



M Ű E G Y E T E M 1 7 8 2

**Budapesti Műszaki és Gazdaságtudományi Egyetem**  
Villamosmérnöki és Informatikai Kar  
MIT Tanszék

Naán Gergely

**GÉNEK ÉS BETEGSÉGEK  
KAPCSOLATÁNAK FELTÁRÁSA GRÁF  
NEURÁLIS HÁLÓK ALKALMAZÁSÁVAL**

KONZULENS

**Gézi András**

BUDAPEST, 2025

# Tartalomjegyzék

|                                                                        |           |
|------------------------------------------------------------------------|-----------|
| <b>Összefoglaló .....</b>                                              | <b>4</b>  |
| <b>Abstract.....</b>                                                   | <b>5</b>  |
| <b>1 Biokémia bevezető.....</b>                                        | <b>6</b>  |
| 1.1 Genom.....                                                         | 6         |
| 1.2 Kromoszómák.....                                                   | 6         |
| 1.3 DNS .....                                                          | 7         |
| 1.4 RNS.....                                                           | 7         |
| 1.5 Gén.....                                                           | 8         |
| 1.6 Nukleotid .....                                                    | 8         |
| 1.7 Fenotípus.....                                                     | 8         |
| 1.8 Pleiotrópia.....                                                   | 8         |
| 1.9 Genetikai variáns .....                                            | 9         |
| 1.10 Gén-betegség kapcsolatok .....                                    | 9         |
| <b>2 Mi az a Gráf Neurális Háló? .....</b>                             | <b>13</b> |
| 2.1 Miért gráfok? .....                                                | 13        |
| 2.1.1 Hogy jelenik meg a gráf a számítógépben? .....                   | 13        |
| 2.1.2 Miért van szükség a gráfokra? .....                              | 14        |
| 2.2 Hogyan tanulható a gráf? .....                                     | 15        |
| 2.2.1 Mit akarunk meghatározni? .....                                  | 15        |
| 2.2.2 Mi a közös a modellekben? .....                                  | 15        |
| 2.2.3 Mitől tanul a modell? .....                                      | 17        |
| <b>3 Hierarchikus osztályozás.....</b>                                 | <b>19</b> |
| 3.1 Mi az a hierarchikus osztályozás?.....                             | 19        |
| 3.2 Milyen működési elvei vannak egy hierarchikus osztályozónak? ..... | 21        |
| 3.2.1 Exclusive Policy .....                                           | 21        |
| 3.2.2 Inclusive Policy.....                                            | 22        |
| 3.2.3 Siblings Policy .....                                            | 23        |
| 3.3 Hogyan alkalmazható egy hierarchikus osztályozó? .....             | 24        |
| <b>4 Adatbázis .....</b>                                               | <b>27</b> |
| 4.1 Encoding .....                                                     | 36        |
| 4.1.1 One-Hot Encoding .....                                           | 36        |

|                                              |           |
|----------------------------------------------|-----------|
| 4.1.2 Word2Vec Encoding.....                 | 37        |
| 4.2 Adathalmaz felosztása .....              | 37        |
| <b>5 A modell felépítése .....</b>           | <b>39</b> |
| 5.1 Mit próbáltam megjósolni? .....          | 39        |
| 5.2 A modell rétegei.....                    | 39        |
| 5.3 A hiperparaméterek kiválasztása .....    | 42        |
| <b>6 Kiértékelés .....</b>                   | <b>44</b> |
| 6.1 Pontosság .....                          | 44        |
| 6.2 F1 pontszám .....                        | 44        |
| 6.3 MCC.....                                 | 46        |
| 6.4 Osztályszintű pontosság.....             | 47        |
| 6.5 Kohen Kappa .....                        | 49        |
| 6.6 Következtetés.....                       | 52        |
| <b>7 Továbbfejlesztési lehetőségek .....</b> | <b>53</b> |
| <b>Irodalomjegyzék.....</b>                  | <b>54</b> |
| <b>Függelék.....</b>                         | <b>56</b> |

# Összefoglaló

Az emberiség folyamatos küzdelmet folytat a betegségekkel szemben. Az utóbbi években egyre nagyobb hangsúlyt kaptak azok a génkutatások, amelyek a különböző betegségek kialakulásának okait vizsgálják, különös tekintettel a gének és a betegségek közötti kapcsolatok feltárására. Dolgozatomban az emberi gének és betegségek közötti asszociációk ontológiai típusainak meghatározását tűztem ki célul, gráf neurális háló alapú hierarchikus osztályozó modell alkalmazásával.

A dolgozatban részletesen bemutatom a biokémiai alapfogalmakat, beleértve a gének, a DNS és a kromoszómák működését, valamint a fenotípusok kialakulásának hátterét. Ezt követően áttekintem a gráf neurális hálók elméleti alapjait és működését. A kapcsolatok összetett ontológiáját hierarchikus osztályozás segítségével modellezem – amelynek elméleti hátterét is részletesen kifejtem.

A kutatás során kidolgozott módszertan lehetővé teszi az emberi gének és betegségek közötti összefüggések pontosabb és átláthatóbb meghatározását. A hierarchikus osztályozás előnyeit és alkalmazási lehetőségeit számos statisztikai mutatókon keresztül szemléltetem, összehasonlítva a hagyományos neurális háló modellekkel.

A dolgozat bemutatja, hogy a gráf neurális háló alapú megközelítések – különösen hierarchikus osztályozókkal kombinálva – hatékony eszközként szolgálhatnak az emberi gének és betegségek közötti kapcsolatok feltárására. A módszer alkalmazása továbbá lehetőséget nyújt más gráfos adatstruktúrával leírható, összetett hierarchikus osztályozási feladatok pontosságának növelésére is.

# Abstract

Mankind continues its ongoing struggle against diseases. In recent years, gene research that investigates the causes of various diseases has gained increasing importance, with a particular focus on uncovering the connections between genes and diseases. In my thesis, I aim to identify the ontological types of associations between human genes and diseases using a hierarchical classification model based on graph neural networks.

In the thesis, I provide a detailed explanation of basic biochemical concepts, including the functions of genes, DNA, and chromosomes, as well as the background of phenotypic development. Following this, I review the theoretical foundations and functioning of graph neural networks. I model the complex ontology of relationships using hierarchical classification, which I also explain in detail in terms of theoretical background.

The methodology developed during the research allows for a more precise and transparent identification of the associations between human genes and diseases. I demonstrate the advantages and application possibilities of hierarchical classification through various statistical metrics, comparing it with traditional neural network models.

The thesis shows that graph neural network-based approaches, especially when combined with hierarchical classifiers, can serve as effective tools for exploring the relationships between human genes and diseases. Furthermore, the application of this method may also provide an opportunity to improve the accuracy of complex hierarchical classification tasks that can be described by other graph-based data structures.

# 1 Biokémia bevezető

A következő fejezetekben igyekszem bemutatni a biokémiai fogalmakat, melyekkel a dolgozatomban a továbbiakban találkozhatunk. Ezen fogalmak tipikusan a biokémia tudományterülethez tartoznak.

## 1.1 Genom

A genom egy élőlény teljes genetikai anyagát jelenti, amely tartalmazza a DNS-t és annak összes génjét. [1]

„AZ EMBERI GENOM (a petesejt vagy hímivarsejt teljes genetikai tartalma) úgy tekinthető, mint egy 23 kötetes könyv. Mindegyik kötet egy kromoszómát képvisel abból a 23 párból, amely minden emberi, testi sejtben megvan. Egy átlagos kötetnek kb. 2000 teljes oldala van, egy oldal minden gén számára. A gének kisebb fejezetekként jelennek meg; a fejezetek a kromoszómán egymás után következő, [többnyire együtt is működő] gének leírásait tartalmazzák. A teljes kötet sorozat, amelyet a szülő a gyerekére örökít, olyan részleteket tartalmaz, amelyek a nagyszülők kötetének véletlenszerű kombinációiból állnak. Egy adott kötet, amely továbbadódik, részleteket tartalmazhat az egyik szülőtől és a másiktól is, keverten, ugyanis a lapok a kötetek között is kicserélődhetnek a rekombináció következtében.” [2]

## 1.2 Kromoszómák

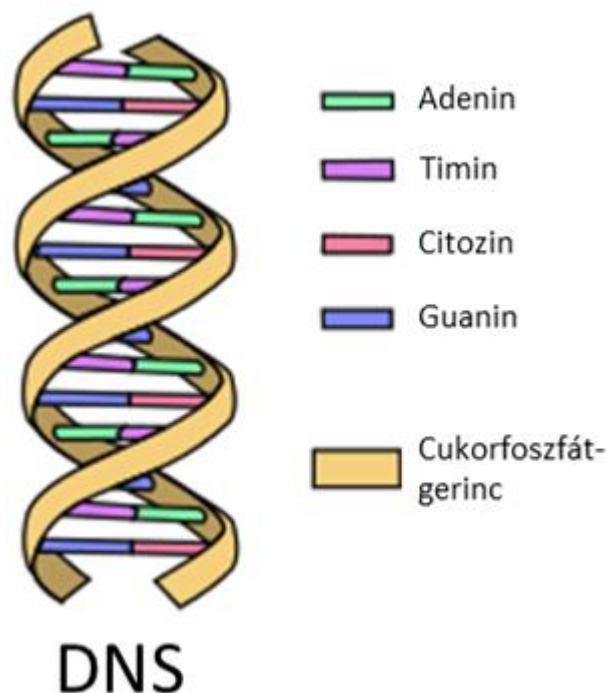
A kromoszómák a genom részei, amelyek a DNS-t és a hozzá kapcsolódó fehérjéket tartalmazzák, összecsomagolva, hogy kezelhetőbbé tegyék a genetikai információ tárolását és továbbítását a sejtosztódás során. (Tehát nem az összes DNS-t, csak egy DNS-szálat tartalmazzák) [3]

Hasonlóan a legtöbb növény- és állatfajhoz az ember is diploid ( $2n$ ), tehát a testi sejtekben minden kromoszómából két példány van jelen. Ezeket nevezzük homológ pároknak. Az embernek 23 homológ párja van, tehát összesen 46 kromoszómája. Az ősoktól történő fenotípusok öröklése a következő okok miatt történik: az emberi petesejtek és hímivarsejtek haploidok ( $1n$ ), tehát a 23 homolog párnak csak az egyik felét

tartalmazzák. A két haploid pár kombinációjaként teljesen új homológ párok jönnek létre, melyek tartalmazzák mindkét szülő örökítő anyagának egy részét. Így örökölhétünk meg fenotípusokat, például: szemszínt, hajszínt, adott esetben betegségre való hajlamot. [4]

### 1.3 DNS

A DNS (dezoxiribonukleinsav) a genom egyik fő összetevője, amely a sejtmagban található, és a genetikai információkat, örökítő anyagot tárolja. [5] „A DNS-ben négyféle nukleobázis: citozin, guanin, adenin és timin fordul elő. Az, hogy ezek egymás után hogyan sorakoznak fel a génen, az határozza meg egy bizonyos fehérje szerkezetének kialakulását. A fehérjék pedig az élő szervezetek legfontosabb anyagai, hiszen ezek a vegyületek az élőlények működésének szinte minden területén kulcsfontosságú szerepet játszanak.” [6] Az 1. ábrán lehet látni a DNS felépítését.



1. ábra: DNS felépítése[6]

### 1.4 RNS

Az RNS (angolul RNA) a DNS információjának közvetítője, amely segít a fehérjék előállításában a sejtekben. Az RNS hasonló a DNS-hez, viszont a DNS-sel

ellentétben egy szálú, és egy másik bázist, az uracilt (U) használja a timin helyett. Az RNS három fő típusa: mRNS (messenger RNA), amely a DNS információját viszi a riboszómákhoz; tRNS (transfer RNA), amely a fehérjeszintézis során az aminosavakat szállítja; és rRNS (riboszomális RNA), amely a riboszómák összetevője, ahol a fehérjeszintézis történik. [7]

## **1.5 Gén**

A gén a DNS egy specifikus szakasza, amely konkrét utasításokat tartalmaz egy vagy több fehérje előállítására. [8] Az öröklődés funkcionális, illetve fizikai alap eleme, amely információt hordoz generációkat átívelően. [9]

## **1.6 Nukleotid**

A nukleotidok a DNS (beleértve a géneket) építőkövei, amelyek a genetikai információk alapját képező kémiai anyagok. A DNS szerkezetét négyféle nukleotid alkotja: adenin (A), citozin (C), guanin (G), és timin (T). [1]

## **1.7 Fenotípus**

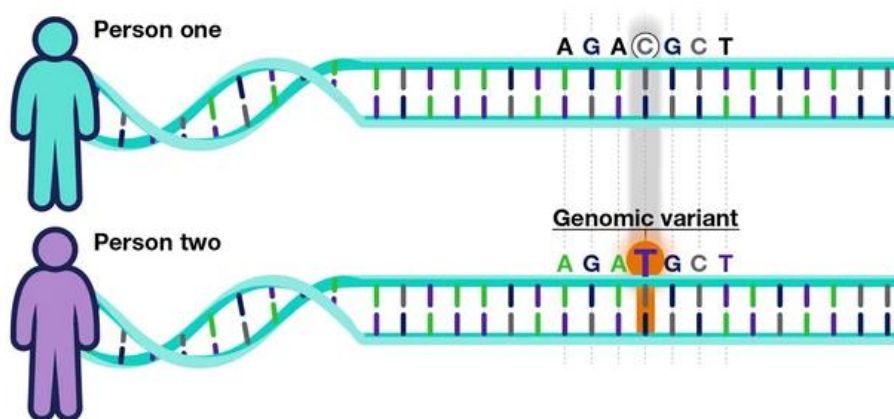
Az élőlények fenotípusát, azaz megfigyelhető jellemzőit, a gének, illetve a gének által kódolt fehérjék és a környezeti tényezők együttesen határozzák meg. [8]

## **1.8 Pleiotrópia**

Azt a jelenséget, amikor egy gén egyszerre több tulajdonságot befolyásol, pleiotrópiának nevezzük, a gének túlnyomó része pleiotróp. Ez azt jelenti, hogy egy gén változása (például egy mutáció) nem csak egyetlen tulajdonságot vagy funkciót befolyásolhat, hanem több különböző jellemzőt is érinthet az élőlényen belül. [10]

## 1.9 Genetikai variáns

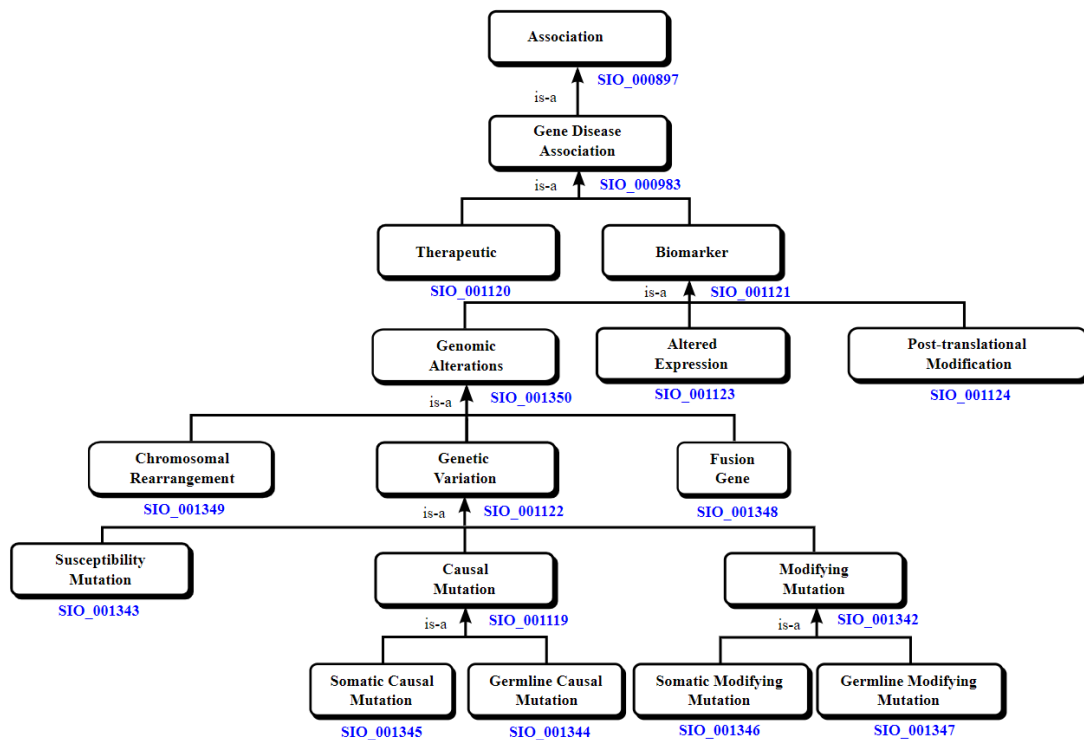
A genetikai variáns, egy DNS szekvenciájának bármilyen változása lehet. Ez a variáns egy adott gén vagy a genom más részein előforduló változás, amely eltér az adott populációban leggyakrabban előforduló szekvenciától. Ezek a variánsok felelősek lehetnek az egyének közötti genetikai különbségekért, és befolyásolhatják az egyének különböző betegségekre való hajlamát vagy rezisztenciáját. A változás természetesen lehet öröklődés eredménye. (Az öröklődés folyamatát a Kromoszómák fejezetben felvázoltam) Azonban lehet nem öröklődésből fakadó, például az adott személy életéből következő változás. Fontos megjegyezni, hogy ezek a változások viszont nem adhatóak tovább öröklés útján. [11] A 2. ábrán látható egy lehetséges variáció leegyszerűsített példája.



2. ábra Egy genetikai variáns leegyszerűsítve[11]

## 1.10 Gén-betegség kapcsolatok

A dolgozatom célkitűzése egy olyan osztályozó elkészítése, amely meghatározza egy gén és betegség közötti kapcsolat milyen ontológiai típusú. A 3. ábrán látható az általam használt ontológiai hierarchia, amelyet a DisGeNET adatbázis biztosított. [12]



3. ábra Asszociációk ontológiai típusa egy fagráfban [12]

A különböző típusok jelentését a következő felsorolásban lehet igazán megérteni.

1. **Therapeutic/Terápiás:** Ez azt jelenti, hogy egy bizonyos gén vagy fehérje segít kezelni vagy enyhíteni egy betegséget.
2. **Biomarker/Biológiai jelző:** Ez arra utal, hogy egy gén vagy fehérje segíthet felismerni egy betegséget, mert vagy részt vesz a betegség kialakulásában, vagy annak jelenléte jelezheti a betegséget.
3. **Genomic Alterations/Genomikus Változások:** Ez azt jelzi, hogy egy betegség kialakulásában vagy hatásában szerepet játszhatnak bizonyos változások a genom szerkezetében. A genomikus változás öse számos olyan asszociáció típusnak, amelyek fontosak a daganatkezelés szempontjából. Ugyanis manapság a daganatok kezeléseit például a vesedaganatos kezeléseket megelőzi egy molekuláris elemzés, amely genomiális, genetikai változásokat keres. [13]
4. **Genetic Variation/Genetikai Variáció:** A genetikai variációt úgy képzelhetjük el, mint bármilyen különbséget, amely két azonos gén vagy DNS-molekula másolata között létezik. A genetikai variáció

legegyszerűbb formája, amelyet egyetlen nukleotid helyen észlelhetünk. [9] Ez az ontológiai típus azt jelzi, hogy egy genetikai eltérés kapcsolatban áll egy betegséggel, de még nem biztos, hogy ez az eltérés okozza a betegséget.

5. **Causal Mutation/Okozati Mutáció:** Ez arra utal, hogy bizonyos genetikai változások közvetlenül okozzák a betegséget.
6. **Germline Causal Mutation/Csíravonal Okozati Mutáció:** Ez azt jelenti, hogy vannak olyan genetikai változások, amelyek a szülőktől öröklődnek és közvetlenül okozzák a betegséget. [14] Ez azt is jelenti, hogy a mutáció a test minden sejtjében jelen van és tovább örökítheti az ő utódjainak. [15]
7. **Somatic Causal Mutation/Somatikus Okozati Mutáció:** Ez arra utal, hogy bizonyos genetikai változások a személy élete során keletkeznek és okozzák a betegséget. Általában nem öröklődnek tovább, mert tipikusan nem fordulnak elő a csírasejtekben. [15]
8. **Chromosomal Rearrangement/Kromoszómális Átrendeződés:** A kromoszómák átrendeződése általános négy csoportba sorolhatóak: amikor egy kromoszómaszegmentum elveszik, amikor egy szegmentum megkettőződik, amikor egy szegmentum megfordul, amikor egy szegmentum áthelyeződik egy másik nem homológ kromoszómára. [9] Ez a kromoszóma módosulás, kapcsolódhat egy betegséghez, akár lehet közvetlenül a betegség okozója. [9]
9. **Fusion Gene/Fúziós Gén:** A szomatikus transzlokációk, azok a genetikai változások amikor a kromoszómák helyet cserélnek a genomban. [9] Ezek gyakoriak számos ráktípusban, és onkogén (rákot okozó gén) fúziós géneket hoznak létre, amelyek az emberi rákmorbiditás körülbelül 20%-ért felelhetnek. [16] Ez az ontológiai típus azt jelzi, hogy két különböző gén összeolvadása kapcsolatban áll egy betegséggel.
10. **Susceptibility Mutation/Hajlamsító Mutáció:** Ugyan a betegségekre való genetikai hajlamnak a kutatása már évtizedekkel ezelőtt kezdődött a

2000-es évekig csak korlátozott sikereket értek el. Azonban a közelmúltban számos DNS variánst találtak, amelyek növelik, illetve csökkentik a betegségekre való hajlamot. [16] Ez az onkológiai típus azt jelenti, hogy egy bizonyos genetikai variáns hajlamosít a betegség kialakulására, de önmagában nem feltétlen okozza azt.

11. **Modifying Mutation/Módosító Mutáció:** Ez arra utal, hogy egy genetikai változás befolyásolja, hogy a betegség milyen formában jelenik meg.
12. **Germline Modifying Mutation/Csíravonal Módosító Mutáció:** Ez egy a Csíravonal Okozati Mutációhoz hasonló örökölt változás, ami annyiban tér el tőle, hogy nem okozza, hanem befolyásolja a betegség megjelenését.
13. **Somatic Modifying Mutation/Somatikus Módosító Mutáció:** Ez a mutáció a Somatikus Okozati Mutációra hasonlít, csak ez a genetikai változás befolyásolja a betegség megjelenését, nem pedig a kialakulását.
14. **Altered Expression/Megváltozott expresszió:** A gén expresszió az a folyamat amikor egy gén DNS szekvenciája átíródik egy RNS-be. [9] Ez az onkológiai típus azt jelzi, hogy egy expresszió megváltozása kapcsolódik egy betegséghez.
15. **Post-translational Modification/Poszt-transzlációs Módosítás:** A fehérje transzlációját követően történő aminosav oldalláncainak módosulását jelenti a Poszt-transzlációs Módosítás. [9] Ez arra utal, hogy egy fehérje működésének megváltozása (például kémiai módosítások révén) kapcsolódik egy betegséghez.

## 2 Mi az a Gráf Neurális Háló?

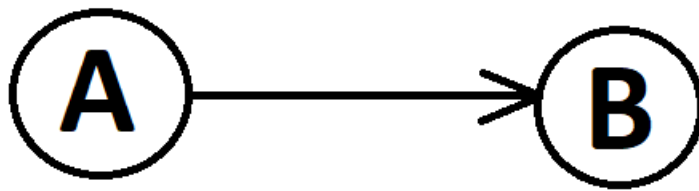
A gráf neurális hálók témakör egy kifejezetten új tudományág a gépi tanuláson és a mélytanuláson belül. Ezen fejezet igyekszik megismertetni az olvasót, a módszeres alkalmazásához szükséges alapvető ismeretekkel, kérdéseken keresztül. Szó lesz egyes algoritmusokról, melyek ismerete szükséges a dolgozatom megértéséhez.

### 2.1 Miért gráfok?

A hagyományos mélytanuló algoritmusaink, tipikusan számokat, szövegeket, listákat, mátrixokat kapnak bemeneti paraméterül és azokon az adatokon végeznek valamilyen bonyolult műveletet, amivel igyekeznek approximációt végrehajtani. Érthetőbben fogalmazva, egy mintából eloszlást készítenek, amely a populáció eloszlását (továbbiakban háttér eloszlás) igyekszik megközelíteni. Joggal felmerülhet a kérdés miért van szükség ezen a bemeneti típuson változtatni, illetve hogyan is lehet ezektől eltérő bemenetet, tehát egy gráfot kreálni egy számítógépben, hiszen Claude Shannon *The Use of Boolean Algebra in the Analysis and Design of Switching Circuits* című műve óta a Boolean algebra elveivel készített digitális eszközök tipikusan a nulla és egy számokat használják. Kezdjük talán az utóbbi kérdéssel.

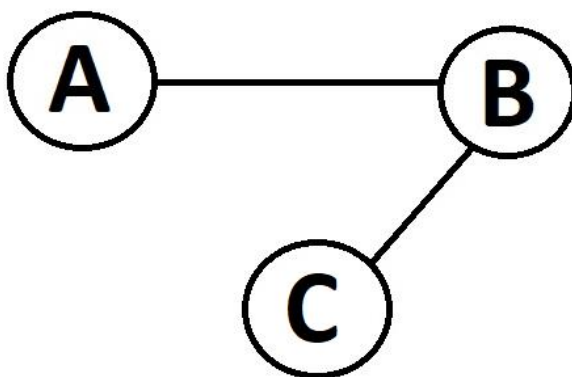
#### 2.1.1 Hogy jelenik meg a gráf a számítógépben?

A gráfok alapvetően 2 típusú elemekből állnak, csomópontokból és élekből. A csomópontokat tipikusan valamilyen entitáshalmaznak szokás elképzelni, míg az éleket az adott egyedek közötti kapcsolathalmaznak. Az entitások rendelkezhetnek különböző paraméterekkel, épp úgy, mint a kapcsolatok. Ezen entitások és élek párosítása által létrejött kapcsolatot nevezzük gráfnak. Fontos kiemelni, hogy egy él rendelkezhet iránnyal. Ahogy a 4. ábrán látható, az A csomópontból egy él vezet a B csomópontba, melybe egy nyíl mutat, míg ez visszafelé nem látható.



4. ábra Irányított kapcsolat

A csomópontok közötti kapcsolat a számítógép nyelvére való leképezésére egy alkalmas eszköz a szomszédsági mátrix elkészítése. Az 5. ábrán látható gráf szomszédsági mátrixának leképezése megmutatja, hogy mivel az A és a B csomópont között található él, ezért az A oszlop és a B sor, illetve a B oszlop és az A sor metszetében álló mezőkön lévő szám egy, míg az A oszlop és C sor, illetve C oszlop és A sor metszetében található mezőkön látható szám nulla, hiszen nem létezik él a két csomópont között. A szomszédsági mátrix lehetőséget ad arra is, hogy kalkulációinkat mátrix műveletekkel végezzük, ahogy azt majd a későbbi fejezetekben láthatjuk.



|   | A | B | C |
|---|---|---|---|
| A | 0 | 1 | 0 |
| B | 1 | 0 | 1 |
| C | 0 | 1 | 0 |

5. ábra Gráf és szomszédsági mátrixa

### 2.1.2 Miért van szükség a gráfokra?

A gráfok adta lehetőség egészen triviális. Így egyfajta kapcsolatot tudunk jelölni az adatpontjaink között és ezen kapcsolat paraméterekkel felvértelve egy kifinomultabb tanulási algoritmushoz vezet. Vegyünk egy egyszerű, talán picit kisarkított példát. Amikor egy 10 tagú csoportban azt szeretnénk meghatározni, hogy ki a legjobb barátja Annának, hagyományos esetben nem ismerjük Anna és a többi csoporttársa kapcsolatait. Következtethetünk a csoporttagok közös tulajdonságaiból, például a kedvenc zenéjükből,

kedvenc ételükből, mely tárgyakat utálják a legjobban. Ezek alapján jogosan gondolhatjuk, hogy ahol a legnagyobb az egyezés Anna és az adott csoporttag tulajdonságai között, akkor az a személy Anna legjobb barátja. De ha ismerjük a csoporttagok közötti kapcsolatokat, például: ki-kit ismer, ki-kinek a barátja, ki-kivel van rosszban. Egy alaposabb következtetésre juthatunk. A gráfok előnye végeredményében ez, egy kifinomultabb következtetés.

## **2.2 Hogyan tanulható a gráf?**

Fontos leszögezni minden egyes gráf neurális háló (továbbiakban GNN) tanítása esetén, hogy mit is szeretnénk elérni, ugyanis több dolgot is meghatározhatunk a segítségével. Rengeteg tanuló modell létezik, mégis mi a közös ezekben, mi a működésük alapja?

### **2.2.1 Mit akarunk meghatározni?**

Hagyományosan három típusa van a GNN modellek következtetésének: csomópont, él és gráf szintű.

Akkor beszélünk csomópont szintű következtetésről, amikor a kapott adat (gráf) alapján a modell egy nem létező csomópontot, vagy esetleg egy létező csomópont tulajdonságait, osztályát következteti ki.

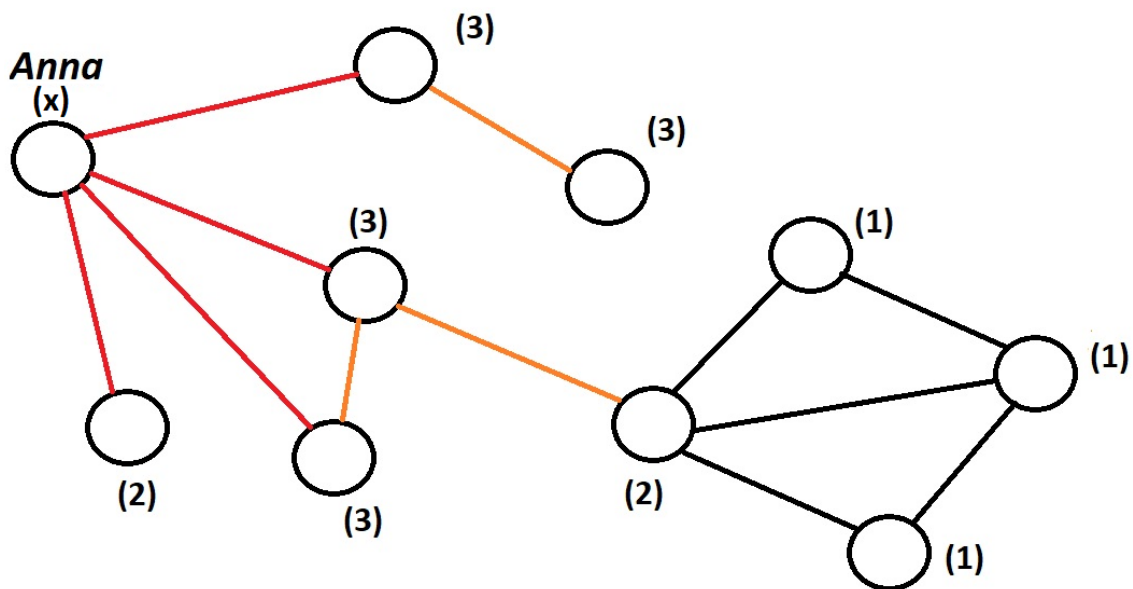
Él szintű következtetésről akkor beszélhetünk, ha a modell egy hiányzó élt következtet ki vagy esetleg egy él hiányzó tulajdonságát, osztályát határozza meg.

Akkor beszélhetünk gráf szintű következtetésről, amikor egy gráf vagy részgráf tulajdonságát, osztályát következteti ki a modell. Például egy molekula gráfja alapján próbálja meghatározni, hogy milyen színű az a molekula.

### **2.2.2 Mi a közös a modellekben?**

Amikor a modellek a bemeneten kapott gráfról szeretnének valamilyen szintű következtetést készíteni, felkell építeniük a csomópontokhoz tartozó beágyazást. Ez a beágyazás, egyfelől igyekszik egyfajta numerikus vektor reprezentációt adni a

kategorikus paramétereknek is, másfelől viszont egy kapcsolódási pontot jelent az adatok feldolgozása szempontjából. Nézzünk egy életről vett példát. Amikor belegondolunk abba, hogy milyen filmeket szeretünk, milyen zenét hallgatunk, mi a kedvenc hobbink vagy akár, mi a családi állapotunk, felfedezhetjük azt az érdekességet, hogy a legközelebbi barátaink ugyanezen tulajdonságaik hasonlóak a miénkhöz. Ennek mi köze van a beágyazott vektorokhoz? Képzeljük el, hogy a 10 tagú csoportba mindenkinek választania kell egy kedvenc színt a piros-zöld-kék lehetőségek közül. Ebből a 10 emberből 9-ét ismerjük is, kivéve Annáét. Vegyünk fel minden ismert csomóponthoz (személyhez) egy-egy hosszú beágyazott vektort. Jelöljük 1-gyel a pirost, 2-vel a zöldet és 3-mal a kéket. Ahogy ezt a képen láthatjuk is, Annának egy  $x$  található a vektorjában. A feladatunk pedig egy gráf neurális háló segítségével Anna  $X$  paraméterének a megjósolása. Képzeljük el egy olyan algoritmust, ami az összes beágyazott érték átlagát venné, ez azt jelentené, hogy Anna kedvenc színe a 2,1 tehát a zöld, holott az 6. ábrán világosan látszik, hogy Annának a barátjának, a barátjának a barátai húzták le az átlagot. Jogos következtetés lenne azt mondani, hogy hibás a modellünk, vagyis inkább az aggregációs függvényünk, ami összegyűjti a szomszédos csomópontok információit. Hiszen valószínű ilyen távoli baráti kapcsolatoknak nem szabadna nagy hatással lenniük Anna kedvenc színére.



6. ábra A 10 tagú csoport barátság gráfja

Vegyünk egy kifinomultabb algoritmust, amely először veszi Anna közvetlen barátainak (piros éllel összekötött) vektorjainak az átlagát.

$$2,75 = \frac{3 + 3 + 3 + 2}{4}$$

Ezekután veszi a közvetlen barátok közvetlen barátait, de csak azokat, akik nem Anna vagy Annának közvetlen barátai és csomópontonként veszi ezek átlagát.

$$3 = \frac{3 + 3}{2} \quad 2,5 = \frac{3 + 2}{2}$$

Ezután veszi ezek végső nagy átlagát, és ez meghatározza Anna kedvenc színét.

$$2,75 = \frac{2,75 + 3 + 2,5}{3}$$

Tehát Anna kedvenc színe a kék. Persze ez nem egy univerzálisan mindig működő implementációja a beágyazott vektorokkal történő kalkulációnak. De egy szemléletes bemutatása annak, hogy nem mindegy milyen aggregációs függvényt választunk a modellünkhöz.

### 2.2.3 Mitől tanul a modell?

A számos tanuló modellek különbözőségét a felépítésük adja. A modellek hagyományosan rétegekből épülnek fel. Ezek a rétegek azok, amelyek valamilyen matematikai egyenletet felhasználva képesek előállítani egy eredményt/jóslatot és ezt az eredményt vagy tovább adják egy másik rétegnek vagy esetlegesen eredményül adják vissza, válaszul egy bemenetre.

Az egyik legtipikusabb réteg a lineáris következtetést alkalmazza. A réteg matematikai felépítését az 7. ábrán lehet látni. A nevéből adódóan a bemeneti adatokat transzformálja lineárisan.

$$y = xA^T + b$$

#### 7. ábra Lineáris transzformáció

Ha szeretnénk pontosabban megérteni, hogyan is működik a lineáris transzformáció, kénytelenek vagyunk megismerkedni részletesebben ennek a matematikai képletével. A képen látható  $x$  a bemeneti jellemző vektor, amely egy  $A$  mátrixszal van súlyozva. Ehhez a szorzathoz hozzá van adva egy  $b$  eltolás vektor. Az  $A$

mátrix és a  $b$  eltolás vektor adja ennek a rétegnek a lényegét, azt a 2 komponenst, amelyet a modellünk tanulás útján módosíthat. Az összeadás eredménye, a kapott  $y$  pedig a kimeneti vektor, ez lesz a következtetésünk. [17]

A GNN-ek világában elterjedtek a konvolúciós rétegek. Ezek tipikusan egy komplex művelettel súlyozzák az adatok közötti kapcsolatokat. [18]

Egy nagyon elterjedt konvolúciós réteg a SageConv, amelynek számos előnye van a hasonló GraphSage architektúrákhoz képest, például az erősebb kifejező képességgel bíró konvolúció használata, ugyanis a SageConv számítása közben súlyozza a szomszédos csúcsokat azok fokszámával. Ha precízebben akarok fogalmazni, a SageConv a szomszédos csomópontok fokszámával normalizált átlagát használja aggregációként, amivel pontosabb következtetések vonhatóak le. [19]

Vizsgáljuk meg a következő 8. ábrán látható képletet. Az  $x_i$  az adott csúcspontoz tartozó beágyazott vektor, ez súlyozásra kerül egy  $W_1$  súlymátrixal, ez egy lineáris transzformáció melynek szerepe, hogy közvetlenül a csomópontoz tartozó információ is tanulással súlyozható legyen. Az érdekesebb rész az összeadás második fele, ahol az adott csúcspont összes szomszédjának a beágyazott vektorának az átlaga kerül súlyozásra. Így egyfajta lokalitása lesz a csomópontnak, ezalatt azt értem, hogy hatnak rá a szomszédjai. Fontos azonban megjegyezni, hogy eltérő a két súlyozás, ez adja ennek a rétegnek az erejét. Vannak ugyanis olyan gráfok, ahol a szomszédoknak kevesebb beleszólásuk lehet az adott csomópont jellemzőibe, azonban lehetséges, hogy az adott csomópont kevés identitással rendelkezik és a szomszédai elnyomják a saját jellemzőit egy erős  $W_2$  és gyenge  $W_1$  esetén. [20]

$$\mathbf{x}'_i = \mathbf{W}_1 \mathbf{x}_i + \mathbf{W}_2 \cdot \text{mean}_{j \in \mathcal{N}(i)} \mathbf{x}_j$$

8. ábra SageConv mögötti képlet

A SageConv általi aggregáció robusztusságának egyik nagy előnye, hogy a zajos és hiányos adatkészleteken is képesek vagyunk nagy pontosságú következtetéseket végezni. [20]

Tehát felhasználva azt az apriori tudást, hogy ismerjük az adatok közötti összefüggést, egy aggregáció segítségével az előnyünkre válthatjuk.

## 3 Hierarchikus osztályozás

A hierarchikus osztályozás egy kevésbé elterjedt módja az osztályozási problémák megoldásának. A dolgozat ezen része ennek az osztályozási típusnak a megismertetésére fókuszál. A továbbiakban szó lesz arról mi ezen módszer eltérése a hagyományos megközelítésektől, illetve arról mennyi különböző szabályrendszer-módszer áll rendelkezésünkre és milyen elgondolások mentén lett a dolgozatban alkalmazott kiválasztva.

### 3.1 Mi az a hierarchikus osztályozás?

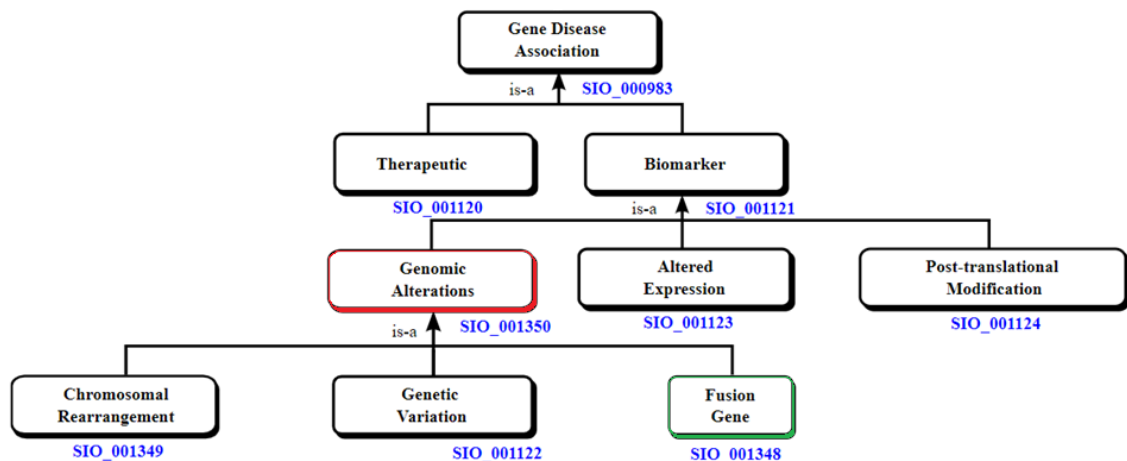
A mindennapokban közel sem mondható a leghasználtabb osztályozási típusnak a hierarchikus osztályozás. Sokkal gyakrabban találkozhatunk a bináris, többosztályos vagy többcímkes osztályozással. A bináris osztályozás esetében két osztály közül választja ki modellünk melyik az adatunkra a legjellemzőbb. Többosztályos esetben viszont számos osztály közül van lehetősége egy osztályt választania, míg többcímkes esetben egy adatra több osztályt is választhat. [21]

Ezek a hagyományos osztályozók, egyenrangú félként kezelik az osztályokat, akár úgy is elképzelhetjük, hogy a választandó osztályok egymástól függetlenül egymás mellett helyezkednek el és a modellünk ezen osztályok közül fog választani egyet, vagy többet.

A hierarchikus esetben a modellek korán sem függetlenek egymástól. Egy hierarchikus rendszerben helyezkednek el egymáshoz képest. Így amikor modellünk kiválaszt egy osztályt a hierarchiából, azzal nem csak azt mondta meg, hogy pontosan melyik osztályba tartozik az adatunk, hanem azt is, hogy az osztályon felül helyezkedő szülő osztályok melyikébe tartozik, illetve, hogy a továbbiakban a gyermekosztályoknak választott adatok is bizony a hierarchiát megtestesítő gráf aktuális szintjén, egy osztályba tartoznak.

A 9. ábrára tekintve láthatjuk, hogy osztályozónk az első osztályozás alkalmával megállapította, hogy az adott gén és betegség között egy Genomic Alteration kapcsolat áll fenn (Ez azt jelzi, hogy egy betegség kialakulásában szerepet játszhatnak bizonyos változások a genom szerkezetében.). Amikor az osztályozó ezt állapította meg, akkor azt

is megmondta, hogy ezen kapcsolat Biomarker is (Ez arra utal, hogy egy gén vagy fehérje segíthet felismerni egy betegséget, mert vagy részt vesz a betegség kialakulásában, vagy annak jelenléte jelezheti a betegséget.). A következő kapcsolatnál, tehát egy másik adat esetében, a Fusion Gene (Ez azt jelzi, hogy két különböző gén összeolvadása kapcsolatban áll egy betegséggel.) lett meghatározva, ami azt is determinálja, hogy ez nem csupán egy génfúzió, hanem egy Genomikus módosulás is.



9. ábra Genomic Alterations és Fusion Gene osztályozása

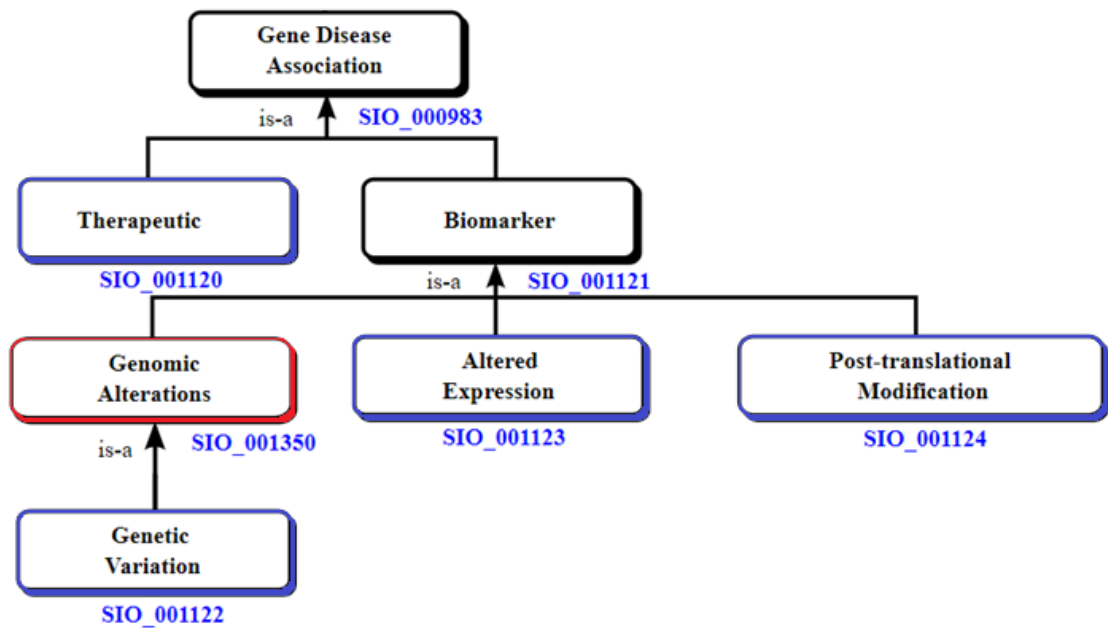
Ha leegyszerűsítve tekintünk erre az osztályozóra akkor azt is mondhatnánk, hogy semmivel sem másabb, mint egy hagyományos többcímkes osztályozó, hiszen amikor egy gyermek csomópontot választ a modellünk akkor automatikus a hierarchián felfelé haladva kiválasztja a szülő hierarchia osztályt is. Ez merőben eltér a hierarchikus osztályozás módszerétől, mert míg hierarchikus esetben a hierarchián fentről lefelé haladva osztályoz a modellünk a megfelelő osztály eléréséig, ahogy azt majd a továbbiakban bemutatom. A többcímkes esetben az egymástól függetlennek tekintett osztályok közül választ egyet és onnan kiindulva annak őseit is kiválasztja. Ezenfelül az első esetben, amikor azt jósolta a modellünk, hogy a kapcsolat Genomikus módosulás akkor azzal azt is determinálta, hogy a kapcsolat jelenlegi tudásunk szerint nem osztályozható ennél specifikusabban, azonban a Genomikus módosulás bármely gyermeke lehet, a többcímkes osztályozás esetében ilyen következtetéseket nem tehetünk meg. [21]

## 3.2 Milyen működési elvei vannak egy hierarchikus osztályozónak?

Számos módszertana-szabályrendszere létezik a hierarchikus osztályozásnak például: Inclusive, Exclusive, Less Exclusive, Less Inclusive, Siblings, Exclusive Siblings Policy. [21] Ezek önmagukban egy teljes dolgozatot érdemelnének, ezért a továbbiakban csupán három szabályrendszerrel ismerkedünk meg, végig gondolva lényegi különbséget közöttük és a belőlük származó előnyt.

### 3.2.1 Exclusive Policy

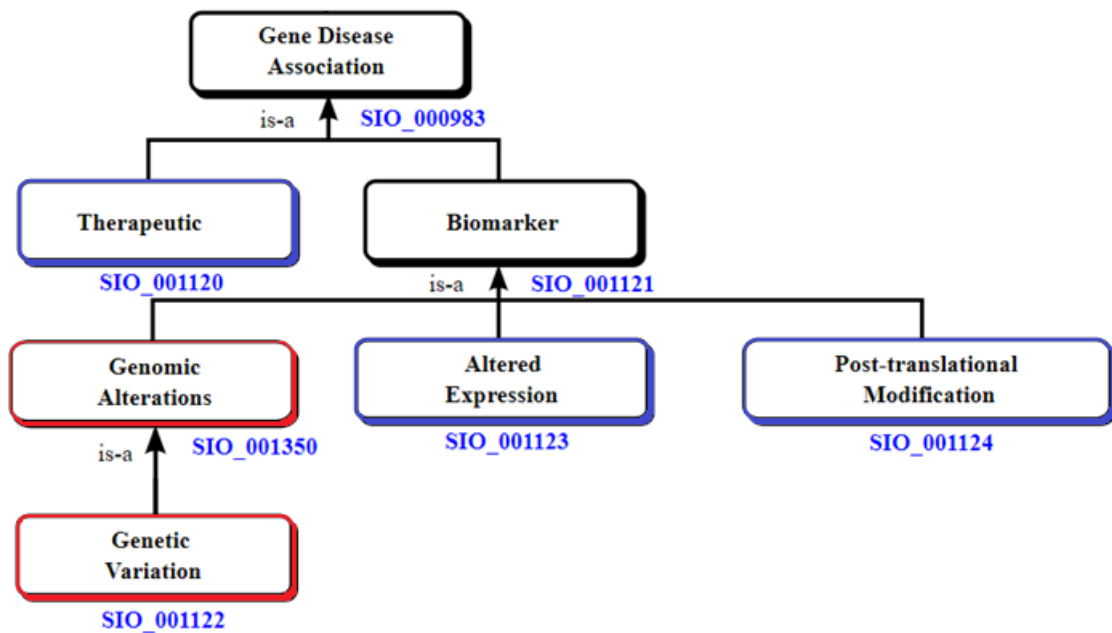
A kizáró szabályrendszer egy szigorú módszer arra, hogyan lehet az osztályozót betanítani. Amikor felosztjuk a tanító adatokat pozitív és negatív példákra, a szabályrendszer azt mondja ki, hogy szigorúan csak azokat a példákat vehetjük pozitív eseteknek, ahol az osztály megegyezik a célváltozóval. [21] Tehát ha a 10. ábrára tekintünk és a tanító adatunk célváltozója Genomic Alteration, akkor szigorúan csak a Genomic Alteration célváltozóval rendelkező adatokat tekinthetjük pozitív eseteknek, kizárva ezzel minden más osztályt, még a gyermekosztályokat is. Amelyek így negatív esetként használatosak, beleértve minden szomszédos osztályt és minden olyan osztályt, amely egy másik gyermeke, illetve annak a gyermeknek az ágán található, a választott csomóponttól a hierarchián felfelé haladva. Az ábrából kiindulva negatív esetek például ezek a célváltozós adatok: Therapeutic, Altered Expression és Genetic Variation. A szülő osztályok semlegesen kimaradnak az osztályozásból. Véleményem szerint ez a szabályrendszer így ellentmondásos, ha a hierarchiánkra úgy gondolunk, mint egy olyan kapcsolatrendszerre, ahol a gyermekosztály egy specifikáltabb szülő osztály. Ezen felül limitálhatja az osztályozónk teljesítményét egy olyan adathalmaz esetén, ahol a célváltozók többsége szülőosztály és nem gyermekosztály.



10. ábra Exclusive Policy esetében csak a Genomic Alterations a pozitív eset

### 3.2.2 Inclusive Policy

A kizáró szabályrendszerrel ellentétben itt egy nagyon megengedő módszerrel ismerkedhetünk meg. Tekintsünk a 11. ábrára, amikor a Genomic Alterations célváltozójú osztályhoz keresünk pozitív eseteket, akkor minden gyermekosztályát is pozitív példának vehetünk. [21] Ha belegondolunk ez egy sokkal megengedőbb szabályrendszer és talán jobban is illik a hierarchikus felfogásunkhoz. Ugyanis amikor azt mondjuk egy adatra, hogy a célváltozója Genetic Variation, akkor implicit azt is kijelentjük, hogy ő bizony egy Genomic Alteration is. Ezért amikor pozitív példákat akarunk gyűjteni, fontos figyelembe vennünk a gyermekosztályokat is. Negatív esetek például a Therapeutic és az Altered Expression célváltozójú adatok. A szülő osztályok itt is semlegesen kimaradnak az osztályozásból.

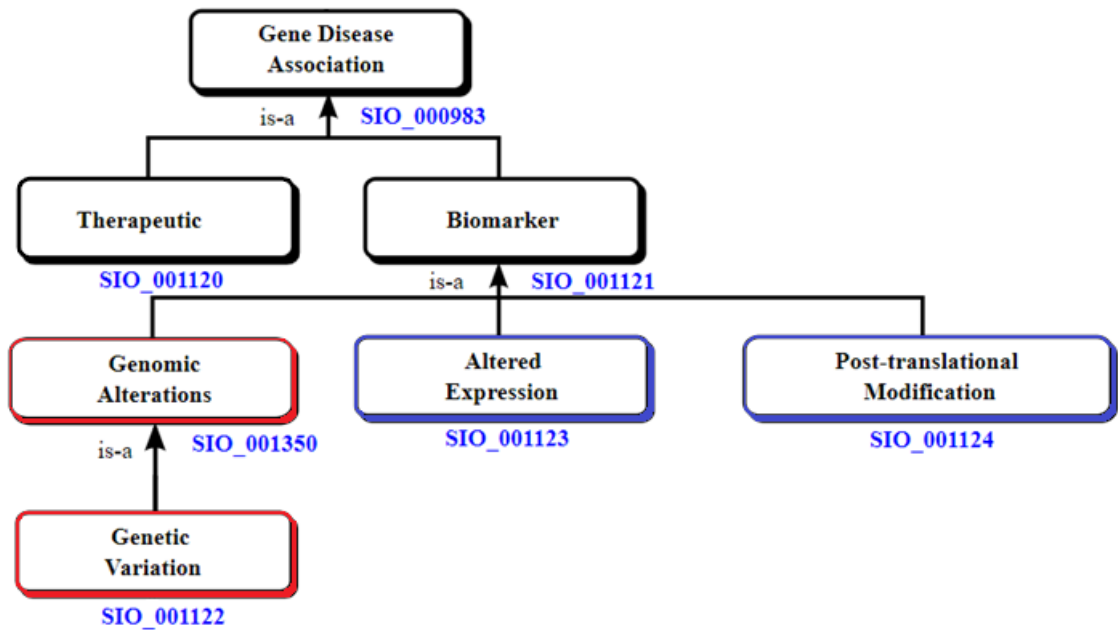


11. ábra Inclusive Policy esetében a Genomic Alterations és leszármazottai a pozitív eset

Ha tovább gondoljuk az osztályozónk működését, nem helyes azt ugyanolyan súlyú hibának tekinteni, ha az osztályozó az adott célosztály helyett annak egy gyerekosztályát jelöli meg, mint ha a hierarchia egy teljesen másik ágáról választana egy osztályt. Logikus, hogy egy közeli (hierarchiában szülő-gyermek kapcsolatban lévő) osztály tévesztése, vagy akár egy nagyobb távolságra lévő mégis egyazon hierarchia ágon található választása, kisebb hibának számít, mint egy teljesen távoli- más ágon található osztály megjelölése.

### 3.2.3 Siblings Policy

Ez a harmadik szabályrendszer nagyon hasonlít az Inclusive Policy-ra, az egyetlen különbség az, hogy nem tartja negatív esetnek a célváltozó ősein felfelé lépkedve azok másik ágain található gyermekeit és azok leszármazottait. [21] Természetesen kivéve a célváltozó testvér osztályait, ahogy azt a 12. ábrán látni lehet.



12. ábra Sibling Policy esetében csak a testvér osztályok a negatív esetek

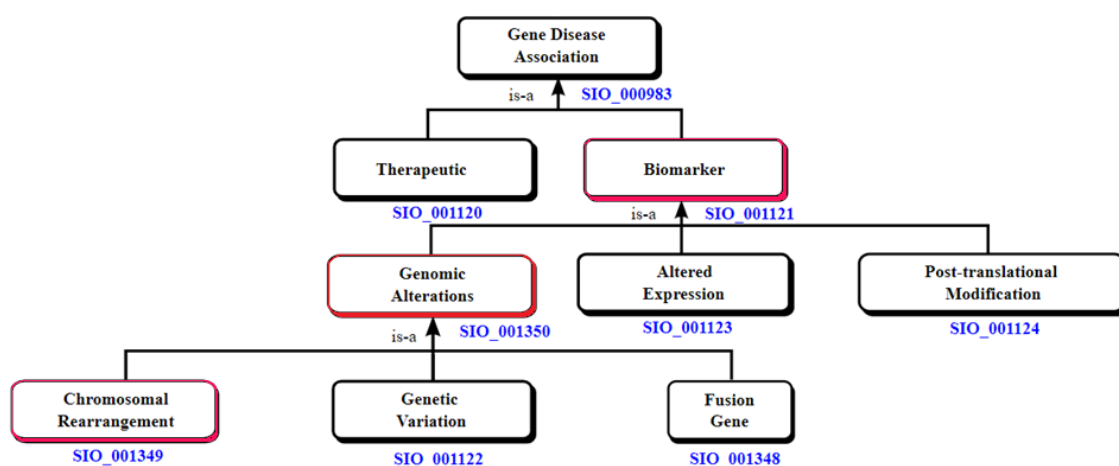
### 3.3 Hogyan alkalmazható egy hierarchikus osztályozó?

Alkalmazhatóság tekintetében is számos módszertan létezik a hierarchikus osztályozás terén. Ez a fejezet a dolgozat sorhasznált *szülőcsomópontonként történő lokális osztályozó* módszer bemutatásával foglalkozik (továbbiakban LCPN = Local classifier per parent node).

Ennek a módszernek a lényege, hogy részekre bontjuk az osztályozási problémát. A megközelítés fentről lefelé történik, tehát elindulunk a gyökér csomópontból, amely a hierarchia teteje és egy osztályozással az osztályozónk eldönti, a hierarchia melyik ágán haladjon tovább. Ez az osztályozás természetesen lehet bináris vagy akár többsztályos. A lényege az, hogy a kiválasztott osztály innentől kezdve az adott adat célváltozója. Ezt az osztályozási folyamatot végig ismétljük a hierarchián lefelé haladva egészen addig amíg egy levél csomóponthoz nem értünk. Ekkor elértük a lehető legspecifikusabb osztályt, amit az osztályozónk adhat. [21] Itt azonban érdemes átgondolni, mit is nyertünk és veszítettünk ezzel a módszerrel. Mindenféleképpen nyertünk azzal teljesítményben, hogy a fának (hierarchiának) csak egy szeletét kell vizsgálnunk, viszont, ha osztályozónk minden alkalommal elmenne a levélig, nem lennének köztes osztályaink, illetve feleslegesen túl specifikus osztályokba sorolnánk olyan adatokat, amelyek lehet, hogy

sekélyes magasságban szerepelnek a hierarchiában. Ezért ez a módszertan javasolja a küszöbértékek bevezetését. [21]

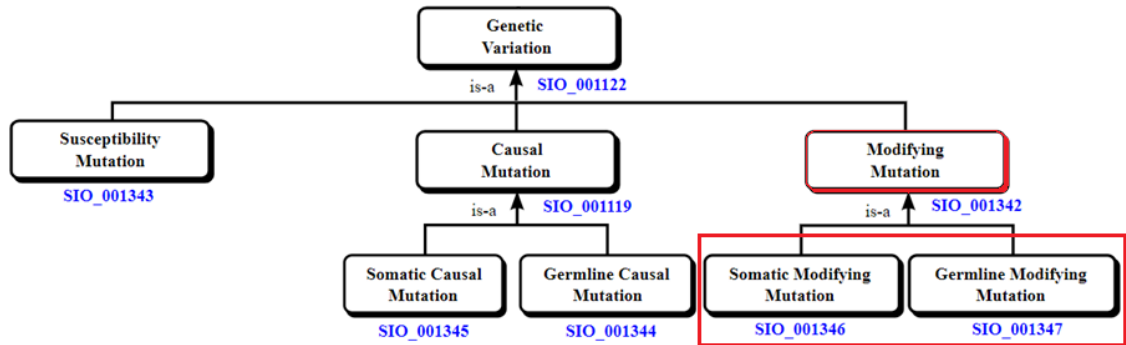
Tekintetünket a 13. ábrára szegezve azt láthatjuk, hogy a modellünk először a Biomarker, majd a Genomic Alterations és végül a Chromosomal Rearrangement mellett döntött. Viszont mi van, ha a valóságban az adat, ahol idáig jutottunk csupán egy Genomic Alteration? Erre megoldásként használhatunk osztályokhoz rendelt küszöbértékeket. Ezek a küszöbértékek segítenek megakadályozni a modellünket abban, hogy a modellünk túl specifikus osztályokat határozzon meg. A működését a következőképpen kell elképzelni. A modell minden szinten eldönti a lehetséges osztályok közül melyik illik a legjobban a vizsgált adatra. A kiválasztott osztály valószínűségét a hozzátartozó küszöbértékhez méri. Ha az osztály konfidencia szintje, tehát a valószínűsége alacsonyabb lesz a meghatározott küszöbértéknél, akkor az osztályozás megáll, és az aktuális osztály (az éppen vizsgált osztály őse) lesz az adat célváltozója. [21]



13. ábra Oszályozás a levélig

Bár Carlos N. Silla Jr. és Alex A. Freitas cikkükben kiemelik, hogy ez a módszertan kifejezetten a Siblings és Exclusive Siblings szabályrendszert támogatja, véleményem szerint itt egy ellentmondás fedezhető fel. Ugyanis amikor a modell egy szinten döntés elé kerül, csak az adott szinten található osztályok közül válogat, tehát úgy is elképzelhetjük, hogy nem ismeri a másik ágon található osztályokat. Ebben az esetben a Siblings egyfajta egyszerűsítő megnevezése lehet az Inclusive szabályrendszernek, de lényegét tekintve egyáltalán nem különbözik tőle.

Tehát a 14. ábrán láthatjuk, amikor a modellünk azt próbálja eldönteni, hogy a Modifying Mutation melyik gyermeke lesz esetlegesen az adat célváltozója, akkor tanítóadatként csak a hierarchia azon ágában található célváltozóval rendelkező adatok állnak rendelkezésre (és természetesen azok gyermekei, ha vannak). Így a Siblings és az Inclusive Policy lényegében megegyeznek ebben a módszertanban.



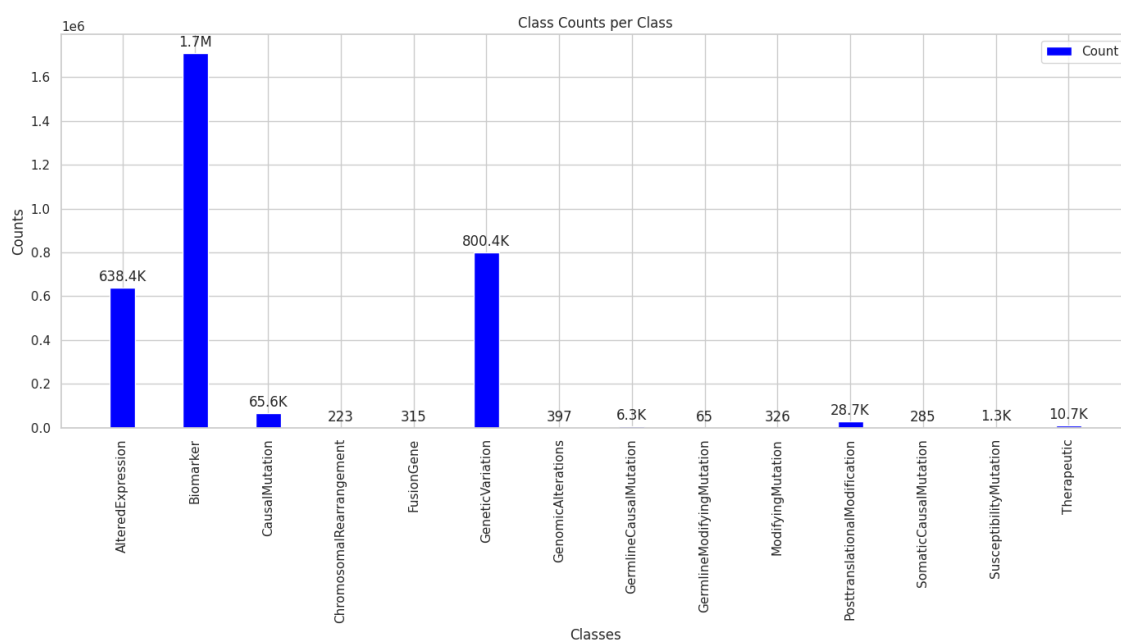
14. ábra LCPN esetében a Sibling és az Inclusive Policy megegyezik

## 4 Adatbázis

A dolgozatomban felhasznált adatokat a DisGeNET adatbázisból szereztem. Ez az adatbázis a világ egyik legnagyobb integrált adatbázisa, amely gén, gén variáns és betegség, illetve ezek közötti kapcsolatokat tartalmaz. A 7-es verziója 1134942 gén-betegség asszociációt tartalmaz, amelyek 21671 gén és 30170 betegség, rendellenesség, tulajdonság és klinikai vagy abnormális emberi fenotípus között oszlanak meg.[22]

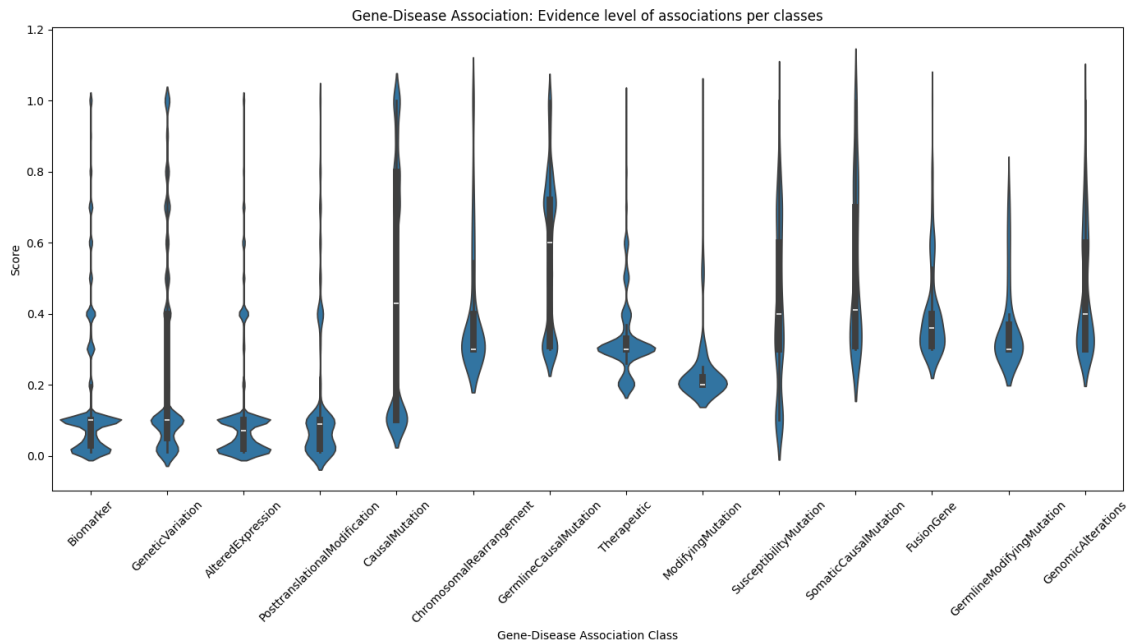
5 adatbázis táblát használtam a dolgozatom során. Az 1. adatbázis tábla a diseaseAttributes tartalmazta a betegségeknek különböző adatait a betegség nevét, típusát. A típus igazából azt jelöli, hogy az adott betegség egy betegség csoport vagy egy konkrét betegség vagy annak egy fenotípusa. A második tábla pedig a geneAttributes volt, amely tartalmazta a géneknek különböző jellemzőit. Ezeknek a tábláknak az összekötésére rendelkezésemre állt talán az egész dolgozat legfontosabb táblája a geneDiseaseNetwork nevű tábla, amely tartalmazta mind a 2 másik táblának az idegen kulcsait, ezzel egyfajta kapcsolatot teremtve a betegségek és a gének között. Illetve a kapcsolatnak egyéb jellemzőit, például az asszociáció típusát, amelyet célváltozóként a modellem igyekezett meghatározni. A géneknek és betegségeknek az asszociációjuknak 15 különböző ontológiai típusa található a DisGeNET adatbázisban. [12]

A 15. ábrán lehet látni, hogy az egyes asszociációs típusokból hány darab található a kapcsolatot jelentő geneDiseaseNetwork táblában. Azt mondhatjuk, hogy elég szélsőséges az adatok eloszlása. Vannak olyan kapcsolat típusok, mint például a Somatic Modifying Mutation, amely meg sem jelenik az adatbázisban. Ez a fajta diverz eloszlás kellően megnehezítette a modell megalkotását.



**15. ábra Gén betegség ontológiai típusok eloszlása**

De a geneDiseaseNetwork nem csak az asszociáció típusát tartalmazza attribútumként, hanem például azt is, hogy egy asszociáció mennyire megalapozott. Ugyanis ez az adatbázis nem csak az evidencia szintjén ismert, bebizonyított kapcsolatokat tartalmazza, hanem az olyanokat is, amelyek csak feltételezéseként léteznek. A DisGeNET egy saját pontozási rendszere alapján egy pontszámot meg is állapít minden egyes asszociációhoz. [23] Minél magasabb egy kapcsolat pontszáma annál biztosabb annak a jelenléte. A 16. ábrán látható a pontszámok ontológiai típusokként való eloszlása egy hegedű ábrán.



**16. ábra Az ontológiai típusok kapcsolatainak bizonyosságának eloszlása**

Érdekes megfigyelés lehet, hogy azok az ontológiai típusok, amelyekből sok adat áll rendelkezésünkre tipikusan alacsony mediánnal rendelkeznek és a hegedű is az alacsonyabb pontszámoknál vastagodik. A mediánt a fehér négyzet jelöli, a hegedű vastagsága pedig az adatok gyakoriságát jelentik. A fehér négyzet körüli vastag fekete sáv az interkvartilis tartományt (IQR) mutatja, ami az értékek középső 50%-ának eloszlását jelzi. Ezt vizsgálva arra a következtetésre juthatunk, hogy a CausalMutation típusnál meglehetősen széles tartományban oszlanak el a pontszámok. Illetve a mediánja is magasabban helyezkedik el a többi frekvenciált osztálynál. A CausalMutation leszármazottjai a GermlineCausalMutation és a SomaticCausalMutation, az ő mediánjaik és IQR-jük is magasabban helyezkedik. Általánosan ez arra enged következtetni, hogy azok a betegség gén kapcsolatok, ahol feltehetően a gén szerepet játszik a betegség kialakulásában azt magasabb biztonsággal jelenthetjük ki, mint a többi kapcsolatot.

A geneDiseaseNetwork tábla tartalmaz még más attribútumokat például az Evidence Level, Evidence Index, pmid, source, sentence és year. Kezdjük az elejéről, az Evidence Level egy ClinGen nevű kutatás által meghatározott mutató, amely egy gén és egy betegség közötti kapcsolat bizonyítékainak erősségét jelzi. [24] Az Evidence Index ezzel ellenkezőleg egy DisGeNET által kreált mutató, aminek a képletét a 17. ábrán lehet látni.

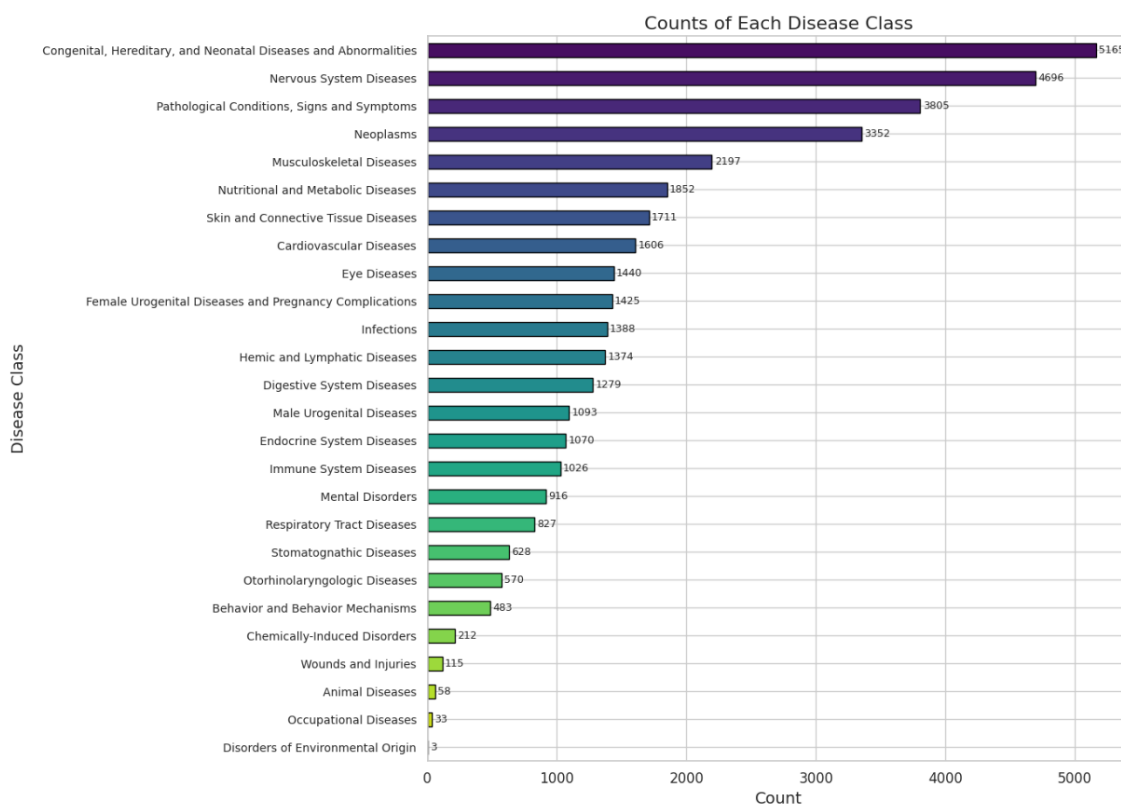
$$EI = \frac{N_{pubs_+}}{N_{pubs_{total}}}$$

17. ábra Evidence Index képlete [23]

Az  $N_{pubs_+}$  változó jelöli azon publikációk számát, amelyek támogatják, hogy létezik ez az asszociáció. Az  $N_{pubstotal}$  pedig az összes publikáció számát jelöli, amely erről az asszociációról szól. A pmid egy olyan azonosító, amivel rá lehet keresni az asszociációra és meglehet találni az összes hozzá kapcsolódó publikációt. A sentence egy általános leíró mondat, amely bemutatja az asszociációt. A year pedig az az év amikor az első publikáció keletkezett erről az asszociációról. [24]

A betegségek besorolhatóak betegség osztályokba, ezeknek az osztályoknak az adatait a diseaseClass nevű tábla tartalmazta. Amelyet egy disease2Class nevű kapcsolótábla köt a diseaseAttributes táblához. A diseaseClass tábla tartalmazza a betegség osztályok nevét, kódját és azt a szójegyzéket, ahol az osztályokat megnevezték. [24] Mivel a szójegyzék mindenhol ugyanaz volt, illetve a kód és a név is egyedi ezért az adatok előkészítésekor megváltam a szójegyzék és a kód oszloptól.

A 18. ábrára tekintve látható a betegség osztályok eloszlása.



**18. ábra Betegség osztályok eloszlása**

A felső 4 betegség típus előfordulása nagyon magas a többi osztályhoz képest. A Congenital, Hereditary, and Neonatal Diseases and Abnormalities (veleszületett örökletes betegsége, születési rendellenességek) osztály előfordulása a legmagasabb. Ilyen betegségek lehetnek akár az akut felső légúti fertőzések vagy bőr ciszták.

A legritkábban előforduló betegség osztály az adatbázisban a Disorders of Environmental Origin (környezetből adódó rendellenességek például a Sugárzás okozta rendellenesség).

Mivel magukból a betegség osztályokból is van szélsőségesen keveset előforduló és az asszociáció típusok eloszlása is végképp egyenletlen eloszlású, érdekes lehet megtekinteni a 19. ábrát.



19. ábra A betegség osztályokhoz tartozó betegségek kapcsolatainak hőtérképe

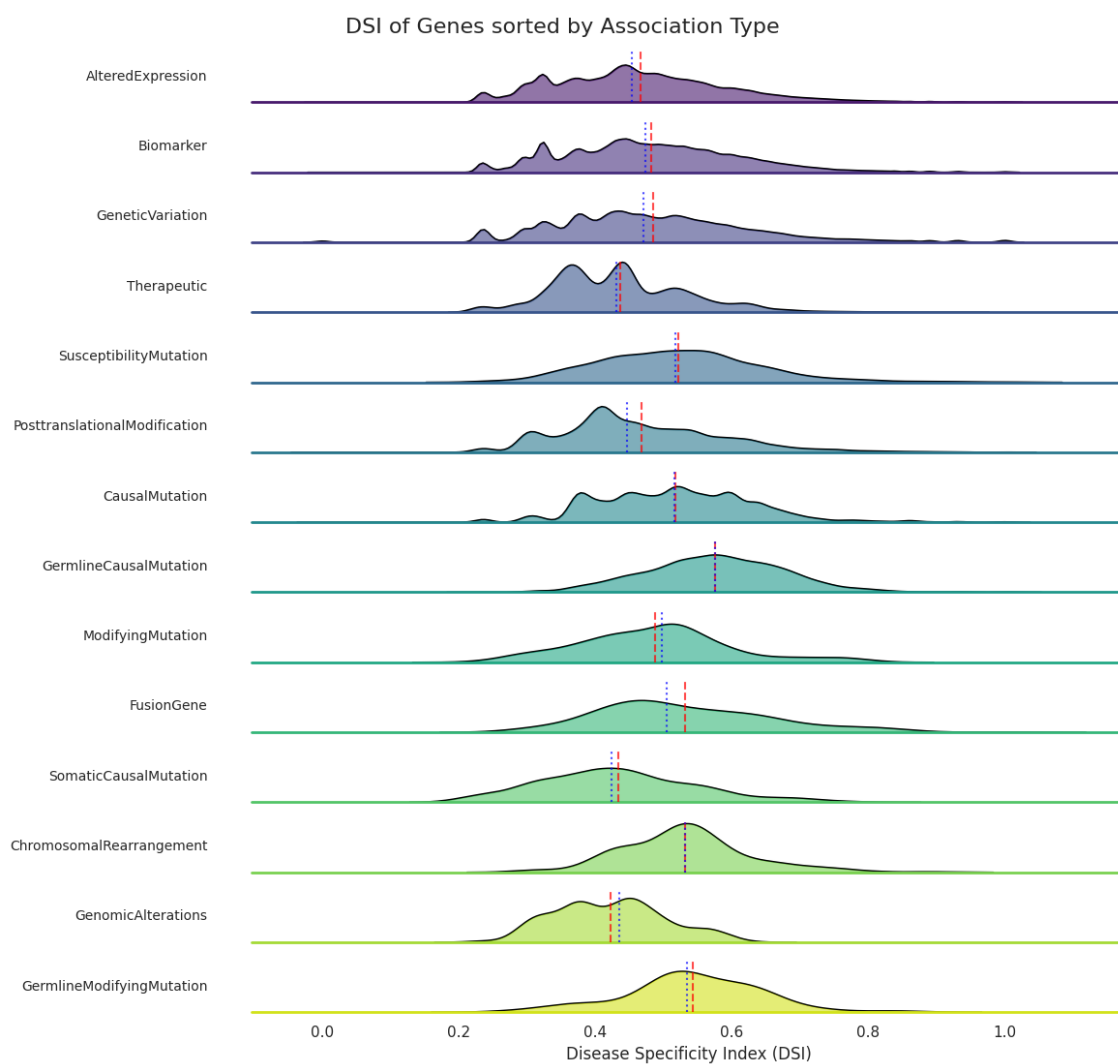
Az általánosságban erről az ábráról is elmondható, hogy azok az asszociáció típusok, amelyek gyakran előfordulnak az adatbázisban, azoknak meglehetősen egyenletesen sok különböző betegség osztályba tartozó betegséggel van kapcsolatuk. Természetesen azok a betegség osztályok, amelyek ritkán fordulnak elő, mint például a sugárzás okozta rendellenesség vagy az állati eredetű betegségek általánosan kevés onkológiai típussal vannak kapcsolatban. A Poszt-transzlációs módosításoknál felfigyelhetünk arra, hogy a neoplazma betegség osztályú betegségből kiugróan sokkal van kapcsolatban. Ilyen betegségek például az akut leukémia, endometrioid daganat.

A geneAttributes táblában található számos metrika, amellyel jellemezni lehet a gének sokszínűségét. Az egyik ilyen a Disease Specificity Index (A gén specifikus betegségekkel való kapcsolatát méri, tehát mennyire sok betegséggel lehet kapcsolatba hozni a gént, továbbiakban DSI). Ennek az indexnek a kiszámítására a következő 20. ábrán látható képlet áll rendelkezésünkre.

$$DSI = \frac{\log_2(\frac{N_d}{N_T})}{\log_2(\frac{1}{N_T})}$$

20. ábra DSI képlete [23]

Itt az  $N_d$  a betegségek száma, amely kapcsolatban áll a génnel, az  $N_T$  az összes betegség száma. [23] Tehát azok a gének, amelyek DSI értéke magasabb azok speciálisabb gének és kevesebb betegséggel hozhatóak kapcsolatba. A gének nem rendelkeznek osztállyal az adatbázisunkban, azonban csoportosíthatjuk őket a kapcsolataik típusa alapján ahogy az a 21. ábrán látszódik.

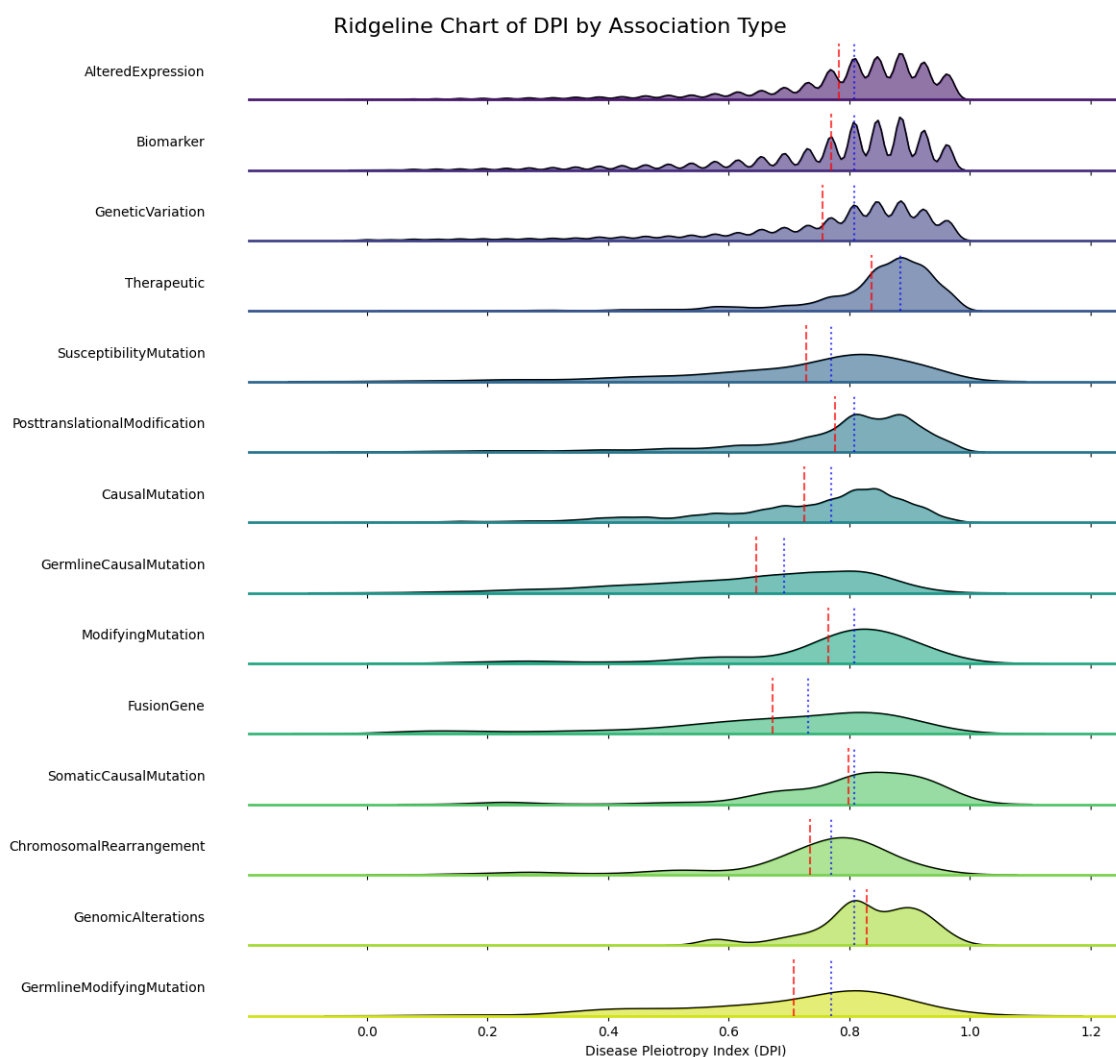


21. ábra Gének DSI értékeinek eloszlása a gén-betegség kapcsolatai szerint

A piros szaggatott vonallal az átlag, míg kék szaggatott vonallal a medián látható. Az elmondható, hogy a 0,5 DSI érték körül koncentrálnak az átlagok, mediánok.

Viszont a gyakoriságok sok helyen egyenletesen el vannak terülve, például a Fusion Gene esetében. Az ábrát nézegetve nem lenne téves azt a következtetést levonni, hogy a Germline Causal Mutation rendelkezik átlagosan a legmagasabb DSI értékekkel, megközelítőleg 0,6 és a gyakoriságot szimbolizáló görbe is szépen csúcsosodik az átlagnál és a mediánál, amelyek közel állnak egymáshoz. Tehát azok a gének, amelyek rendelkeznek Germline Causal Mutation ontológiai típusú kapcsolattal valamilyen betegség felé, azok általában elég speciális gének és kevés más betegséggel lehet őket kapcsolatba hozni. Ez kevésbé igaz a Genomic Alterations típusú kapcsolattal rendelkező génekre, ahol a DSI átlag közel két tizeddel marad el az előbb említettet képest.

A DSI-hez nagyon hasonló mutatóval rendelkeznek még a gének, ez pedig a Disease Pleiotropy Index (továbbiakban DPI). Ez az index azt méri, hogy az adott gén mennyire különböző osztályú betegségekkel van kapcsolatban. Ha sok eltérő osztályú betegséggel jelenthetünk ki asszociációt, magas lesz a gén DPI-je, ellenkező esetben alacsony. Az 1.8 fejezetben bemutatottak alapján a magas DPI jelentősége az, hogy az előlány számos egymástól függetlennek tűnő betegségeiért felel. A 22. ábrán látható a gének DPI értékeinek eloszlása a gén-betegség kapcsolatai szerint.



**22. ábra Gének DPI értékeinek eloszlása a gén-betegség kapcsolatai szerint**

Továbbra is piros szaggatott vonallal az átlag, míg kék szaggatott vonallal a medián látható.

Talán félre vezető lehet, a 19. ábrán nem ugyanaz látható más típusú diagramon. Ott a betegség osztályok szerint csoportosított betegségek arányát lehet látni, a hozzájuk tartozó asszociáció ontológiai típusa szerint lebontva. Itt azonban azt láthatjuk, hogy a gének, asszociációik szerint csoportosítva milyen DPI értékkel rendelkeznek, amely azt jelöli mennyire eltérő betegség osztályokkal vannak kapcsolatban. Meglepő módon átlagosan a terapeuta kapcsolattal is rendelkező gének, általában több különböző osztályú betegséggel vannak kapcsolatban. Ezek a kapcsolatok természetesen nem feltétlen kell terapeuta típusúak legyenek. Ahogy a 21. ábrán lehetett látni a Germline Causal Mutation kapcsolattal rendelkező gének kifejezetten specifikusak és itt is az

mutatkozik meg, hogy a legalacsonyabb átlagos DPI-vel rendelkeznek, tehát kevés különböző osztályú betegséggel hozhatóak kapcsolatba.

Szembevetendő, hogy az ábra tetején látható AlteredExpression, Biomarker, GeneticVariation ontológiai típusú kapcsolattal rendelkező gének DPI értékének eloszlása hullámzó. Ez véleményem szerint azért alakulhatott így mert, ezek az osztályok nagyon magas előfordulás számmal bírnak. Így könnyen lehet, hogy az ilyen típusú kapcsolattal rendelkező gének vegyes DPI-vel rendelkeznek és egy részük csak kevés betegségsoporthoz kapcsolódik, tehát alacsony DPI-értékkel rendelkezhet. Míg egy másik részük, amely számos, különböző betegségsoporthoz kötődik, magasabb DPI-értéket mutathat. Az ilyen heterogenitás több kisebb csúcsot eredményezhet az eloszlásban.

De ha pusztán az AlteredExpression osztályból indulunk ki, egy gén expressziójának megváltozása sok betegséghez társulhat, míg más gének expressziójának változása csak specifikus betegségsoportokra korlátozódik. Ez a sokszínűség adja az eloszlás hullámzást.

Tehát a geneAttributes tábla tartalmazza a génnek nevét, DSI, DPI pontszámát, egy rövid leírást a génről, illetve egy pLI pontszámot. Ez a pLI pontszám azt mutatja meg, hogy egy gén mennyire érzékeny egy olyan változásra, amely megváltoztatja a működését. Egy ilyen változás lehet akár egy génszakaszának elvesztése. Ha alacsony (0,1 vagy kevesebb), akkor a gén jobban elviseli ezeket a változásokat, ellenkező esetben (0,9 vagy annál nagyobb) a gén nagyon nehezen viseli a változásokat. [24]

## 4.1 Encoding

Mivel ezek a táblák sok esetben szöveges adatokat tartalmaztak, ezért szükséges volt ezeket a szöveges adatokat valamilyen encoding módszer segítségével numerikus alakba transzformálni.

### 4.1.1 One-Hot Encoding

Ez egy olyan technika, ahol kategorikus változókat ábrázolunk numerikusan. Ezek a kategorikus változók tipikusan több diszkrét értéket vehetnek fel, ezeket bináris

vektorok formájába kódolja. Ahol a kategóriák közötti viszonyt a nullák és az egyek reprezentálják. Ezzel ellehetetlenítve a numerikus sorrendbe rendezhetőség problémáját és a kategóriák közötti távolság is teljesen értelmetlen lesz. [25]

Például egy olyan szín kategorikus változót szeretnénk encodeolni, amely tegyük fel, hogy a korábbi példában említett három piros, zöld, kék színt veheti fel. Egy One-Hot encoder a következőképpen is ábrázolhatná őket:

piros = [1, 0, 0]

zöld = [0, 1, 0]

kék = [0, 0, 1]

#### 4.1.2 Word2Vec Encoding

A Word2Vec segítségével szavakat tudunk numerikus vektorokkal ábrázolni. Ezek a vektorok igyekeznek információval szolgálni nem csak a szóról, hanem azt azt körülvevő mondat szerkezetéről. [26]

A kódban a Word2Vec modellt a gének és betegségek tábláinak szöveges attribútumain, például a geneName, geneDescription és diseaseName került használatba, a beágyazott vektor reprezentációjának létrehozásakor. A szöveges oszlopok szavait a modell 25-dimenziós vektorokká alakítja, amelyek a szavak jelentését hordozzák.

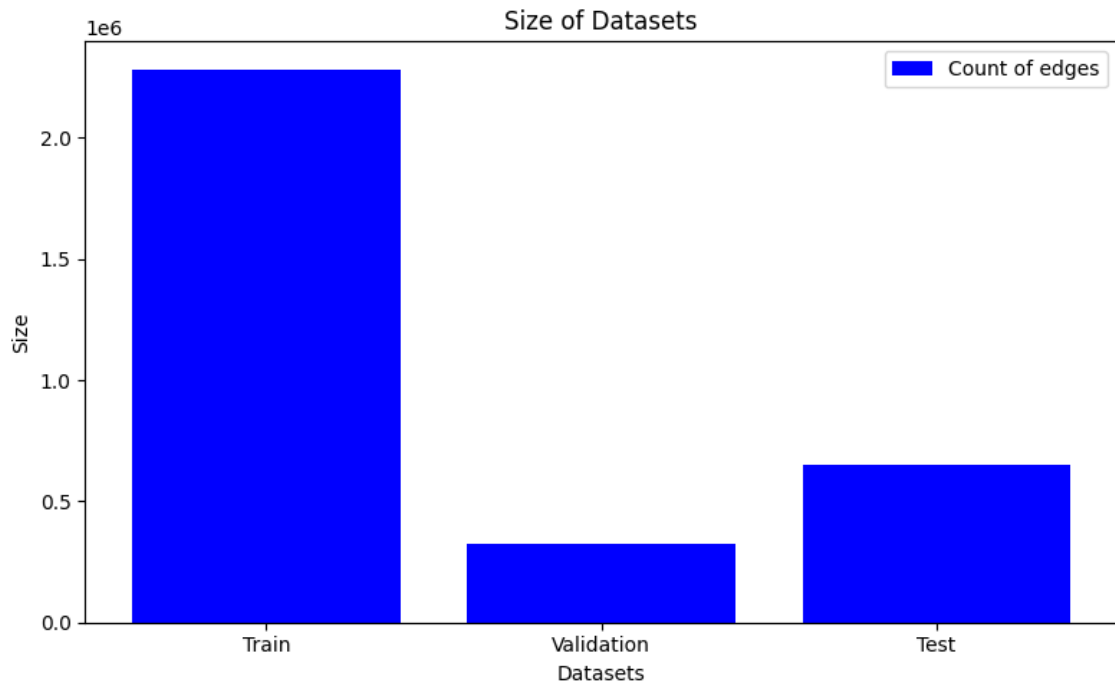
Az egyes szövegek teljes vektorábrázolásához a minden szó vektorát átlagolja egy függvény, hogy egyetlen átlagolt vektort kapjunk. Így a szövegeket tömör, numerikus formában ábrázolja, ezzel is csökkentve az adathalmaz méretét.

### 4.2 Adathalmaz felosztása

Az adatok transzformációját követően, egy gráf neurális hálók által feldolgozható struktúrába kellett alakítanom és fel kellett osztanom 3 részre: tanító, tesztelő és validációs adathalmazokra. A tanító ciklusokban a modell természetesen a tanító adathalmazon dolgozott, azonban a modell sikerességének megítéléséhez a validációs adathalmazon történt a kiértékelés. Ezek alapján a kiértékelések alapján optimalizáltam a modell paramétereit, hiperparaméter optimalizálással, amiről az 5.3-as fejezetben lehet olvasni. A teszt adat szerepe a nevéből fakadó tesztelés, amikor végeztünk a modell

tanításával és a megfelelő hiperparamétereket is megtaláltuk egy a tanítástól eltérő adaton való tesztelés.

Az adatokat az alábbi módon osztottam fel: a teljes adathalmaz 70% tanító adat, 20% teszt adat és a maradék 10% validációs adat. A 23. ábrán látszódik az élek adathalmazok szerinti eloszlása.



23. ábra Élek adathalmazok szerinti eloszlása

Amikor az adatok felosztását készítettem a Scikit train\_test\_split függvényét használtam, amely rendelkezik egy stratify nevű paraméterrel. Ennek a paraméternek a szerepe, hogy az eredeti adathalmaz kiválasztott paraméterének az eloszlását megtartja a felosztott adathalmazokban is. [27] A dolgozat esetében ez az associationType volt, tehát a célosztályunk. Így elkerülhető volt, hogy a ritkán előforduló ontológiai típusok ne jussanak minden adathalmazba.

## 5 A modell felépítése

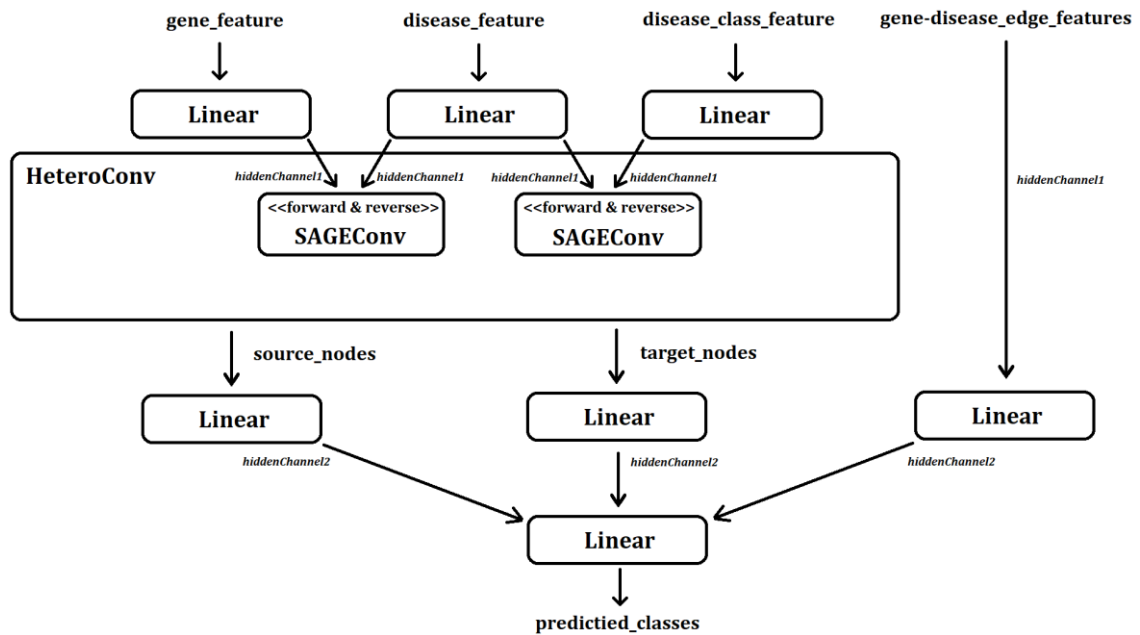
Ebben a fejezetben a modell felépítésének és megalkotásának a részleteivel lehet megismerkedni, beleértve a modell struktúráját és paramétereit, illetve a tanító ciklushoz tartozó információkat. A megvalósításhoz a PyTorch Geometric-et használtam, amely egy a PyTorch könyvtáron alapuló gráf neurális hálók tanítására tervezett könyvtár. [28]

### 5.1 Mit próbáltam megjósolni?

A modellem egy élek szintű következtetést próbált megvalósítani. Megpróbálta megjósolni a gének és a betegségek közötti kapcsolatnak az asszociációs típusát. Ez az asszociációs típus, eredetileg egy szöveges adat volt, melyet a korábbiakban ismertetett módon numerikus formára transzformáltam.

### 5.2 A modell rétegei

A modellem számos rétegből áll, amelyek különböző kombinációkban kerültek alkalmazásra az adatokon. Tipikusan a Linear és a SageConv rétegeket alkalmaztam a HeteroConv segítségével. A 24. ábrán látható a modell osztályozását egyszerűsített formában bemutató folyamatábrája.



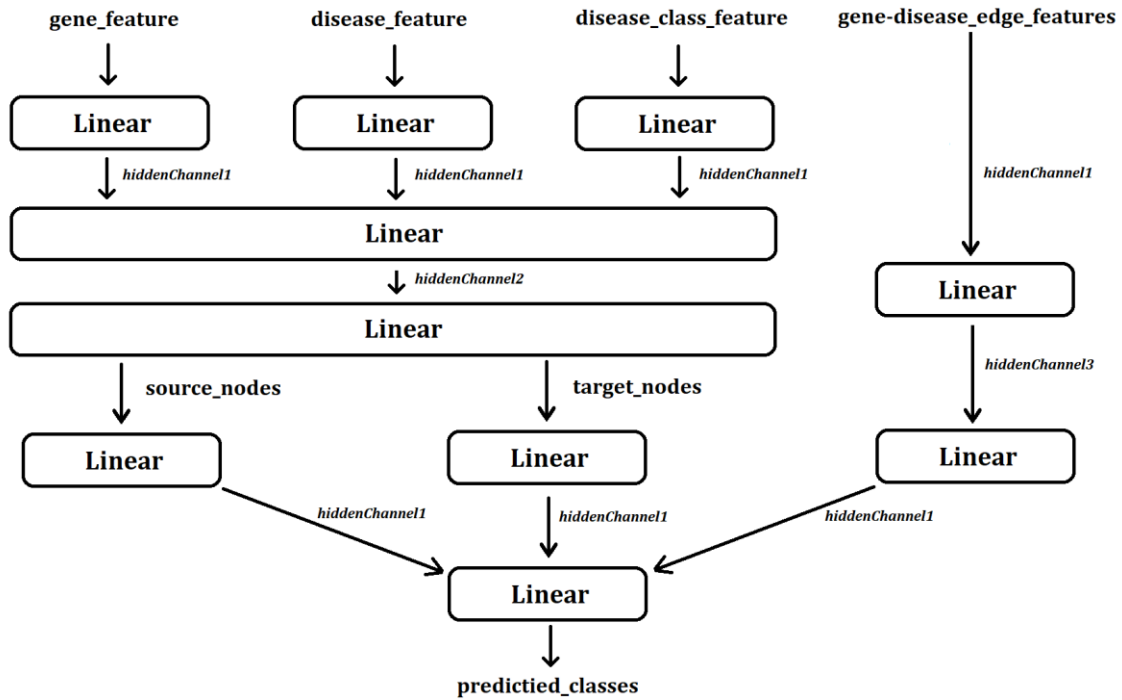
24. ábra GNN Modell felépítése

A három különböző típusú csomópont attribútumait egy lineáris rétegen keresztül vezeti a modellem. Ezek azok az attribútumok melyeket a 4. fejezetben mutattam be, pontosabban a geneAttributes, diseaseAttributes, diseaseClass táblák attribútumai. Ezek eredményeit pedig tovább adja a GNN konvolúciós rétegének, kapcsolatok szerint. Tehát gén- betegség kapcsolatokat egy SAGEConv-nak és a betegség-betegség osztály kapcsolatokat egy másik SAGEConv-nak. Az ábrán 2 SAGEConv réteg látható, de valójában 4db található a modellben, ugyanis a csomópontok közötti kapcsolat oda vissza irányú és ezt a két irányú kapcsolatot hivatott jelölni a <<forward & reverse>>.

A HeteroConv, amiben a SAGEConvok találhatóak igazából egy csomagoló réteg, amelyre azért van szükség mert alapvetően a GNN modellek homogén gráfot várnak, tehát ugyanolyan csomópontokat és éleket. Viszont a mi gráfunkban három különböző csomópont van és két különböző él, itt az információkat valahogy tudnunk kell aggregálni az információkat és erre ad lehetőséget a HeteroConv réteg. [29]

A HeteroConv kimenete az információ forrás és cél csomópontja, ezek a mi esetünkben a gén és betegség pontjai a gráfban. Ezek külön-külön lineáris transzformáláson esnek át, párhuzamosan a csomópontok közötti élek attribútumaival. Az élek attribútumai geneDiseaseNetwork táblán található attribútumok. Végül ezeket a transzformált adatokat konkatenálja a modell és a konkatenált adatokat még egyszer lineáris transzformáció alá veti. Ezen transzformációból kapjuk meg a célváltozó logitjait, amelyekből softmax függvény segítségével valószínűségeket generálunk.

A modellem kiértékeléséhez szükségem volt egy hagyományos Neurális hálóra, erről részletesebben a 6. fejezetben lehet olvasni. A 25. ábrán lehet látni a Neurális háló felépítését.



25. ábra NN Modell felépítése

Itt is alkalmaztam hiperparaméter optimalizációt, a megfelelő eredmények elérése végett. Természetesen a legszembetűnőbb változás a GNN-hez képest, hogy eltűnt a gráf konvolúciós réteg és csak Lineáris rétegek találhatók a modellben. Az, hogy nincs más réteg a modellben nem egy törvényszerű dolog, egész egyszerűen az optimalizálás következtében ilyen formájú modell készült.

A különböző rétegeket követően dropout függvényt alkalmazva igyekeztem megtartani a modellek általánosító képességét. Amely annyit csinál, hogy a tanító adatokat random módon kitörli, ezzel elvéve a lehetőséget a modellek elől a túltanulásra. [30]

A dropout függvényt leszámítva aktivációs függvény is alkalmazásra került. A hiperparaméter optimalizáláskor a legjobb eredményeket a ReLU függvénnyel érték el a modellek. A 26. ábrán látható a függvény képlete, ahol észrevehetjük, hogy ha a bemenet pozitív, akkor az értéket változatlanul továbbítja, de ha negatív, akkor nullára állítja. Így azt mondhatjuk, a ReLU előnye a nemlinearitás, ami kiváltképp fontos egy mélytanuló modellnél. [31]

$$\text{ReLU}(x) = (x)^+ = \max(0, x)$$

26. ábra ReLU aktivációs függvény képlete [31]

### 5.3 A hiperparaméterek kiválasztása

A modell hiperparamétereinek a meghatározásához hiperparaméter optimalizálást alkalmaztam. Mivel ez a téma is egy saját dolgozatot érdemelne, ezért nem férne bele ezen dolgozatban a hiperparaméter optimalizálás részletes kifejtése.

Pongyolán elegendő annyit tudni róla a dolgozat megértése szempontjából, hogy a hiperparaméterek lehető legjobb kombinációját lehet vele megtalálni. Felveszi a paraméterek értékkészletének valamennyi értékét és az alapján a lefuttatja a modellt és a legjobb eredményt adó kombinációt fogja ajánlani a fejlesztő számára. [32] Ennek köszönhetően tudtam meghatározni, az iterációk számát, a rétegek közötti csatornák számát, magát a rétegeknek a számát. Továbbá az Adam optimalizációs függvényhez tartozó tanulási rátát és a súlyt.

Illetve az osztályozó egy szintén kiemelten fontos elemét, a küszöbértékek kiszámításának a függvényét, amelyet a 27. ábrán lehet látni, is hiperparaméter optimalizálással találtam meg.

$$threshold = \frac{1}{[classCount]^{thresholdController}}$$

27. ábra Hierarchikus küszöbértékek függvénye

Tehát az osztályok előfordulásának a számát egy hiperparaméterrel hatványozza a függvény, ezt a hiperparamétert kerestem meg hiperparaméter optimalizálással.

A 28. ábrán lehet látni a GNN modell esetében keresett paraméterek lehetőségeit.

| dropoutRate | learningRate | weightDecay | epochs | hiddenChannels1 | hiddenChannels2 |
|-------------|--------------|-------------|--------|-----------------|-----------------|
| 0.2         | 0.0001       | 1,00E-06    | 500    | 64              | 64              |
| 0.3         | 0.001        | 1,00E-05    | 1000   | 128             | 128             |
| 0.5         | 0.005        | 0.0001      | 2000   | 256             | 256             |
| 0.6         | 0.05         | 0.001       | 5000   | 512             | 512             |

28. ábra GNN modell lehetséges hiperparaméterei

A zölddel kiemelt cellákban található paramétereket választott az optimalizáló. A hiddenChannel a 28. és 29. ábrákon látható paraméterek, melyek a rétegek között átadott adatok dimenzióját jelölik. A 29. ábrán látható az NN modell esetében keresett paraméterek lehetőségeit.

| dropoutRate | learningRate | weightDecay | epochs | hiddenChannels1 | hiddenChannels2 | hiddenChannels3 |
|-------------|--------------|-------------|--------|-----------------|-----------------|-----------------|
| 0.2         | 0.0001       | 1,00E-06    | 500    | 64              | 64              | 64              |
| 0.3         | 0.001        | 1,00E-05    | 700    | 128             | 128             | 128             |
| 0.5         | 0.005        | 0.0001      | 900    | 256             | 256             | 256             |
| 0.6         | 0.05         | 0.001       | 1100   | 512             | 512             | 512             |
| 0.7         | 0.01         | 0.005       | 1300   |                 |                 |                 |
|             |              |             | 1500   |                 |                 |                 |
|             |              |             | 2000   |                 |                 |                 |

29. ábra NN modell lehetséges hiperparaméterei

Az optimalizáló az NN modell esetében se volt képes 512-es hiddenChannel méretet elérni. Egy nagyobb GPU RAM-mal rendelkező gép esetében érdemes lehet kipróbálni ezeket az optimalizációkat.

## 6 Kiértékelés

Ebben a fejezetben a modell teljesítményének a kiértékelését szeretném bemutatni. Ez egy kiváltképp érdekes feladat volt számomra, ugyanis nem minden mérőszám alkalmas egy ilyen adatbázis esetén. Amikor különböző metrikákat vizsgálva próbáljuk megítélni a modell eredményességét, érdemes valamilyen viszonyítási alapot kínálni a GNN-nek, ezért készítettem egy hagyományos neurális hálót (továbbiakban NN). Az NN-en is természetesen alkalmaztam hiperparaméter optimalizálást, hogy a lehető legjobb eredményt nyújtsa.

### 6.1 Pontosság

A modell megfelelő kiértékelése egy kifejezetten nagy kihívás elé állított, ugyanis nem lehet egy ennyire kiegyensúlyozatlan eloszlásra egyszerűen a pontosság alapján megítélni a modellem helyességét. Ha belegondolunk ez a pontosság karakterisztikájából adódik, ugyanis a pontosság egy olyan teljesítmény mutató, amely a modell által összes helyesen besorolt minta arányát mutatja az összes mintához képest, ahogy ezt a 30. ábra mutatja. [33]

$$accuracy = \frac{\text{count of correct predictions}}{\text{count of all predictions}}$$

30. ábra Pontosság képlete

Mivel a cél változó, tehát az éleken található asszociációs típusnak az eloszlása nagyon egyenlőtlen volt, ezért a pontosság önmagában nem elegendő mérőeszköz arra, hogy megállapítsuk a modellem hatékony-e vagy sem. [33]

### 6.2 F1 pontszám

Figyelemmel kísértem az F1 pontszámot is a tanítás alatt. Az F1 pontszám megértéséhez szükséges megismernünk a recall, vagyis az érzékenység és a precision, azaz a precizitás mutatóit.

Szeretném bemutatni őket egyszerű példákon keresztül. A precision-nél képzeljük el, hogy egy páciens lehet beteg vagy egészséges. És a mi modellünkkel szeretnénk megmondani, hogy egészséges vagy beteg a páciens. A precision-nel azt tudjuk meghatározni, hogy a modell által egészségesnek ítélt páciensek között valójában mekkora volt az egészségesek aránya. Tehát ahogy a 31. ábrán is lehet látni, a helyesen egészségesnek ítéltet lehet megfeleltetni a TP (True Positive) változónak. A Hamisan egészségesnek ítéltet pedig a FP (False Positive) változónak. [33]

$$Precision = \frac{TP}{TP + FP}$$

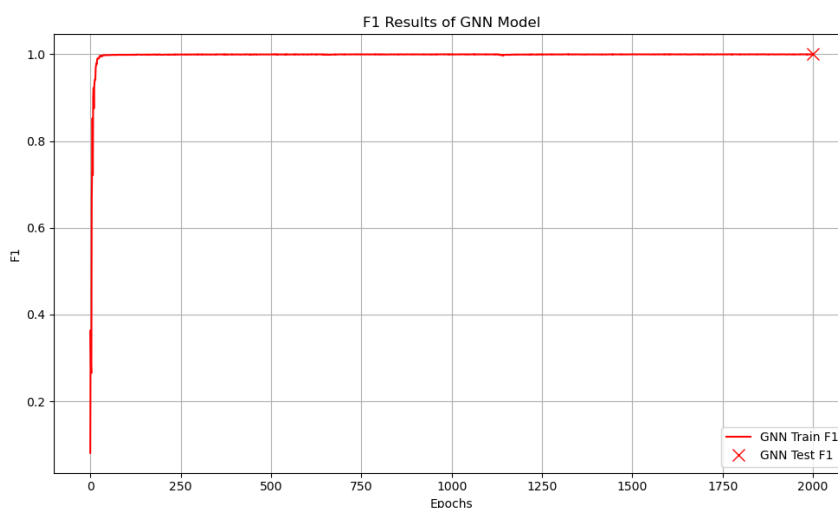
31. ábra Precizitás képlete [33]

A recall mutatóval pedig azt tudnánk meghatározni, hogy a valójában egészséges betegek közül mennyit sikerült a modellnek eltalálnia. [33] A 32. ábrán látszódik a képlete, annyi a változás a 31. ábrához képest, hogy a FP-t felváltotta az FN (False Negative), amely a hibásan betegnek ítéltet szimbolizálja.

$$Recall = \frac{TP}{TP + FN}$$

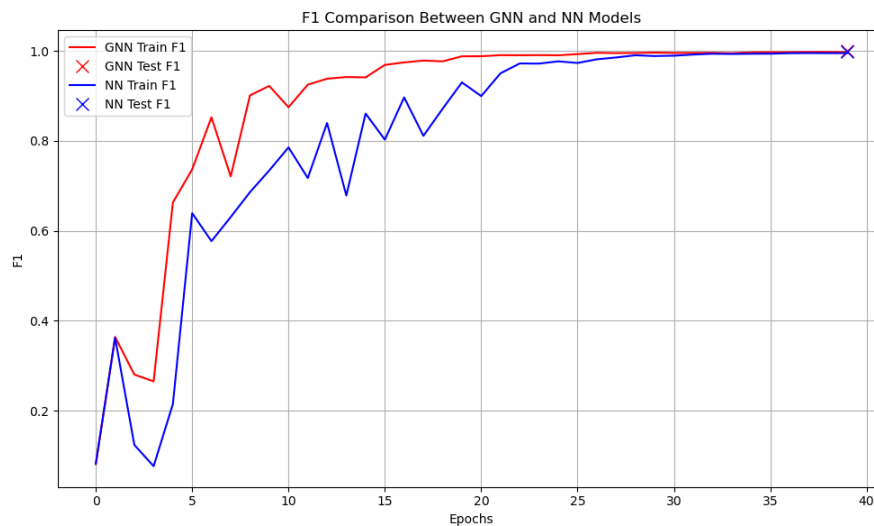
32. ábra Érzékenység képlete [33]

Az F1 pontszám a precizitás és a recall harmonikus középértéke. [33] Ha a 33. ábrára tekintünk, láthatjuk a modellünk F1 pontszámának görbét.



33. ábra GNN modell F1 görbéje 2000 hosszú iteráción

Ez a diagram önmagában nem rendelkezik sok információval. Látható, hogy nagyon korán nagyon magas F1 pontszámot ér el a modellünk. Ez önmagában persze egy jó hír, azonban érdemes összevetnünk egy hagyományos neurális háló által elért F1 pontszámokkal, egy kisebb iteráción, ahogy azt a 34. ábrán lehet látni



**34. ábra GNN és NN modellek F1 görbéje 40 hosszú iteráción**

A két modell viszonylag szimmetrikusan kezdi a tanulást, azonban a GNN hamarabb kezd el kiegyenesedni a maximális F1 pontszám mentén. Azonban látható az is, hogy egy bizonyos iteráció után nem lesz igazán különbség a két modell eredményei között. A legutolsó iteráció eredményei a teszt adatokon például 0.9997 a GNN esetében és 0.9984 a NN esetében, ez egy csupán 0.0013-as különbség a GNN javára, amely talán elhanyagolhatónak tűnhet első látomásra, de a továbbiakban kiderül mi okozza ezt a különbséget és miért jelentős ez önmagában mégis.

## 6.3 MCC

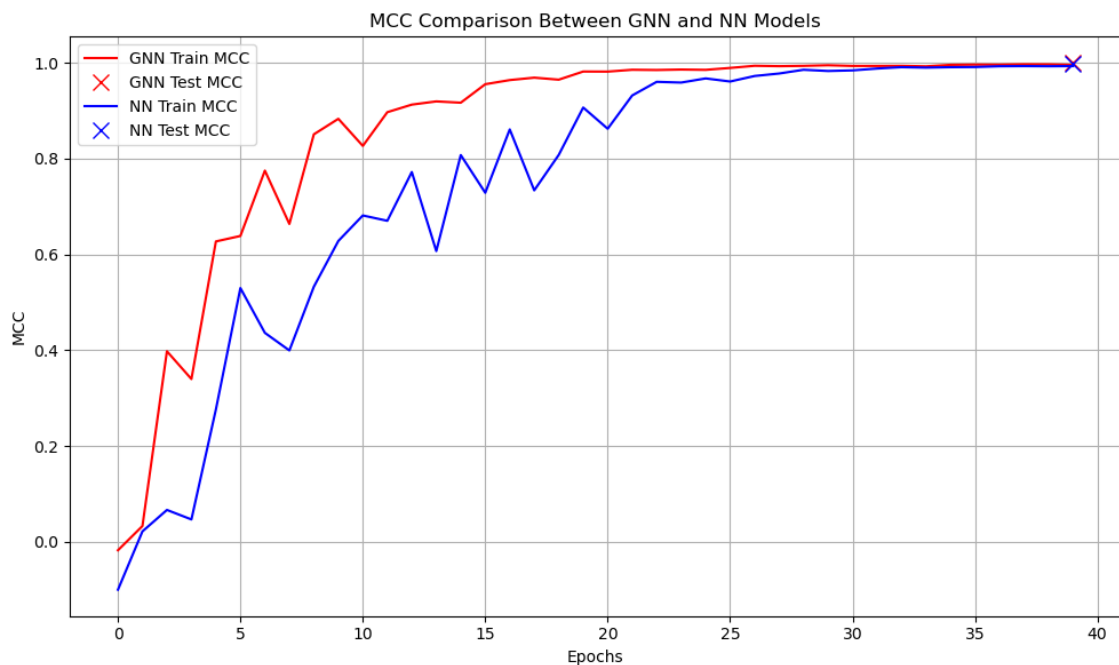
A modell kiértékelésekor igyekeztem minél több metrikát figyelembe venni, hogy pontosabb képet kapjak a modellem hatékonyságáról. Egy ennyire kiegyensúlyozatlan eloszlással bíró adathalmaz esetében azt a következtetést kellett levonnom, hogy az F1 pontszám sem a legmegfelelőbb mutató arra, hogy bemutassuk a modell pontosságát. A Matthews Correlation Coefficient (továbbiakban MCC) azonban egy fokkal biztonságosabb metrika, ugyanis figyelembe veszi a tévesztési mátrix minden celláját, tehát az összes hibát is. Így képes az összes osztályon átívelő teljesítményt mutatni. A 35.

ábrán látszódik az MCC képlete, az egyetlen változó, amelyet még nem ismerünk a TN (True Negative), amely a korábbi példából következtetve a helyesen betegnek ítélteteket jelöli. [33]

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FN)(TP + FP)(TN + FN)(TN + FP)}}$$

35. ábra MCC képlete [33]

Azonban az MCC gyengesége, hogy értékei kiváltképp alacsonyak lehetnek egy egyenlőtlen eloszlású adathalmazon. [33] A 36. ábrán egy nagyon hasonló szituáció tárul elénk, mint amit az F1 pontszámánál a 34. ábrán láthattunk.



36. ábra GNN és NN modellek MCC görbéje 40 hosszú iteráción

Ezen a ponton felmerülhet bennünk a gondolat, nagy epoch szám meghatározásakor van-e lényegi különbség a GNN és az NN között? A válasz az osztályszintű pontosságban rejlik.

## 6.4 Osztályszintű pontosság

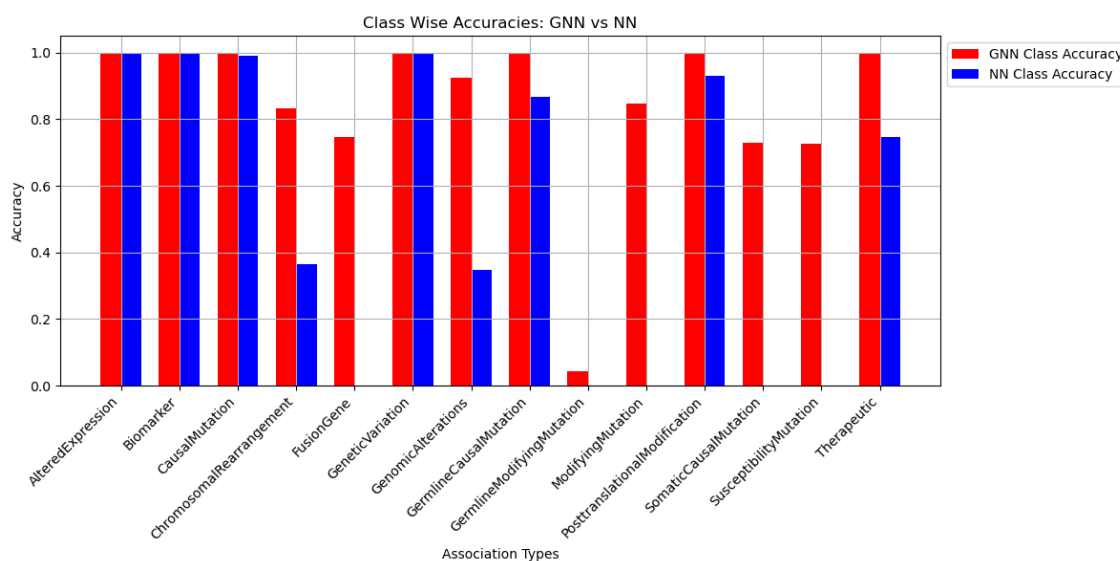
Az osztályszintű pontosság abban különbözik a fejezet elején ismertetett pontosságtól, hogy itt nem az általános minden osztályon átívelő helyes osztályozott példák számát osztjuk az összes példával, hanem csak az osztály helyes predikcióit az

osztály minden példájának számával. A 37. ábrán látható az osztályszintű pontosság képlete.

$$\text{class wise accuracy} = \frac{\text{count of correct predictions by class}}{\text{count of all predictions by class}}$$

37. ábra Osztályszintű pontosság képlete

A 38. ábrára tekintve láthatjuk a GNN és az NN modellek osztályszintű pontosságát. Mielőtt belemennénk a mélyebb elemzésbe, talán célszerű egy pillantást vetnünk a 15. ábrára, ahol láthattuk az osztályok eloszlását.



38. ábra GNN és NN modellek osztályszintű pontosságának diagramja

Első ránézésre is látszódik egy markáns különbség a két osztályozó eredményei között. Nincs olyan osztály az adatbázisban, ahol az NN pontossága megelőzné a GNN -ét. Az nagyon jól megfigyelhető, hogy mind a két osztályozó magas értékeket ér el azon osztályoknál, amelyeknek magas az előfordulása, például a Biomarker vagy a Genetic Variation.

Az NN és a GNN modellnél is érdemes felfigyelni arra, hogy a Susceptibility Mutation esetszáma magasabb jóval (közel tízszerese), mint a Chromosomal Rearrangement-nek, mégis az utóbbi osztályszintű pontossága az erősebb. Tehát nem egyenes arányos az osztályszintű pontosság az osztályok előfordulásával. Mivel ezen osztályok esetszáma kombinálva sem éri el a teljes gén-betegség asszociációk 0,2%-át, nem mernék messzemenő következtetéseket levonni, miért is alakulhatott így ennek a metrikának az eloszlása. Azonban fontos tényező lehet az elhelyezkedésük a

hierarchiában, ugyanis a Chromosomal Rearrangement egy szinttel feljebb helyezkedik el. Lehetséges, hogy a Genetic Variation, mely a Chromosomal Rearrangement testvére és a Susceptibility Mutation öse, olyan küszöbértékkel rendelkezik, amely megfogja az osztályozót az alacsonyabb szintekre való lejutástól. Továbbá a 21. ábrára tekintve láthatjuk, hogy a Susceptibility Mutation-hoz tartozó gének specifikációjának eloszlása kiváltképp egyenletes, amely esetlegesen nem ad megfelelő karakterisztikát ennek a gén-betegség kapcsolatnak.

Ha megfigyeljük az MCC és az F1 metrikákban lévő minimális különbségek, ebben a mérőszámokban csúcsosodnak ki. Itt látszódik igazán, hogy azokat az osztályokat is, amelyek minimális előfordulással rendelkeznek, sokkal nagyobb pontossággal találja el a GNN. De ezek összeadva se érik el az össz esetszám 1%-át, ezért az általánosított mérőszámoknak, mint az MCC és az F1, nem volt esélyük kimutatni ezt a különbséget.

## 6.5 Kohen Kappa

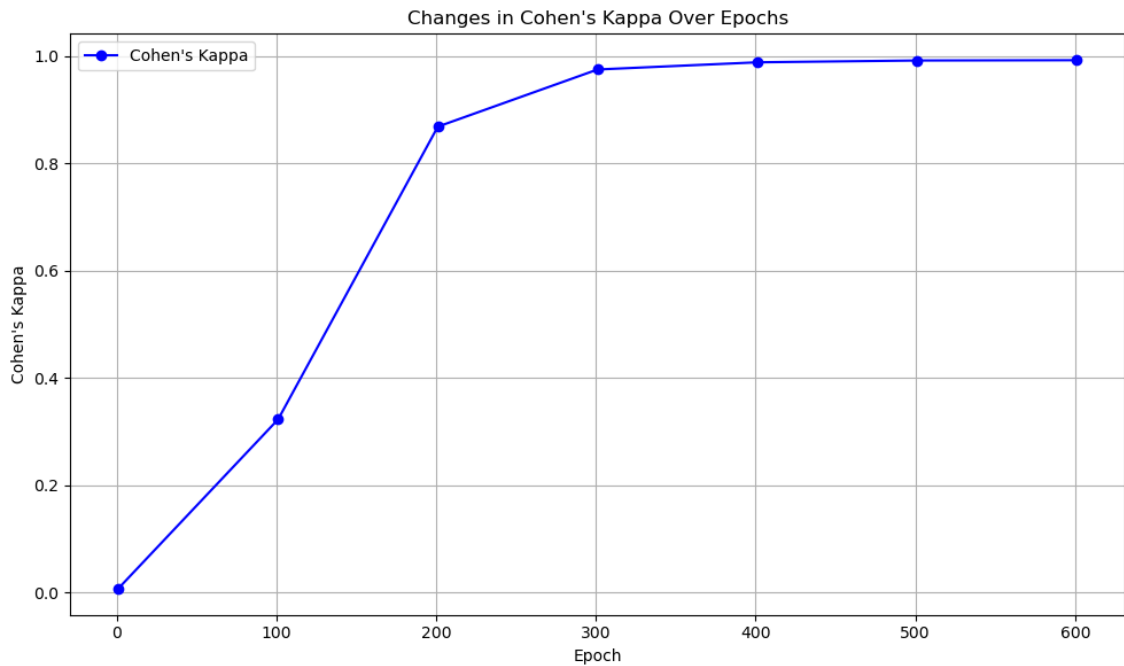
A modell kiértékeléséhez szükséges utolsó metrika a Kohen Kappa, amely a két osztályozó közötti kapcsolatot mutatja be. Pontosabban fogalmazva, bemutatja mennyire a hasonló a két osztályozó osztályozása, a véletlen egyezéseket is bele számítva. Előnye a többi mutatóhoz képest, hogy figyelembe veszi a várható egyezést, nem csak a helytelen és helyes osztályozásokat. [33] A 39. ábrán látható a Kohen Kappa képlete.

$$K = \frac{P_o - P_e}{1 - P_e}$$

39. ábra Kohen Kappa képlete [33]

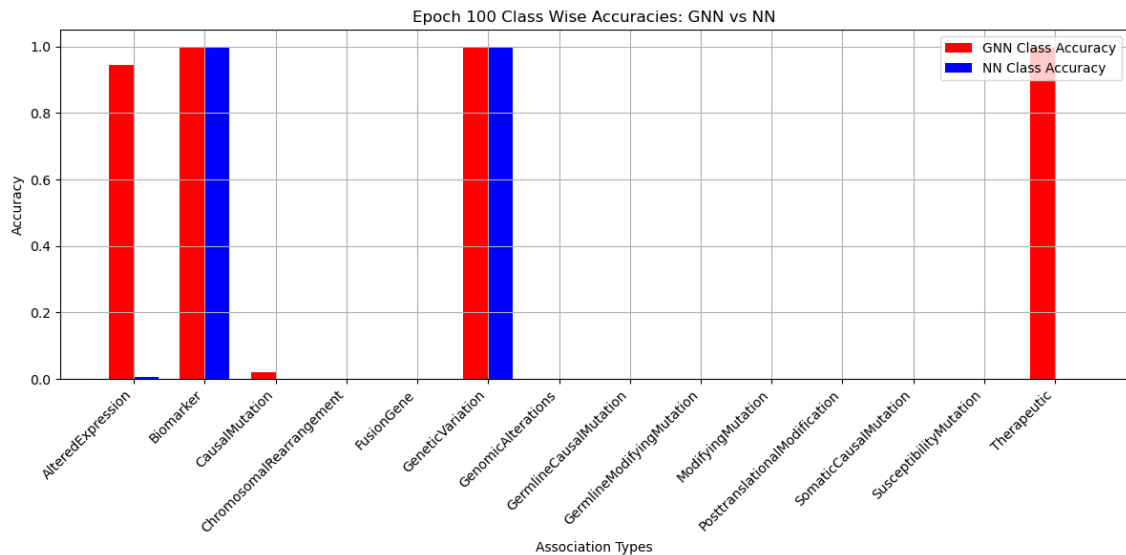
A  $P_o$  változó a két osztályozó közötti egyező osztályozások össz osztályozáshoz viszonyított arányát szimbolizálja. Vagyis a két osztályozó hány esetben osztályozta ugyanazt az ontológiai típust az asszociációkra az összes asszociáció számához képest.  $P_e$  változó ehhez képest a várható pontosságot szimbolizálja. Tehát annak az arányát, hogy mennyit fog eltalálni a modell az asszociációs típusokból az eloszlásuk alapján. [33]

A 40. ábrán látható a GNN és az NN osztályozó közötti egyezés tehát, hogy mennyire osztályoznak hasonlóan az iterációk előrehaladtával.



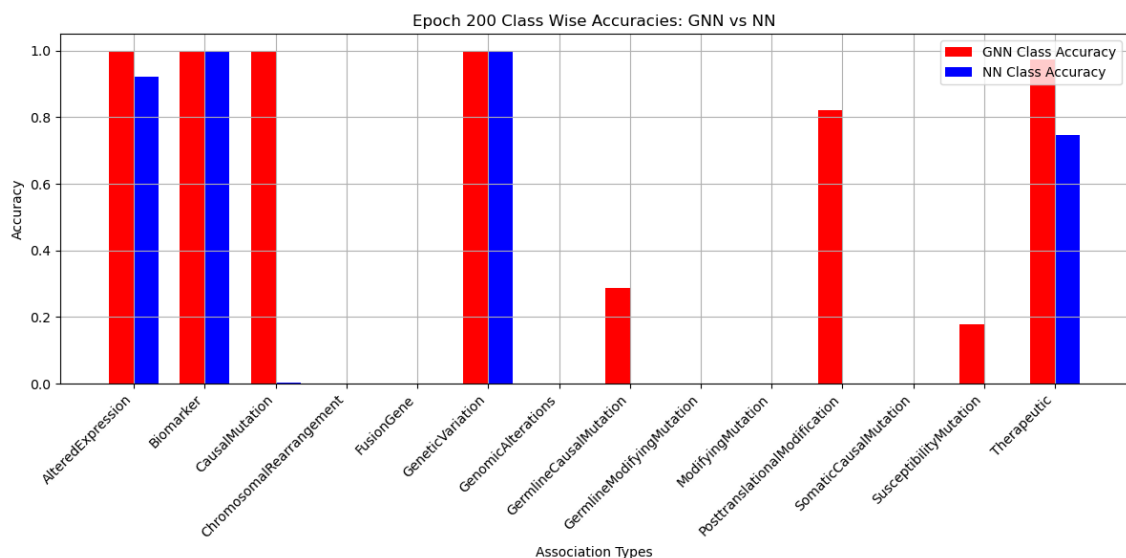
**40. ábra Kohen Kappa görbéje**

Meglepő eltérés lehet a többi ábrához képest, hogy itt elég lassan közelíti az egyenesünk a maximális értéket. A 34 ábrán például mind a két modell a 20. epoch környékén már az 1-hez közeli F1 pontszámmal rendelkezik. Az első 100 iteráció alatt csupán 0,3 körüli egyezést ér el a két modell. Az ezt követő 100 iterációban történik egy erős ugrás, ahol megközelítőleg 0,6-tal emelkedik meg az egyezések értéke. Ezután egy stabilizálódás látható, amely szépen közelíti a tökéletes egyezést megtestesítő 1-es értéket. Ez a korábban látható magasan reprezentált osztályok eltalálása miatt könnyen elérhető. A 41. ábrán látszódik az osztályszintű pontossága a két modellnek a 100. iterációban.



**41. ábra GNN és NN modell osztályszintű pontossága a 100. iterációban**

Erős különbség látható itt még a leggyakoribb osztályok között is, például az AlteredExpression esetében az NN modell nem képes felvenni a versenyt. Tehát a 100. iterációban a két osztályozó közötti egyet nem értés a 40. ábrán nem véletlen. A 42. ábrán viszont már a 200. iterációban láthatjuk az osztályszintű pontosságokat.



**42. ábra GNN és NN modell osztályszintű pontossága a 200. iterációban**

Itt már komolyan felzárkózott az NN a GNN-hez a magasan reprezentált osztályokban. Érdekes azonban, hogy a Terápiás ontológiai típusokat hamarabb kezdte el felismerni, mint az Okozati Mutációkat, pedig megközelítőleg hatszor több Okozati Mutáció típusú kapcsolat van az adatbázisban.

## 6.6 Következtetés

A korábbi ábrák csak megerősíthetnek minket abban a hitben, hogy nagyon alaposan kell körül járnunk a kiértékelés témáját. Számos különböző metrikát érdemes alkalmaznunk, ha meg akarunk győződni a modellünk helyességében.

A GNN adta lehetőség, hogy az adatokat gráfként kezeljük és a gráf elemei kiterjesszék a karakterisztikájukat és kölcsönösen hatással legyenek egymásra egy erős fegyvernek bizonyult. Az ábrák első ránézésre azt a képet festhetik le, hogy nincs olyan nagy különbség egy hagyományos neurális hálós megvalósítás és egy gráf neurális hálós között. Azonban nem győzöm hangsúlyozni, hogy a döntő különbséget a két modell között az alulreprezentált osztályoknál lehet megtalálni, a sokadik iterációban. A GNN képes olyan kapcsolatok sikeres osztályozására is, amelyek előfordulási gyakorisága nagyságrendekkel kisebb, mint a dominánsabb osztályoké, például a harmadik leggyakoribb osztály, megközelítőleg tízezerszer többször fordul elő, mint a legritkább. Véleményem szerint az aggregáció az a kulcsfontosságú folyamat, amely lehetővé teszi, hogy ezeknek a ritkább osztályoknak a tulajdonságai is terjedjenek a gráf struktúráján keresztül és valamelyest egyensúlyozzák ezt az egyenlőtlen eloszlást.

Tehát ezen dolgozat egyik legfontosabb következtetése számomra az lehet, hogy a gráf neurális hálók nem csak általánosan, hanem kifejezetten az alulreprezentált osztályok osztályozásában is hatékonyan teljesíteni, még torz vagy kiegyensúlyozatlan eloszlások esetén is.

## 7 Továbbfejlesztési lehetőségek

A dolgozat megkezdése óta megjelent egy újabb verziója a DisGeNET adatbázisnak, amely számos újítást tartalmaz. Természetesen megnőtt az adatbázis mérete, illetve az ontológiai hierarchia kiegészült újabb osztályokkal. Ezen felül újabb táblák kerültek be az adatbázisba. Ezeknek a módosításoknak a bevonásával tovább lehetne finomítani az osztályozó eredményességét.

A dolgozatban egyetlen módszertan szerint készítettem el a hierarchikus osztályozót. Érdemes lenne más lehetséges formákat kipróbálni.

A kiértékelést tovább lehetne finomítani, más modellek összehasonlításával is, tehát nem feltétlenül csak neurális hálóval.

Abban is rejlik továbbfejlesztési lehetőség, hogy ha nem kifejezetten az éleknek az asszociációs ontológiai típusát próbálnánk meghatározni, hanem például a betegségekhez tartozó gének, különböző attribútumait vagy akár magát a gént is.

## Irodalomjegyzék

- [1] [Szabad János – Sejtbiológia És Molekuláris Genetika jegyzet](#), 24. oldal 2012 ősz
- [2] Magyar Elektronikus Könyvtár: A humán genom, [A humán genom | Magyar Elektronikus Könyvtár](#) (utolsó letöltés: 2024. 04. 13.)
- [3] Wikipédia: *Kromoszóma*, [Kromoszóma | Wikipédia](#) (utolsó letöltés: 2024. 04. 13.)
- [4] Khan Academy: *A kromoszómák*, [A kromoszómák | Khan Academy](#) (utolsó letöltés: 2024. 04. 13.)
- [5] [Szabad János – Sejtbiológia És Molekuláris Genetika jegyzet](#), 23. oldal 2012 ősz
- [6] BioKemOnline: *A DNS-től a fehérjéig*, [A DNS-től a fehérjéig | BioKemOnline](#) (utolsó letöltés: 2024. 04. 13.)
- [7] Wikipédia: *Ribonukleinsav*, [Ribonukleinsav | Wikipédia](#) (utolsó letöltés: 2024. 04. 13.)
- [8] [Szabad János – Sejtbiológia És Molekuláris Genetika jegyzet](#), 45. oldal 2012 ősz
- [9] Griffiths, A. J. F., Wessler, S. R., Carroll, S. B., Doebley, J.: *Introduction to Genetic Analysis*, 11th edition, ISBN-13: 978-1-4641-0948-5, W. H. Freeman and Company A Macmillan Education Imprint, 2015
- [10] [Szabad János – Sejtbiológia És Molekuláris Genetika jegyzet](#), 72. oldal 2012 ősz
- [11] StoryMD: *What is a gene variant and how do variants occur?*, [What is a gene variant and how do variants occur? | MedlinePlus](#) (utolsó letöltés: 2024. 04. 13.)
- [12] DisGeNET: *The DisGeNET Association Type Ontology*, [The DisGeNET Association Type Ontology | DisGeNET](#), (utolsó letöltés: 2024.05.17)
- [13] Springer Link: *Genomic alterations and diagnosis of renal cancer*, [Genomic alterations and diagnosis of renal cancer | Springer Link](#) (utolsó letöltés: 2024.10.17)
- [14] National Cancer Institute: [NCI Dictionary of Cancer Terms, NCI Dictionary of Cancer Terms | National Cancer Institute](#) (utolsó letöltés: 2024.10.17)
- [15] Sullivan, K. E., Stiehm, E. R.: *Stiehm's Immune Deficiencies*, ISBN: 978-0-12-405546-9, Elsevier, 2014
- [16] Strachan, T., Lucassen, A.: *Genetics And Genomics In Medicine*, 2nd edition, ISBN 9780367490812, CRC Press, 2023
- [17] PyTorch: *Linear*, [Linear | PyTorch](#) (utolsó letöltés: 2024.05.17)

- [18] Medium: *Graph Convolutional Networks: Introduction to GNNs*, [Graph Convolutional Networks: Introduction to GNNs | Medium](#) (utolsó letöltés: 2024.05.17)
- [19] Medium: *Exploring SageConv: A Powerful Graph Neural Network Architecture*, [Exploring SageConv: A Powerful Graph Neural Network Architecture | Medium](#) (utolsó letöltés: 2024.05.17)
- [20] Hamilton, W. L., Ying, R., Leskovec, J.: *Inductive Representation Learning on Large Graphs*, Department of Computer Science Stanford University, 2018
- [21] Silla Jr., C. N., Freitas, A. A.: *A survey of hierarchical classification across different application domains*, Data Mining and Knowledge Discovery - Springer Link, Vol. 22, 2010, pp 31-72
- [22] DisGeNET: *Home*, [Home | DisGeNET](#), (utolsó letöltés: 2024.05.17)
- [23] DisGeNET: *Metrics*, [Metrics | DisGeNET](#), (utolsó letöltés: 2024.10.22)
- [24] DisGeNET: *Data Attributes*, [Data Attributes | DisGeNET](#) (utolsó letöltés: 2024.10.26)
- [25] Wikipedia: *One-hot*, [One-hot | Wikipedia](#), (utolsó letöltés: 2024.10.22)
- [26] Wikipedia: *Word2vec*, [Word2vec | Wikipedia](#), (utolsó letöltés: 2024.10.22)
- [27] Scikit Learn: *train\_test\_split*, [train\\_test\\_split | Scikit Learn](#), (utolsó letöltés: 2024.10.26)
- [28] Pytorch Geometric: *PyG Documentation*, [PyG Documentation | Pytorch Geometric](#), (utolsó letöltés: 2024.10.23)
- [29] Pytorch Geometric: *conv.HeteroConv*, [conv.HeteroConv | Pytorch Geometric](#), (utolsó letöltés: 2024.10.23)
- [30] Pytorch: *Dropout*, [Dropout | Pytorch](#), (utolsó letöltés: 2024.10.21)
- [31] Pytorch: *Relu*, [Relu | Pytorch](#), (utolsó letöltés: 2024.10.21)
- [32] AWS: *What is Hyperparameter Tuning?*, [What is Hyperparameter Tuning? | AWS](#), (utolsó letöltés: 2024.10.21)
- [33] Grandini, M., Bagli, E., Visani, G.: *METRICS FOR MULTI-CLASS CLASSIFICATION: AN OVERVIEW*, 2020

## **Függelék**

A dolgozathoz felhasználtam a ChatGPT mesterséges intelligencia alapú eszközt, nyelvhelyesség korrigálása céljából.