# python3

# postgresql

## Adatbázis létrehozása

Az előzetes tervekhez képest változtattam az adatmodellen.

* A Score táblában lesznek a játékosokhoz tartozó pontszámok. Ez fogja az összehasonlítások alapját képezni.
* A Player táblában találhatóak a játékosok személyére vonatkozó adatok.
* A Club táblában található a játékos csapata; a liga, amelyben játszik és hogy melyik ország, hanyadik divíziójában található (Pl.: Spanyol 2.osztály )



A több adatpont miatt egy másik adatforrást is választottam mellé (<https://www.kaggle.com/datasets/ktyptorio/football-manager-2020>). Ugyanannak az adatbázisnak egy későbbi (2020-as) verziója.

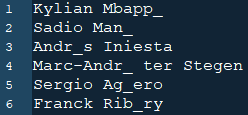
1. Adattisztítás

Az új (2020-as) adatforrás nem megfelelő karakterkódolással lett eredetileg elmentve, ennek következményeképp a játékosok nevében lévő speciális karakterek nem jól jelennek meg:



Megoldás:

* 1. Exportáltam az új adatforrásból érkező neveket egy csv 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 ’\_’ karaktert helyettesített.



* 1. Ezt az új adathalmazt visszatöltöttem a PostgreSQL szerverre.
  2. Az régebbi (első) adathalmaz helyesen tartalmazza ezeket a neveket. A helyes neveket összevetettem a helytelen nevekkel és ahol csak egy karakter eltérés volt két név között, azt megtalálta.

(Az SQL-ben a string összehasonlításoknál a ’\_’ karakter azt jelenti, hogy annak a helyén bármilyen karakter szerepelhet. Ezért ezt helyettesítettem be a helytelen karakterek helyére).

* 1. A hibás nevek helyére behelyettesítettem a javított változatokat.

Ezzel a megoldással több, mint 10.000 értéket sikerült kijavítanom. További 9000 hibás adatot el kellett távolítanom, mivel azok nem szerepeltek a korábbi adathalmazban (valószínűleg ezek a játékosok befejezték az aktív pályafutásukat).

Ezenkívül voltak olyan játékosok, ahol nem volt megadva csapat. Itt egységesen ’Unknown’ értéket adtam meg nekik, hogy a megjelenítésnél látszódjanak ezek az értékek.

1. Adatstruktúra kialakítása

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

Létrehozok egy nagy ideiglenes táblát, amibe beleteszem mindkét adathalmaz adatait (játékosnév szerint megfeleltetve egymásnak). Ezt az ideiglenes táblát fogom darabolni és a végső táblákká alakítani.

* Először a Club táblát állítom össze, mivel ez a legkisebb és nincsen idegenkulcsa.
* Másodikként a Score táblát állítom össze. Ennek a mezőit mellékletként fejtem ki, a dokumentum végén.

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 lőni 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: Menniyre 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
* Professional: a játékos mennyire számít profinak
* Sportsmanship: Mennyire sportszerű a pályán a játékos
* Temperament: Mennyire könnyen veszíti el a hidegvérét a játékos
* Controversy: A játékos mennyire vállalja fel a konfliktusokat
* PositionsDesc:

A játékos pozíciójára vonatkozó adatok:

* Goalkeeper: Mennyire jól teljesít kapusként
* Sweeper: Mennyire jól teljesít söprögetőként (a kapus és a védők között)
* Striker: Mennyire jól teljesít támadóként
* AttackingMidCentral: Mennyire jól teljesít középső támadó középpályásként
* AttackingMidLeft: Mennyire jól teljesít baloldali támadó középpályásként
* AttackingMidRight: Mennyire jól teljesít jobboldali támadó középpályásként
* DefenderCentral: Mennyire jól teljesít középső védőként
* DefenderLeft: Mennyire jól teljesít baloldali védőként
* DefenderRight: Mennyire jól teljesít jobboldali védőként
* DefensiveMidfielder: Mennyire jól teljesít védekező középpályásként
* MidfielderCentral: Mennyire jól teljesít középső középpályásként
* MidfielderLeft: Mennyire jól teljesít baloldali középpályásként
* MidfielderRight: Mennyire jól teljesít jobboldali középpályásként
* WingBackLeft: Mennyire jól teljesít baloldali szárnyvédőként
* WingBackRight: Mennyire jól teljesít jobboldali szárnyvédőként
* Utolsóként állítom össze a Player táblát, így be tudom rajta állítani a már meglévő táblákra vonatkozó idegenkulcsokat is.

1. s
2. s

# django

telepíteni kell a következő csomagokat: pip install pandas, sqlalchemy, sklearn, psycopg2

mvt leírás

dtl DJANGO TEMPLATE LANGUAGE

első django alkalmazás egy lightweight serverrel - localhost:8000, amit a django biztosít -> megvizsgál, hogy elég lesz e vagy kell az apache

első alkalmazás -> Hello World

Ciklus, if, template generálás

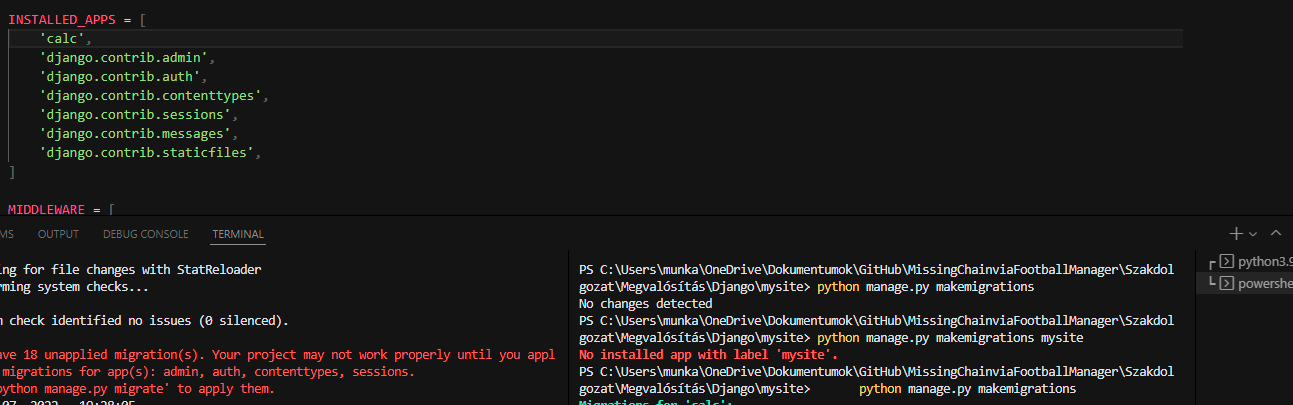
ORM – Object Relationship Mapper -> KIFEJTÉS

Adatbázis kapcsolathoz kell

Tábla = Osztály, Mező = Tulajdonság, Oda-Vissza működik

1. Modell létrehozás

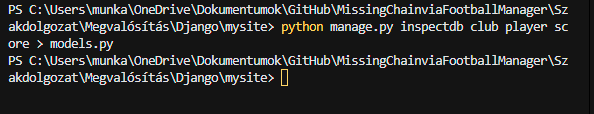




# 



Adatbázis táblából Model generálás:



# apache mod\_wsgi xampp

# Python

Külső függvényeket kell meghívni, amik visszaadják az ajánlott játékosok ID-ját.

#1:

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, referenciaként szolgáló játékos helyére.

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:

1. Alapvetően nem az összes tulajdonságot vettem figyelembe, hanem amikben a kiválasztott játékos a legjobb. A többi, gyengébb képességei nem relevánsak.
   1. Meghatároztam a kiválasztott játékos legjobb 20 (ez a viszonylag magas szám azért kell, hogy a játékos sok tulajdonságát figyelembe vehessük) tulajdonságát. Ez úgy történik, hogy a *score* táblára egy PIVOT átalakítást (elforgatást) végeztem, egy mezőbe rakva a játékos tulajdonságait, amit sorba rendezve kiolvastam első 20 legjobbat.
   2. A többi játékosra ezáltal már el tudtam végezni egy szűrést. A feltétel szerint a játékosok ezen 20 tulajdonsága legyen jobb a 7 értéknél. Ez az érték alacsonynak számít, de arra jó, hogy a teljesen más profilú játékosokat kiszűrje. (Például, ha egy csatárhoz hasonló játékosokat keresünk, akkor a kapusok és védők egyből kiesnek, mert a csatárokra jellemző tulajdonságok náluk alacsony pontszámot kaptak). Ezzel a módszerrel a nagyságrendileg 140.000 sorral rendelkező táblát, sikerült néhány ezerre redukálni.
   3. A szóba jöhető játékosok szűrésének következő lépésként, szerettem volna megkeresni a maradék játékosok közül, hogy kiknek egyezik meg a 20 legjobb tulajdonsága a kiválasztott játékos 20 legjobb tulajdonságával. Sajnos jellemzően olyan kevés egyezést találtam, hogy azokra nem lehetett volna egy gépi tanuló algoritmust alkalmazni.

Ennek megoldásaként kevesebb egyező tulajdonsággal is megelégszem. Az alsó elfogadási határt 100 véletlenszerű játékos példáját tekintve határoztam meg.

x.x ábra

Az x.x ábrán látható 100 játékos átlagát figyelembe véve, hogy a megkövetelt egyező tulajdonságok számával hogyan csökken az egyező játékosok száma. Minél magasabb az egyező tulajdonságok száma, annál jobban hasonlít a vizsgált játékos a referenciaként szolgáló játékosra.

Mivel ez egy átlagolt érték, bizonyos játékosok esetén kevesebb a hozzá hasonló játékosok száma. A 11-et választottam elvárt egyezési számnak, így biztosan maradni fog több száz tovább vizsgálható játékos.

Ez azt jelenti, hogy csak azokat a játékosokat vizsgáljuk, akiknek a legjobb 20 tulajdonsága közül legalább 11 megegyezik a referenciaként szolgáló játékos legjobb 20 tulajdonságával.

* 1. Az így kapott néhány száz játékosra már könnyen lehet gépi tanuló algoritmusokat alkalmazni.

1. Az algoritmus lényege, hogy szeretnénk a játékosokat különböző csoportokba sorolni a képességeik alapján. A felhasználó az általa kiválasztott játékossal azonos csoportban lévő többi játékost fogja ajánlásként visszakapni. Ezt a feladatot valamilyen Clustering algoritmussal tudjuk megoldani:
   1. A elterjedt Clustering algoritmusok jellemzően 2 adatdimenzióra értelmezhetők. Mivel mi a játékosok (a referenciaként szolgáló játékos legjobb 20 tulajdonságával megegyező) 20 tulajdonságát vizsgáljuk, ez a szükséges 2-vel szemben 20 dimenziót jelent.
   2. Ki kell választani a megfelelő cluster számot
   3. Ha megtaláltuk a megfelelő clustereket, megnézzük, hogy melyik clusterbe került a kiválasztott játékos és visszaadjuk a többi játékos ID-ját ugyanabból a clusterből.

*A Clustering lényege, hogy az algoritmus az egyedeket a tulajdonságaik alapján halmazokban rendezi, ezeket a halmazokat viszont nekünk kell tudni értelmezni.*

#2 Alulértékelt játékosok keresése