Budapesti Műszaki és Gazdaságtudományi Egyetem
Villamosmérnöki és Informatikai Kar
Hálózati Rendszerek és Szolgáltatások Tanszék

Kányádi Richárd

FORGALMI TORLÓDÁS DETEKCIÓ GÉPI TANULÁSON ALAPULÓ ESZKÖZÖK SEGÍTSÉGÉVEL

KONZULENS

Dr. Simon Vilmos

Bereczki Norman Zoltán

BUDAPEST, 2023

Tartalomjegyzék

Összefoglaló **4**
Abstract **5**
1 Bevezetés **6**
2 Motiváció **8**
3 Szakirodalmi áttekintés **10**
4 Összehasonlított anomália detekciós modellek bemutatása **12**
4.1 Döntési fa 12
4.2 Logisztikus regresszió 13
4.3 Support Vector Machine 15
4.4 XGBoost 16
5 Torlódás detekciós módszerek kiértékelése **18**
5.1 Szimuláció 18
5.2 Eredmények 25
6 Összegzés **36**
Irodalomjegyzék **38**
Ábrajegyzék **40**
Táblázatjegyzék **41**

HALLGATÓI NYILATKOZAT

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

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

Kelt: Budapest, 2023. 12. 08.

...…….

Kányádi Richárd

Összefoglaló

A városok folyamatos terjeszkedése és fejlődése, valamint a jelenséget kísérő határtalan mobilitási igények miatt kiemelten fontos a közlekedési rendszerek hatékony irányítása. Egy rosszul menedzselt forgalmi infrastruktúrában gyakoriak a torlódások, mindennaposok a késések, ami számos probléma forrása lehet az emberek életében, pl.: időveszteség, idegesség, balesetek kockázata.

A közlekedés minősége javítható, például infrastruktúra fejlesztésével: új forgalmi sávok, körforgalmak bevezetése. További javulás érhető el az utakon lévő járművek számának csökkentésével vagy a forgalom összetételének megváltoztatásával, például tömegközlekedés, kerékpárok vagy elektromos rollerek népszerűsítése, mint utazási eszközök. A felsorolt opciók mind lehetséges alternatívák a közlekedés jobbá tételére, de tulajdonságaikból adódóan csak korlátolt hatáskörrel bírnak.

A hosszútávú és megbízható megoldást a forgalmi stratégiák és a forgalomirányító rendszerek (ITS - Intelligent Transportation Systems) optimalizálása jelenti. A modern technológiáknak köszönhetően manapság már mindenhol találhatóak elektromos eszközök, amiket adatforrásként felhasználhatunk. Kitelepített szenzorok és kamerák, mobilkészülékek, de akár már a jármű is szolgálhat adatforrásként. Mérhető a járművek sebessége, a forgalom sűrűsége, az utak kihasználtsága, az időjárási viszonyok és még számtalan más paraméter is.

A begyűjtött adatok feldolgozásán és elemzésén keresztül feltárhatóak és kezelhetőek a közlekedési rendszerek gyengeségei. Ha képesek vagyunk megbízható módon felismerni a torlódott állapotokat, visszakereshetjük azok forrását, terjedését és előre jelezhetünk jövőbeli állapotokat. A felsorolt képességek lehetővé teszik az ITS rendszerek működésének optimalizálását.

A dolgozat négy különböző gépi tanulási modellt (döntési fa, logisztikus regresszió, Support Vector Machine, XGBoost) alkalmaz egy saját szimuláció által generált, forgalmi anomáliákat tartalmazó adathalmazon. Az elkészült modelleket a klasszifikáció területén elfogadott metrikák alapján kiértékeli, majd összegzi az eredményeket. A cél az, hogy megállapítsa az összehasonlított modellek alkalmasságát torlódás felismerése épülő feladatok tekintetében.

Abstract

Due to the continuous expansion and development of cities, along with the boundless mobility demands accompanying this phenomenon, effective management of transportation systems is of paramount importance. In poorly managed traffic infrastructures, where congestion is frequent, daily delays are commonplace, leading to various issues in people's lives, such as time loss, stress, and increased risk of accidents.

Improving the quality of transportation can be achieved through infrastructure enhancements (introduction of new traffic lanes, roundabouts). Further improvement can be attained by reducing the number of vehicles on the roads or altering the composition of traffic, for example, promoting public transportation, bicycles, or electric scooters as means of travel. The listed options are all potential alternatives to enhance transportation, but due to their inherent characteristics, they have limited scopes of impact.

The long-term and reliable solution is the optimisation of traffic strategies and traffic management systems (ITS - Intelligent Transportation Systems). Thanks to modern technologies, electric devices are now everywhere and can be used as a source of data. Sensors and cameras, mobile devices and even vehicles can be used as data sources. Vehicle speed, traffic density, occupancy, weather conditions and many other parameters can be measured.

Through the processing and analysis of the data collected, weaknesses in transport systems can be identified and addressed. By being able to reliably identify congestion, we can trace its source and spread and predict future conditions. These capabilities will enable the optimisation of ITS systems.

The thesis applies four different machine learning models (decision tree, logistic regression, Support Vector Machine, XGBoost) to a dataset containing traffic anomalies generated by a simulation. The developed models are evaluated based on popular metrics in the classification domain, and the results are summarized. The objective is to determine whether the compared models are suitable for solving congestion detection based tasks.

1 Bevezetés

A folyamatos piaci és munkahelyi verseny okozta megfelelési kényszer, a modern technológia nyújtotta fizikai korlátok nélküli lehetőségek és az egyre újabbat és többet igénylő fogyasztói igények a napjainkra jellemző rohanó életstílus kialakulásához vezettek, melyben a legfontosabb értékek közé tartozik az idő hatékony kihasználása. A gépjárművek által nyújtott gyors, kényelmes és rugalmas közlekedés lehetőségei ugyan ideálisan illeszkednek ehhez a modern életstílushoz, ugyanakkor népszerűsége egyben a legnagyobb hátránya is, hiszen az utak fokozatosan túltelítetté válnak. A nagyvárosok esetén ezek a hatások sokszorozottan érvényesülnek, ezért ezeken a településeken kiemelten fontos a forgalmi infrastruktúra és a közlekedés megfelelő minőségű menedzselése.

A közlekedés hatékonyságának javítása érdekében többféle megközelítés is létezik, melyek együttes alkalmazása kulcsfontosságú az optimális működés elérése érdekében. Az egyik lehetőség a közlekedési infrastruktúra javítása, ami jelentheti például a kiemelten fontos kereszteződések fejlesztését vagy új forgalmi sávok bevezetését az utak kapacitásának növelése érdekében. A hatékony városi közlekedéshez elengedhetetlen a gyors és megbízhatóan működő tömegközlekedési rendszer kiépítése, ami csökkentheti a személygépjárművek számát az utakon, javítva azok kihasználtságát. A zsúfoltság csökkentésére további lehetőséget jelenthet a bicikliutak bevezetése és népszerűsítése. A közlekedők kerékpározásra ösztönzése nemcsak a környezetbarát szemléletet erősíti, hanem segít tehermentesíteni a közlekedési dugókat is.

A másik szemlélet az okos városok által biztosított, modern adatgyűjtési és elemzési technológiákon alapszik. Az utak mentén elhelyezett szenzorok és kamerák segítségével folyamatosan, valós időben gyűjthetünk hatalmas mennyiségű információt a közlekedés állapotáról. Ilyen adattípusnak számít a járművek sebessége, a közlekedés sűrűsége vagy például az időjárási viszonyok. Ezen tulajdonságok összességéből képezett adathalmaz feldolgozásán és elemzésén keresztül mélyebben megismerhetjük az adott hálózatra jellemző tulajdonságokat és forgalmi mintákat, ami lehetővé teszi a forgalmi torlódások felismerését, valamint kialakulásának és terjedésének előrejelzését. Az információk megbízható ismeretében képesek lehetünk hatékonyabb közlekedési stratégiákat kidolgozni és implementálni, optimalizálni az intelligens forgalomirányító

rendszereket, illetve alternatív útvonalakat javasolni a közlekedők számára dinamikus módon.

A fentebb említett módszerek közül néhány alkalmasnak vélt modellt alkalmazok és tesztelek, egy általam készített forgalmi anomáliákat tartalmazó szimulációból generált adathalmazon. A kapott eredményeket összevetve és kiértékelve célom egy olyan módszer keresése, amely megbízható pontossággal és hatékonysággal képes torlódás detekcióra, anomáliától és forgalmi helyzettől függetlenül, így egy kiváló alapot biztosítva a torlódás predikció és a forgalom optimalizálás folyamataihoz.

2 Motiváció

A forgalmi torlódások jelenlétének számos negatív következménye van, társadalmi és egyéni szinten egyaránt. Torlódások esetén a megszokottnál jelentősen alacsonyabb közlekedési sebességet, hosszabb utazási és sorban állási időket, gyakori fékezési kényszert és rövidebb követési távolságot tapasztalhatunk. Ezen tényezőkből következően a legegyértelműbb probléma az időveszteség. Azt az időt, amit a dugóban várakozva elvesztegetünk, tölthetnénk a családunk társaságában, pihenéssel vagy akár produktív munkatevékenységgel. Az idő elvesztegetése továbbá stresszt és türelmetlenséget válthat ki az egyénből, ami azon kívül, hogy negatív egészségügyi következményekkel járhat, növeli a balesetek bekövetkezésének valószínűségét.

Gazdasági szempontból nézve a hosszabb utazással töltött idő több üzemanyagfelhasználással jár, ami plusz költséget jelent. A késések, amellett, hogy plusz idővel járnak a közlekedő számára további közvetett idő és pénzügyi kieséssel járhatnak más személyek számára is, gondoljunk itt például egy üzleti találkozóra. A már említett stressz és bizonytalanság elégedetlenséghez vezet, ami további negatív hatást gyakorolhat az egyén munka és produktivitási moráljára. Végezetül meg kell említeni a környezetvédelmi szempontokat is, hiszen a lassú közlekedés miatt elkerülhetetlenül több káros anyag kerül a levegőbe, további negatív egészségügyi hatásokat generálva az ember és környezete számára egyaránt.

A modernkorra jellemző digitális lefedettség rengeteg különálló adatforrás jelenlétét biztosítja. Minden olyan digitális eszköz, ami alkalmas hálózati kapcsolat létesítésére és adatok továbbítására, értékes információt nyújthat. A legjelentősebb adatforrások, a forgalom- és térfigyelő kamerák és az útmenti, illetve járműveken található szenzorok. Az autókon számos szenzortípus található, melyek könnyebbé és biztonságosabbá teszik a mindennapi közlekedést, például a kerék blokkolását monitorozó érzékelők, amik a blokkolásgátló működéséhez szükségesek, esőszenzor az ablaktörlő automatikus törléséhez vagy az autón található képfelismerő érzékelők, amik a sávtartás, illetve parkolás asszisztens funkcióit teszik lehetővé.

További adatgyűjtési és -terítési lehetőség a V2X (Vehicle-to-Everything) néven ismert modern technológiacsoport alkalmazása, amely a korábban felsorolt negatív következmények csökkentésére és a biztonságosabb, gazdaságosabb közlekedés

elősegítésére irányul [1]. A V2X kifejezés magába foglalja a járművek minden más elemmel történő kommunikációját, beleértve a hálózatot, gyalogosokat, infrastruktúrát és más járműveket is. Segítségével valós időben (milliszekundumos késleltetéssel) gyűjthető össze a jármű környezete által biztosított széles információhalmaz. Információt kaphatunk például olyan kereszteződések állapotáról, amire sem a sofőrnek, sem az érzékelő radaroknak nincsen rálátása, ezáltal lehetőségünk van, hogy a megfelelő döntést hozzuk meg. A technológia elengedhetetlen részét képezi az okos városokhoz tartozó innovációknak és fejlődésével újabb és újabb fejlesztési lehetőségek nyílnak meg, közelítve az önvezető járművek realitását.

A V2X szerteágazó funkcionalitásánál egy jóval egyszerűbb, ugyanakkor már évek óta működő megoldásnak számítanak a közösségi érzékelésen alapuló alkalmazások. Ebbe a kategóriába tartozik a Waze, mely hazánkban is népszerű és több százezres felhasználóbázissal rendelkezik [2]. Működése a felhasználói tapasztalatokon alapszik, ezért lehetőséget biztosít az utakon észlelt torlódások, balesetek, útlezárások, kátyúk, hatósági sebességmérő műszerek és veszélyhelyzetek rögzítésére, ezáltal információt biztosíthatunk más járművezetők számára az optimális közlekedési viselkedésről.

Az előző fejezetek által bemutatott módon, tehát a forgalmi torlódások negatív hatásainak kezelésére és megelőzésére számos lehetőség létezik. Dolgozatom keretein belül, a torlódás detekció témakörét vizsgálom részletesen, különböző módszerek alkalmazásán és összehasonlításán keresztül. Olyan gépi tanulási modelleket választottam, melyek a szakirodalom szerint megfelelően alkalmazhatóak a kitűzött feladat szerkezetéhez és komplexitásához, így alkalmasak lehetnek a vizsgálni kívánt szempontok összehasonlításához.

3 Szakirodalmi áttekintés

Az utóbbi években megnövekedett érdeklődés, a közlekedési dugók okozta negatív hatású állapotok javítása érdekében, számos forgalmi torlódás detektálásával foglalkozó tanulmány és módszer kidolgozásához vezetett. A detekciós módszerek fejlődésének köszönhetően megbízhatóan végezhető torlódás predikció, akár már valós időben is [3].

T. T. Sigurdsson a Stockholmban található útmenti szenzorok hálózatát egy irányított súlyozott gráfként értelmezte, melyben a csomópontok szenzorok, az élek pedig az azokat összekötő útszakaszok, ezáltal a torlódás detekciót egy adatfolyam-gráf feldolgozásának problémakörére vetítve [4]. A szenzorok által gyűjtött járműszám és sebesség adatokon több algoritmust is alkalmazott, melyek közül a legsikeresebbnek, ≈94%-os pontossággal (*accuracy*), a Congested Components algoritmus bizonyult. Az algoritmus célja felismerni a nem torlódott útszakaszokat, majd eltávolítani a hozzájuk tartozó éleket a gráfból, így a megmaradó élek a torlódott szakaszok láncolatát fogják ábrázolni. További jó teljesítményt nyújtott a DenGraph, ami egy sűrűség alapú közösségi érzékelési (Community Detection) algoritmus, ≈ 87.5%-os pontosságot (*accuracy*) érve el [5].

A V. Simon és M. Bawaneh által kidolgozott statisztikai módszereken alapuló Traffic Congestion Detection (TCD) algoritmus először választ egy forgalmi paramétert (pl.: sebesség, telítettség), kiszűri az ahhoz tartozó zajokat, majd a minta első deriváltjának segítségével, anomália jelenlétének valószínűsége alapján klasszifikálja azt [6]. A derivált számítás azért szükséges, mert egy idősor első deriváltjának kiugró mértékű ingadozása, a sorozat értékei közötti jelentős változást jelenti, ami anomália jelenlétét feltételezheti. A megoldás, amellett, hogy eredményei alapján felveszi a versenyt más módszerek teljesítményével, egyszerű, számítási komplexitásának köszönhetően kiválóan alkalmas valós idejű felhasználásra is. A tanulmány emellett egy másik, különböző típusú forgalmi paramétereket is felhasználó módszerről, Ensemble Based Traffic Congestion Detection (EB-TCD), is értekezik, amely a különböző forgalmi paraméterekből önállóan kinyert következtetéseket megfelelően súlyozva összegzi, és csak azután klasszifikál. Az EB-TCD pontosságát tekintve ugyan valamivel gyengébben

teljesített a TCD-hez viszonyítva, viszont képes volt felismerni a forgalmi torlódásokat még mielőtt azok kialakultak volna.

A forgalmi torlódások és anomáliák detektálására kiváló eszközt nyújtanak a különféle gépi tanuláson alapuló módszerek. R. Sujatha és társai döntési fa alapú klasszifikációt használtak Dél-Afrikában [7], O. ElSahly és A. Abdelfatah random forest alapján kategorizálták a közlekedési baleseteket kimagasló sikerrel [8]. S. Sharma és társai Support Vector Machine (SVM) módszert alkalmaztak Bengaluru városában készült közlekedési felvételeken, aminek következtében klasszifikálni tudták a forgalom sűrűségét [9].

A deep learning modellek felhasználását illetően, J. Kurniawan és társai a képfeldolgozás témakörében népszerű megoldás, a Convolutional Neural Network (CNN) segítségével vizsgáltak egy 1000 CCTV (Closed-Circuit Television - zártláncú televíziós rendszer) kamera szolgáltatta képfolyamhalmazt és végeztek rajta torlódás klasszifikációt 89.50%-os pontossággal (*accuracy*) [10]. Y. Liu, Z. Cai és H. Dou a shanghaji autópályahálózat torlódottságát vizsgálta, autoencoder neurális háló segítségével, amely egy felügyelet nélküli “önkódoló” modell [11]. A kutatás célja kidolgozni egy, a valós torlódottsági szintet megfelelően reprezentáló indexet, aminek köszönhetően pontosan és időszerűen lehet mérni a torlódásokat.

4 Összehasonlított anomália detekciós modellek bemutatása

Az összehasonlításhoz olyan anomália detekciós modelleket azonosítottam a szakirodalomban, melyek képesek megbízható teljesítményt nyújtani, ugyanakkor könnyen értelmezhetőek és viszonylag alacsony futásidővel rendelkeznek. Az alacsony futási idő kiemelten fontos szempont, ugyanis valós időben kell detektálnunk a torlódásokat, hogy a forgalomirányító hatóságok valós időben be tudjanak avatkozni, megakadályozva ezzel a torlódás kialakulását, illetve a súlyosabbá válását. Ez lehetővé teszi az elosztott módszerek használatát is, a modellek folyamatos újratanítása, értékelése futhat helyi, kisebb számítási kapacitással rendelkező komposenseken is, nem kell mindent központilag futtatni. A felületes gépi tanulási módszerek (*shallow machine learning*) tulajdonságai megfelelnek a felsorolt választási szempontoknak, ezért ebből a csoportból választottam négy különböző modellt és azokat hasonlítottam össze.

4.1 Döntési fa

A döntési fák az egyik legnépszerűbb modellnek számítanak a felügyelt gépi tanulási algoritmusok körében [12]. Legnagyobb előnyei közé tartozik, hogy könnyen kezelhető és értelmezhető, nem igényel nagy mennyiségű adatot vagy számítási erőforrást, valamint klasszifikáció mellett kiválóan alkalmas regressziós problémák megoldására is. A modell működésének értelmezését elősegíti, hogy működése grafikusan, egy fagráfként ábrázolható. A fa gyökér eleme a kiindulási pont, vagyis a teljes adathalmaz, a belső csomópontok döntési elágazások, melyek a megfelelő logika alapján részhalmazokra bontják az adathalmazt, míg végezetül a levelekben található csoportok már az algoritmus végeredményét reprezentálják. Klasszifikációs problémakör esetén a tanulási folyamat megkezdése előtt az adatokat elő kell készíteni úgy, hogy minden mintát megfelelően ellássunk címkékkel a vizsgált tulajdonságaik alapján. A modell célja a tanítás során, hogy hasonló tulajdonsághalmazzal rendelkező részhalmazokat képezzen, ezáltal egy tisztább és homogénebb állapotot létrehozva. A szétválasztás pontossága és hatékonysága több kritérium alapján is vizsgálható [13]. Az egyik megközelítés a Gini impurity, melynek értéke egy 0 és 0.5 közötti szám, ami azt jellemzi, hogy mekkora a valószínűsége, annak, hogy egy véletlenszerűen választott

elemet, tévesen klasszifikál a modell. Egy adott csomópontban, minél kisebb az impurity értéke, annál valószínűbb, hogy a csoportba tartozó minták helyesen vannak osztályozva, vagyis a 0 Gini impurity érték az ideális esetet, azaz a tökéletes klasszifikációt jelenti. A másik tisztaságot vizsgáló mérték az entrópia, melynek értéke az elemek rendezetlenségét jellemzi. Az entrópia értékének nagysága azt mutatja, hogy az elemek mennyire egyenletesen vannak elosztva a területen. A magas fokú elosztottság növeli a bizonytalanságot, ami gyengített döntésképességhez vezet. Összevetve a két módszert, a Gini impurity pontosabb részhalmazokat képez és kevesebb számítási kapacitást igényel, ezzel szemben az entrópia egy kiegyensúlyozottabb adathalmazt alkot. A pontosság maximalizálása érdekében a modell detekciós problémán történő alkalmazása során Gini impurity-t fogok használni. Szétválasztás esetén, az algoritmus tehát sorra veszi a lehetséges "gyerek" csomópontok tisztaságát és elemszámát, majd a kapott értékeket összevetve dönt. Regressziós feladatok esetén a célváltozó értékkészlete nem egy meghatározott halmaz, hanem egy folytonos érték, éppen ezért másfajta kiértékelés szükséges. A regresszió esetén használt metrika, az MSE (átlagos négyzetes hiba), melynek csomópont-beli értéke, az összes hozzátartozó elem célváltozójának átlaga. A tanulás célja természetesen itt is ennek a hibának minimalizálása. A döntési fák egyik problémája, hogy az adathalmazban található zajok miatt hajlamosak lehetnek a túltanulásra (*overfitting*), aminek következtében, ismeretlen adatok esetén gyengébben teljesíthetnek. A probléma elkerülése érdekében alkalmazható nyesés, azaz pruning, ami a fa lényegi információt nem tartalmazó vagy a komplexitást túlzottan növelő részeinek eltávolítását jelenti. Egy másik lehetőség a túltanulás javítására, ha a probléma jellegéhez igazodva leállási kritériumokat alkalmazunk, például meghatározunk egy minimális elem számot a levelekben és a szétbontandó csomópontokban, vagy szabályozzuk a fa maximális mélységét. Berry és Linoff cikkükben a levelek szabályozására a teljes adathalmaz 0.25 és 1.00% közötti részét javasolják, ezzel csökkentve a túltanulás és az alultanulás (*underfitting*) valószínűségét [14].

4.2 Logisztikus regresszió

A logisztikus regressziót bináris vagy többosztályos célváltozó klasszifikációjára használják és a felügyelt gépi tanulási modellek csoportjába tartozik [15]. A célváltozó felépítése szerint 3 részre osztható. Abban az esetben, ha a lehetséges eredményhalmaz kételemű (pl.: igaz-hamis, 0-1), akkor a regresszió bináris, ha pedig három vagy több (pl.: egy személy hajszíne), akkor multinominális. Legalább három kategória esetén, ha a

célváltozók egy adott logika szerint egyértelműen sorrendbe állíthatóak (pl.: iskolai végzettség), akkor ordinális regresszióról van szó. Lineáris regresszió alkalmazása esetén az eredmény értéke folytonos, így a logisztikus megoldással szemben nem alkalmas klasszifikációs problémák megoldására. A logisztikus modell további előnyei közé tartozik, hogy kezelni képes a kategorikus független változókat, valamint rugalmasabban kezeli az adatelemzést torzító kirívó értékeket. A logisztikus függvény, más néven Sigmoid számításának képlete, lásd ***1. egyenlet***.

$$P = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_N X_N)}}$$

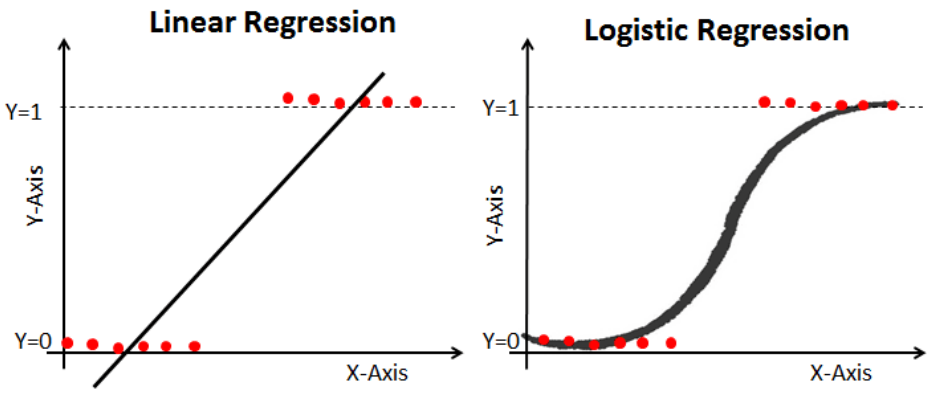
1. egyenlet: Sigmoid függvény

A logisztikus függvény lineáris regresszió egyenletére épül és magába foglalja azt, lásd ***2. egyenlet***.

$$y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_N X_N$$

2. egyenlet: Lineáris regresszió egyenlete

Ahogy az ***1. ábra*** szemlélteti, a függvény 0 és 1 közé szorítja a lineáris regresszió értékkészletét, egy S alakú görbe segítségével.



1. ábra: Sigmoid függvény[1]

A logisztikus függvény nem linearitása miatt a lineáris regressziónál használt költség függvény ebben az esetben alkalmatlan, mert az MSE konkáv alakzatok esetén

[1] Kép forrása: Datacamp: [Online] https://www.datacamp.com/tutorial/understanding-logistic-regression-python [Hozzáférés dátuma: 2023.11.27.]

elakadhat lokális minimumban. A problémára megoldást jelent a Log-loss, ami lényegében “a klasszifikációs problémák MSE-je", hiszen mindkét kiértékelési módszer a predikciók valós értéktől való eltérését vizsgálja. Kiegyensúlyozatlan adathalmaz esetén az eredmények a valós teljesítménytől eltérőek lehetnek, így körültekintően kell kezelni a kapott értékelést.

4.3 Support Vector Machine

Ahogyan az előbbiekben említett modellek, a Support Vector Machine (SVM) is egy felügyelt gépi tanulási modell, amelyet elsősorban klasszifikációra használnak, de alkalmazható regresszió esetén is [16]. Az algoritmus célja, hogy találjon egy hipersíkot, ami két csoportra választja szét az adathalmazt. Hipersík alatt a tér dimenziójánál eggyel kisebb dimenziójú altereket értjük, ami pl.: 2D esetén egy egyenest, 3D esetén pedig egy síkot jelent. Ilyen altérből végtelen létezhet, ezért fontos, hogy olyat keres, ami lehetőleg a két csoport közötti távolság (margó) felénél található. Nagyobb margó esetén a csoportok egyértelműbbek, még akkor is, ha így néhány elem a margók határain belülre vagy a túloldalára kerül. A hipersík meghatározásában a margó szélein található minták (“support vektorok” - innen ered az elnevezése) segítenek. A modell teljes mértékben erre a két vektorra épül, ezért, ha eltávolítjuk őket az adathalmazból, egy új hipersíkot és ezáltal új modellt kapunk. A többi minta nem befolyásolja a modell működését. Mivel tehát a modell alapvetően az adatok közötti távolságra épül, elengedhetetlen, hogy az attribútumok azonos skálán legyenek. Mivel az attribútumok a feladat jellegétől függően igencsak eltérő tartományban mozoghatnak, ezért még a tanítási fázis megkezdése előtt fontos a megfelelő skálázás alkalmazása, megelőzve ezzel az algoritmus torz értékeken tanítását [17]. A két leggyakoribb skálázási módszernek számít a normalizáció, ami 0 és 1 közé szűkíti az értékkészletet és a standardizáció, ami az átlag (μ) és szórás (σ) felhasználásával úgy alakítja át az adathalmazt, hogy a középértéke 0, szórása pedig 1 legyen, lásd **3. egyenlet** és **4. egyenlet**.

$$y = \frac{x - \min(x)}{\max(x) - \min(x)}$$

3. egyenlet: Normalizáció képlete

$$z = \frac{x - \mu}{\sigma}$$

4. egyenlet: Standardizáció képlete

Attól függően, hogy az adatok lineárisan ketté oszthatóak-e vagy sem, a modellt Support Vector Classifier-nek (SVC) vagy Support Vector Machine-nek nevezzük. Amennyiben az adathalmaz lineárisan nem szeparálható, úgynevezett kernel trükk alkalmazható, melynek lényege, hogy transzformálja az adatokat magasabb dimenzióba ezáltal, az eddig rejtett határok mentén lehetővé téve a szétválasztást. A *polinomial, RBF* és a *sigmoid* mind a nem-lineáris kernelek közé tartoznak. Az *SVM* előnyei közé sorolható, tehát, hogy a hatékonyan képes kezelni a nem-lineáris vagy magas dimenziójú adatokat, könnyen általánosítható és kevésbé érzékeny az adathalmazt érő változásokra. Nem alkalmas azonban hatalmas vagy erősen zajos adatbázisok kezelésére, igényli az adatok skálázását és nehéz lehet megtalálni az adott problémához illő ideális hangolási paramétereket.

4.4 XGBoost

A 2014-es bemutatkozása óta az XGBoost, teljes nevén Extreme Gradient Boosting, korunk vezető gépi tanulási modelljének számít regresszió és klasszifikáció terén egyaránt [18]. Népszerűségét a kiváló skálázhatóságnak, pontosságnak, valamint az ezekhez társuló kimagaslóan gyors futtatási sebességnek köszönheti. A modell elosztott felépítése lehetővé teszi a párhuzamos működésű futtatást, ami előnyt jelent más modellek számítási sebességével szemben. A *gradient-boosted decision tree* modellek (*GBDT,* köztük az XGBoost is) *ensemble learning* algoritmusok, azaz többféle modellt (pl.: K-Nearest Neighbors, Support Vector Machine, döntési fa, random forest) használnak, majd súlyozva összegzik azok eredményeit [19]. A részmodellek csökkentett mélységgel és leállási paraméterekkel rendelkeznek, úgynevezett *weak learner-ek* (gyenge osztályozó). Weak learner alatt olyan osztályozókat (classifier) értünk, amelyeknek gyenge a korrelációja a valós értékekkel. Egy weak learner önállóan tekintve nem elég erős, viszont több együttes kombinálásával és a hibáik felismerésével létrehozható egy nagy teljesítményű *strong learner* (erős oszályozó) modell. XGB esetén az alap döntéshozó algoritmus általában a döntési fa. Mivel a döntési fák általában magas varianciával dolgoznak, ezért ennek csökkentésére *bagging* és *boosting ensemble learning* technikákat alkalmaznak. A *bagging* (Bootstrap Aggregating) módszer esetén a részmodellek egymástól függetlenül, véletlenszerűen választott részhalmazokon tanulnak és azok eredményeinek átlaga adja meg a kimenetet. Az XGB, ahogy a neve is jelzi *boosting*-ot alkalmaz, ami azt jelenti, hogy a tévesen klasszifikált minták nagyobb, a helyesen pedig kisebb súlyokat kapnak. Fontos kiemelni, hogy *boosting* során nem az

előző hibák kijavításán, hanem az azok eredményeiből szerzett információ további döntések alkalmával történő felhasználásán van a hangsúly. Az algoritmus végén a jobb teljesítményű részmodellek eredménye nagyobb súllyal fog számítani az eredmények végső aggregálásánál. A *gradient boosting* olyan *boosting* technika, ahol a hibák minimalizálására és súlyok hangolására *gradient descent* (legmeredekebb sülyedés) módszert használnak. A *gradient descent* egy olyan optimalizációs algoritmus, melynek fő célja, hogy megtalálja egy függvény lokális minimumát, ami csökkenti a költségfüggvény értékét. Az XGBoost robosztus, kevésbé érzékeny a zajos adatokra és nem lineáris problémák esetén is jól teljesít.

5 Torlódás detekciós módszerek kiértékelése

Az előző fejezetben bemutatott gépi tanulási modellek hatékonyságának összehasonlításához szükségem volt egy torlódások detektálására alkalmas adathalmazra. Sajnos a való életből származó adatok privát és bizalmas jellegükből adódóan általában nehezen hozzáférhetőek, ezért ilyen típusú adatok nem álltak rendelkezésemre. Szerencsére számos forgalomszimulációs program létezik, hálózatok modellezésére, forgalom szimulálására és a szimulációs adatok begyűjtésére, amelyek kitűnő alternatívát nyújtanak az adathalmaz forrásaként.

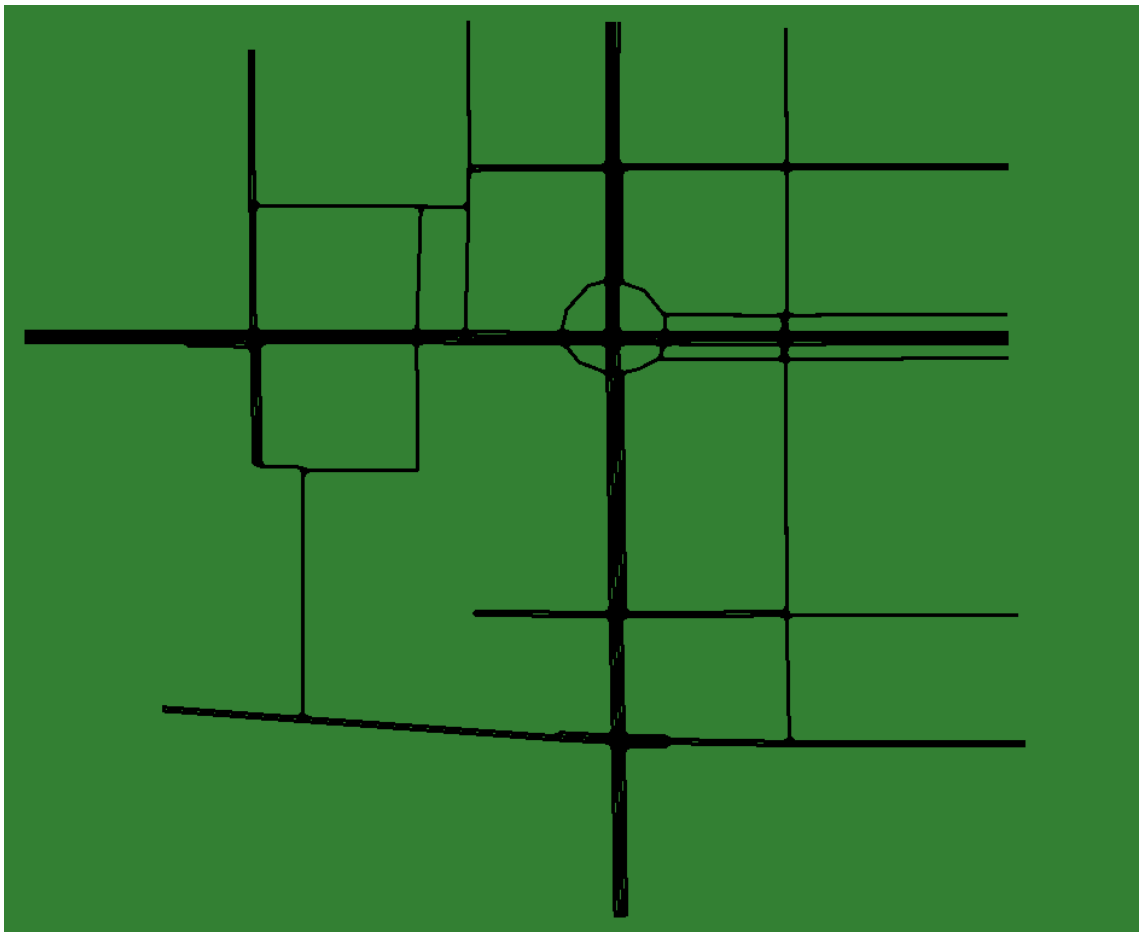
5.1 Szimuláció

Dolgozatomban egy népszerű keretrendszert, a SUMO-t (Simulation of Urban MObility) használtam, ami egy nyílt forráskódú, nagyobb méretű hálózatok kezelésére tervezett, könnyen hordozható városszimulációs csomag [20]. Lehetővé teszi különböző közlekedési és forgalomirányítási stratégiák működésének tesztelését, így a szerzett tapasztalatok hozzájárulhatnak a közlekedési infrastruktúra javításához, valamint a közlekedésirányító rendszerek optimalizálásához egyaránt. A szimuláció teljes egészében mikroszkopikus szinten működik, tehát minden egyes jármű önállóan definiálva van saját útvonallal és tulajdonságokkal. Az önállóságnak köszönhetően a szimuláció valósághűbb és lehetőség van részletesebb elemzésre is. A SUMO alapértelmezett módon tartalmaz különféle járműtípusokat, például személyautókat, teherautókat, sürgősségi járműveket, sőt még vonatokat és hajókat is. A beépített járműtípusok funkciójuknak megfelelően különböző paraméterekkel (maximális sebesség, gyorsulás, jármű forma, szín stb.) és viselkedési mintákkal (követési távolság, hajlamosság sávváltásra, gyorshajtás stb.) rendelkeznek. Ezek a tulajdonságok a felhasználó igényei szerint paraméterezhetőek, de lehetőség van saját típusú közlekedési eszközök létrehozására is. A különböző járművek egyesével is a szimulációhoz adhatóak, de forgalmat generálni legegyszerűbben flow tulajdonságú járműtípusok hozzáadásával generálhatunk. A flow által kibocsátott járművek gyakorisága pontosan beállítható azáltal, hogy meghatározzuk, a járművek számát időegységenként. Ezen kívül lehetőség van véletlenszerű viselkedésre is, ahol a szimuláció minden pillanatában azt határozhatjuk meg, hogy milyen valószínűséggel induljon el egy jármű. Fontos megemlíteni, hogy minden hozzáadott jármű vagy jármű flow számára meg kell adnunk egy egyértelmű úticélt vagy egy előre kijelölt pontos

útvonalat, amit követni tud. A szimulációnak tehát nincs lehetősége arra, hogy spontán módon, úticélokat fogalmazzon meg, viszont úticél vagy érintett szakaszok kijelölése esetén előbb kiszámítja a legrövidebb távolságot, majd azon keresztül vezérli céljához az autókat. Az utakat illetően, tetszőleges számú és formájú útszakaszokból és forgalmi sávból álló hálózat alkotható. A kereszteződésekbe forgalmi lámpák telepíthetőek, melyeknek működése (állapotok váltakozása, időtartama) szintén teljes mértékben szabályozható. A teljesség igénye nélkül további közlekedési elemek a SUMO-ban, STOP tábla, gyalogos-átkelőhely, buszmegálló stb.

Valós adatokhoz ugyan nem volt hozzáférésem, viszont szerettem volna minél valóságközelibbé tenni a szimulációt, ezért a forgalmi hálózatot egy létező területről, a Budapesten található Oktogon térről és környékéről mintáztam. A SUMO csomagban beépített módon található egy olyan python szkriptek összességéből álló program (OSMWebWizard), amely lehetővé teszi létező területek és hozzájuk tartozó különféle járművekből álló forgalom modelljének részletgazdag importálását egyszerű módon. Azért döntöttem mégis a hálózat manuális megépítése mellett, mert jelen esetben fontosabb volt számomra a modell letisztultsága, így nem volt szükségem az OSMWebWizard nyújtotta részletességre, valamint szerettem volna az építés tapasztalata által részletesen is megismerni a hálózatot.

Az infrastruktúrát tehát saját kezűleg, a SUMO által biztosított grafikus hálózat-szerkesztő felület, a *netedit* segítségével építettem meg. Annak érdekében, hogy az Oktogon területének modellje helyesen legyen implementálva, a Google Maps térkép szolgáltatását használtam referenciaként. Egy hálózat építésekor a SUMO környezet saját logikája szerint automatikusan létrehoz és alkalmaz különböző forgalmi szabályokat, mint például bizonyos forgalmi sávok összeköttetésének engedélyezése vagy tiltása. Ezek a generált szabályok természetes módon nem feleltek meg teljes mértékben a valóságnak, ezért manuálisan korrigálnom kellett őket. Hogy melyik útszakaszon milyen kanyarodási, továbbhaladási és egyéb közlekedési szabályok érvényesek, azt a térkép utcakép funkciójának részletes vizsgálata alapján határoztam meg. Az útszakaszok haladási irányát figyelemebe véve, a hálózat 10 darab belépési és 12 kilépési pontot tartalmaz, melyek közül 8 darab kétirányú. A főbb kereszteződésekben (5 helyen) közlekedési lámpák irányítják a forgalmat. A hálózatban található továbbá 2 zsákutca, melyek közül az egyik kilépési és belépési pontként egyaránt szolgál. Az elkészített szimulációs hálózatot a **2. ábra** szemlélteti.

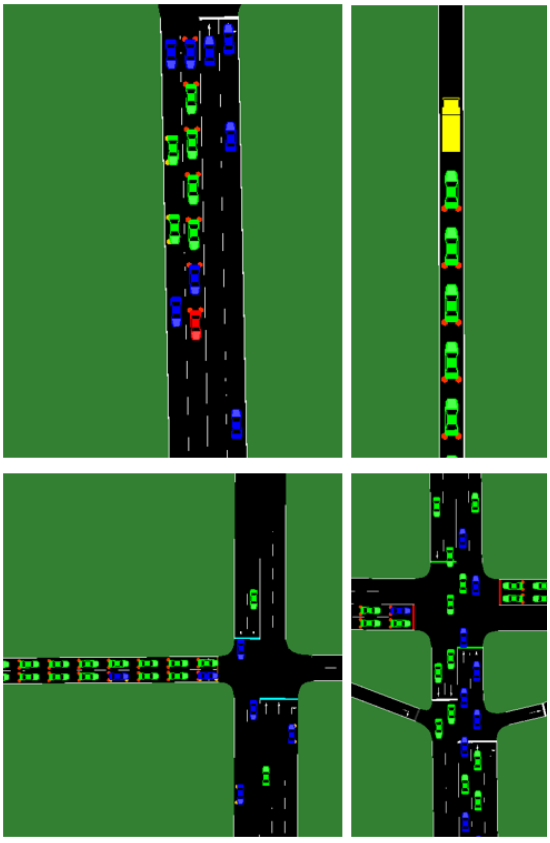


2. ábra: Szimulált úthálózat

A hálózat forgalmát alapvetően átlagosan (zöld jármű) és annál lassabban közlekedő (kék jármű) személygépjárművekből álló *flow* elemek alkották. A különbséget az elérhető maximális sebesség, illetve a gyorsulás mértéke közötti eltérő szabályozás jelentette. Véletlenszerű forgalomra volt szükségem, ezért minden flow eloszlását valószínűség alapján paramétereztem. A főútvonalak esetében a sűrűbb forgalom elérése érdekében nagyobb (~10 %), míg a mellékutcák tekintetében alacsonyabb (~1-3 %) értékeket határoztam meg. A valószínűségeket a szimuláció futtatása során gyűjtött megfigyelések alapján fokozatosan úgy finomhangoltam, hogy az útszakaszok fontosságát figyelembe véve minél több jármű indulhasson, elkerülve a jelentősebb torlódás kialakulását. A célom egy olyan alapforgalom kialakítása volt, ahol a kritikus csomópontokon átfutó forgalom alapos vizsgálata által egy free-flow-szerű állapot tapasztalható, úgy, hogy közben a fő és mellékutakon közlekedő járművek aránya valóságszerű marad. A free-flow fogalmának többféle értelmezése is létezik, jelen esetben a járműfolyam gördülékeny és akadálymentes áramlását jelenti. A normális forgalom mellett definiáltam egy, a csúcsforgalomnak megfelelő állapotot is, ezáltal

lehetőségem volt nagyobb forgalom esetén is tesztelni a hálózatot. A végeláthatatlan torlódott állapotok feloldozása érdekében a SUMO alapértelmezett beállítása szerint, ha egy jármű várakozási ideje meghalad egy adott küszöbértéket, akkor egyszerűen elteleportálja onnan azt. Ezt a viselkedést a valósághű megvalósítás szemléletét figyelembe véve kikapcsoltam.

Mivel a hálózat az előbb említett módon javarészt torlódásmentes, ezért különböző forgalmi anomáliák alkalmazásával mesterséges módon akadályoztam a forgalom szabad áramlását. Négy különböző, a mindennapi életből átvett anomáliát valósítottam meg és alkalmaztam változó körülményekkel, lásd **3. ábra.** A bal felső képen egy lerobbanó jármű látható, amely egy hirtelen fékezést követően adott ideig egyhelyben áll, majd alacsony sebességgel közlekedve elhagyja a szimulációt. A jobb felső ábra a hulladékot gyűjtő kukásautót ábrázolja, amely a kijelölt útszakaszokon hosszuktól függő alkalommal ismételten megáll, majd tovább halad. Az éjjeli időszakot vagy áramszünetet szimuláló kikapcsolt állapotú forgalmi lámpa a bal alsó képen látható. A negyedik eset, amikor az időjárási viszonyok miatt csak jelentősen lassabban közlekedhetnek a járművek, egyedül a járművek paraméterezésénél jelentkezik, vizuálisan nincs eltérés a normális működéshez képest (jobb alsó ábra).



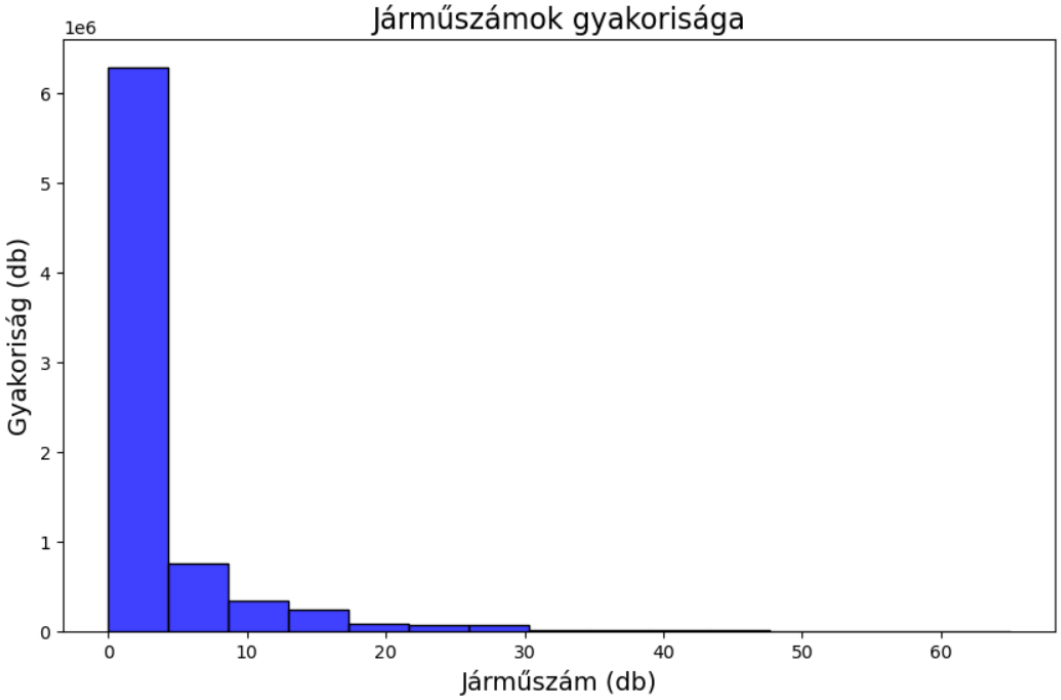
3. ábra: Forgalmi anomália típusok

Az adathalmaz generálásához 24, külön eseteket tartalmazó, egyenként egy-egy órának megfelelő szimulációt készítettem. Vannak anomáliamentes, csak egy vagy két típusú anomáliát tartalmazó és mindet egyszerre alkalmazó esetek is a lehetséges összetételeknek megfelelően. A legtöbb szituációt normális és csúcsforgalommal egyaránt teszteltem. Minden szimuláció 3600 lépésig tartott, ami nagyjából 330000 sornyi adatnak felelt meg. A szimulációk elkészültével aggregáltam az adathalmazt, így összeségében majdnem 8 millió adat állt rendelkezésemre.

Az anomáliákat leíró, illetve felhasználó szkripteket *Python3* segítségével valósítottam meg [21]. A *Python3* egy népszerű felhasználóbarát programozási nyelv, széleskörű támogatottsággal a gépi tanulás és mesterséges intelligenciát illető könyvtárak tekintetében. Az anomáliákat tartalmazó tesztesetek szimulációba integrálására a SUMO dedikált felhasználói interfészét, a *TraCI-t* (Traffic Control Interface) használtam, ami lehetővé teszi a szimuláció valós időben történő irányítását, illetve a forgalmi adatok lekérését. Az adathalmaz könnyű kezelhetősége érdekében a *Pandas* nevű adatmanipulációs könyvtár 1.4.4-es verzióját használtam [22]. A könyvtár címkézhető sorokkal és oszlopokkal dolgozó egydimenziós (*Series*) és többdimenziós (*DataFrame*) adatszerkezeti támogatottságának köszönhetően rendkívül rugalmas és szinte bármilyen feladat kezelésére alkalmas. Számos beépített funkcióval rendelkezik, amik jelentősen egyszerűsítik az adatok feldolgozásának folyamatát, például indexelhetőség, hiányzó adatok kezelése, adatok aggregálása, statisztikai analízis készítése. Az adatok tömbszerű kezelhetőségéhez az 1.23.5-ös verziójú *Numpy* könyvtárat, grafikus megjelenítésükhöz pedig a 3.5.2-es verziójú *Matplotplib* és a 0.11.2-es *Seaborn* könyvtárakat használtam [23][24][25].

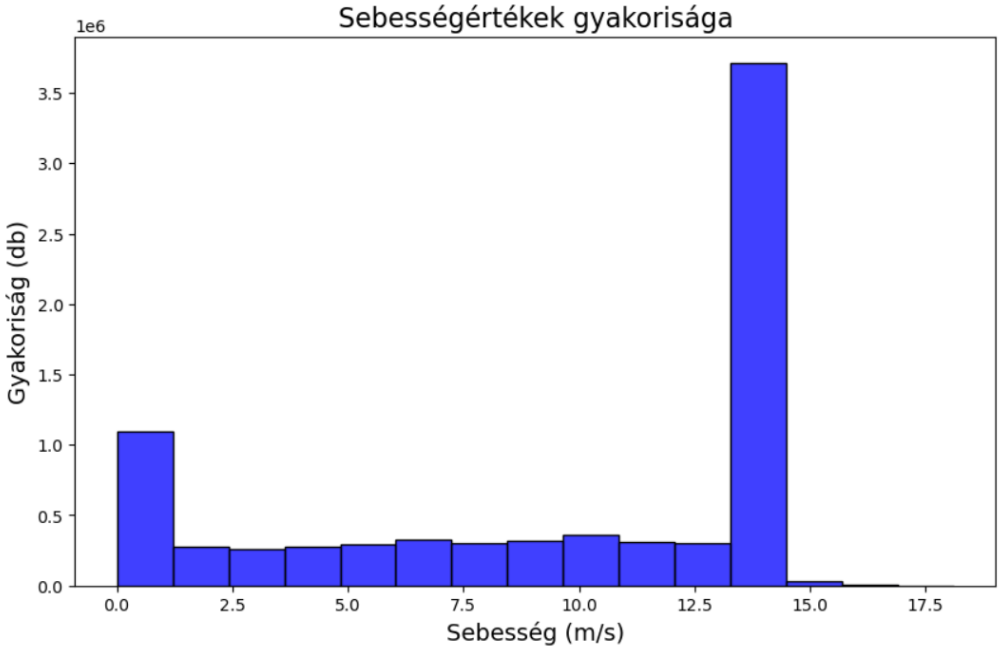
Az elkészült adathalmaz elemzését elsősorban a Pandas *profiling report* funkciójának segítségével végeztem. A *profiling report* automatikusan generál egy összefoglaló jelentést az adathalmazról, ami statisztikai leírást tartalmaz az adattípusok közötti korrelációról, a hiányzó és ismétlődő értékek eloszlásáról és további tulajdonságokról. Az elemzésből kiderült, hogy az összes útszakaszt együttesen vizsgálva, az időpillanatok ~43%-ban üresek, az esetek ~1%-ban pedig 90%-nál magasabb kihasználtságúak az utak. A jelenséget vélhetően a mellékutcák alacsonyabb frekventáltsága okozza. Jelen kontextusban a hálózat készítésekor elhelyezett csomópontokat összekötő területek számítanak útszakaszoknak.

A **4. ábra.** az egyes útszakaszokra jellemző járműszámokat aggregálja és ábrázolja gyakoriságuk függvényében.



4. ábra: Járműszámok gyakorisága

Észrevettem továbbá, hogy abban az esetben, ha egy útszakaszon nem található jármű, az átlagsebesség értékéhez mindig 13.89 m/s-ot (50 km/h) rendel a SUMO, ami megegyezik a sebességkorláttal, lásd **5. ábra**. Ez egy logikus értékválasztás, hiszen, ha nincsen egy másik autó sem az úton akkor elvárható, hogy a közlekedő sebessége a maximálisan megengedett értékhez közeli legyen, így nem változtattam az alapbeállításon.



5. ábra: Sebességértékek gyakorisága

A megtisztított adathalmaz címkézését kétféleképpen, bináris és többosztályos megközelítés alapján egyaránt elvégeztem, ezáltal két különböző klasszifikációs problémakört létrehozva. A New Jersey-i Közlekedési Minisztérium torlódás definíciója alapján, bináris klasszifikáció esetén *torlódott* (congested) osztályba tartoztak a 30%-os telítettségnél nagyobb értékkel rendelkező állapotok (0.3 < *occupancy*), míg a 30%-os vagy annál kisebb értékek a *nem torlódott* osztályba kerültek [26]. Többosztályos osztályozás esetén a *torlódott* (0.3 < *occupancy* <= 0.7) állapot mellett, egy új, *erősen torlódott* (0.7 < occupancy) osztály vezettem be a részletesebb felbontás érdekében. A telítettség a SUMO értelmezésének megfelelően az útszakasz kihasználtságának százalékos értékének felel meg, úgy, hogy a járművek szorosan egymás mögött helyezkednek el és nem hagynak távolságot egymás között.

Az adathalmaz 67%-át (5896800 minta) használtam tanítóadatként és 33%-át (1965600 minta) tesztadatként. Annak érdekében, hogy a pozitív és negatív minták aránya a szétválasztás után is megfeleljen az eredeti adathalmaz arányának, az sklearn beépített paraméterét, a stratify-t használtam.

A modellek implementációját a scikit-learn (1.3.2) python segédkönyvtár segítségével végeztem [27]. Azért ezt a könyvtárat választottam, mert jól dokumentált, használata egyszerű és intuitív, valamint az alkalmazott modellek teljesítményének kiértékeléséhez jól definiált, széleskörű metrikákat biztosít.

5.2 Eredmények

Az átlagsebesség, az utazási idők, a járműszám (*flow*) és forgalomsűrűség (*density*), valamint az útszakaszok telítettsége mind a torlódás detekciós algoritmusok alapjait képező jellemzők [28]. Különböző megoldásokat nyújtó tanulmányok és alkalmazási módok vizsgálják az említett jellemzők valamely összetételének alkalmazhatóságát, viszont nehezen megfogható jellege miatt a torlódás detekcióra nem létezik egy egyértelműen legjobb módszer. Dolgozatomban, a címkézést telítettség (occupancy) alapján végeztem, ezért azt a tanítás folyamán nem lett volna alkalmas felhasználnom, így a felismerési minta alapját átlagsebesség és járműszám adatok képezték. Az előbb említett két fő jellemző mellé hozzávettem az útszakaszokon mért zajszintet is, mint egy viszonylag kevésbé korreláló, de plusz információt hordozó tulajdonságot.

A gépi tanulási modellek kiértékelésére számos módszer létezik [29]. A különböző metrikák alkalmassága a feladat jellegétől függően változó. Más módszereket alkalmaznak regressziós és másokat klasszifikációs problémák esetén. Klasszifikáció terén az egyik legkifejezőbb eszköz a *konfúziós mátrix*. A konfúziós mátrix lényegében egy táblázat, ami kombinálja az előre jelzett és a valós címkéket. Bináris osztályozás esetén a táblázat elemei a következőek **1. táblázat.** True negative (TN), a helyes csoportba sorolt negatív (0) elemek száma. False negative (FN), a helytelenül, vagyis negatívként klasszifikált pozitív (1) elemek száma. False positive (FP), a helytelenül, vagyis pozitívként klasszifikált negatív elemek száma. True positive (TP) a helyes csoportba sorolt pozitív elemek száma.

		Valós címkék	
		0	1
Előrejelzett címkék	0	True Negative (TN)	False Negative (FN)
	1	False Positive (FP)	True Positive (TP)

1. táblázat: Konfúziós mátrix

A konfúziós mátrixban lévő értékek amellett, hogy átláthatóan bemutatják a klasszifikáció eredményeit, lehetővé teszik további metrikák számítását is, amelyek további segítséget nyújtanak a modell teljesítményének részletesebb értékelésében. A

különböző mérőszámok számítási módja eltérő, viszont értékkészletük egységesen egy 0 és 1 közé eső valós szám, ahol az 1-es érték a tökéletes eredményt jelenti.

Az első mutató az *accuracy,* ami a helyesen osztályozott minták és az összes minta hányadosaként számolható, lásd **5. egyenlet** és megmutatja, hogy milyen mértékben predikált helyesen a modell. Kiegyensúlyozott (balanced) adathalmazok esetén egy nagyon hasznos metrikának számít, hiszen a modell ítélőképességét kiválóan jellemzi. Sajnos azonban a valóság gyakran kiegyensúlyozatlan (imbalanced) szituációkat eredményez, amik esetén az accuracy értéke nem megfelelően jellemzi a valóságot. Például, ha egy adathalmaz 90%-a pozitív osztályba tartozik, akkor a tanulási folyamat során túltanul a példákon és ez esetben, ha minden elemet a pozitív csoportba sorolna, akkor 0.9-es eredményt érne el, ami egy ismeretlen adathalmaz esetén jelentősen gyengébb lenne. Mivel az általam szimulált adathalmaznál a pozitív-negatív arány ~15:85, ezért az accuracy értéke az előbb említett példához hasonlóan várhatóan túlságosan magas lesz, ezért a teljesítmény helyes megítéléséhez szükség lesz további szempontok vizsgálatára is.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

5. egyenlet: Accuracy számítási képlete

Egy másik fontos értékelési kritérium, a *precision*, ami megadja, hogy a pozitív osztályba sorolt minták közül milyen arányban voltak az elemek ténylegesen pozitívak, lásd **6. egyenlet.** A metrika segítségével felmérhető, hogy modell pozitív értékeket tekintő predikciói milyen mértékben megbízhatóak. A *precision* értékének vizsgálata kiemelten fontos azon problémakörök esetén, amik jellegükből adódóan érzékenyek a pozitív minták helytelen osztályozására (FP).

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

6. egyenlet: Precision számításának képlete

A következő elemzőeszköz a *recall,* melynek számítási módja és logikája nagyon hasonló a precision értékéhez, azzal a különbséggel, hogy a klasszifikáció által pozitív osztályba sorolt elemek száma helyett a valós pozitív elemek száma kerül az osztóba, lásd **7. egyenlet.** Ennek a változtatásnak köszönhetően a hányados értéke azt mutatja meg, hogy a modell milyen mértékben volt képes felismerni a pozitív mintákat. Mivel a dolgozat

témáját képező összehasonlítás alapja a torlódások felismerésére való képesség, a *recall* értéke kifejezetten fontos lesz számomra.

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

7. egyenlet: Recall számításának képlete

Az *F1-score* az előbb említett metrikákat egyesíti és összehangoltan felhasználja, ezzel kiaknázva azok sajátos gyengeségeit. Az összehangoltságnak köszönhetően kiemelkedően kezeli az adathalmaz kiegyensúlyozatlanságát, ezáltal egy reprezentatív képet nyújt a modell teljesítményéről. Értéke a *precision* és *recall* mutatók harmonikus közepeként számolható, ami a két érték közel azonos fontosságát feltételezi, lásd **8. egyenlet**. Az *F1-score* használata kevésbé alkalmas, ha a felhasznált értékek között jelentős prioritásbeli eltérés van, illetve, ha többosztályos klasszifikáció a feladat.

$$\mathrm{F1} - score \; = \; 2 \; \times \; \frac{Precision \; \times Recall}{Precision \; + Recall}$$

8. egyenlet: F1-score számításának képlete

Összegezve tehát a felsorolt metrikák tulajdonságait, alkalmazási területtől függően mindegyik módszernek léteznek előnyei és hátrányai, viszont együttes alkalmazásuk által egy átfogó és megbízható elemzés alkotható a modell teljesítményéről. Az eredmények kiértékelésére tehát az előbb felsorolt 5 eszközt alkalmaztam.

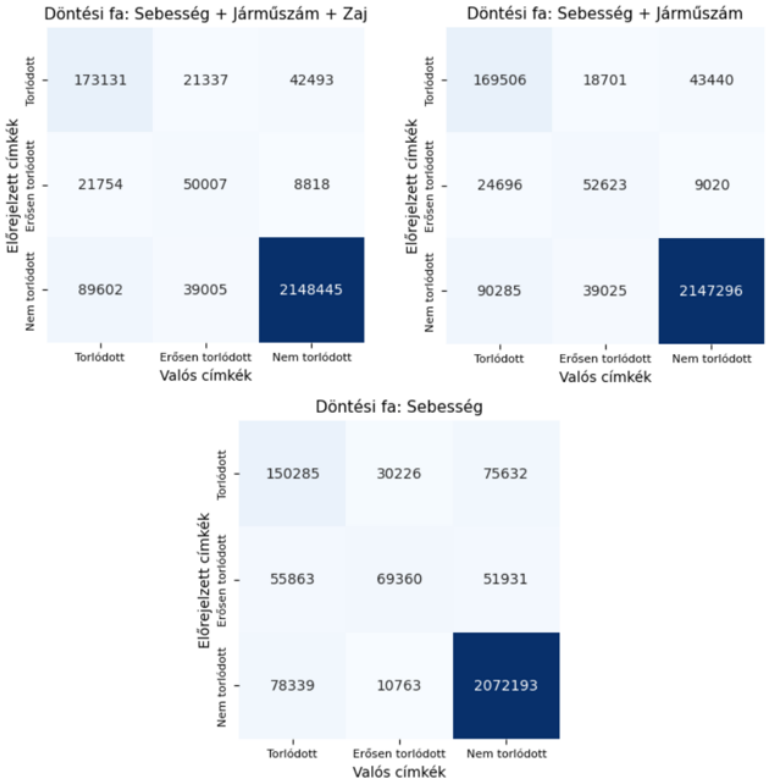
Többosztályos klasszifikáció

A többosztályos klasszifikáció teljesítménye bármelyik paraméterezést tekintve jelentősen elmaradt az általam előzetesen elvárt eredményektől, lásd **2. táblázat.** A **6. ábra** által bemutatott konfúziós mátrixok értelmezéséből kiderül, hogy az erősen torlódott állapotok felismerésében a csak sebességre épülő szcenáriók (1.teszteset) teljesít a legjobban. A sebességet és járműszámot is használó (2.teszteset) és a mindhárom tulajdonságot együttesen alkalmazó (3.teszteset) verziók csak a helyes minták kevesebb mint felét találják meg. Az 1.teszteset esetén több a tévesen torlódottként klasszifikált (FP) állapot, így a *precision* és ennek köszönhetően az F1-score értéke is jelentősen alacsonyabb lesz, mint a több paramétert használó alternatíváknál. A 2. és 3. eset

eredményei minden metrikát tekintve azonosak, így ebben az esetben kijelenthető, hogy a zajszint nem nyújtott elegendő új információt a teljesítmény növeléséhez. A többi gépi tanulási modell esetén is teszteltem az eredményeket, de hasonlóan gyenge mutatókat kaptam. A rossz teljesítmény miatt a továbbiakban elvetettem a több osztályos klasszifikációt és a bináris változatra fókuszáltam.

Döntési fa				
Felhasznált tulajdonságok	*Accuracy*	*Precision*	*Recall*	*F1-score*
Sebesség	0.88	0.65	0.70	0.66
Sebesség + Járműszám	0.91	0.76	0.68	0.72
Sebesség + Járműszám + Zajszint	0.91	0.76	0.68	0.72

2. táblázat: Döntési fa eredményei többosztályos klasszifikáció esetén



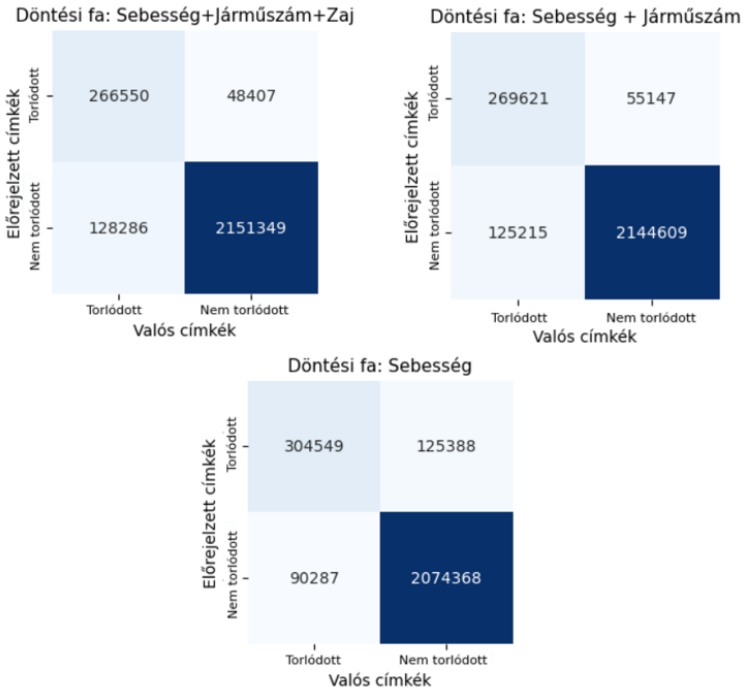
6. ábra: Többosztályos döntési fához tartozó konfúziós mátrixok

Bináris klasszifikáció

Döntési fa esetén a túltanulás jelenségének csökkentése érdekében a 4.1 fejezetben említett módon szabályoztam az egyes levélcsomópontokban lévő minták minimális számát az adathalmaz 1%-nak megfelelően. A szabály bevezetésével korlátozható a fa növekedése és javítható az általánosító képessége, így jobb eredményeket sikerült elérnem. A **3. táblázat** recall értékeit vizsgálva, a többosztályos változatnál szintén felismert tendencia tapasztalható, vagyis a csak sebességre épülő modell több torlódást ismer fel, de többet is hibázik, lásd **7. ábra.** A többi metrikát illetően a több paramétert tartalmazó adathalmazok jobb eredményeket hoztak és itt már a zajszint bevezetése minimális teljesítménybeli javulást jelentett.

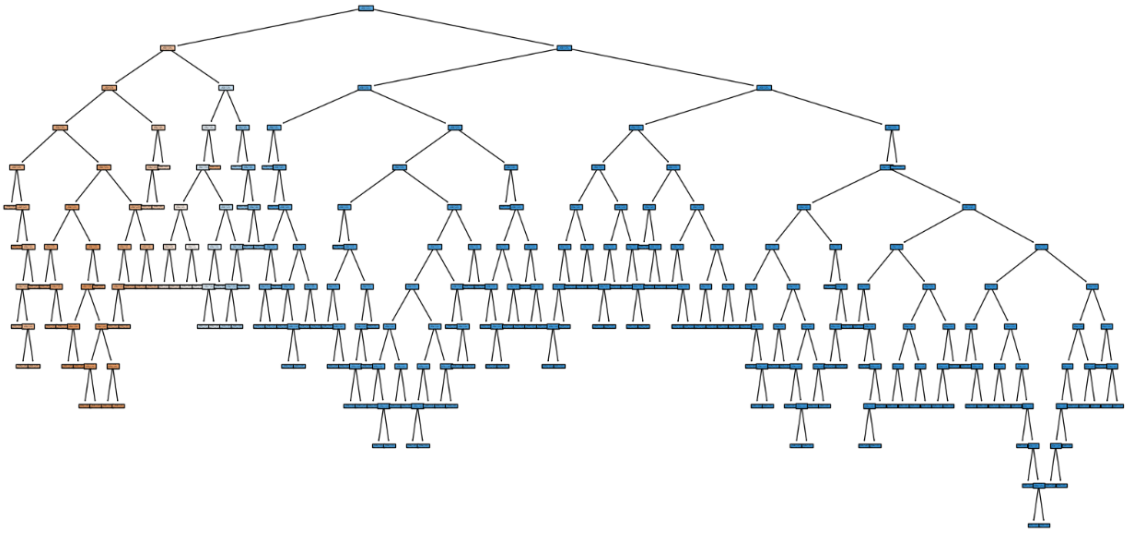
Döntési fa				
Felhasznált tulajdonságok	*Accuracy*	*Precision*	*Recall*	*F1-score*
Sebesség	0.92	0.83	0.86	0.84
Sebesség + Járműszám	0.93	0.89	0.83	0.85
Sebesség + Járműszám + Zajszint	0.93	0.90	0.83	0.86

3. táblázat: Döntési fa eredményei bináris klasszifikáció esetén



7. ábra: Bináris döntési fához tartozó konfúziós mátrixok

A mindhárom tulajdonságot felhasználó modell alkotta döntési fa elágazási struktúráját, a **8. ábra** jeleníti meg. Könnyedén látható, hogy a modell 14 szint mélységig ágazott el, aminek köszönhetően specifikusabb csoportokat hoz létre, ezáltal romlik az általánosító képessége.

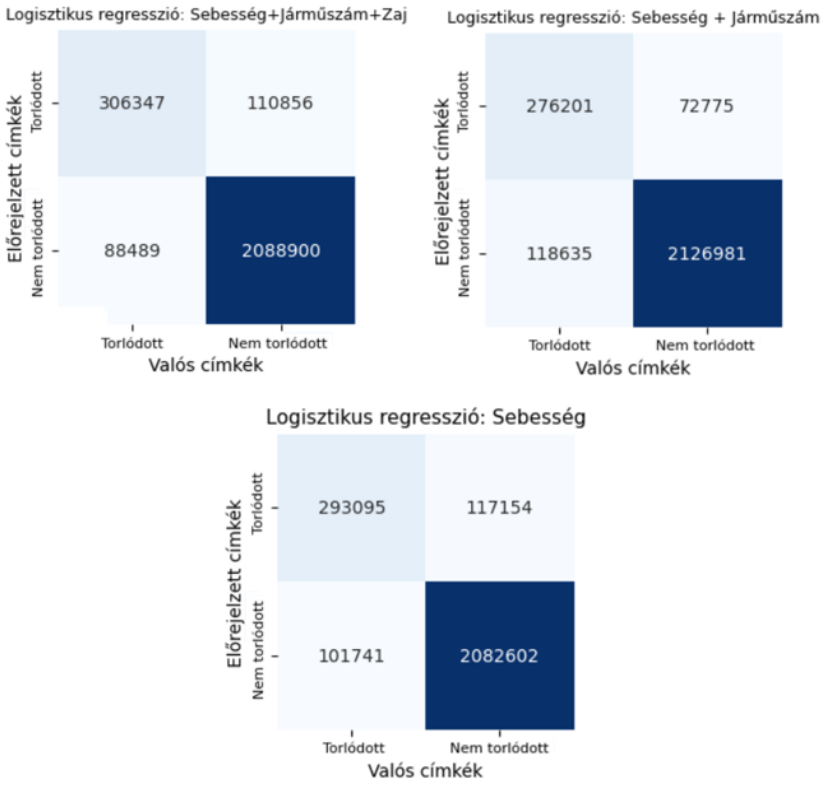


8. ábra: A 3.tesztesethez tartozó döntési fa

Az *accuracy* értékek a döntési fák eredményeihez hasonlóan logisztikus regresszió esetén is kiemelkedően magasak voltak, lásd **4. táblázat.** Ennek magyarázata, ahogyan a metrikák bemutatásánál már említettem, az *accuracy* kiegyensúlyozatlan adathalmazokat érintő gyengeségére, a túltanulás jelenségére vezethető vissza. Az *accuracy* értéke tehát valójában nem mérvadó, ezért érdemesebb a többi metrika eredményét figyelembe venni. A döntési fánál tapasztalt tendenciától eltérő módon, regresszió esetén a 3.tesztesethez tartozó *recall* érték volt a legmagasabb. A **7. ábra** tanulsága, hogy a 2.teszteset lényegesen kevesebb elemet osztályozott torlódottként. Az első esetben mintegy 15%-kal, míg a harmadik esetben mintegy 17%-kal kevesebb mintát sorolt a pozitív osztályba. Bináris klasszifikációból adódóan, ez azt fogja eredményezni, hogy a nem torlódott osztály előrejelzéseinek száma viszont növekedni fog. Több negatív klasszifikáció pontosabb klasszifikációt fog jelenteni, hiszen a valós adatok túlnyomó része a negatív osztályba tartozik. A megfigyeléseket alátámasztják a **4. táblázat** 2.sorában található magasabb *accuracy* és alacsonyabb *recall* értékek. Mivel a 3.teszteset accuracy és recall tekintetében is alulteljesített a 2. tesztesettel szemben, kijelenthető, hogy logisztikus regresszió tanítása esetén a zajszint adatok felhasználása rontott a modell teljesítményén.

Logisztikus regresszió				
Felhasznált tulajdonságok	*Accuracy*	*Precision*	*Recall*	*F1-score*
Sebesség	0.92	0.83	0.84	0.84
Sebesség + Járműszám	0.93	0.87	0.83	0.85
Sebesség + Járműszám + Zajszint	0.92	0.85	0.86	0.85

4. táblázat: Logisztikus regresszió eredményei



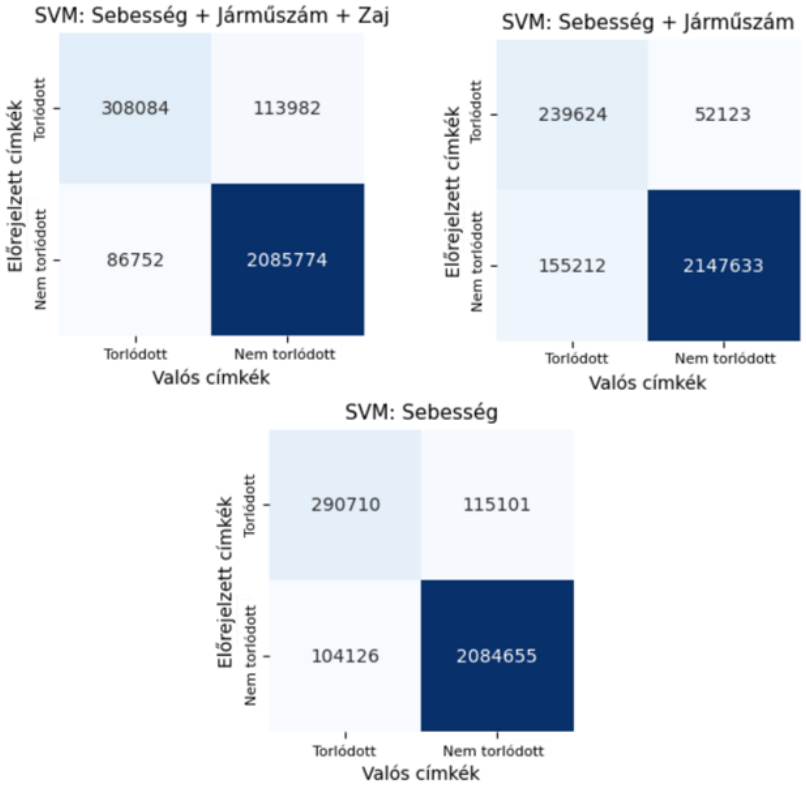
7. ábra: Logisztikus regresszióhoz tartozó konfúziós mátrixok

Support Vector Machine használata előtt az adatok skálázása szükséges, ezért az *sklearn* csomag *StandardScaler* metódusával standardizáltam az adathalmazt, ezzel biztosítva a hatékony működést. Sorra véve az eddig vizsgált modellek eredményeit, az **5. táblázat** elemei között figyelhető meg a legnagyobb szórás. Az *accuracy* értékek változatlanul magasak és emellett jelen esetben meg is egyeznek. A többi metrikát tekintve az 1. teszteset mindenhol közepesen teljesít. A 2. eset egyszerre produkált kirívóan jó *precision* és rossz *recall* mutatókat. A 3. teszteset a *precision* kivételével mindenben a legjobban teljesített, így egyértelmű, hogy SVM alkalmazása esetén a

zajszint felhasználása javít a modell teljesítményén. Az eredményeket kiegészítő konfúziós mátrixokat a **8. ábra** szemlélteti**.**

Support Vector Machine				
Felhasznált tulajdonságok	*Accuracy*	*Precision*	*Recall*	*F1-score*
Sebesség	0.92	0.83	0.84	0.84
Sebesség + Járműszám	0.92	0.88	0.79	0.83
Sebesség + Járműszám + Zajszint	0.92	0.85	0.86	0.85

5. táblázat: SVM eredményei

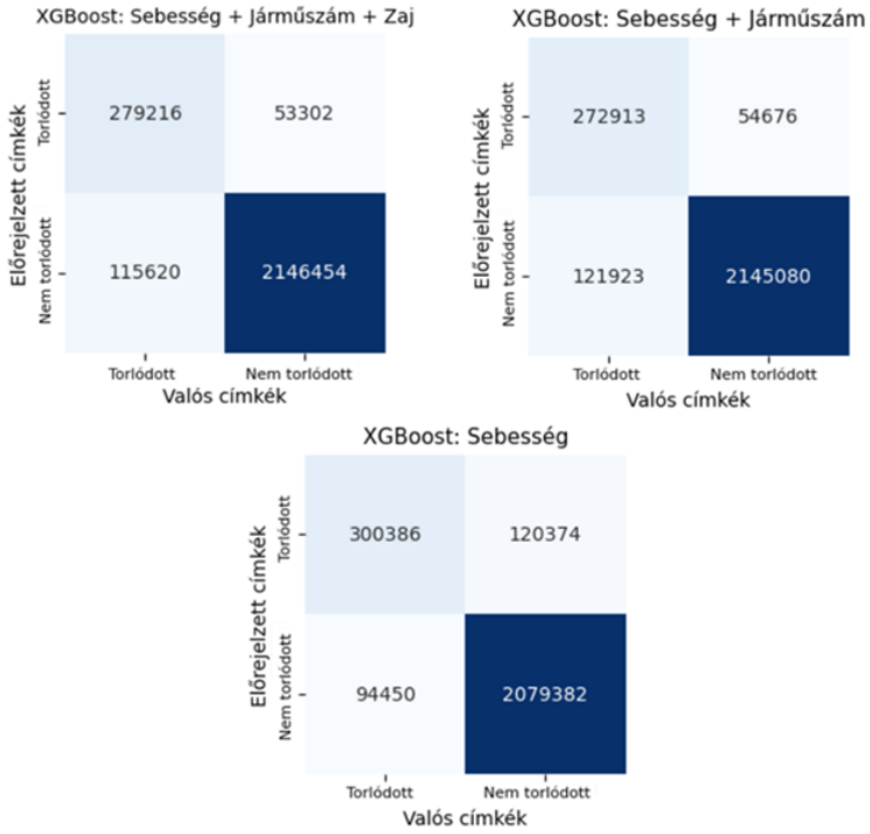


8. ábra: SVM-hez tartozó konfúziós mátrixok.

A **6. táblázat** értékei a döntési fa eredményeihez (3. táblázat) hasonlóan viszonyulnak egymáshoz. Az 1.teszteset eredményei a recall-tól eltekintve valamivel gyengébbek a több elemből álló változatokhoz képest. A 2. és 3. tesztesetek mind értékelési szempontok, mind eloszlás tekintetében (**9. ábra)** közel megegyező eredményeket mutattak. A konfúziós mátrixok összehasonlításából kiderül, hogy mindössze pár ezer minta eltérő osztályozása jelentette a különbséget.

XGBoost				
Felhasznált tulajdonságok	*Accuracy*	*Precision*	*Recall*	*F1-score*
Sebesség	0.92	0.84	0.85	0.84
Sebesség + Járműszám	0.93	0.89	0.83	0.86
Sebesség + Járműszám + Zajszint	0.93	0.89	0.84	0.86

6. táblázat: XGBoost eredményei

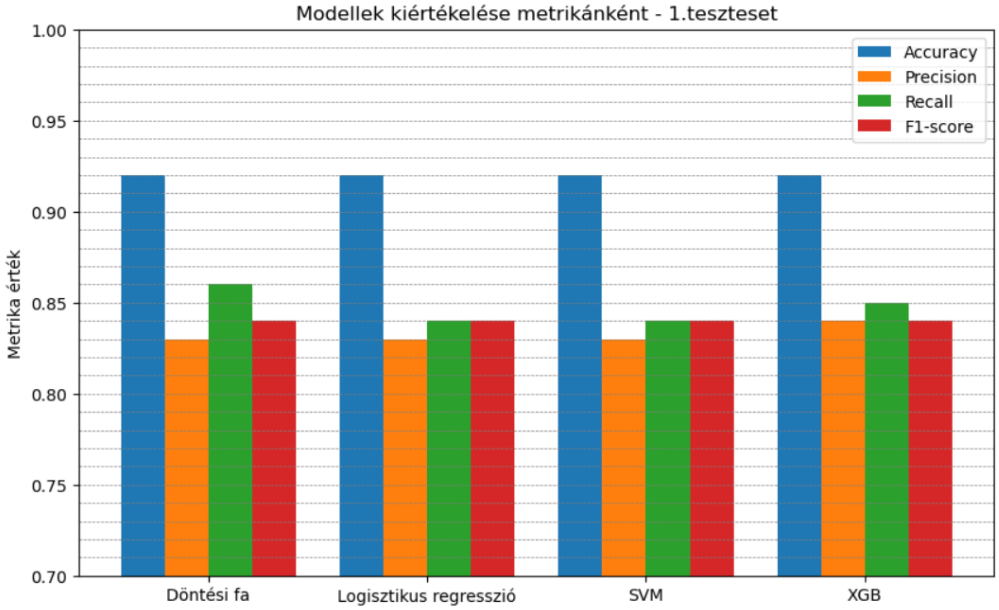


9. ábra: XGBoost-hoz tartozó konfúziós mátrixok

Minden modell esetén a 3. tesztesetet vizsgálva lemértem a tanítási fázis időtartamát. A döntési fa és logisztikus regresszió modellek voltak a leggyorsabbak, 18 és 20 másodperces tanulási idővel. Az XGB tanításra nagyjából ezen értékek ötszörösének megfelelő időt, vagyis 1 perc 50 másodpercet használt fel, míg az SVM-nek több, mint 40-szer ennyi időre, azaz 12 perc 25 másodpercre volt szüksége.

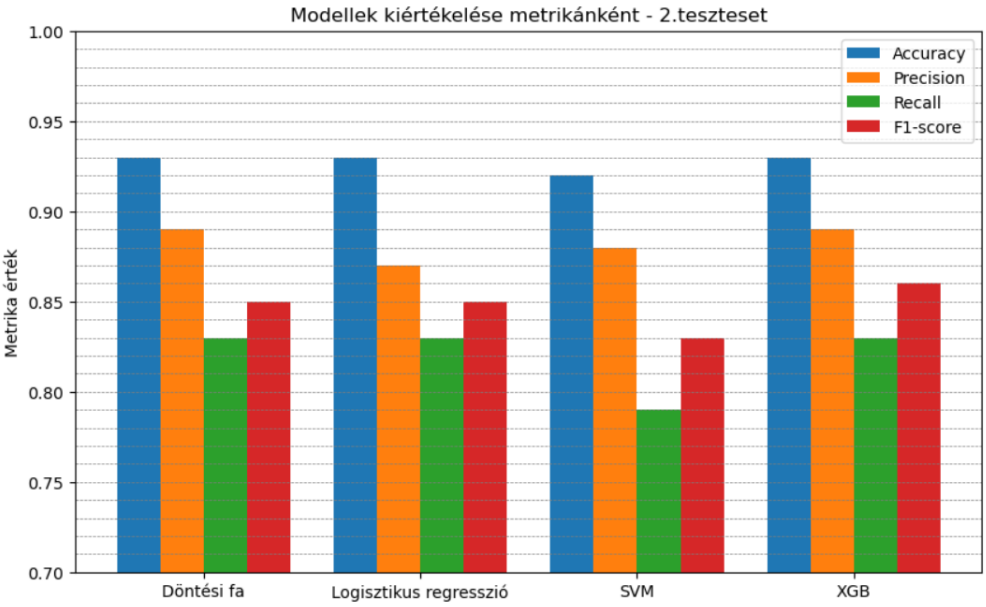
A **10. ábra** összehasonlítja a vizsgált metrikák eredményeit a sebességet, mint egyedüli paramétert használó teszteset esetén. A különböző modellekhez tartozó grafikon

csoportok első pillantásra, akár teljesen azonosnak is tűnhetnek, hiszen nagyon sok átfedés található köztük. A logisztikus regresszió és az SVM eredményei minden tekintetben megegyeznek, míg a döntési fa és XGB értékei egyedül a *precision* és *recall* mutatókban különböznek. Az *accuracy* és az *F1-score* értéke mind a 4 modell esetén azonos volt. *Precision* tekintetében az XGB 0.1-gyel jobban teljesített, mint a többi modell, míg recall tekintetében hasonló különbséggel a döntési fa bizonyult a legjobbnak. A szituáció győztes modelljei egyértelműen a döntési fa és az XGB, melyek között különbséget tenni egyedül a precision és recall értékek közötti eltérő priorizáció tehet.



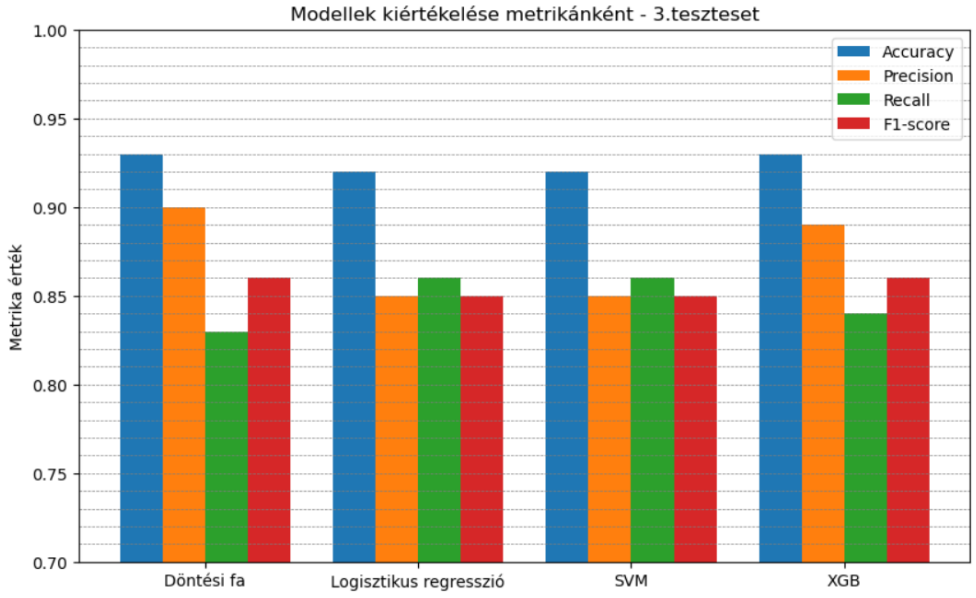
10. ábra: Összegzés az 1.teszteset eredményeiről

A járműszámon is tanuló tesztesetek eredményei nagyobb varianciával dolgoztak, mint a csak sebességre épülő megoldások, lásd **11. ábra**. Ebben az esetben minden modell eredményhalmaza egyedi volt és legalább egy metrika tekintetében eltért a többitől. Az SVM eredményei a többi modellhez képest szembetűnően gyengébbek, egyedül a precisiont vizsgálva teljesít jobban regressziós társánál. A döntési fa és az XGB között ezúttal is sok volt az egyezés, viszont az XGB F1-score értéke egy fokkal jobbnak bizonyult, így a 2. szcenárió legjobb modellje az XGB volt.



11. ábra: Összegzés a 2.teszteset eredményeiről

A sebesség, járműszám és zajszintre együttesen építő verzió eredményei egymáshoz viszonyítva különösen hasonlítanak az 1. tesztesetnél tapasztalt kapcsolatokra. A logisztikus regresszió és az SVM eredményei identikusak, a döntési fa és az XGB között pedig ugyancsak egyedül a precision és recall kritériumokat érintő apró különbségek vannak, így összeségében az F1-score értékük szintén megegyezik.



12. ábra: Összegzés a 3.teszteset eredményeiről

6 Összegzés

A szakdolgozatom elején ismertettem a napjaink közlekedésére jellemző problémákat és azok következményeit. A továbbiakban bemutattam lehetséges intézkedéseket, amik által csökkenthető a közlekedési infrastruktúrára nehezedő több millió autó terhe. Bevezettem továbbá az intelligens rendszerek és adatgyűjtő szenzorok lehetőségein alapuló rendszerek modern, jövőbe mutató fejlesztéseit. A képességet, hogy felismerjük a torlódások létrejöttét, terjedését, kialakulásának okát, helyét vagy visszatérő mintázatait.

Mivel valós adatbázishoz nem volt hozzáférésem, szimulációs környezet segítségével építettem egy saját úthálózatot és összegyűjtöttem a rajta generált forgalom adatait. Annak érdekében, hogy az adathalmaz torlódás detekcióra alkalmas legyen, vagyis biztosan tartalmazzon megfelelő mennyiségű és minőségű torlódott állapotot, forgalmi anomália szcenáriókat hoztam létre és alkalmaztam a szimuláción. Vizsgáltam az utak kihasználtságát és a tartózkodó járművek számát, sebességét és az általuk kibocsátott zaj erősségét.

Az összegyűjtött adatokat feldolgoztam, telítettség értékek alapján felcímkéztem *torlódott* és *nem torlódott* állapotokra, majd átadtam a modelleknek tanításra. Négy különböző gépi tanulási módszert alkalmaztam, melyek a döntési fa, a logisztikus regresszió, a Support Vector Machine és az XGBoost voltak. Mindegyik modellt három különböző tulajdonsághalmazzal teszteltem. A tesztesetek a sebesség, sebesség és járműszám, valamint a sebesség, járműszám és zajszint együttes alkalmazása voltak. A modellek teljesítményének kiértékelésére a klasszifikációs problémák esetén használt, konfúziós mátrixból könnyedén kinyerhető metrikákat használtam (*accuracy, precision, recall, F1-score*).

Végezetül az eredményeket összegeztem és a megérthetőség és átláthatóság egyszerűsítése érdekében vizuálisan is megjelenítettem. Az összefoglalóból kiderült, hogy a logisztikus regresszió kivételével minden modell esetén a zajszintet is tartalmazó paraméterezés teljesített a legjobban. A zajszintet nem ismerő tesztesetre az említett regressziós esettől eltekintve leginkább, mint a 3.teszteset egy csökkentett verziójára lehet tekinteni, hiszen az értékeik nagyon hasonlóak, de az esetek nagyobb részében minimálisan mégis elmarad a másik értékétől. A csak sebesség alapján tanuló eset

általában *recall* tekintetében jól teljesített, a többi mutató terén viszont csak közepesen. Mivel a generált adathalmaz nem volt kiegyensúlyozott, az accuracy mutatók értékei minden esetben magasak voltak és kevésbé hordoztak magukban értékes információt. Többosztályos klasszifikációval is lefuttattam a modelleket, de gyengébb eredményeket, hozott a bináris változatnál, ezért nem folytattam azt az irányt. Meglepő módon a logisztikus regresszió és a SVM eredményei, az 1. és 3. tesztesetnél teljesen megegyeztek. A döntési fa és az XGB eredményei is általánosan nagyon hasonlóak voltak, ugyanakkor emellett kimagaslóan teljesítettek a modellek.

Összegezve a leírtakat, először is zajszint vizsgálata javulást hozott a modellek teljesítményét illetően. A feladatot legjobban teljesítő modellnek az XGBoost bizonyult, amit szorosan követ a döntési fa. A logisztikus regresszió többnyire jó eredményeket ért el és gyorsan is futott a tanítási fázis. A Support Vector Machine eredményei lettek a leggyengébbek ráadásul a tanításhoz tartozó futási idő is sokszorosa volt a többi modellének. Az összehasonlított modellek közül, tehát az XGBoost és döntési fa alapú torlódás detektáló algoritmusokat tartom a legcélravezetőbbnek. Ezen eredmények minden kutatónak és forgalomirányítónak hasznos és megbízható kiindulási pontot biztosítanak torlódás felismerén terén, annak függvényében, hogy milyen prioritások mentén szeretnék befolyásolni a forgalom változását.

Irodalomjegyzék

[1] Petkovics, Á., Szabó, Cs. A., Wippelhauser, A., Varga, N., Bokor, L.: *A V2X járműkommunikáció alapjai*, Útügyi Lapok, 8. évfolyam, 14. szám, 2020

[2] Waze: [Online] Elérési út: https://www.waze.com/hu/live-map/ [Hozzáférés dátuma: 2023.12.03.]

[3] Tseng, F. -H., Hsueh, J. -H., Tseng, C. -W., Yang, Y. -T., Chao, H. -C., Chou, L. -D.: *Congestion Prediction With Big Data for Real-Time Highway Traffic*, IEEE Access, Vol. 6, 2018, pp. 57311-57323

[4] Thorri Sigurdsson, T.: Road traffic congestion detection and tracking with Spark Streaming analytics (Dissertation), 2018

[5] Falkowski, T., Barth, A., Spiliopoulou, M.: *DENGRAPH: A Density-based Community Detection Algorithm*, IEEE/WIC/ACM International Conference on Web Intelligence (WI'07), 2007, pp. 112-115

[6] Bawaneh, M., Simon, V.: *Novel traffic congestion detection algorithms for smart city applications*, Concurrency Computat Pract Exper., 2023, 35:e7563

[7] Sujatha, R., Nithya, R. A., Subhapradha, S., és Srinithibharathi, S.: *Decision tree classification for traffic congestion detection using data mining*, International Journal of Engineering and Techniques, Vol. 4, 2018, 166-172

[8] ElSahly, O., Abdelfatah, A.: *An Incident Detection Model Using Random Forest Classifier*, Smart Cities, Vol. 6, 2023, pp. 1786-1813

[9] Sharma, S., Meenakshi, V., Apurva, Chakrasali, S., Satish, S.V.: *Real-time Traffic Congestion Detection using Combined SVM*, International Journal of Scientific & Engineering Research, Vol. 4, issue 11, 2013

[10] Kurniawan, J., Syahra, S. G., Dewa, C. K.: Traffic congestion detection: learning from CCTV monitoring images using convolutional neural network, Procedia computer science, Vol. 144, 2018, pp. 291-297

[11] Liu, Y., Cai, Z., Dou, H.: Highway traffic congestion detection and evaluation based on deep learning techniques, Soft Computing, Vol. 27, pp. 12249-12265

[12] Song, Y. -Y., Lu, Y.: *Decision tree methods: applications for classification and prediction*, Shanghai Arch Psychiatry, Vol. 27, 2015, pp. 130-135

[13] Yuan, Y., Wu, L., Zhang, X.: *Gini-Impurity Index Analysis*, IEEE Transactions on Information Forensics and Security, Vol. 16, 2021, pp. 3154-3169

[14] Berry, M. A., Linoff, G. S.: Mastering Data Mining: *The Art and Science of Customer Relationship Management*, Industrial Management & Data Systems, Vol. 100 No. 5, 2000, pp. 245-246

[15] Abadeh, S. S., Esfahani, P. M. M., Kuhn, D.: *Distributionally Robust Logistic Regression*, Advances in Neural Information Processing Systems, 2015, pp. 1576-1584

[16] Cervantes, J., Garcia-Lamont, F., Rodríguez-Mazahua, L., Lopez, A.: *A comprehensive survey on support vector machine classification: Applications, challenges and trends*, Neurocomputing, Vol. 408, 2020, pp. 189–215

[17] Singh, D., Singh, B.: *Investigating the impact of data normalization on classification performance*, Applied Soft Computing, Vol. 97, Part B, 2020, pp. 105-524

[18] XGBoost: [Online], Elérési út: https://xgboost.readthedocs.io/en/stable/ [Hozzáférés dátuma 2023.12.01.]

[19] Polikar, R.: *Ensemble Learning*, Zhang, C., Ma, Y. (szerkesztők): Ensemble Machine Learning, 2012, pp. 1-34

[20] SUMO: [Online], Elérési út: https://sumo.dlr.de/docs/index.html [Hozzáférés dátuma 2023.11.18.].

[21] Python: [Online] Elérési út: https://www.python.org/about/ [Hozzáférés dátuma 2023.11.26.].

[22] Pandas: [Online] Elérési út: https://pandas.pydata.org/docs/user_guide/index.html [Hozzáférés dátuma 2023.11.26.]

[23] NumPy: [Online] Elérési út: https://numpy.org/ [Hozzáférés dátuma 2023.11.28.]

[24] Matplotlib: [Online] Elérési út: https://matplotlib.org/ [Hozzáférés dátuma 2023.11.26.]

[25] Seaborn: [Online] Elérési út: https://seaborn.pydata.org/ [Hozzáférés dátuma 2023.11.26.]

[26] New Jersey Department of Transportation: [Online] Elérési út: https://www.nj.gov/transportation/commuter/roads/I78/occupancy.htm [Hozzáférés dátuma 2023.12.07.]

[27] Scikit-learn: [Online] Elérési út: https://scikit-learn.org/stable/ [Hozzáférés dátuma: 2023.12.01.]

[28] Kumar, N., Raubal, M.: *Applications of deep learning in congestion detection, prediction and alleviation: A survey*, Transportation Research Part C: Emerging Technologies, Vol. 133, 2021, pp. 103432

[29] Hossin, M., Sulaiman, M.N.: *A Review on Evaluation Metrics for Data Classification Evaluations*, International Journal of Data Mining & Knowledge Management Process, Vol. 5, No.2, 2015, pp. 1–11

Ábrajegyzék

1. ábra: Sigmoid függvény 14
2. ábra: Szimulált úthálózat 20
3. ábra: Forgalmi anomália típusok 21
4. ábra: Járműszámok gyakorisága 23
5. ábra: Sebességértékek gyakorisága 24
6. ábra: Többosztályos döntési fához tartozó konfúziós mátrixok 28
7. ábra: Logisztikus regresszióhoz tartozó konfúziós mátrixok 31
8. ábra: SVM-hez tartozó konfúziós mátrixok. 32
9. ábra: XGBoost-hoz tartozó konfúziós mátrixok 33
10. ábra: Összegzés az 1.teszteset eredményeiről 34
11. ábra: Összegzés a 2.teszteset eredményeiről 35
12. ábra: Összegzés a 3.teszteset eredményeiről 35

Táblázatjegyzék

1. táblázat: Konfúziós mátrix 25
2. táblázat: Döntési fa eredményei többosztályos klasszifikáció esetén 28
3. táblázat: Döntési fa eredményei bináris klasszifikáció esetén 29
4. táblázat: Logisztikus regresszió eredményei 31
5. táblázat: SVM eredményei 32
6. táblázat: XGBoost eredményei 33