, de személyes adatok nincsenek róluk.

Mivel az alkalmazás a pontszámok elemzésének bemutatására fókuszál, ezért benne hagytam őket az adathalmazban, de valamilyen karaktert kellett helyettesítenem az üres értékek helyére.

⁶ 7.1 ábrán található erre példa.

6.2 Django webalkalmazás tervezése

A webalkalmazás tervezésénél összeszedtem az általam előre meghatározott funkciókat, amelyeket teljesítenie kell. Próbáltam felhasználói fejjel megvizsgálni, hogy hogyan lehet egyszerűsíteni a munkafolyamatokat, illetve ergonomikusan kialakítani a felhasználói felületet.

Az előre meghatározott funkciók a következők:

- Az adatbázisban tárolt adatok megjelenítése
 - A játékosokat megjelenítése táblázatban
 - A pontszámokat megjelenítése táblázatban
 - A csapatokat megjelenítése táblázatban
 - Mindegyik megjelenítésnél az oszlopokat egyenként el lehessen rejtetni je-lölőnégyzetek segítségével
- Szűrők elhelyezése
 - A játékosokat és a csapatokat lehessen szűrni a tulajdonságaik alapján
- Pontszám szerinti rendezés
 - A játékosoknál meg lehessen határozni egyszerre legalább 5 olyan tulajdonságot, amely szerint rendezve lesznek (prioritás szerint, az első megadott érték lesz a rendezésben az elsődleges szempont, az utolsó érték lesz az utolsó szempont)
- Legyen lehetőség a Gépi Tanuló komponens használatára
 - Egy külön képernyőn lehessen felparaméterezni és az általa adott ajánlásokat megjeleníteni

6.3 Gépi tanuló komponens tervezése

Ez a rész két funkciót valósít meg:

- A felhasználó kiválaszt egy játékost, akire a profilja szerint szüksége lenne a csapatában. Gépi tanulás segítségével a program felméri a kiválasztott játékos képességeit, majd hasonló profilú játékosokat fog kilistázni neki.
(Ez akkor lehet hasznos, ha van egy kiszemelt játékos, de nem lehetséges a szerződtetése. Ilyenkor alternatívákat kaphat a felhasználó)
- A felhasználó kiválaszt egy pozíciót. Gépi tanulás segítségével a program kilistázza a számára a 100 legjobban alulértékelt játékost a kiválasztott pozíció szerint.

6.4 Adatbetöltő script tervezése

A scriptnek kezelnie kell az új adatok betöltését amellet, hogy a korábbi adatok ne vesszenek el és a későbbiekben a historikus riportokat lehessen rájuk ültetni.

Emiatt létrejön kettő új tábla:

- Player_hist: A játékosok korábbi személyes adatainak tartalmazó tábla
- Score_hist: A játékosok korábbi pontszámait tartalmazó tábla

Mindkét új tábla az eredetihez képest kiegészül egy *ValidTo* mezővel, amely jelzi, hogy meddig volt érvényes az adat. Így pontosan lehet látni, hogy hogyan változtak a játékosok képességei és személyes tulajdonságai (például magasság, testsúly) az idők során.

A scriptnek egy .csv kiterjesztésű fájlt kell megadni paraméterként, amiből be tudja importálni a sorokat az adatbázisba.

7. MEGVALÓSÍTÁS LEÍRÁSA

7.1 Adatbázis kialakítása

A forrásadatok két `.csv` kiterjesztésű fájlban álltak rendelkezésre. A két fájl (közel) ugyanannak az adathalmaznak a néhány évvel későbbi változata, ezért az összefésülésük nem okozott problémát.

Importáltam mindkét adathalmazt a PostgreSQL szerverre, ahol a mezők típusait már a bennük szereplő adatok szerint határoztam meg.

Terv szerint létrehoztam egy nagy ideiglenes táblát, amibe beletettem mindkét adathalmaz adatait (játékosnév szerint megfeleltetve egymásnak). Később ezt az ideiglenes táblát daraboltam és alakítottam ki a végső táblákat.

Ezt akadályozta, hogy az egyik fájl rossz karakterkódolással volt mentve, ezért a nevekben különleges karakterek helytelenül jelentek meg. Ez egyértelműen akadályozta, hogy a név szerint kapcsoljam össze a táblákat. Ez nagyságrendileg 10.000 adatpont vesztesét jelentette volna.

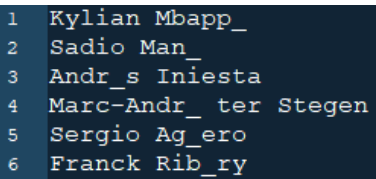


ClÃ©ment ChantÃ©me
ClÃ©ment Maury
Claudiu Belu
Clebinho
Clemens Walch
Conor Hazard
Cristián NÃ©ez

7.1 ábra – Hibás kódolással rendelkező karakterek a nevekben

Megoldás:

1. Exportáltam az új adatforrásból érkező neveket egy `.csv` kiterjesztésű fájlba.
2. Írtam egy rövid python scriptet, ami beolvassa ezt az adathalmazt és kiválogatja belőle a hibás neveket.
3. Ahol hibás karaktert talált, ott a karakter helyére egy alsó kötőjel karaktert helyettesített.



1	Kylian Mbapp_
2	Sadio Man_
3	Andr_s Iniesta
4	Marc-Andr_ ter Stegen
5	Sergio Ag_ero
6	Franck Rib_ry

7.2 ábra – A hibás karakterek alsó kötőjelre cserélve

4. Ezt az új adathalmazt visszatöltöttem a PostgreSQL szerverre.
5. Az régebbi (első) adathalmaz helyesen tartalmazza ezeket a neveket. A helyes neveket összevettem a helytelen nevekkal és ahol csak egy karakter eltérés volt két név között, azt megtalálta.
(Az SQL-ben a karakterlánc összehasonlításoknál a _ karakter azt jelenti, hogy annak a helyén bármilyen karakter szerepelhet. Ezért ezt helyettesítettem be a helytelen karakterek helyére).
6. A hibás nevek helyére behelyettesítettem a javított változatokat.

Ezzel a megoldással az összes hibás nevet sikerült kijavítanom és utána el tudtam végezni velük a táblák összekapcsolását.

Voltak olyan játékosok, akik nem szerepeltek mindkét adatbázisban (valószínűleg ezek befejezték az aktív pályafutásukat), de mivel a pontszámaik megvoltak, úgy döntöttem, hogy megtartom őket. Ezeknél a tulajdonságokat egységesen 'Unknown' felirattal töltöttem fel.

A kialakult ideglenes táblát már szét bírtam bontani a szükséges mezők szerint.

7.2 Webalkalmazás kialakítása

7.2.1 Környezet kialakítása

A webalkalmazást Django keretrendszerben írtam. A Django egy Python-alapú ingyenes és nyílt forráskódú keretrendszer, emiatt a környezet kialakításaként a Python legfrissebb verzióját telepítettem.

A következő Python csomagokat használtam fel a fejlesztés során, ezeket külön kell telepíteni: *pandas*, *sqlalchemy*, *sklearn*, *psycpg2*.

Ezekután telepítettem a Django keretrendszert. A Djangoval együtt automatikusan települ egy úgynevezett lightweight webservert, amit a fejlesztés során lehet használni, de a tényleges publikáláshoz egy másik webservert telepítésére van szükség. [8]

A Django alkalmazást a projektkönyvtárban a „*python manage.py runserver*” paranccsal tudjuk elindítani.

Ezután bármilyen böngészőben a *127.0.0.1:8000* címen érjük el az alkalmazást.

7.2.2 Django projekt felépítése

A Django egy úgynevezett MVT (Model View Template) tervezési mintát használ. A minta három fontos rétegből tevődik össze:

- Model:

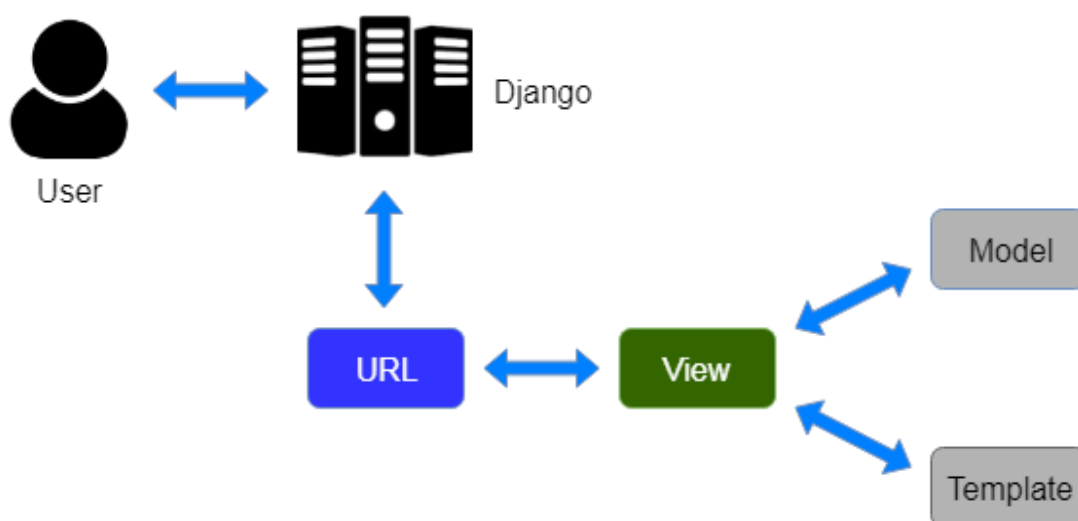
Ez a réteg felelős az adatbázissal való kapcsolatért, az adatok megszerzéséért és az adatmodell reprezentálásáért a Django projekt számára.

- View:

Ez a réteg valósítja meg az üzleti logikát, kezeli a Model által biztosított adatokat, feldolgozza a kéréseket és kommunikál a Template-tel.

- Template:

Ez a réteg felelős teljes egészében a felhasználóval való kommunikációért és a felhasználói felület megjelenítéséért. [1]



7.3 ábra – MVT működése [1]

7.2.3 Felhasználói felület

Játékosok

A szükséges funkciók meghatározása után kialakítottam a 7.4 ábrán is látható felhasználói felületet, először a tényleges funkciók nélkül.

Személyes adatok

Név

Csapat

< Kor <

Pozíció

< Magasság <

Nemzetiség

< Súly <

Ügyesebb láb

Oszlopok megjelenítése

☒ Név ☒ Csapat ☒ Kor ☒ Születési dátum

☒ Magasság ☒ Súly ☒ Pozíció

☒ Nemzetiség ☒ Ügyesebb láb ☒ Piaci érték

☒ Kereset

< Piaci érték € <

< Kereset € <

Pontszámok megjelenítése

☒ Kapusokra vonatkozó ☒ Technikai ☒ Szellemi

☒ Fizikai 1 ☒ Fizikai 2 ☒ Pozíciókra vonatkozó

Rendezés tulajdonság szerint

1. Szempont ☐ Csökkenő

2. Szempont ☐ Csökkenő

3. Szempont ☐ Csökkenő

4. Szempont ☐ Csökkenő

5. Szempont ☐ Csökkenő

Kiegészítő lehetőségek

Hasonló játékosok keresése

Alulértékelt játékosok keresése Pozíció

Név	Csapat	Kor	Születési Dátum	Magasság	Súly	Pozíció	Nemzetiség	Ügyesebb láb	Piaci érték	Kereset	AerialAbility	CommandOfArea	Communication	Eccentricity
Morten Sundli	Ull Kisa	26	31-03-1990	188	83	D (C)	NOR	Right	28500 €	375 €	1	2	2	3
Elvedin Buljubic	SC Neusiedl See	17	25-04-1999	178	75	AM (R)	AUT	Right	1500 €	300 €	3	3	3	2

7.4 ábra – Teljes felhasználói felület

A 7.5 ábrán látható Személyes adatok panel lehetőséget biztosít szűrő értékek megadására és a játékosok tulajdonságainak megjelenítésére/elrejtésére.

Személyes adatok

Név

Csapat

< Kor <

Pozíció

< Magasság <

Nemzetiség

< Súly <

Ügyesebb láb

Oszlopok megjelenítése

☒ Név ☒ Csapat ☒ Kor ☒ Születési dátum

☒ Magasság ☒ Súly ☒ Pozíció

☒ Nemzetiség ☒ Ügyesebb láb ☒ Piaci érték

☒ Kereset

< Piaci érték € <

< Kereset € <

7.5 ábra – Személyes adatok panel

A 7.6 ábrán látható Pontszámok megjelenítése panel lehetőséget biztosít a játékospontszámok megjelenítésére/elrejtésére. Mivel túl sok tulajdonság van ahhoz, hogy mindegyik saját jelölőnégyzetet kapjon, csoportonként lehet állítani. A csoportban található tulajdonságok leírása az 1. Mellékletben található.

38

Pontszámok megjelenítése

☒ Kapusokra vonatkozó
☒ Technikai
☒ Szellemi

☒ Fizikai 1
☒ Fizikai 2
☒ Pozíciókra vonatkozó

7.6 ábra – Pontszámok megjelenítése panel

A 7.7 ábrán látható Rendezés tulajdonság szerint panel lehetőséget biztosít a rendezési szempontok meghatározására úgy, hogy az 1. szempont lesz az elsődleges, a 2. a másodlagos és így tovább. Minden szemponthoz be lehet állítani, hogy csökkenő sorrendben legyenek rendezve az adatok az adott szempont szerint. Az alapértelmezett minden szempontnál a növekvő sorrend. A mezőbe kattintva egy lenyíló listából lehetőség nyílik a látható mezők kiválasztására.

Rendezés tulajdonság szerint

1. Szempont

☐ Csökkenő

2. Szempont

☐ Csökkenő

3. Szempont

▼

Kor

Magasság

Súly

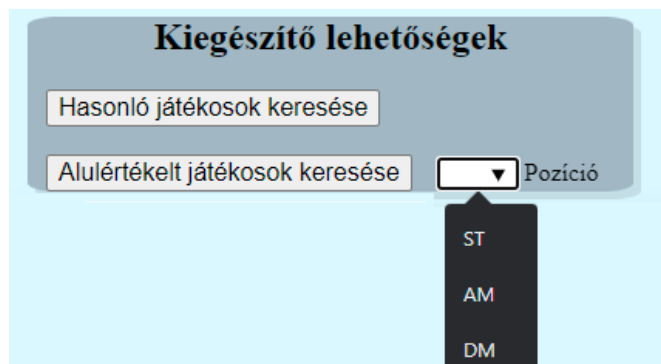
Érték

4. Szempont

5. Szempont

7.7 ábra – Rendezés tulajdonság szerint panel

A 7.8 ábrán látható Kiegészítő lehetőségek panel lehetőséget biztosít a gépi tanuló komponens használatára. A *Hasonló játékosok keresése* gomb megnyomása után a felhasználó a kiválasztott játékos nevére kattintva indíthatja el a folyamatot. Néhány másodperccel később a gépi tanuló komponens által visszaadott játékosok jelennek meg a lenti táblázatban. Az *Alulértékelt játékosok keresése* gombra kattintva, a gomb melletti mezőben megadott pozíció szerint, a gépi tanuló komponens által visszaadott alulértékelt játékosok jelennek meg a lenti táblázatban.



7.8 ábra – Kiegészítő lehetőségek panel

A felhasználói felület kialakítása után megvalósítottam a Model rétegben az adatkapcsolatot és az adatmodellezést.

Az adatok rendelkezésre állása után megjelenítettem a játékosok adatait is a felső panelek alatt.

Mikor az adatok és a funkció nélküli felhasználói elemek is rendelkezésre álltak, elkezdtem megvalósítani a funkciókat.

Mivel a szűrő és sorba rendező funkciók mind a megjelenítendő játékosok halmazát változtatják, az SQL lekérést kell módosítani a felhasználó cselekedeteinek megfelelően.

- A személyes adatok szűrése a lekérdezés WHERE része után szűr be felételeket
Ha a felhasználó helytelen karaktert, érvénytelen értéket ad meg, a lekérdezés nem fog találni egyező értéket, de hibát nem fog kapni.
- A megadott rendezési szempontok pedig a lekérdezés ORDER BY része után szűrődnek be, annak megfelelően, hogy a *csökkenő* jelölőnégyzet be van-e pipálva.
Ha a felhasználó helytelen karaktert, érvénytelen értéket ad meg, egy vizsgálatnak köszönhetően a lekérdezés figyelmen kívül fogja hagyni a megadott rendezési értéket.

A Pontszámok megjelenítése és az Oszlopok megjelenítése panelek az SQL lekérdezést nem módosítják, hanem a Django template-ben figyelmen kívül hagyják az adott oszlop megjelenítését. Erre a 7.9 ábrán látható egy példa a kódból. Ha az *age*, *born* és *height* mezők láthatósága nem *True*, akkor egyszerűen nem fognak megjelenni. Ezek a láthatóságok a **View** rétegben lévő változóknak tárolódnak.


```
{% if column_visibilityes.age == True %} <td> {{ p.age }} </td>{% endif %}
{% if column_visibilityes.born == True %} <td> {{ p.born }} </td>{% endif %}
{% if column_visibilityes.height == True %} <td> {{ p.height }} </td>{% endif %}
```

7.9 ábra – Django példakód a láthatóságok megoldására

Mivel a pontszámok láthatóságát is kell állítani, de azokból túl sok van hozzá, hogy mindegyik külön gombot kapjon (több, mint 60 darab), ezért ezeknek a láthatóságát csoportonként kezelem. Az 1. Mellékletben megtalálhatók a pontok leírása és csoportosítása, ennek megfelelően egy-egy csoporthoz rendeltem egy gombot. A csoporthoz tartozó gomb a csoportban lévő összes mező láthatóságát azonosan befolyásolja.

Csapatok

A csapatok a játékos neve mellett a csapat nevére kattintva érhetők el. A 7.10 ábrán látszódik, hogy a csapatok neve egy linkként szolgál. A 7.11 ábrán látható módon csapatok felületen kezdetben csak a kiválasztott csapat található, de a szűrőből az értékét kitörölve megkapjuk az összes csapatot, a ligát, amiben játszanak és a csapat székhelyét. Ez egy vizsgált játékos esetében plusz információval szolgálhat.

Név	Csapat	Kor	Születési Dátum	Magasság	Súly	Pozíció	Nemzetiség	Ügy láb
Morten Sundli	Ull Kisa	26	31-03-1990	188	83	D (C)	NOR	Righ
Elvedin Buljubasic	SC Neusiedl See	17	25-04-1999	178	75	AM (R)	AUT	Righ
Jale Naskali	FC Kiffen 2	25	22-09-1990	180	0	ST (C)	FIN	Righ

7.10 ábra – Játékosok megjelenítése

Szűrők

Huracán (Montecarlo)

Név

Liga

Székhely

Frissít

Név	Liga	Székhely
Huracán (Montecarlo)	Argentine Torneo Federal C	Argentina (Fifth Division non AFA)

7.11 ábra – A Huracán(Montecarlo) csapat megjelenítése

7.2.4 Logika

A felhasználói interakciók feldolgozása és a megfelelő adatok megjelenítése mögött bonyolult logikák futnak le. Ezek a **View** rétegben hajtódnak végre, de kiszerveztem őket egy külön **Logic** programozási osztályba. Az osztály példányosítása után a megfelelő **View** metódusban már meg tudom hívni az üzleti logika függvényeit, amik így nem közvetlenül a **View**-ban vannak.

Ennek az átláthatóság mellett az is az előnye, hogy a később tárgyalt üzleti logika tesztelése teljesen független az azt használó alkalmazástól.

Működés szerint:

1. **View** megkapja a Web-kérést, benne az összes szükséges paraméterrel és beállítással, amiket a felhasználói interakció generált.
2. A **View** átadja a paramétereket a tőle különálló üzleti logikának és visszakapja a logika által előállított SQL lekérdezést.
3. Az SQL lekérdezést a **Model** segítségével lefuttatja, megkapva így a megjelenítendő játékosok halmazát.
4. A megkapott játékosokat átadja a **Template**-nek, ami a beágyazott dinamikus kódokkal előállítja a weblapot, amit a **View** a Web-kérésre válaszolva visszaküld a felhasználónak.

7.3 Gépi tanuló komponens

A Gépi tanuló komponens funkciói a belső logikájukban különböznek, de azonos elven segítik a játékosok megjelenítését:

1. A felhasználó valamilyen interakciójának hatására létrejön az az adat, ami az algoritmusok alapjául fog szolgálni (játékos azonosító vagy pozíció, a kiválasztott funkciótól függően)
2. A **View** a Web-kérésből kiolvassa ezt a paramétert és meghívja a megfelelő külső Gépi Tanuló programot
3. A Gépi Tanuló program a belső logikája szerint műveleteket végez és visszaadja a megjelenítendő játékosok azonosítóját
4. Miután a **View** megkapta ezt a listát, beágyazza egy SQL lekérdezésbe és a **Model** réteg segítségével beolvassa az megjelenítendő játékosokat.
5. A megkapott játékosokat átadja a **Template**-nek, ami a beágyazott dinamikus kódokkal előállítja a weblapot, amit a **View** a Web-kérésre válaszolva visszaküld a felhasználónak.

Ezáltal a gépi tanuló komponens a folyamat üzleti logikai részének megvalósítását veszi át, a **View** réteg pedig egy központi irányítóként szerepel.

7.3.1 Játékos ajánlás

A lényege, hogy a felhasználó kiválaszthat egy játékost, aki illene a csapatába, de valamilyen ok miatt nem lehetséges a megszerzése. A program keres olyan játékosokat, akik hasonló tulajdonságokban erősek, így alternatívát kínál az eredeti játékos helyett.

Az eljárás logikája:

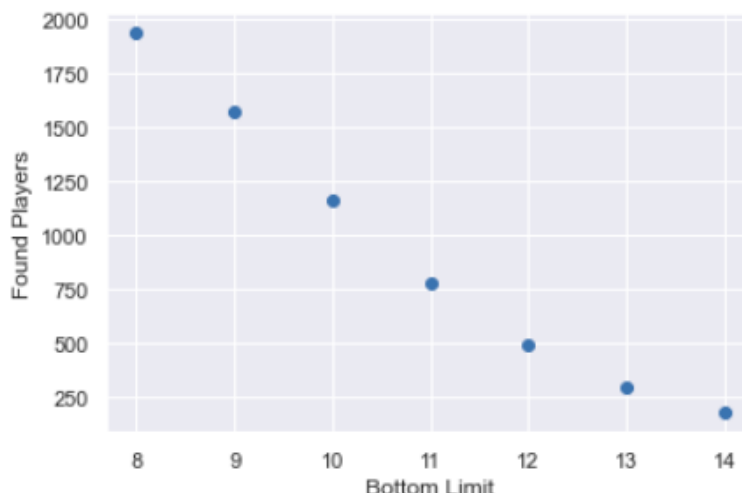
1. Megkeressük az eredeti játékos 20 legjobb tulajdonságát.
2. A többi játékos közül olyanokat keresünk, akiknek a legjobb 20 tulajdonsága közül legalább 13 (a 7.12 ábrán látható okok miatt) szerepel az eredeti játékos 20 legjobb tulajdonságai között.
3. A leszűrt játékosokból a tulajdonságaik alapján egy klaszterizáció kerül végrehajtásra, ami a játékosok halmazokba rendezését jelenti. A klaszterizációt K-Means algoritmussal valósítottam meg.
4. Az eredeti játékossal megegyező halmazban található játékosok az eredetivel hasonló profilúak, őket adja vissza eredményként az algoritmus.

Mivel sok adatsor (játékos) van az adatbázisban nagyon sok tulajdonsággal, ezért a modellt kezelhetetlen ideig kellene futtatni. A futási idő javítása érdekében leszűrtem a szóba jöhető játékosok halmazát:

- Meghatároztam az eredeti játékos 20 legjobb tulajdonságát.
- Sok olyan játékos van, akit alaptól ki lehet zárni, mert más posztion játszik vagy annyira gyengék ezek a képességei, hogy nem érdemes ajánlani őket. Emiatt leszűrtem a játékosokat úgy, hogy a figyelembe vett 20 tulajdonságban az 1-20 terjedelmű skálán legalább a 7 értéket éri el.
- Ezután végrehajtottam az eljárás logikájában leírt 2. lépést
- Az így kapott néhány száz nagyságrendű adatra már könnyen lehetett gépi tanuló algoritmust alkalmazni.

A közös értékek alsó határának meghatározására elvégeztem egy 100 véletlenszerű játékosra vonatkoztatott felmérést. A 7.12 ábrán látszik, hogy a megkövetelt közös tulajdonságok növelésével hogyan csökken a talált játékosok száma.

A 13-as elfogadási határnál még közel 250 játékos felel meg átlagosan. Ezek, a képességeik alapján, már hasonlítani fognak az eredeti játékoshoz, de a klaszterizáció tovább szűkíti a lehetőségeket.



7.12 ábra – Hasonló játékosok a közös tulajdonságok számának függvényében

Klaszterek számának meghatározása – Sziluett Módszer

A halmazokba rendezett pontok Sziluett értéke megmutatja, hogy a pontok mennyire illenek a saját halmazukba a többi halmazhoz képest. Ez egy -1 és $+1$ közötti valós szám. A $+1$ -hez közelítő érték azt jelenti, hogy a pontok jól illenek a saját halmazukba.

Az ideális klaszterszám megállapításához létre kell hozni több K-Means modellt, különböző klaszterszámokkal (> 2). Az a klaszterszámú modell a legoptimálisabb, ahol a legnagyobb a Sziluett érték.

7.3.2 Alulértékelt játékosok keresése

A lényege, hogy a felhasználó meghatároz egy pozíciót és a Gépi tanuló algoritmus visszaadja számára a 100 legjobban alulértékelt játékost a kiválasztott pozícióban.

Az alulértékelt játékos azt jelenti, hogy a többi játékos piaci árát figyelembe véve az alulértékelt játékosnak a jelenleginél magasabb piaci árral kellene rendelkeznie. Minél nagyobb a különbség a becsült és tényleges piaci ára között (a becsült javára), annál alulértékeltőbb.

Az eljárás logikája:

1. A játékosok közül ki kell szűrni azokat, akik a felhasználó által megadott pozícióban játszanak.

- a. Ha a megadott pozíció „GK” (, tehát kapusok keresése a feladat), akkor a játékosoknak csak a kapusokra vonatkozó tulajdonságait kell figyelembe venni.
 - b. Ha a megadott pozíció nem „GK” (, tehát úgynevezett mezőnyjátékosok keresése a feladat), akkor a kapusokra vonatkozó tulajdonságok kivételével az összes tulajdonságot figyelembe kell venni.⁷
2. A szűrés után kapott játékosokra egy Lineáris Regressziós modellt kell építeni, ahol a magyarázó változók a játékosok tulajdonságai (az 1. lépésben leírtak szerint), a célváltozó pedig a játékosok piaci ára.
 3. A modellnek meg kell adni ugyanazokat a játékosokat, akik alapján készült a modell, így megkapható, hogy a Regresszió szerint mennyinek kellene lennie a játékosok piaci árának a többi játékos ára szerint.
 4. A száz legalulértékeltőbb játékos megtalálásához a játékosokat sorba kell rendezni a becsült és tényleges piaci árak különbsége szerint, csökkenő sorrendben és ezek közül az első százat kiválasztani.

7.4 Komponensek közötti kapcsolatok kialakítása

7.4.1 Adatbázis – Webalkalmazás

A Django keretrendszer egyik legnagyobb erőssége az ORM (Object-Relational Mapper). Az ORM egy programozási technika adatok konvertálására adatbázisok és objektumorientált programozási nyelvek között. Az általunk írt programban egy objektum leképezést hoz létre, aminek a felépítése megegyezik az adatbáziséval. [17]

A használata jelentősen csökkenti a megírandó kód mennyiségét.

A Django Model rétegében létre kell hoznunk az modelleket. Ez valójában osztályok definiálását jelenti, amelyek tulajdonságai megegyeznek az osztályok által modellezett táblák mezőivel.

Ezt nem muszáj kézzel megírni, van lehetőség meglévő adatbázis táblákból Django modelleket generálni és fordítva is. [17]

Mivel az adatbázisban már létre voltak hozva a táblák a megfelelő kulcsokkal, ezért ezekből generáltam a Django modelleket:

1. A Django projektben található *settings.py* fájlban be kell állítani az adatbázis elérését (cím, felhasználónév, jelszó, adatbázis)
2. A projektmappában a következő parancsokat kell végrehajtani:
 - `python manage.py makemigrations`

⁷ Az 1. lépésben említett pontszámok leírása az 1. Mellékletben található meg.

- `python manage.py sqlmigrate [Projekt neve]`

Ezek hatására a projektkönyvtárban a *migrations* mappán belül megjelenik egy fájl, ami tartalmazza az automatikusan generált modelleket. Ezt természetesen le kell ellenőrizni, különös tekintettel a mezők típusára. Ha megfelelő a modell, akkor a fájl tartalmát be kell másolni a *models.py* fájlba és innentől az adatbázis struktúra leképezhető a Django-n belül. [17]

A Python biztosít eszközöket az adatbázis lekérdezések kezelésére, de én jobban szeretem a nyers SQL lekérdezést saját magam megírni és úgy lefuttatni. Ez a megoldás biztosította számomra a szabadságot, hogy tetszőleges lekérdezéseket generáljak különböző logikák alapján.

Például: `players = Player.objects.raw('SELECT * FROM player')`

A fenti programkód eredményeként a *players* változó tartalmazni fogja az összes játékost a tulajdonságaikkal együtt.

7.4.2 Adatbázis – Gépi tanuló komponens

Itt az *sqlalchemy* Python csomag *create_engine* metódusát használtam. Ezzel a metódussal létrehoztam egy engine objektumot, aminek paraméterként meg kellett adni a kapcsolódás adatait (cím, felhasználónév, jelszó, adatbázis). Az így létrejött engine objektumnak meghívtam a *.connect()* metódusát, ezáltal kaptam egy connection objektumot.

A Python *pandas* csomagjában található *read_sql()* metódussal tudtam elérni az adatbázist. Ennek paraméterként egy SQL lekérést kellett megadni és az előbb létrehozott connection objektumot. Mivel az SQL lekérést itt is én állítottam elő, kihasználtam ennek az előnyét és a megadott logikák szerint általam írt metódusokkal generáltattam.

Itt a webalkalmazással ellentétben nem kellett Modelleket definiálni és leképezni az adatbázis tábláit osztályokba. Ebben az esetben a főként *pandas* csomagban található *DataFrame* és *List* adattípusokon hajtottam végre a műveleteket.

7.4.3 Webalkalmazás – Gépi tanuló komponens

A Webalkalmazás a megfelelő funkció meghívásakor a Gépi Tanuló komponenshez fordul. A Gépi Tanuló komponens különálló, *.py* kiterjesztésű Python programokat jelent.

Azért, hogy tényleg különálló legyen, alpból egy másik mappába helyeztem és nem behívtam a Webalkalmazásba és ott futtattam belőle a megfelelő függvényeket, hanem a Windows környezetben futtattam. Ez azért előnyös, mert a két alkalmazás így független egymástól. Egy másik nyelven íródott futtatható fájlt (, amelynek az input és output értékei megegyeznek) is be lehet rakni a helyére.

A Python *subprocess* könyvtárával van erre lehetőség. A Webalkalmazásban a *subprocess.run()* metódust kell futtatnunk. Több paramétert meg lehet adni, az én esetemben így nézett ki:

```
subprocess.run(['python', 'ML\FindSimilarPlayer.py', playerid], capture_output=True)
```

A *FindSimilarPlayer.py* program megkapja a kiválasztott játékos azonosítóját argumentumként (*playerid*), feldolgozza, majd visszaadja a hasonló játékosok azonosítóját, amik egy listában tárolódnak, majd épülnek be az SQL lekérdezésbe.

A Gépi tanuló komponensben kezelni kell, hogy kapni fog argumentumként egy azonosítót. A megtalált játékosok azonosítóját nem tudja közvetlenül visszaadni az őt meghívó Webalkalmazásnak, mivel nem függvényhívás történt, hanem a Windows környezetben lett elindítva az alkalmazás. Ezáltal viszont a *print()* metódusok eredményeit, amik Windows konzolból elindítva a konzolra íródnának ki, kapja meg a Webalkalmazás.

Ebben az esetben a *print()* függvénnyel adtam vissza értékeket. Figyelni kellett rá, hogy pontosan egy darab *print()* legyen a kódban, mert ha több mindent „ír ki” a program, akkor a webalkalmazás nem fogja tudni feldolgozni.

7.5 Adatbetöltő script

Működése:

1. A script a paraméterként megadott, .csv kiterjesztésű fájl tartalmát beolvassa egy Pandas DataFrame típusú objektummá.
2. A Players és Score tábla tartalmát kimentti egy ideiglenes táblába, majd üríti ezeket a táblákat.
3. Az mezők neve szerint megpróbálja betölteni az adatokat a megfelelő táblába.
4. Ha sikeres volt a betöltés, néhány teszt lekérdezést futtat, hogy ténylegesen a megfelelő mennyiségű és minőségű adat került-e be.
5. Ha az ellenőrzés sikeres, akkor az ideiglenes táblákból áttölti az adatokat a Players_hist és Score_hist táblákba úgy, hogy a *ValidTo* oszlopot az aktuális dátummal tölti ki.
6. Ha az ellenőrzés vagy az adatbetöltés során hiba merül fel, akkor az ideiglenes táblából az adatokat visszatölti az eredeti Player és Score táblákba a felhasználónak visszaadja a hibaüzenetet.

Ezzel a megoldással a Player és a Score táblákban az aktuális adatok szerepelnek, amik az alkalmazás elsődleges felhasználási módjához szükségesek, az időbeli változásokat lekövető riportokhoz felhasználható adatok pedig a Player_hist és Score_hist táblákban.

8. TESZTELÉS

8.1 Tervezés

A három nagy komponenst külön kell kezelni tesztelési szempontból. Ha mindegyik maradéktalanul eleget tesz az előre meghatározott funkcióknak, akkor rendszer egészének működését az alkalmazás használatával lehet tesztelni.

Ha a komponensek külön-külön átmennek a teszteken, akkor az előre meghatározott funkció lista teljesülését fogom figyelembe venni. A funkciók működésének teljesülése után elkezdtem a rendszer aktív használatát több használati esettel, hogy minél több hibát derítsek fel.

8.2 Adatbázis

Az adatbázisban csak lekérdezések fognak futni, adatmanipuláció nem. A lekérdezések tesztelését a lekérdezést generáló függvényekben végeztem úgy, hogy összevettem a függvények által visszaadott adathalmazt a közvetlenül az adatbázison lefuttatott lekérdezések eredményével.

Adatbetöltés csak az erre a célra készített adatbetöltővel történhet, ennek a tesztelését a 8.4 fejezet tárgyalja.

8.3 Webalkalmazás és Gépi Tanuló komponens tesztelése

A Webalkalmazás és a Gépi Tanuló komponens üzleti logikájának függvényeire úgynevezett egységteszteket írtam, amelyek az adott függvények bemenetekre adott kimenetét vizsgálják. Több fajta bemeneti paraméternél kell vizsgálni a kimenetet, attól függően, hogy milyen lehetséges értékeket kaphat az adott függvény.

Ezek az egységtesztek egy külön, python környezetben futtatható fájlban találhatók. A python biztosít egy Unittest nevű tesztelési csomagot, amely a Test előtaggal kezdődő függvényeket automatikusan felismeri, és lefuttatja a bennük megírt egységteszteket. Az egységtesztek leírása és eredménye az 2. Mellékletben található.

8.4 Adatbetöltő tesztelése

Az adatbetöltő scriptet a tényleges működése és használata során tudtam tesztelni. A fejlesztés során is figyelembe vettem a hibalehetőségeket. Annak elkerülése érdekében,

hogy hibás vagy hiányos adatok kerüljenek a rendszerbe, a script bármilyen ellenőrzés közben fellépő hiba esetén visszafordítja a folyamatot és jelez a felhasználónak a hibával kapcsolatban. Ha a fájlban található adatok a korábban említett adatforrásokkal megegyező struktúrájúak, akkor hiba nélkül történik az adatbetöltés.

9. AZ EREDMÉNYEK BEMUTATÁSA, ÉRTÉKELÉSE

A munka során az elsődleges célom az volt, hogy a feladatkiírásban szereplő pontok teljesüljenek és egy használható rendszert hozzak létre. Ezeket sikerült teljesítenem.

9.1 Eredmények

Az rendszert a terveknek megfelelően valósítottam meg, minden előzetes elvárásnak megfelel és plusz funkcióként az adatbetöltés lehetősége is megjelent.

Sikerült olyan eszközt készíteni, amely segítségével a felhasználók hozzáférhetnek a Football Manager nevű játék adatbázisában rejlő értékes információkhoz. Programozási és adatbázis tudás nélkül is dinamikusan tudják változtatni a lekérdezéseket, ezáltal a megjelenített játékosokat.

9.1.1 Komponensek

Az adatbázisban kialakítottam az adatmodellnek megfelelő struktúrát és az adattisztítás után betöltöttem az adatokat, létrehoztam a kulcsokat és indexeket.

A Webalkalmazást olyan módon írtam meg, hogy minden előre meghatározott igényt kielégít. A felhasználói interfészen kívül kommunikál az adatbázissal és a Gépi Tanuló komponenssel, és megvalósítja az üzleti logikát.

A Gépi Tanuló komponens a megkapott paraméterek alapján elvégzi az üzleti logikának megfelelő műveleteket, majd a Webalkalmazásnak visszaadja az eredményt.

Az adatbetöltő lehetőséget ad az új adatok betöltésére, a régiek megtartása és időbeli változások vizsgálatának biztosítása mellett.

9.2 Értékelés

A játékosok összehasonlítása egy fontos szempont lehet, amely által nagyságrendekkel csökkenthetik az átigazoláskor szóba jöhető játékosokat. Az **adatbázis és az alkalmazás erőssége, hogy gyorsan és könnyen ki lehet szűrni az elvárásoknak nem megfelelő játékosokat, illetve objektíven össze lehet hasonlítani őket a képességeik alapján.** Amiatt, hogy ez egy webes alkalmazás, a felhasználónak nem kell semmilyen programot telepítenie, egyszerűen elérheti a böngészőjén keresztül. Sajnos mobil eszközön a megjelenítés nem kényelmes, mert a táblázatban sok oszlop van

egymás mellett, ez pedig csak hordozható vagy asztali számítógépen teszi lehetővé az alkalmazás használatát.

9.2.1 Felhasználási lehetőségek

A rendszer a meghatározott céloknak megfelelően alkalmas arra, hogy olyan játékosokat azonosítsanak a segítségével, akik tehetségesek, de még alulértékelték. A játékosok képességeinek egymáshoz viszonyítása objektíven megmutatja a köztük lévő különbségeket. A gépi tanuló algoritmusok és a szűrő mezők segítségével a felhasználó olyan játékosokat tud saját maga felfedezni, akik megtalálásához hosszabb kutatómunkák lennének szükségesek.

Ezek alapján a rendszer jó lehetőséget nyújt azon edzők/játékosmegfigyelők számára, akiknek nincs lehetőségük több száz játékos megvizsgálására. A rendszer nyújtotta lehetőségekkel a lecsökkenthetik a szóba jöhető és megvizsgálandó játékosok számát néhány tízes nagyságrendűre. Láthatják számok formájában, hogy a játékosok milyen területen erősek és milyen területen gyengék. Ez nagy lehetőséget jelent a tényleges üzleti felhasználásban.

A valós csapatok edzőin kívül segítséget nyújthat azoknak a játékosoknak, akik valamilyen virtuális, futball tematikájú játékban keresnek új játékosokat vagy szeretnék összehasonlítani őket a csapatuk összeállítása érdekében.

A 3.3 fejezetben célként kitűzött felhasználási lehetőségeknek eleget tesz az elkészült alkalmazás.

10. MEGVALÓSÍTÁS ELEMZÉSE, TOVÁBBFEJLESZTÉSI LEHETŐSÉGEK

A megvalósítás hatékony, mivel a felhasználó egy intuitív, könnyen használható felületet kap, ahol az interakciói szerint hajtódnak végre a háttérfolyamatok.

Az adatok megfelelő, könnyen hozzáférhető struktúrában vannak tárolva és a 7.4.1 fejezetben kifejtett ORM technológiának köszönhetően nagyon egyszerűen képződnek le a Webalkalmazás és a Gépi tanuló komponens számára.

A megvalósítás során törekedtem a gyakorlatias megoldások keresésére. A felesleges adatmozgatások elkerülése végett az adatlekéréseknél már az adatbázisban megtörténnek a szűrések és csak azok az adatok mozognak, amelyre a felhasználónak ténylegesen szüksége van. A komponensek közötti kommunikáció során is csak a játékosok azonosítói mozognak, nem a teljes adatszerkezetek.

A függvények írásánál igyekeztem az Egyetlen Felelősség elvét (Single Responsibility) betartani és így jól tesztelhető programrészeket írni.

A fejlesztés során mindig az felhasználói igény kielégülése, a hatékonyság és a tesztelhetőség voltak az elsődleges szempontok. Ezeknek az alkalmazás minden részénél sikerült megfelelni.

Továbbfejlesztési lehetőségek

Továbbfejlesztési lehetőségként szeretném megemlíteni a historikus adatokra építhető riportok lehetőségét. Sok információ kinyerhető belőle, ha látszódik a játékosok fejlődése. Megfelelő számú historikus adat alapján egy előrelejző rendszer építésére is lehetőség lenne, amely megmutatja, hogy a játékosok várhatóan milyen teljesítményt érhetnek el a jövőben.

A felhasználói keresések eredményének mentése és a találatként kapott játékosok táblázatos formába történő exportálása egy fontos funkció lehet.

Továbbfejlesztésként lehet még említeni, hogy az alkalmazás rendszeres időközönként frissítse a játékosok fizetésére vonatkozó adatokat egy megbízható forrásból, hogy mindig naprakész legyen.

11. A SZAKDOLGOZAT TARTALMI ÖSSZEFOGLALÓJA MAGYAR ÉS ANGOL NYELVEN

Magyar nyelven

A szakdolgozatom céljául tűztem ki:

- az adatalapú játékosmegfigyelés ismertetését a futball tekintetében,
- a Football Manager nevű játék adatainak megfelelő struktúrában történő eltárolását,
- ezen adatok elérhetővé tételét egy webalkalmazás segítségével,
- szűrő és rendező lehetőségek biztosítása az adatokra,
- a gépi tanulás használatának lehetősége a felhasználó számára, a megfelelő játékosok megtalálásának érdekében.

Ez idő közben kiegészült egy plusz funkcióval: új adatok betöltésének lehetősége a játékosadatok frissen tartása érdekében.

Ezek mellett a játékosok értékelése, piaci árak kialakulása is kifejtésre került. A Django, a PostgreSQL és a külső Python programok együttműködésének és megoldásainak bemutatása is fontos része a dolgozatnak.

Kitérek arra is, hogy miért ezeket az adatokat használtam és milyen jellegű üzleti értéket lehet kinyerni belőlük.

In english

The aim of my thesis was:

- to exposit data-driven scouting in football,
- to store the data of Football Manager in the fitting structure,
- to create a web application to make the data reachable,
- to provide data filtering and sorting,
- to provide machine learning usage to the user to find the adequate players.

A new feature has been added: the opportunity of import new data to keep up to date the player informations.

In addition player ratings and their price development were explained. The presentation of cooperation and solutions of Django, PostgreSQL and external Python programs is an important part of the thesis.

I have also explained why I used this data and what kind of business value can be extracted from it.

12. IRODALOMJEGYZÉK ÉS HIVATKOZÁSOK

- [1] Adrian Holovaty : Definitive Guide to Django, *Springer-Verlag Berlin and Heidelberg GmbH & Co. KG*, 2009
- [2] Advantages and Disadvantages of ASP.NET, (<https://www.software-developer-in-dia.com/advantages-and-disadvantages-of-asp-net/>), utoljára megtekintve: 2021.12.04
- [3] Comparing Database Management Systems: MySQL, PostgreSQL, MSSQL Server, MongoDB, Elasticsearch, and others (<https://www.altexsoft.com/blog/business/comparing-database-management-systems-mysql-postgresql-mssql-server-mongodb-elasticsearch-and-others/>), utoljára megtekintve 2021.12.03
- [4] Concept work „Data-driven Scouting in Football” (<https://millerton.de/en/2020/07/03/konzeptarbeit-datengetriebenes-scouting-im-fussball/>), utoljára megtekintve 2021.11.14
- [5] Daniel Memmert & Dominik Raabe : Data Analytics in Football: Positional Data Collection, Modelling and Analysis, *Routledge*, 2018
- [6] Dave Braines & Joe Pavitt & Richard Tomsett : Applied AI Letters, *John Wiley & Sons, Ltd.*, 2021
- [7] Devndra Ghimire : Comparative study on Python web frameworks: Flask and Django, 2020, (https://www.theseus.fi/bitstream/handle/10024/339796/Ghimire_Devndra.pdf?sequence=2&isAllowed=y)
- [8] Django Documentation, Writing your first Django app, part 1 (<https://docs.djangoproject.com/en/4.0/intro/tutorial01/>), utoljára megtekintve: 2022.03.03.
- [9] Füredi Balázs: Legyőzheti-e egy kis csapat a Liverpoolt az adatok segítségével? (<https://blog.hiflylabs.hu/hu/2020/06/11/legyozheti-egy-kis-csapat-a-liverpoolt-az-adatok-segitsegevel/>), utoljára megtekintve: 2021.12.06
- [10] Here’s why England star Jude Bellingham’s No.22 was retired at Birmingham City (<https://www.birminghammail.co.uk/sport/football/football-news/jude-bellingham-birmingham-city-dortmund-20776728>), utoljára megtekintve: 2021.11.13
- [11] Javas Spring Pros and Cons (<https://www.javatpoint.com/java-spring-pros-and-cons>), utoljára megtekintve: 2021.12.03
- [12] Joe Bertuzzi : Soccer Scouting Guide, *MICL*, 1999
- [13] Justin Harper: Data experts are becoming football’s best signings, (<https://www.bbc.com/news/business-56164159>), utoljára megtekintve: 2021.12.06
- [14] Luca Pappalardo & Paolo Cintia & Paolo Ferragina & Emanuele Massucco & Dino Pedreschi & Fosca Giannotti: PlayeRank : Data-driven Performance Evaluation and

Player Ranking in Soccer via a Machine Learning Approach, 2019 (<https://dl.acm.org/doi/pdf/10.1145/3343172>)

[15] Martijn Ernest: The importance of scouting, (<https://ifbi.brussels/programme/the-importance-of-scouting/>), utoljára megtekintve: 2021.12.06

[16] Oliver Müller & Alexander Simons & Markus Weinmann : Beyond crowd judgements: Data-driven estimation of market value in association football (<https://www.sciencedirect.com/science/article/pii/S0377221717304332>), utoljára megtekintve: 2021.12.02

[17] Shabda Raaj : Django ORM Cookbook, *Agiliq Info Solutions India Pvt Ltd*, 2018

[18] Tom Hamilton & Tor-Kristian Karlsen : Football Manager 2021 could find next Messi, but would club scouts use it? (<https://www.espn.com/soccer/soccer-transfers/story/4242573/football-manager-2021-could-find-next-messibut-would-club-scouts-use-it>), utoljára megtekintve: 2021.11.08

[19] Victor Rak, Pros and Cons of Ruby on Rails (<https://sloboda-studio.com/blog/pros-and-cons-of-ruby-on-rails/>), utoljára megtekintve: 2021.12.03

[20] A Football Manager 2017-es adatainak elérése (<https://www.kaggle.com/ajinkyab-laze/football-manager-data>), Utoljára megtekintve: 2021.12.08.

[21] A Football Manager 2020-as adatainak elérése (<https://www.kaggle.com/datasets/kyptorio/football-manager-2020>). Utoljára megtekintve: 2022.03.10.

13. MELLÉKLETEK

1. Melléklet – Játékosok pontszámai

Mindegyik játékos rendelkezik az alábbi tulajdonságokkal. Ezek 1-20 között egész számú értékek lehetnek, arányosan a játékos képességével.

Az 1 a legrosszabb, a 20 a legjobb adható pontszám.

Kapusoknál lényeges tulajdonságok

AerialAbility	Milyen messzire ér el a kapus ugró helyzetben
CommandOfArea	Mennyire valószínű, hogy egy levegőben ívelt labdát megszerez a kapus
Communication	Mennyire jól tudja a kapus szervezni a védelmet
Eccentricity	Mennyire valószínű, hogy a kapus jól reagál a váratlan helyzetekben
Handling	Mennyire biztosan tudja elkapni a labdát
Kicking	Milyen messzire tudja rúgni a labdát a kapus
OneOnOnes	Mennyire jól teljesít a kapus az egy-az-egy elleni szituációkban
Reflexes	Milyen gyorsan képes reagálni a lövésekre
RushingOut	Mennyire jól tudja felmérni a kapus, hogy kirohanjon-e a kapujából megszerezni a labdát
TendencyToPunch	Mennyire valószínű, hogy a kapus csak kiüt egy fogható labdát, ahelyett, hogy elkapná azt
Throwing	Mennyire pontosan tudja a kapus a csapattársaihoz dobni a labdát

Technikai tulajdonságok

Corners	Mennyire pontosan tudja elvégezni a játékos a szögletet
Crossing	Milyen pontosan tudja a játékos a pálya széléről beívelni a labdát
Dribbling	Mennyire hatékonyan tudja kezelni a játékos a labdát, amikor nála van
Finishing	Mennyire pontosak a játékos kapura tartó lövései
FirstTouch	Mennyire hatékonyan tudja átvenni és továbbítani a labdát a játékos
Freekicks	Mennyire pontosan tudja elvégezni a szabadrúgásokat a játékos
Heading	Mennyire pontosan tudja a kívánt helyre fejelni a labdát a játékos
LongShots	Milyen pontosan tud kapura löni messziről a játékos
Longthrows	Mennyire messzire tudja bedobni a labdát a játékos
Marking	Mennyire jól tud helyezkedni a játékos, hogy könnyen eljuthasson hozzá a labda
Passing	Mennyire pontosan tudja passzolni a játékos a labdát
PenaltyTaking	Mennyire pontosan tud elvégezni a játékos egy büntető rúgást

Tackling	Mennyire hatékonyan tudja elvenni a játékos a labdát az ellenféltől, anélkül, hogy szabálytalankodna
Technique	Mennyire jól tud magasabb technikai szintű

Szellemi tulajdonságok

Aggression	Mennyire valószínű, hogy a játékos belemegy egy fizikai konfliktusba
Anticipation	Mennyire jól tudja a játékos felmérni a csapattársai és ellenfelei következő lépését a játékos
Bravery	Mennyire hajlamos a játékos belemenni olyan szituációba, ami fájdalommal vagy sérüléssel járhat
Composure	Mennyire jól tűri a játékos a mentális nyomást meccs közben
Concentration	Mennyire tud koncentrálni a játékos a mérkőzés egésze alatt
Vision	Mennyire jól látja meg a játékos a megjátszható csapattársait
Decisions	Mennyire jó döntéseket hoz a játékos a labdával
Determination	Mennyire ad bele mindent a játékos a siker érdekében
Flair	Mennyire tud a játékos váratlanokat húzni a labdával
Leadership	Mennyire inspiráló és motiváló a játékos személyisége
OffTheBall	Mennyire jól tud a játékos a mozgásával zavart kelteni az ellenfélnél
Positioning	Mennyire jól helyezkedik a játékos labda nélkül
Teamwork	Mennyire pontosan követi a játékos a taktikai utasításokat
Workrate	Mennyire jól teljesít fizikálisan a játékos a meccs alatt

Fizikai tulajdonságok

Acceleration	Mennyire gyorsan éri el a játékos a maximális sebességét
Agility	Mennyire jól tud a játékos megállni, elindulni és irányt váltani
Balance	Mennyire jól tud állva maradni a játékos fizikai nyomás alatt is
Jumping	Milyen magasra tud ugrani egy játékos fejelésnél
LeftFoot	Mennyire ügyesen használja a bal lábát a játékos
NaturalFitness	Mennyire jól tudja fenntartani a játékos az erőnléti állapotát meccsek között
Pace	Mennyire gyorsan tud futni egy játékos, ha eléri a maximális sebességét
RightFoot	Mennyire jól használja a jobb lábát a játékos
Stamina	Mennyire nehezen fárad ki a meccs közben a játékos
Strength	Mennyire tud a játékos fizikai előnyt szerezni egy ellenféllel szemben
Consistency	Mennyire valószínű, hogy a játékos a ki tudja aknázni a képességeit a meccsen
Dirtiness	Mennyire gyakran szokta a játékos megszegni a szabályokat, hogy ezzel előnyt szerezzen
ImportantMatches	Mennyire jól teljesít a játékos a nagy, fontos meccseken
InjuryProness	Mennyire valószínű, hogy a játékos megsérül meccsen vagy edzésen
Versatility	Mennyire jól teljesít a játékos egy számára idegen pozícióban
Adaptability	Mennyire könnyen tud a játékos új országban letelepedni
Ambition	A játékos mennyire törekszik a siker érdekében
Loyalty	A játékos mennyire becsületes, hűséges a csapatával szemben

Pressure	A játékos mennyire jól teljesít nyomás alatt
Sportsmanship	Mennyire sportszerű a pályán a játékos
Temperament	Mennyire könnyen veszíti el a hidegvérét a játékos
Controversy	A játékos mennyire vállalja fel a konfliktusokat

2. Melléklet – Egységtesztek leírása, eredménye

Webalkalmazás üzleti logikájának egységtesztjei

Függvény neve	Leírás	Bemenet	Kimenet	Tesztek kimenete
CheckboxValueToBoolean	A http kérésből a megadott jelölőnégyzet paraméter értékét Igaz/Hamis értékké alakítja, attól függően, hogy a jelölőnégyzet be van-e pipálva	Http kérés, Jelölőnégyzet neve	Igaz/Hamis	Mivel http kérést kezel, nem tudtam egységtesztet írni rá, de a tesztesetek során hiba nélkül működött
ColumnVisibilitiesInit	Az alkalmazás összes betöltésekor a mezők láthatóságát Igaz értékre állítja	Oszlopok láthatóságát tartalmazó lista	Oszlopok láthatóságát tartalmazó lista, Igazra állított értékekkel	SIKERES
CreateOrderByList	A http kérés alapján egy listát készít a sorba rendezési mezők értékeivel	Http kérés	A sorba rendező mezők értékeit tartalmazó lista	Mivel http kérést kezel, nem tudtam egységtesztet írni rá, de a tesztesetek során hiba nélkül működött
	A sorba rendező mezők értékeit tartalmazó	A sorba rendező mezők	SQL szintaktikájú sorba	SIKERES

CreateOrderByString	listából egy SQL szintaktikát követő sorba rendezési (ORDER BY) feltételt állít elő	értékeit tartalmazó lista	rendezési feltétel	
CreateWhereString	A szűrő mezők értékei alapján egy SQL szintaktikájú (WHERE) lekérdezési felételt hoz létre	Http kérés, szűrő mezők nevei	SQL szintaktikájú lekérdezési feltétel	Mivel http kérést kezel, nem tudtam egységtesztet írni rá, de a tesztesetek során hiba nélkül működött

Gépi Tanuló komponens üzleti logikájának egységtesztjei

Függvény neve	Leírás	Bemenet	Kimenet	Teszt kimenete
FindTargetAbilities	A megadott játékosnak megkeresi azt a 20 tulajdonságát, amiben a legjobb	Játékos azonosítója	Játékos 20 legjobb tulajdonságát tartalmazó lista	SIKERES
FindPlayersWithSimilarAbilities	Megkeresi azokat a játékosokat, akiknek 20 legjobb tulajdonsága közül legalább 13 szerepel a paraméterként kapott tulajdonság listában	20 tulajdonságot tartalmazó lista	Játékosazonosítók listája	SIKERES

FilterPlayers	Megkeresi azokat a játékosokat, akik a megadott tulajdonságokban legalább 7 értékkel rendelkeznek	Tulajdonságokat tartalmazó lista	Játékosazonosítók listája	SIKERES
DetermineOptimalKValue	A kapott játékos entitások tulajdonságait figyelembe véve meghatározza, hogy hány halmazba lehet őket optimálisan besorolni	A játékos entitások (Pandas Dataframe adattípus)	Halmazok optimális száma	SIKERES
CompareLists	Összehasonlítja két megadott listát és meghatározza, hogy hány közös elemük van	Lista1, Lista2	Közös elemek száma	SIKERES
GetPlayersOnPosition	Kiválogatja a megadott pozícióban játszó játékosokat	Pozíció	Játékosazonosítók listája	SIKERES