Diákok teljesítményének osztályozása életviteli mutatók alapján

Marsó Máté
NKDG9I
Óbudai Egyetem
Alba Regia Műszaki Kar
Székesfehérvár, Magyarország
marso.mate@stud.uni-obuda.hu

Pásztóhy László Ádám
H3A8UV
Óbudai Egyetem
Alba Regia Műszaki Kar
Székesfehérvár, Magyarország
pasztohylaszlo@stud.uni-obuda.hu

Absztakt - Ez a dokumentum diákok tanulási- és munkabéli teljesítményének, koncentrációjának alakulását vizsgálja az életükben történő különböző események, és mutatók, valamint a családi hátterük alapján. Az ezen eseményeket és mutatókat tartalmazó adathalmazzal feltanításra került egy neurális háló, mely ezáltal képes megjósolni egy tanuló jövőbeli teljesítményét a meghatározott adatok alapján. Kutatásunk célja, hogy elemezni tudjuk, hogy miként változik a diákok teljesítménye az életbeli mutatóik alapján, valamint hogy a neurális háló segítségével jóslatokat tudjunk tenni ezen teljesítmény alakulására.

Kulcsszavak: ***tanulás, neurális háló, életvitel, diákok***

I. Bevezetés

A projekt célja egy teljesítmény prediktáló osztályozó neurális háló felépítése, mely a kapott adatok elemzésével megadja az aznapi teljesítményt. Ezzel szeretnénk egy olyan szoftvert létrehozni, amivel egy életviteli mutatókból álló adathalmaz alapján osztályozni tudjuk a diákok teljesítményét. A kutatásunkban keressük az optimális állapotot a legjobb teljesítmény eléréséhez. A projekt prediktív mivoltja azt takarja, hogy a kutatás végeztével birtokunkban lesz egy olyan program, mely képes a fentebb említett mutatók megadása után előrejelezni a lehetséges teljesítményt. A motivációnk a diákok szokásainak teljesítményükre gyakorolt hatásának feltérképezése.

II. A téma bemutatása

A. Big Data

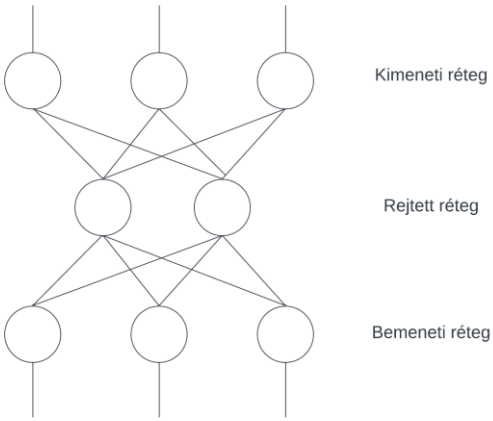
Egy olyan kifejezés, amelyet egy hatalmas mennyiségű strukturált és/vagy strukturálatlan adatra használnak, amelyből ismereteket nyernek ki. Ennek a mennyiségnek olyan óriási méretei vannak, hogy nagyon nehéz feldolgozni őket a hagyományos adatbázis- és szoftvertechnikák segítségével. [1]

B. Neurális hálók

A neurális hálók az emberi agy matematikai szimulációi. Az emberi agy megközelítőleg 10^{11} neuronból áll, melyek kémiai reakciók során kommunikálnak egymással (neurotranszmitterek). Bizonyos bemenetek hatására az adott neuron – ha azok átlépik annak ingerküszöbét – kimenő jelet küld a körülötte álló neuronoknak, így létrehozva egy kommunikációs hálót. [2]

Ennek gépi szimulációja, a mesterséges neuron *n* bemenetet hatására produkál egy kimenetet (vagy kimenetek sorozatát). Ezek a bemenetek érkezhetnek a hálón kívülről, vagy a hálóban szereplő többi, szomszédos neurontól. Egy neuronnak lehet aktiválási függvényt meghatározni, mely meghatározza, hogy milyen erősségű bemeneti jel szükséges az adott neuron aktiválásához. Ezzel kiküszöbölhető a neuronok túlérzékenysége. [2]

Ezek a neuronok rétegekbe vannak szervezve, és ezáltal kapcsolódnak egymáshoz. Általában létezik egy bemeneti réteg, valamint egy kimeneti réteg. Ezek között találhatóak meg a rejtett rétegek. Az összes réteg együttesen alkotja a neurális hálót, melynek több változata is van: létezik generatív háló, mely létrehoz adatokat (szöveget, képeket, stb…), valamint létezik osztályozó háló, mely a kapott adathalmazt dolgozza fel, és a megtanult adatok alapján sorolja osztályokba az újonnan beérkező adatokat. [2]



1. ábra: A legegyszerűbb, háromrétegű neurális háló felépítése forrás: saját szerkesztés a **[2]** *melléklet alapján*

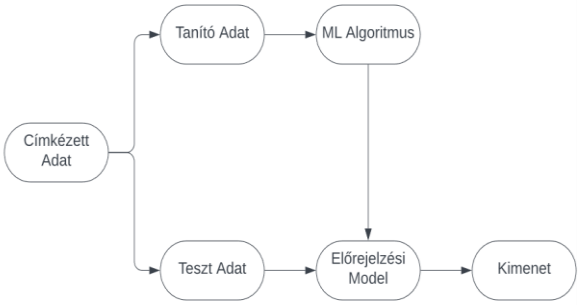
Ebben a kutatásban egy osztályozó hálót fogunk használni, melynek tanítási folyamatát, valamint az eredmények kiértékelését tartalmazza ezen dokumentum.

C. Felügyelt tanulás

A gépi tanulási algoritmusok minden adatpontot az adatkészletben (Dataset) tulajdonságok gyűjteményeként kezelnek. [3]

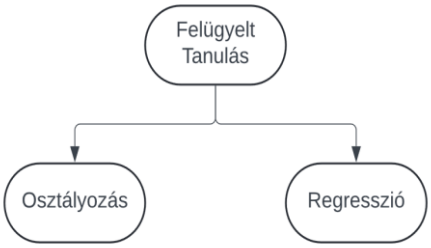
Ha az adatpontok címkézettek, akkor ezt a tanulási típust felügyelt tanulásnak nevezik. A felügyelt tanulás a modellek tanítását jelenti címkézett adatokon, majd a tesztelésüket címkézetlen adatokon. [3]

A felügyelt tanulás alapvető architektúrája a következő lépéseket foglalja magában: adathalmaz gyűjtése, az adathalmaz felosztása tesztelési és tanítási adatokra, majd az adat előfeldolgozása. A kinyert tulajdonságokat beillesztik egy algoritmusba, majd a modellt megtanítják azoknak a tulajdonságoknak a megtanulására. [3]



2. ábra: A felügyelt tanulás alapvető architektúrája **[3]**

Az osztályozás és a regresszió a felügyelt tanulás két nagy típusa. A következőkben az osztályozással fogunk foglalkozni.



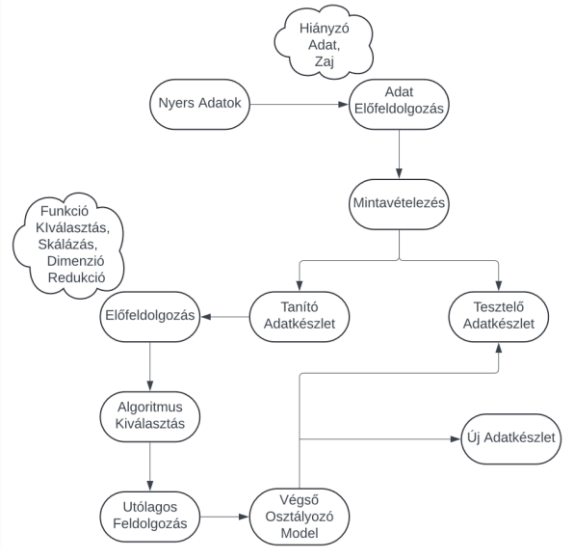
3. ábra: Felügyelt tanulás típusai **[3]**

D. Osztályozási modell

A felügyelt osztályozás az úgynevezett Intelligens Rendszerek által leggyakrabban végzett feladatok egyike. [4]

A klasszifikáció során egy modell ismeretlen értékeket (kimenetek halmaza) jósol egy ismert értékek halmaza (bemenetek halmaza) alapján. Ha a kimenet kategorikus formában van, akkor a problémát klasszifikációként említjük. Általában a klasszifikáció során az adathalmaz példáit meghatározott osztályokba sorolják. [3]

Egy klasszifikációs modell egy leképzési függvényt használ, amelyet a modell a tanítóadatokból következtet, hogy megjósolja a tesztadatok osztálycímkéjét. Végül egy jellemző társul az adathalmazhoz, amely segít egy pontos előre jelző modell létrehozásában. [3]



4. ábra: A felügyelt tanulásban történő osztályozás munkafolyamata **[3]**

III. KUTATÁSMÓDSZERTAN

A. Az elkészítendő neurális háló

Az előző fejezetekben kifejtett okokból egy *felügyelt tanítás* módszerével betanított *neurális háló* létrehozása a cél, mely a hallgatókat teljesítményük alapján képes *osztályozni*. A tanítás elvégzéséhez azonban adatokra van szükség.

B. Felhasználható adatok keresése

A kutatás céljául kitűzött neurális háló létrehozásához szükség van nagy mennyiségű adatra (Lásd *II.A. Big Data*).

Ezen adatok összegyűjtése többféle módon valósulhat meg. A legalapvetőbb módszer az egyetemi csoporttársaink, valamint körülöttünk élő emberek megvizsgálása, vagy megkérdezése lett volna. Ezen módszerrel azonban nem lehetett volna nagyobb időszakot átívelő, konzisztens és pontos adathalmazt összegyűjteni. A hallgatótársak több napon, vagy akár héten keresztüli monitorozása túl hosszadalmas műveletnek bizonyult a kutatás elvégzésére kapott idő korlátossága miatt.

Másik lehetőség egy már elvégzett kutatás adathalmazának felhasználása, melynek előnyei, hogy bárki számára elérhető, megbízható információt tartalmaz, valamint felhasználásra alkalmas formátumban van. Ezen okok miatt egy létező adathalmazzal történik a neurális háló tanítása.

Ez a *Student performance* [5] nevű adathalmaz, mely két portugál középiskola diákjainak adatait tartalmazza. Az adathalmazban megtalálható 649 diák adata, valamint diákonként 30 tulajdonság, és 3 különböző teszt eredménye. Ez *csv* formátumba exportálás után már használható is, azonban ezen kutatás egyetlen egy végeredmény osztályozására irányul. Ennek érdekében az előre elkészített adathalmazban szereplő 3 különböző teszt eredményéből egy átlagolt eredmény kiszámítása szükséges.

C. *Adatok átalakítása*

A kiválasztott adathalmazt személyre kell szabni, mivel sok adatra nincs szükség, vagy nem olyan formában kell felhasználni, amilyenben kaptuk. Az adatok transzformációját egy Python script segítségével oldottuk meg. Azért a Python nyelvet alkalmaztuk, mert ez egy viszonylag könnyen programozható, rövid feladat és az ilyenekre kiválóan használható ez a nyelv. A modell tanítását is Pythonban programoztuk.

A student.csv adathalmazból csináltunk egy selected.csv adathalmazt.

Először kiszelektáltuk azokat az adatokat, melyekre a mi feladatunk szempontjából nem volt szükség.

A kiinduló adathalmaz utolsó három oszlopa három eredményt jegyez. Ezt a három eredményt alakítottuk át egy 1 – 5-ig terjedő egész számra. Ezek az osztályozási modell osztályai. Az eredményt a következőképpen állítottuk elő:

```
if (not firstLine):
    if words[len(words)-1] and words[len(words)-2] and words[len(words)-3]:
        x = ((int(words[len(words)-1]) / 3) + (int(words[len(words)-2]) / 4) +
             (int(words[len(words)-3]) / 4)) / 3
        if x > 5:
            x = 5
        if x < 1:
            x = 1

        x = str(int(round(x, 0)))
    else:
        x = ''  # Vagy valamilyen más alapértelmezett érték, ha az adat nem érvényes
```

5. ábra: Eredmény képzése
Forrás: Saját szerkesztés

A fent említett kódrészlet leírja, hogy a három számnak az átlagát vettük, lenormáltuk egy 1 és 5 közötti számra, majd megnéztük, hogy megfelelő értéket kaptunk-e. Ha nem, akkor kijavítottuk azt. Az algoritmust úgy terveztük meg, hogy lehetőleg minden adatból kapjunk, így nem pontosan átlagot számoltunk. A folyamat végén egész számra kerekítettünk, így kaptuk meg a Result oszlop eredményét.

Az osztályozási modellt úgy lehet feltanítani és tesztelni, ha minden adat szám. Így át kellett alakítani az összes szöveges adatot számmá. Ezt is a kódban oldottuk meg. Minden szöveges adatot megfeleltettünk egy számnak. Ennek a dokumentációját meg lehet találni a Python kódban.

A kódot függvények használatával tettük struktúrálttá és újrafelhasználhatóvá. A transform_data nevű függvény A szöveget számokká alakítja. A calculate_result nevű függvény az eredményt számolja ki. A process_file függvény pedig a filekezelésért felel. Az így megalkotott három függvénnyel könnyen manipulálható az adat.

A programban minden fontosabb lépés kommentálva van, így könnyen értelmezhető.

Az így kapott selected.csv adathalmaz megfelel a kívánt tanítóhalmaznak. A továbbiakban ezzel fogunk tovább dolgozni.

D. *A felhasznált adatok*

A megszűrt és a kutatás céljainak megfelelően átalakított adathalmaz a diákok következő adatait tartalmazza:

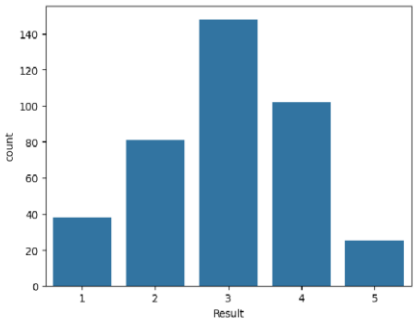
- **Alapvető adatok:** a diák neme, életkora, élőhelye (város vagy vidék)
- **Családi adatok:** diák családjának mérete, szüleinek együttélése (együtt élnek, vagy külön), anyjának legmagasabb képzettsége, apjának legmagasabb képzettsége, törvényes képviselője (apa vagy anya)
- **Iskolával és tanulással kapcsolatos adatok:** iskola kiválasztásának oka (pl.: iskola hírneve), mindennapi utazási idő a diák otthonától az iskoláig, hetente tanulással eltöltött idő, diák múltbéli bukásainak száma, iskola általi extra tanulmányi segítség (igen vagy nem), szülők általi extra tanulmányi segítség (igen vagy nem), fizetett különórákon való részvétel (igen vagy nem), tanórákon kívüli szakkörökön való részvétel (igen vagy nem)
- **Egyéb személyes adatok:** járt-e óvodába, szeretne-e felsőoktatásban tanulni, van-e internet-hozzáférése, van-e párkapcsolata, a diák családi kapcsolatainak minősége (1 és 5 között értékelve)
- **Szabadidőre vonatkozó adatok:** diák iskola utáni szabadideje (1 és 5 között értékelve), barátokkal való találkozás gyakorisága (1 és 5 között értékelve), munkanapokon való alkoholfogyasztás gyakorisága (1 és 5 között értékelve), hétvégéken való alkoholfogyasztás gyakorisága (1 és 5 között értékelve)
- **Egészségre vonatkozó adatok:** a diák jelenlegi egészségügyi állapota (1 és 5 között értékelve), iskolából hiányzott napok száma

Ezek alkotják a tanítási adatokat. Ezen kívül még minden diákhoz tartozik egy célattribútum, mely a fentebb említett módszerrel kiszámított végeredmény, melyek szerint a neurális háló osztályozza a diákok teljesítményét.

IV. Kutatás megvalósítása

A. *Az adatok elemzése*

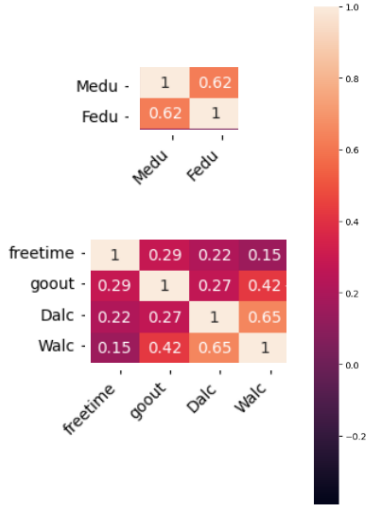
Az osztályozó neurális háló felépítésére a Google Colab által bárki számára elérhető futtatókörnyezetben [6] került sor, Python programozási nyelven. A kutatás számára megfelelően átalakított adatokat .csv formátumban be kellett olvasni, és ezt egy speciális adathalmazként, úgynevezett DataFrame-ként felhasználni a következőkben. Ebben a DataFrame-ben külön szerepelnek a megtanulandó változók egy táblázatban, valamint ezek eredményei egy egysoros táblázatban (vektorban) [5. ábra], az osztályozó neurális hálók felügyelt tanítási módszerének megfelelően.



Megszerzett jegy	1	2	3	4	5
Létszám	38	81	148	102	25

5. ábra: Az adathalmazban szereplő tanulók eredményeinek eloszlása
Forrás: saját szerkesztés

Az adatok beolvasása után következik ezek elemzése, melyhez a Seaborn Python csomag nyújt segítséget. A Seaborn egy Matplotlib-en alapuló adatvizualizációs könyvtár, mellyel a modellben felhasznált adatok összefüggéseit lehet vizsgálni [7]. Így az adatokat egy hőtérképen ábrázolva megtudhatjuk, hogy mely adatok milyen mértékben függenek egymástól. Ez segít ezen adatok további elemzésében, valamint túlságosan nagy összefüggési mutató esetén bizonyos adatok elhagyhatóak a modell tanítása során. A jelenlegi modell hőtérképe azt mutatja, hogy a tanuló apjának és anyjának legnagyobb végzettsége mutat összefüggést [6. ábra], amelyből arra következtethetünk, hogy általában hasonló edukációs hátterű emberek alkotnak egy párt. Hasonló összefüggés figyelhető meg továbbá a tanuló hétköznapi-, és hétvégi alkoholfogyasztása között, valamint ezen mutatók és a tanuló szabadidejében barátaival való találkozásainak gyakorisága, és a tanuló szabadideje között [6. ábra].



6. ábra: Összefüggés bizonyos adatok között. Minél nagyobb az érték, annál nagyobb az összefüggés mértéke
Jelmagyarázat: Medu = anya végzettsége; Fedu = apa végzettsége; freetime = tanuló szabadideje; goout = tanuló barátokkal töltött ideje; Dalc = tanuló hétvégi alkoholfogyasztása; Walc = tanuló hétköznapi alkoholfogyasztása
Forrás: részletek a Seaborn könyvtár által készített hőtérképből, saját szerkesztés

Bár ezen összefüggések felfedezhetőek, mértékük mégsem olyan magas, hogy bármely adatok törlését vagy mellőzését indokolnák.

B. *A neurális háló felépítése*

Az adatok elemzése után megkezdődhet a neurális háló felépítése. Ehhez először szét kell választani az adathalmazt úgy, hogy legyen egy tanuló adathalmaz, ennek célértékei, valamint egy validációs (tesztelő) adathalmaz, és ennek cél-értékei. Ennek elérése érdekében a beépített *train_test_split* függvényt használtuk [7. ábra], melynek paramétereként megadtuk a szétválasztás mértékét. Ez alapján a modell a teljes adathalmaz 75 százalékát használja fel tanításra, 25 százalékát pedig a megtanult minták hatékonyságának tesztelésére. Ennek eredményeképp a modell képes lesz saját hatékonyságának ellenőrzésére, mely segítségével még pontosabb eredmények jöhetnek létre.

```
X_train, X_test, y_train, y_test =
  train_test_split(df, target, test_size = 0.25, random_state=12345)
```

7. ábra: Az adathalmaz ketté választása: 75% tanuló, és 25% tesztelő részre (test_size). A random_state paraméter a szétválasztás véletlenszerűségét befolyásolja
Forrás: saját programkód

Az adathalmaz szétválasztása után következik a modell betanítása. Ehhez az SKLearn SVC (Support Vector Classification) [8] könyvtárát használjuk, mely képes az adatok lineáris osztályozására. Ezt a könyvtár beépített *fit* metódusával [8. ábra] lehet megtenni, mely a tanuló adathalmazt és a célértékeket felhasználva megtanulja az adatok megfelelő kombinációinak illesztését.

```
svc_model.fit(X_train, y_train)
```

8. ábra: A modell tanítása a korábban meghatározott tanuló adathalmazzal és annak célértékeivel
Forrás: saját programkód

C. *A betanított háló kipróbálása*

Az elkészült osztályozó modell működésének tesztelésére is létezik beépített metódus, mégpedig a *predict*. Ez megvizsgálja a teszt adathalmaz adatait, és azokra megpróbálja ráilleszteni a korábban megtanult mintákat. Ha ezekből a jósolt eredményekből, és a teszt adathalmaz tényleges célértékeiből is készítünk egy táblázatot (DataFrame-et), és ezeket egymás mellé tesszük, akkor megfigyelhető a tényleges és a jósol eredmények közötti különbség. Ez a folyamat látható a 9. ábrán.

Jóslatok elvégzése a teszt adathalmazon:

```
y_predict = svc_model.predict(X_test)
```

Táblázat készítése a teszt adathalmaz tényleges célértékeiből (Result-okból):

```
test_df = pd.DataFrame(y_test,columns=['Result'])
```

Táblázat készítése a jósolt eredményekből:

```
predict_df = pd.DataFrame(y_predict,columns=['predict'],index=test_df.index)
```

Két táblázat összevetése:

```
result_df = pd.concat([test_df,predict_df],axis=1)
result_df
```

	Result	predict
68	2	3
203	2	3
363	4	4
356	4	4
391	4	4
...	...	...
33	3	3
377	3	3
4	2	3
268	3	3
16	4	4

*9. ábra: A modell kipróbálásának folyamata. A **Result** nevű oszlop tartalmazza a tényleges, a **predict** nevű oszlop pedig a jósolt értékeket*
Forrás: saját programkód alapján saját szerkesztés

Ebből az összesített táblázatban látható az, hogy a ténylegesen 3 és 4 eredményeket elérő tanulók esetén a

jóslatok helyesek, a szélsőségeket közelítve viszont akár egy teljes jegy eltérés is előfordulhat a tényleges és a jósolt eredmény között. Ennek oka feltételezhetőleg az, hogy 1-es és 5-ös megszerzett jeggyel kevés tanuló rendelkezik [lásd 5. ábra]. Így ezen szélsőséges adatok mintázatainak megtanulása nem sikerült teljes mértékben a megfelelő adatok alacsony száma miatt.

D. Saját adatokkal való kipróbálás

A betanított modellt, a tanítás és a tesztelés után saját adathalmazon is kipróbáltuk. Létrehoztunk egy data.csv file-t, amiben felvezettük a saját adatainkat.

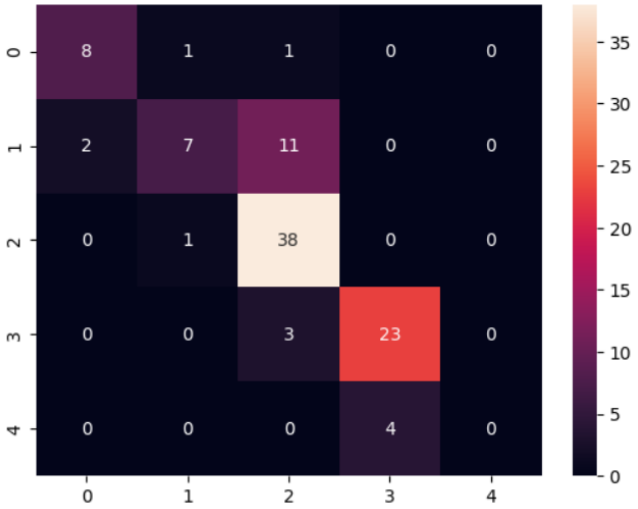
Az adathalmazt tíz valódi ember adatain teszteltük.

A data.csv file-nak létrehoztunk egy változót a programban és a teszt halmaz helyére behelyettesítettük a kódot, így azt tudtuk vizsgálni.

A vizsgálat eredménye, hogy a modell képes osztályozni a kapott adatokat a megadott skálán.

E. Modell optimalizálása

A kapott modellünk eredménye a következő:



6. ábra: Modell tesztelésének eredménye
Forrás: Saját munka

Ez azt jelenti, hogy az átlóban találhatóak azon eredmények, melyek a teszthalmazon végzett teszten találtak. Minél messzebb vannak a számok, annál rosszabb megoldást adott a modell a tesztre. A fent említett ábrán látszik, hogy a tanított modellünk egészen jól eltalálta a teszthalmazon tesztelt elemeket, azonban maximális eredményt senkinek sem adott. Ez abból eredhet, hogy kevés a tanító halmazon a maximális pontszámot elért emberek száma. Ezt a tanítóhalmaz javításával lehetne fejleszteni.

Az eredmény javítására a következő metódusokkal lehet javítani:

- Normalizálás: Az adatokat egy 0 – 1 közötti értékké normáljuk, így a program hatékonyabban tud tanulni.
- Paraméterek beállítása
- Döntési fa: Lehet más módszert alkalmazni a modell tanítására
- Legközelebbi szomszéd osztályozó algoritmus

V. Konklúzió

A korábban mutatott ábrák is azt mutatják, hogy közepes teljesítményű tanulóból van a legtöbb. Így a betanított modell a tanulók többségének eredményét sikeresen meg tudja jósolni, ezáltal a tanulók többségét sikeresen be tudja helyezni a definiált eredményosztályokba. A pozitív és negatív szélsőségek (1-es és 5-ös eredmény) alacsony létszáma a tanuló halmazban azt eredményezte, hogy a modell ezen szélsőségeket nehezen ismeri fel, mert nincs hozzá elegendő viszonyítási alapja. Így a létrehozott neurális háló modell használható saját célra. Segítségével megfigyelhető a tanuló családi hátterének befolyása a diák iskolai teljesítményére. Ezen kívül a tanuló szabadidejét, barátokkal való találkozásainak számát, tanulással eltöltött idejét, valamint alkoholfogyasztásának mennyiségét valós időben kitöltve a felhasználó egy jóslatot kaphat ezen tevékenységek melletti tanulmányi eredményére vonatkozóan. Ezáltal optimalizálható a diák időbeosztása, és könnyebben elérhető egy bizonyos tanulás-szabadidő egyensúly a lehető legjobb eredmény elérésének érdekében.

A modell továbbfejlesztésének érdekében szükséges lenne egy olyan adathalmaz létrehozása, melyben arányaiban hasonló mennyiségben szerepelnek az összes eredménykategória képviselői. Ezáltal a háló az összes kategória adatait azonos mértékben tudná megtanulni, így nem lenne elfogult egyikkel szemben sem.

VI. Összefoglalás

Az adatok alapján elkészítettünk egy osztályozó neurális hálót, amely képes megjósolni a diákok teljesítményét az életviteli mutatók alapján. A kutatás során egy portugál középiskolás diákok adatait tartalmazó adathalmazzal dolgoztunk, mely 649 diák adatait és többek között családi, iskolai és egészségügyi mutatókat tartalmazott. Az adatokat Python segítségével alakítottuk át és vizsgáltuk meg. A neurális háló betanítása és tesztelése után saját adatokat is felhasználtunk a modell kipróbálására, mely jól működött. Az eredmények alapján a modell jól teljesített a tesztelő adathalmazon, bár maximális pontszámot nem ért el senki, ami a tanító adathalmazban található kevés maximális pontszámot elért diáknak tudható be. A továbbiakban az eredmények optimalizálására lehetőség van a tanító adathalmaz bővítésével, az adatok normalizálásával és más modellek vagy paraméterezési módszerek kipróbálásával. A kutatás eredményei alapján a neurális háló segítségével hatékonyan lehet előrejelezni a diákok teljesítményét az életviteli mutatók alapján.

VII. HIVATKOZÁSOK

[1] C. Shavari, . S. Vijender K. és J. Madhuri S., The Basics of Big Data and Security Concerns, 2017.

[2] S. Mengel, W. Lively, Using a Neural Network to Predict Student Responses, 1992.

[3] S. Dridi, “SUPERVISED LEARNING - A SYSTEMATIC LITERATURE REVIEW,” 2021.

[4] K. S. B., Z. I. D. és P. P. E., „Machine learning: a review of classification,” 2007.

[5] Cortez,Paulo. Student Performance. UCI Machine Learning Repository. https://archive.ics.uci.edu/dataset/320/student+performance

[6] “colab.google,” *colab.google*. https://colab.google/

[7] seaborn, “seaborn: statistical data visualization — seaborn 0.9.0 documentation,” *Pydata.org*, 2012. https://seaborn.pydata.org/

[8] scikit learn, “1.4. Support Vector Machines — scikit-learn 0.20.3 documentation,” *Scikit-learn.org*, 2018. https://scikit-learn.org/stable/modules/svm.html