BUDAPESTI GAZDASÁGI EGYETEM

PÉNZÜGYI ÉS SZÁMVITELI KAR

SZAKDOLGOZAT

Ecsédi András

Nappali munkarend

Gazdaságinformatikus szak

Üzleti adatelemző informatikus specializáció

2024

BUDAPESTI GAZDASÁGI EGYETEM
ALKALMAZOTT TUDOMÁNYOK EGYETEME

PÉNZÜGYI ÉS SZÁMVITELI KAR

BUDAPESTI GAZDASÁGI EGYETEM

PÉNZÜGYI ÉS SZÁMVITELI KAR

Hajó a láthatáron: tengeri fegyveres támadások előrejelzése a Szingapúr-és Malaka-szorosokban gépi tanulás segítségével

Belső konzulens: Dr. Kovács Endre

Külső konzulens: Csicsman József

Ecsédi András

Nappali munkarend

Gazdaságinformatikus szak

Üzleti adatelemző informatikus specializáció

2024

1149 Budapest, Buzogány utca 10-12.
Telefon: (+36-1) 469-6600
Fax: (+36-1) 469-6610
www.uni-bge.hu

NYILATKOZAT

Alulírott ECSEDI ANDRÁS büntetőjogi felelősségem tudatában nyilatkozom, hogy a szakdolgozatomban foglalt tények és adatok a valóságnak megfelelnek, és az abban leírtak a saját, önálló munkám eredményei.

A szakdolgozatban felhasznált adatokat a szerzői jogvédelem figyelembevételével alkalmaztam.

Ezen szakdolgozat semmilyen része nem került felhasználásra korábban oktatási intézmény más képzésén diplomaszerzés során.

Tudomásul veszem, hogy a szakdolgozatomat az intézmény plágiumellenőrzésnek veti alá.

Budapest, 2024 év december hónap 6 nap

.....

hallgató aláírása

1 Tartalom

2 Bevezetés

A Szingapúr-és Malaka-szorosok (Straits of Singapore and Malacca, a továbbiakban SOMS) együttesen egy Szingapúr, Indonézia és a Maláj-félsziget által körülölelt vékony, 805 km hosszú vízfelület (Paterson, 2023), mely létfontosságú a világkereskedelem szempontjából: a nemzetközileg gazdát cserélő termékek 40%-a ezen területen megy keresztül (Paterson, 2023).

Utóbbi információ tükrében érthető módon aggodalomra ad okot a SOMS területén történő tengeri fegyveres támadások száma - 2023-ban 63 db (ReCAAP ISC, 2024) - és növekvő tendenciája - 2022-ben 55 volt (ReCAAP ISC, 2024).

Szakdolgozatom során ezen incidensek előfordulását kísérlem meg előjelezni: ennek érdekében a különféle térségbeli országok statisztikai hivatalaitól, világszintű adatbankokból és regionális szervezetektől begyűjtött adatokat tisztítok meg és készítek elő többféle gépi tanulás-alapú modellek azokon való lefuttatásához.

Ezen modellek segítségével különféle következtetéseket igyekszem levonni: van-e kapcsolat az időjárási viszonyok és az incidensek gyakorisága közt? Egy adott hónap melyik időszakában, egy hét melyik napján, egy nap melyik napszakjában a leggyakoribbak a támadások? Milyen pontossággal jelezhető elő az egy hónapra jutó incidensek száma? Szakdolgozatomban ezen és más hasonló kérdésekre kísérlek meg választ találni.

Mivel a probléma, ahogy a 4. fejezetből ki fog derülni, nem újkeletű, az elmúlt években kialakultak megoldások és azokat végrehajtó szervezetek, növekedett a regionális kooperáció és információmegosztás mértéke.

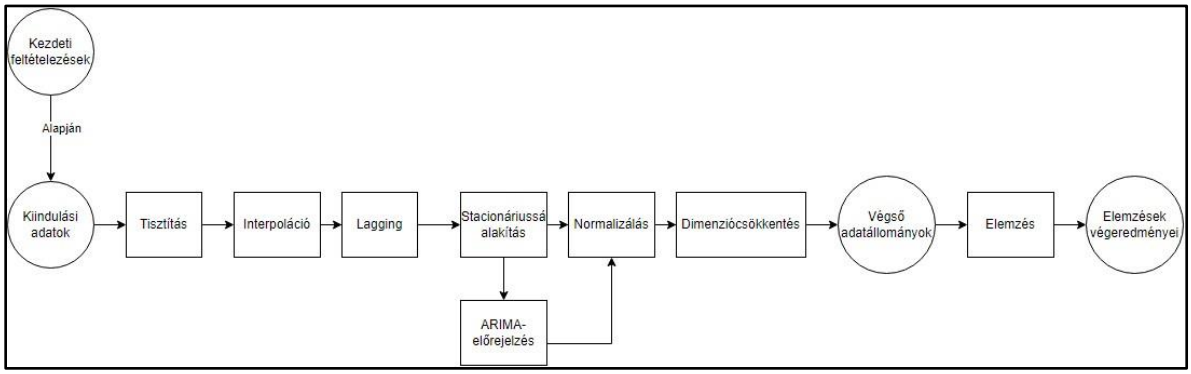
Reményeim szerint szakdolgozatom és annak eredményei segítséget nyújthatnak a térségben aktív, zászlajukra a kalóz-és tengeri tengeres támadások visszaszorítását tűző szervezeteknek erőforrásaik hatékonyabb felhasználásában. Ez azért is kiemelten fontos, mert a térség felügyeletét célul kitűző szervezeteknek nem áll rendelkezésére elegendő erőforrás elegendő számú őrjárat elvégzéséhez: véges hajóhaddal, véges üzemanyaggal, véges járőrözésre fordítható idővel és munkaerővel rendelkeznek. Ilyen körülmények között az erőforrások felhasználásának optimalizációja kiemelten fontos.

A szakdolgozatomnak úgy vélem, további jelentőséget és létjogosultságot ad, hogy bár a probléma alkalmas arra, hogy gépi tanuláson keresztül vizsgálják meg, kiaknázatlan terület; kevés tanulmány született róla.

2.1 A szakdolgozat felépítése

A bevezetés lezárulta után először az elemzés céljait fogom kifejteni, majd a térségbeli tengeri fegyveres támadások szakirodalom által feltételezett okairól fogok írni. Ez utóbbit az elemzésre kerülő adatok összeszedésének, és elemzésre való alkalmassá tételének kifejtése követi. Ezután az elemzési módszerek leírása, az azokhoz köthető hiperparaméterek kiválasztása, optimalizációja, illetve a modellek és az azok eredményeiből levonható következtetések kiértékelése kerül sorra, melyek alapján a legjobban teljesítő modellek segítségével 2025 decemberéig fogok prognózist vonni.

Az alábbi ábrán a szakdolgozatban felhasznált adatok életciklusa kerül szemléltetésre.



Forrás: saját szerkesztés

A szakdolgozatban 7.8. fejezetében megjelenő adatok, amennyiben nem tartozik hozzájuk külön forrásmegjelölés, az 5. fejezetben kialakításra kerülő adatállományokon lefuttatott modelljeim eredményeiből származnak. A modelleket, adatállományokat és a tisztító lépéseket tartalmazó mappa a szakdolgozat számára általam létrehozott GitHub-tárhelyen (András, 2024) elérhető.

3 Az elemzés céljai

Szakdolgozatom három fő célt tűz ki maga elé:

a. A hajók elleni tengeri fegyveres támadások (továbbiakban ARAS, azaz Armed Robbery Against Ships) és kalóztámadások okairól fennálló elméletek tesztelése. Pl. kimutatható-e kapcsolat a környékbeli országok politikai stabilitási indexe és az ARAS incidensek száma közt?
b. Pontos előrejelzések készítése. Ezt klasszifikációs, illetve numerikus célváltozós modellek segítségével próbálom elérni, előbbi esetben alacsony, illetve magas számú ARAS-incidensű hónapokat megkülönböztetve. Ezen modellek segítségével havi alapon szeretném előrejelezni a tengeri fegyveres támadásokat, majd azokat további nem havi, hanem az adott incidenshez kötődő adatok (pl. az adott incidens alatti időjárási viszonyok) segítségével pontosítani.
c. Előrejelzés 2025 végéig: az első és második célkitűzés során létrehozott modellek segítségével egy, az ARIMA-módszerrel 2025 decemberéig kiterjesztett idősoron futtatom le a legjobban teljesítő modelleket (egy numerikus célváltozójút és egy klasszifikációs modellt), majd vizualizálom az eredményeket.

Összesítve az elemzés célja a már meglévő elméletek tesztelése, illetve pontosabb előrejelzések elkészítése és azok gyakorlati alkalmazása.

4 Az elemzett helyzet ismertetése

Bár az ázsiai kalóz-és tengeri fegyveres támadásokat nyilvántartó szervezet, a RECAAP, más ázsiai vizekről is rendelkezik adatokkal, elemzésemben csupán a Szingapúr-és Malaka-szorosokban történő incidensekre fókuszálok, mivel annak körülményeire vonatkozó adataim és tesztelendő hipotéziseim vannak, melyek nem feltétlenül relevánsak a nagyobb ázsiai helyzet megértésére, feltérképezésére: támadások általános ázsiai trendjei több helyen eltérnek a SOMS-étól, 2021-ben például a Szingapúr-és Malaka-szorosok kivételével mindenhol a pandémia előtti szintre csökkent a támadások száma. (Storey, 2022) A továbbiakban a SOMS-ról, illetve a tengeri fegyveres támadások természetéről írok le releváns, az elemzés későbbi lépéseit meghatározó adatokat.

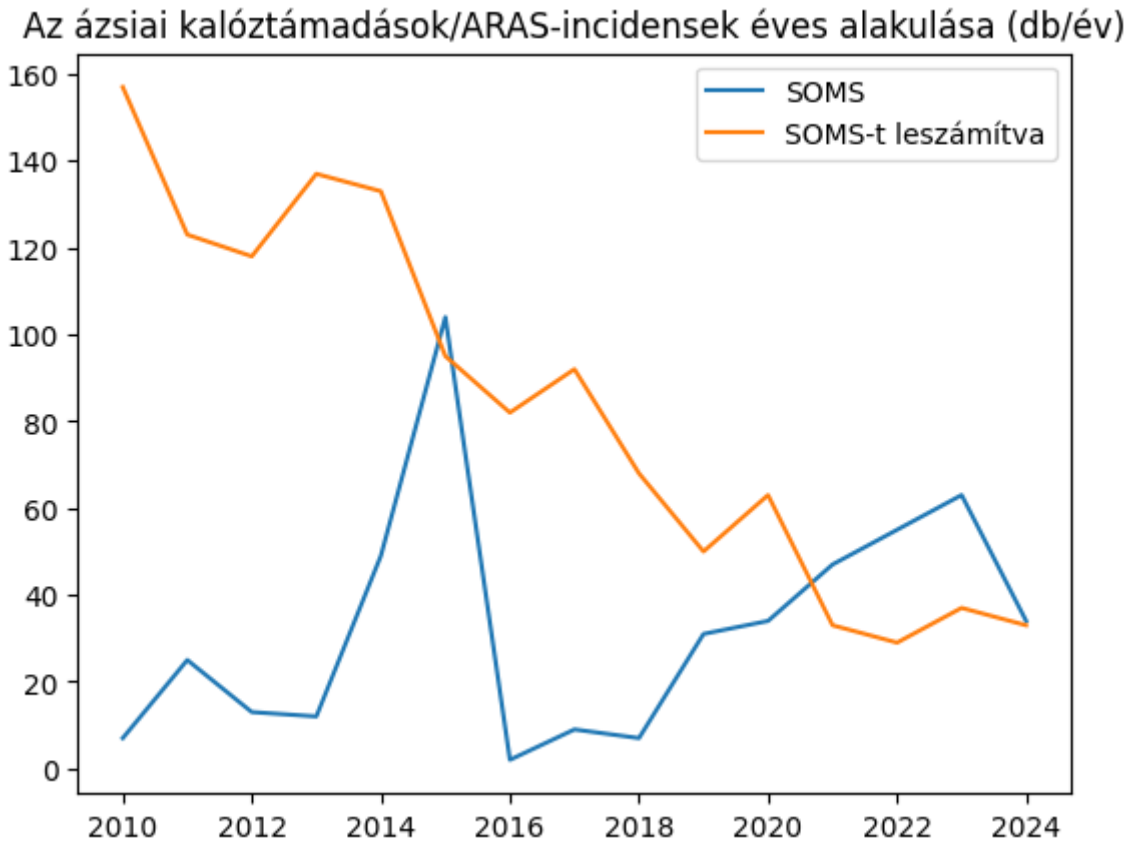
4.1 A SOMS rövid jellemzése földrajzi, gazdasági és nemzetközi jogi szempontokból

Az északról Malajzia és Szingapúr, délről pedig az Indonéz-szigetvilág által körülölelt, 805 km hosszú, legrövidebb pontján 600 méter széles, legsekélyebb pontján pedig 25 méter mély (The Nippon Foundation, dátum nélk.) SOMS nem elhanyagolható világgazdasági jelentőséggel bír, hisz az Indiai-óceán és a Csendes-óceán összekötőjeként (Haire, 2021) a tengeri kereskedelem majdnem fele ezen az útvonalon keresztül történik. Egyes országok számára ezen túl is fontos: Japán olajimportjainak több, mint 80%-a a SOMS-on keresztül jut az országba (The Nippon Foundation, dátum nélk.). Emellett a többi nagyobb ázsiai gazdaság – Tajvan, Kína, India, Szingapúr, Vietnám, Dél-Korea, Tájföld, Indonézia, Malajzia és a Fülöp-szigetek – összeköttetésében is fontos szerepe van.

A Szingapúr-és Malaka-szorosok három ország – Szingapúr, Malajzia és Indonézia – felségvizeihez tartoznak, így a terület ellenőrzésének felelőssége 1994 óta rájuk hárul (The Nippon Foundation, dátum nélk.).

4.2 SOMS-béli kalózkodás és tengeri fegyveres támadások: okok, megoldások és trendek

Mielőtt kifejteném a térséget érinti kalóz- és tengeri fegyveres támadások okait, trendjeit és az azokra adott megoldásokat, magyarázatot szolgáltatok a kalóztámadás és a tengeri fegyveres támadás közti különbségre: előbbi nemzetközi vizeken történik, utóbbi egy adott ország felségvizein (ReCAAP ISC, 2024). A 2019 és 2023 közt a térségben feljegyzett támadások 98%-a ARAS volt, a maradék kalóztámadás (ReCAAP ISC, 2024). A továbbiakban ezeket együtt kezelem, mert az elemzésem egyik kiindulópontja, hogy az őket meghatározó ösztönzőkben nincs különbség.



Forrás: saját szerkesztés (a ReCAAP ISC 2010-2024 közti éves adatait felhasználva

Ahogy a fentebbi, a RECAAP nevű szervezet által összeállított éves kimutatások alapján készített ábrán is látszik, az elmúlt években jelentősen megnőtt a SOMS-béli kalóztámadások száma. A most is tartó növekvő tendencia 2018-ban indult, ezelőtt két nagyobb kiugrást figyelhettünk meg, 2011-ben, illetve 2013 és 2015 közt.

A térségben nem újkeletű jelenség a kalózkodás (már a 14. században is akadt rá példa (Bileta, 2022)), aminek több oka is van. Ezeket négy kategóriába sorolnám:

a. Potenciális haszon: ahogy a fentebb leírtakból kikövetkeztethető, a térségen sok hajó halad keresztül, melyek jó része értékes árut szállít.
b. Földrajzi adottságok: a SOMS vékony, hajóktól sűrű csatornájában nehéz gyorsan haladni (Paterson, 2023), ami megkönnyíti a hajók megtámadását. Emellett a szoros megannyi kis szigetnek ad otthont (Paterson, 2023), illetve sok folyó is csatlakozik hozzá (Watson, 2013), ami alkalmassá teszi az elrejtőzésre és az elmenekülésre.
c. Az illetékes hatóságok felelőssége: a SOMS felett rendelkező és azért felelősséget vállaló államokhoz kötődő szervezetek, úgymint a haditengerészetek és a partiőrségek, illetve a civil rendfenntartó ügynökségeknek limitált a rendszeres járőrözésre való

képessége (Storey, 2022). A helyzetet tovább rontja, hogy több rendfenntartót rajtakaptak már kalózokkal való információmegosztáson (Spiess, 2019). Jogi vetülete is van a problémának: Malajzia törvénye például csak a kalóztámadásokra tér ki, az ARAS-ra nem (Panneerselvam & Ramkumar, 2023), Indonéziát pedig több szakértő felől is az a vád érte, hogy kalózellenes törvényeik nem hatékonyak és hogy az indonéz bíróságok által a kalózokra kiszabott büntetések enyhébbek a térség többi országához viszonyítva (Panneerselvam & Ramkumar, 2023).

d. Gazdasági okok: az uralkodó nézetek szerint kapcsolat figyelhető meg a térség gazdasági stabilitása és a kalóztámadások gyakorisága közt: az 1997-es ázsiai gazdasági krízist követő politikai destabilizáció és gazdasági visszaesés nyomán a kilencvenes évek végén megemelkedett a kalóztámadások száma; a nehézségek hatására sok, Indonézia és Malajzia partvidékein élő lakos fordult a kalózkodáshoz, aminek kivitelezését megkönnyítette a politikai instabilitás, főleg Indonéziában (Raymond, 2009). Mivel zömében a szegény partvidéki lakosok állnak kalóznak, a támadások gyakoriságát meghatározó tényező a halászati ipar teljesítménye (Paterson, 2023).

Mivel a térség gazdasági adatairól áll rendelkezésemre a legtöbb adat, szakdolgozatomban azokra fogok fókuszálni: egyfelől igyekszem bebizonyítani a kapcsolatot a támadások gyakorisága és a d. pontban említett tényezők közt, másfelől pedig megpróbálok azok alapján pontos előrejelzéseket alkotni.

A támadások elterjedésére adott válaszlépések közül a legfontosabb a ReCAAP (Regional Cooperation Agreement on Combating Piracy and Armed Robbery against Ships in Asia) létrejötte: ez egy multilaterális egyezmény, mely 2006-ban 10 ázsiai ország közt köttetett meg, mára pedig 21 tagra bővült. Az egyezmény célja a regionális kooperáció elmélyítése. Ennek nyomán jött létre a ReCAAP ISC (Information Sharing Centre), mely a támadásokhoz köthető incidenseket jegyzi fel, teszi elérhetővé és veti alá elemzésnek (https://www.recaap.org, dátum nélk.).

5 Adatok

A következő fejezet tárgyai a modellek által feldolgozandó adatállományok. Mivel előzetesen a gazdasági, társadalmi és politikai trendekből havi alapú előrejelzéseket tartottam egyszerre reálisnak és a szakdolgozat végcéljai szempontjából érdemesnek, ezért az efféle adatokat egy havi alapú adatállományba gyűjtöttem össze, az ezekkel elvégzett előrejelzések finomítása céljából összegyűjtött, az incidensekhez kötődő adatokat pedig egy másodikba.

A havi alapú gazdasági, politikai és szociális adatok négy országból származnak: Malajzia, Indonézia, Szingapúr és Thaiföld, mivel ezen országok veszik körül a Malaka-szorost. Előzetes, az olvasott irodalom alapján kialakult sejtésem az volt, hogy Thaiföld jelentősége kisebb a másik három országénál (elvégre nem kapott említést), ezért a modelleket olyan adatállományokon is lefuttattam, melyek rendelkeztek thai adatokkal és olyanokon is, amelyek nem.

5.1 Adatgyűjtés

5.1.1 Incidensekre vonatkozó adatok

A kalóz- és tengeri fegyveres támadásokra vonatkozó adatok két fő forrásból származnak: ReCAAP ISC és OpenMeteo.

A ReCAAP ISC PDF-formátumú, 2010 és 2021 közti éves jelentéseiben egy táblázatban szerepelnek az incidensek és az azokhoz köthető adatok, ami nem ideális, ha az adatokat táblázatos formában szeretném kezelni, azonban, mivel a ReCAAP-től nem kaptam választ a kommunikációs felületükön megosztott adatigénylésemre, nem volt más választásom, mint ezekből kiindulni. A 2010 és 2021 közti incidenseket a ReCAAP jelentései közt található éves jelentésekből (www.recaap.org, dátum nélk.), a 2022-es (www.recaap.org), 2023-as (recaap.org) és szeptember 22-vel bezárólag a 2024-es évre (www.recaap.org) vonatkozó incidensek adatait pedig a szervezet honlapjának főoldaláról letölthető PDF-ekből gyűjtöttem ki.

A PDF-ekben megtalálható táblázatok kinyerése nehézséget jelentett: az Adobe Acrobat online felületén ingyenesen elérhető, PDF-eket Excel-fájllá alakító program (www.adobe.com, dátum nélk.) segítségével vittem véghez: az eredeti, a RECAAP oldaláról letöltött jelentésekből kivágtam azon oldalakat, melyek a fent említett táblázatot tartalmazták, majd azokat az Adobe felületére feltöltve Excel-táblázattá alakítottam őket. A folyamat nem volt tökéletes: akár egy adott táblázaton belül is volt, hogy két sorban két különféle módon oszlottak el az oszlopok, az

egyik változót volt, hogy csak színnel jelezték, emellett a szövegfeldolgozás is hagyott némi kívánnivalót maga után (pl. O-k 0-nak való értelmezése és fordítva).

Az előbbi két problémára egyedül manuális választ tudtam adni, kézzel vontam össze és szedtem szét cellákat, hogy a végén egy működőképes táblázatot kapjak. Egy módszer, ami nagy segítséget jelentett, az Excel-táblázat CSV-fájllá konvertálása, majd visszaalakítása .xlsx formátumra, így kiszűrve az üres cellákat (volt táblázat, ahol ez a megoldás nem volt életképes, mivel az eredeti dokumentumban stilizáltan szaggatott vonallal jelzett táblázat-határokat az Adobe programja nem tudta rendesen feldolgozni).

Az így kapott fájlban több példa volt olyan cellákra, melyek elvileg egy cellát kellett volna, hogy alkossanak, ennek a problémának a megoldására viszont már alkalmasnak bizonyult a Python Pandas (pandas.pydata.org, dátum nélk.) nevű, adatelemzésre és adatmanipulációra szolgáló könyvtára: egy adatkeretbe helyezve a fentebbi folyamat révén kapott táblázatot, annak vertikális határait felismerve (az incidens súlyosságát jelző oszlopban mindig az adott sor tetején szerepelt a „CAT 1-4” kifejezés - ez a későbbi változók közt említésre kerül majd) egy programmal sikerült az eredeti PDF-ben megtalálható táblázat cellahatárait követő táblázattá alakítanom. A második problémára adott megoldásaimat az 5.2.1. fejezetben részletezem.

A táblázaton további változásokat eszközöltem: az eredetileg egy-egy oszlopba tartozó dátum és időt, illetve hosszúság-szélességet kettébontottam (utóbbi a „terület” oszloppal is egy oszlopban volt, az egyszerűség kedvéért ezt is külön oszlopba szedtem). Ezt, illetve a dátum egységes formátumra hozását és a hosszúság-szélesség átkonvertálását fok-perc-másodperc formátumról decimális fokokra (ez a későbbi API-hívásokhoz volt szükséges) szintén a Pandas segítségével végeztem el.

Az így kapott adatállomány változói a következők:

- Súlyosság: az incidens súlyossága, CAT 1-4 közti skálán mozog, ahol az 1 a legsúlyosabb (az elkövetők fel vannak fegyverkezve késekkel és/vagy lőfegyverekkel, a legénységet pedig erőszak érte, legyen az elrablás, sérülés vagy halál), a 4 pedig a legkevésbé súlyos (az elkövetők nem voltak felfegyverezve, vagy semmit nem loptak el, vagy csak csekély értékű holmit). A változó által felvehető érték továbbá az „Attempted”, azaz „megkísérelt”: a meghiúsult támadások súlyosság szerinti kategorizálása nem lehetséges.
- Támadásban érintett hajó(k) típusa: az adott támadásnak áldozatul eső hajó (vagy több esetben hajók) fajtája angolul. Pl. Oil tanker, General Cargo Ship.

- Dátum: a támadás dátuma YYYY/MM/DD formátumban.
- Idő: a támadás kezdetének időpontja (több órásra nyúló támadások esetén a végpont is szerepel).
- Hosszúsági pont
- Szélességi pont
- Terület: a terület, ahol a támadás történt. Pl. SOMS, Dél-Kínai tenger, Szingapúr stb. A SOMS-ra való szűréshez szükséges változó.

Az incidensekhez kötődően időjárási adatokat az OpenMeteo nevű, globális időjárási adatokhoz való hozzáférést lehetővé tevő szervezet API-ának segítségével nyertem ki: a Python-béli openmeteo_requests könyvtár által biztosított API kliens segítségével írtam egy, az adott szélességi és hosszúsági pontra, dátumra és időpontra (ahol több órán át tartott az incidens, ott a kezdőpontra) vonatkozó időjárási adatokat kinyerő és eltároló programot. A kliens működése miatt először napi szinten szereztem meg az adatokat, majd az adott napon belül szűrtem rá az adott órára.

Az OpenMeteo oldalán megtalálható változókból (open-meteo.com, dátum nélk.) azokat választottam ki, amik segíthettek egy potenciális támadónak döntést hozni abban, hogy végrehajtsa-e a támadást. Az így kapott változók a következők:

- Cloud_cover_low: alacsonyan lévő felhők és köd 3 km-es magasságig, az adott területhez mérten, százalékosan megadva.
- Temperature_2m: a hőmérséklet 2 méterrel a talajszint felett, Celsius-fokban megadva.
- Wind_speed_10m: Szélsebesség 10 méterrel a talajszint felett, km/órában megadva.
- Wind_speed_100m: Szélsebesség 100 méterrel a talajszint felett, km/órában megadva.
- Weather_code: az adott nap legsúlyosabb időjárási állapota, a WMO kódjai (WMO Code Table 4677, dátum nélk.) alapján. Az adataim közt előforduló kódokat és jelentésüket az alábbi táblázatban ismertetem.

Kód	Jelentés
0	Nem figyelhető meg felhőképződés.
1	A jelenlévő felhők szertefoszlóban, visszafejlődésben vannak.
2	Az ég állapota változatlan.
3	Felhőképződés.
51	Folyamatos, szitáló, nem jeges, a megfigyelés idején kis mértékű eső.

53	Folyamatos, szitáló, nem jeges, a megfigyelés idején közepes mértékű eső.
55	Folyamatos, szitáló, nem jeges, a megfigyelés idején nagy mértékű eső.
61	Folyamatos, nem jeges, a megfigyelés idején kis mértékű eső.
63	Folyamatos, nem jeges, a megfigyelés idején közepes mértékű eső.
65	Folyamatos, nem jeges, a megfigyelés idején nagy mértékű eső.

- Wind_gusts_10m: az előző órában a talaj felett 10méterrel mért széllökések maximális sebessége km/órában megadva.
- Rain: Az előző órában mért esőzés mértéke milliméterben megadva.

A két adatállomány összekapcsolásából kaptam meg az incidensekre vonatkozó adatállományt, melyben az incidensek sorszámmal vannak egyértelműen azonosítva. Az így kapott adatállomány hiányos: egyes esetekben a pontos idő és a koordináták nem kerültek megadásra a RECAAP által biztosított jelentésekben, így az ezekre támaszkodó időjárási adatok is hiányoznak.

Az adatállományon szűrést hajtottam végre, hogy csak a SOMS-béli, illetve a Szingapúri incidensek maradjanak meg. Bár utóbbi nem tartozik a SOMS-hoz, elég közel van hozzá ahhoz, hogy megérje az ott történő támadásokat előbbihez sorolni. Mindezt a „Terület” változó segítségével tettem meg. Összesen 492, a SOMS-ban (vagy Szingapúrban, de ez elenyésző részét képezi az eseteknek) történő incidens és az azokhoz tartozó adatok alkotják a végső adatállományt.

5.1.2 Országokra vonatkozó adatok

A térségben lévő országok – Malajzia, Indonézia, Szingapúr és Thaiföld – adatait több forrásból gyűjtöttem össze: az egyes országok statisztikai hivatalaiból, a Világbank DataBankjából, a St. Louis-i Szövetségi Jegybank FRED nevű gazdasági adatbázisából, az IMF DataMapper nevű felületéről, illetve az OECD Data Explorer platformjáról.

Mivel ezen adatok kigyűjtése és a belőlük létrejövő országos, illetve azokat összesítő adatállományok létrehozása nem jelentett jelentős informatikai kihívást, illetve az azokból kiválasztott releváns változók a szakdolgozat későbbi részeiben említésre és megtárgyalásra kerülnek, így ezen adatok részletes ismertetését a mellékletben végeztem el.

5.2 Adatok előkészítése a modellekhez

5.2.1 Adattisztítás

Ahogy az incidensekre vonatkozó adatok összegyűjtésénél említésre került, a RECAAP éves jelentéseiből az Adobe Acrobat PDF-ből Excel-táblázatokat alkotó programjával történő adatkinyerés nem volt zökkenőmentes. Az alábbiakban a táblázatokban megtalálható és a végső adatállományba bekerülő változókat érintő adattisztítási folyamatot fogom kifejteni.

A legnagyobb probléma a karakterfelismerésben rejlett: sok példa akadt 0-k O-ként való, 1-ek és 7-ek L-ként való értelmezésére és fordítva. Ezek elsősorban a dátum, idő és a hosszúság-szélesség esetén jelentettek nehézséget, de a területeknél is (pl. a SOMS rövidítés többször S0MS-ként került feldolgozásra). A hosszúság és szélesség esetén a „°” szimbólum is problémát jelentett, sokszor 0-ként vagy O-ként lett feldolgozva.

A nehézségekre hibrid megoldást alkalmaztam, Excelben és Pythonban oldottam meg a problémát. Voltak speciális, egyszer-kétszer előforduló hibák, ezeket Excelben javítottam ki, mert úgy gondoltam, nem éri meg programot írni rájuk. Arra is akadt példa, hogy egész oszlopot érintő változtatást eszközöltem Excelben (a koordináták esetén említett problémát például itt oldottam meg).

A dátumoknál, az időnél és a koordinátáknál, melyeknek elviekben nem szabadott nem numerikus karaktert tartalmaznia (pár kivétellel, pl. az óra esetén a „hrs”, vagy a 2022-24 közti adatok esetén a dátum-oszlop hónapjai esetén, melyeket ott szövegesen jelöltek), rászűrtem azokra, amik tartalmaztak ilyet, így felismerve, majd Excelben kézileg kijavítva a problémás eseteket – azért kézileg, mert az 1 és a 7 is gyakran L-ként lett feldolgozva, így azok kijavítását csak az eredeti RECAAP-jelentések átnézése után tehettem meg.

A hajótípus kiszedése során is meghaladandó nehézségekbe ütköztem. Az első ilyen probléma az volt, hogy a hajótípusok egy cellában voltak más értékekkel (hajó neve, azonosítója, zászlója), melyek nem voltak relevánsak, a köztük lévő határt pedig nem lehetett felismerni tömegesen, algoritmus segítségével. Ez azért jelentett nehézséget, mert nem rendelkeztem listával az összes hajótípusról.

A problémát több lépcsőben oldottam meg: a 2022-24 közti támadásokról készült RECAAP-es jelentések táblázataiban a hajótípus már külön oszlopban szerepelt. Ezeket listába szedve algoritmikusan végigmentem a 2010 és 2021 közti adatokon, ahol a cellában stringként szerepelt bármelyik a listában szereplő hajótípusok közül, ott egy új oszlopba bekerült az.

Itt mellékes problémát jelentett, hogy az egyes hajótípusok neveiben volt egyezés, pl. a „tanker” szó mind a „chemical tanker”, mind a „tanker” hajótípusban szerepelt. Ezt úgy oldottam meg, hogy hossz szerint csökkenő sorrendbe rendeztem a listában lévő hajótípusokat, amikor pedig ezen a listán haladt végig az algoritmus, az első egyezés után kilépett (ez nem jelentett problémát, mert nem volt olyan cella, amiben pl. „chemical tanker” és „tanker” is lett volna).

Mindezek eredményeképp az incidensek jelentős részében el tudtam különíteni a hajótípust, a maradék felismert esetet (például ahol a szövegfeldolgozó indokolatlanul szóközt helyezett két szó közé vagy a már korábban is említett hasonszőrű problémák közül fordult elő az egyik) manuálisan, Excelben javítottam ki, majd a kijavított Excel-tábla alapján létrehozott adatkereten újra lefuttattam a programot.

5.2.2 Interpoláció

Az országokra vonatkozó adatok közül a legtöbb nem havi, hanem éves, negyedéves (ritkábban két-hároméves) időszakokra vonatkozott. Viszont mivel a modelljeimmel havi pontosságot kívántam elérni, havi idősort kellett alkotnom. Ezt az interpoláció nevű módszerrel értem el.

Az interpoláció lényege, hogy két pontot egy adott függvényosztályból (pl. lineáris) származó görbével összekötünk, így megbecsülve a két pont közti ismeretlen értékeket (byjus.com, dátum nélk.). A Python Pandas könyvtára lehetőséget ad az interpolációra: az országos adatállományok létrehozásakor havi alapúra alakítottam át azokat, az így keletkezett üresedéseket pedig a pandas.DataFrame.interpolate() metódus segítségével, két módszerrel töltöttem ki, lineáris és időalapúval. Utóbbi az elsőhöz hasonlóan szintén egy lineáris görbét illeszt a pontokra, azonban olyan tényezőket (pl. a hónapok hossza) is figyelembe vesz, amiket a lineáris módszer nem és amelyek időalapú indexelésnél számítanak.

Bár mindkét interpolációs módszert elvégeztem, végül csak az időalapú interpolációval előállított adatokat használtam fel.

5.2.3 Lagging

Bár a kalóz-és tengeri fegyveres támadásokra vonatkozó adataim 2010-ben kezdődnek, az országos adatállományokból létrehozott összesített adatállományban vannak változók, melyeknek kezdőpontja 2008. Ennek oka, hogy mivel a szakdolgozatom egyik célja a különféle gazdasági mutatók és az incidensek közti kapcsolat letesztelése, a 2008-as gazdasági világválságnak otthont adó és az arra következő év adatait hibának éreztem kihagyni.

Ezt a lagging módszerével (GeeksForGeeks, 2024) oldottam meg: az adatállományon végighaladva azon oszlopok alapján, amiknek első nem-null értéke 2008-ra esik, lagged oszlopokat hoztam létre, melyek 24 hónappal eltolják az eredeti oszlop értékeit, tehát egy adott oszlop 2008. januári értéke 2010 januárjára kerül ebben az oszlopban.

A lagging nyomán két adatállományt hoztam létre: egy olyat, amiben azon oszlopokból, melyeknek sima és lagged változata is van, csak utóbbit hagytam meg, illetve egy olyat, amiben mindkettőt.

5.2.4 Stacionaritás

A regressziós modellek egyik alapfeltétele, hogy az idősor, ami alapján betanítjuk őket, stacionárius legyen, azaz olyan idősor, amely nem változik az idő múlásával (Faridi). Bár nem minden modellem regressziós (pl. van köztük döntési fa-alapú), két okból mégis csak stacionárius adatokra futtattam le minden modellem: egyfelől a szakdolgozat harmadik céljában említett, 2025-ig előrejelzett értékekkel dolgozó idősor előállítása ARIMA-val történik és mint ilyen, stacionárius, másfelől pedig a döntési fa-alapú modelleknek nehézséget jelent a trend extrapoláció, melyre gyógyírt jelenthet az idősor stacionáriussá tétele.

Az idősort a következőképpen tettem stacionáriussá: a pmdarima.arima.utils könyvtár ndiffs metódusának segítségével minden oszlopról megállapítottam, hányszor kéne differenciálni őket, hogy a bennük lévő idősor stacionárius legyen (Gupta, 2024), majd annyi alkalommal elvégeztem azt. Bár a differenciálásnak alávetett adatállományokban voltak változók, amik eredetileg diszkrétek voltak (pl. bűnesetek száma egyes indonéz provinciákban), az interpolációval folytonossá tettem őket, az így kapott idősort pedig már szabad volt stacionáriussá tenni.

5.2.5 Normalizálás

Az idősorban esetlegesen előforduló kiugró értékek kezelését skálázással értem el, azaz úgy alakítottam át az adatokat, hogy bizonyos keretek közé férjen. Három módszerrel végeztem el a skálázást:

- Min-Max skálázás: az adatok 0 és 1 közti értéket vesznek fel azáltal, hogy egy adott adathalmaz minden értékéből kivonásra kerül az adathalmaz minimuma, az így kapott érték pedig elosztásra kerül az

adathalmaz maximális értékének és minimumértékének kivonásából származó értékkel (Nalcin, 2022).

- Robosztus skálázás: ezen skálázási módszer lényege, hogy eltávolítja a mediánt és az interkvartilis tartomány alapján skálázza az adatokat. Nevét onnan kapta, hogy ellenállóbb a kiugró értékekkel szemben. (Nalcin, 2022)
- Standardizálás: ezen módszer szerint az adott adathalmaz összes értékéből kivonásra kerül az adathalmaz átlaga, majd az így kapott eredmény elosztásra kerül az adathalmaz szórásával. (Nalcin, 2022)

A Python scikit-learn könyvtára mindhárom módszerre lehetőséget biztosít.

Ezen túl zscore-normalizálással is próbálkoztam (Bobbitt, 2021). A módszer lényege, hogy az adott adatállomány minden értékét úgy alakítja át, hogy azok átlaga 0, átlagos eltérése 1 legyen. Az így kapott adatsorokban a 3 feletti értékek outliereket jelentenek.

Az oka annak, hogy ezt a módszert nem használtam a végső adatállományok kialakítása során, hogy mivel száz feletti dimenziószámú adatállományaim voltak, ha egy adott idősorban kevés kiugró érték is volt, azok felgyülemlettek, az ezen értékeket tartalmazó sorok kiszűrése pedig egy jelentősen kisebb adatállományt hagyott maga után.

5.2.6 Dimenziócsökkentés és jellemzőkiválasztás

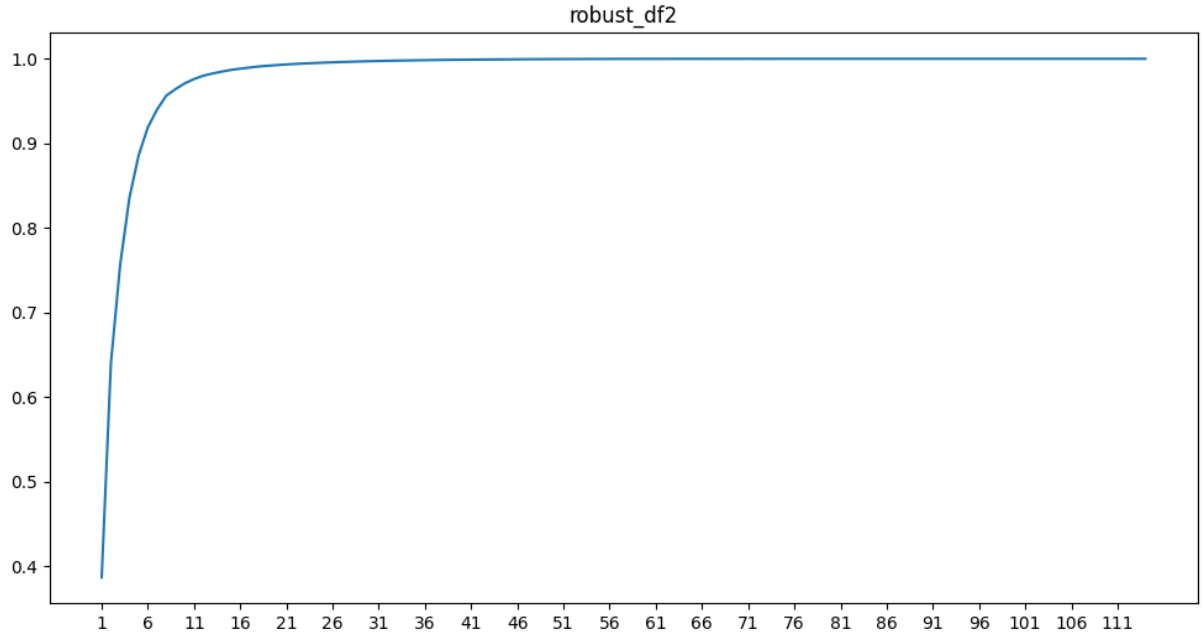
Az eddig leírt folyamatok eredményeképp több száz változós adatállományokat kaptam. Ezeket több okból is érdemes volt lecsökkentenem: egyfelől a modellek gyorsabb lefutása és alacsonyabb memóriaigénye, másfelől pedig a redundáns és irreleváns jellemzők kezelése érdekében.

A dimenziócsökkentést két fő módszerrel végeztem el: PCA-val és AutoEncoderekkel. A továbbiakban ezek elméleti hátterét és gyakorlati megvalósulását fogom kifejteni.

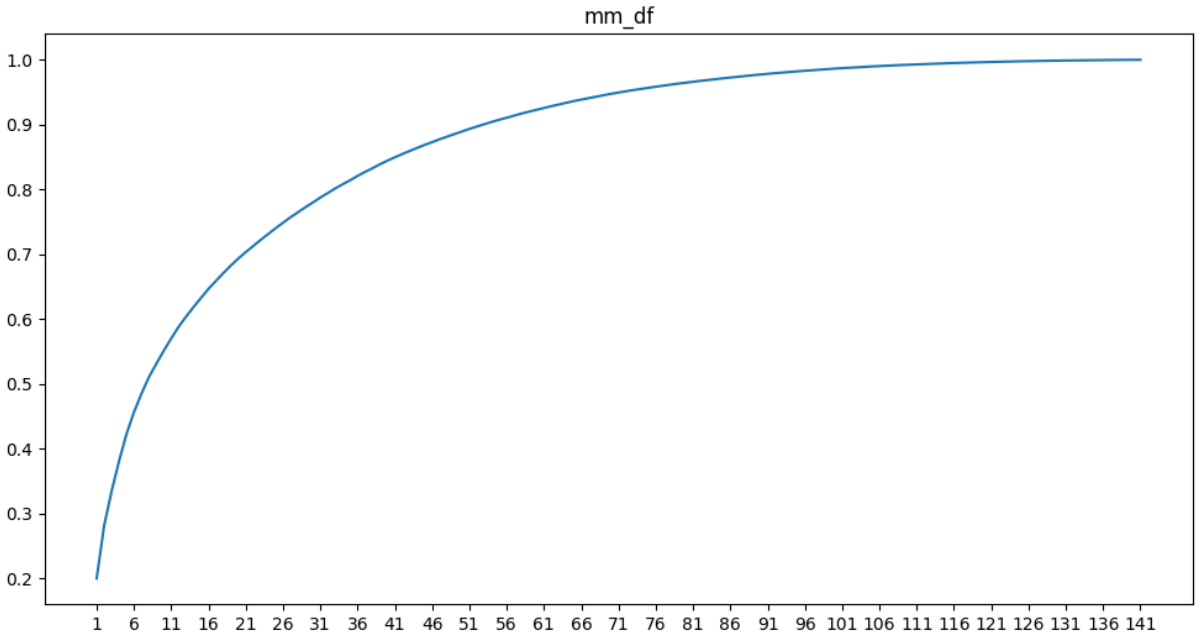
5.2.6.1 Főkomponens-elemzés

A PCA (Főkomponens-elemzés) nevű lineáris dimenziócsökkentő eljárás lényege, hogy egy nagy adathalmaz információtartalmát adott számú, egymással a lehető legkevésbé korreláló, az eredeti változók lineáris kombinációjaként előálló komponensekbe szervezi (IBM, 2023).

A főkomponensek ideális számát a könyökmódszer (Mangale, 2020) segítségével állapítottam meg: az adott adatállományon lefuttattam a főkomponens-elemzést 0 és a maximális lehetséges dimenziószámmal (ez az eredeti adatkeret sorai és oszlopai közül a kisebbik értékkel egyenlő), majd vizualizáltam az adott dimenziószám mellett a modell által megmagyarázott varianciát. Két példa az eredményekre:



A könyökmódszer-vizualizáció eredménye a maláj, indonéz és szingapúri adatokat tartalmazó adatállományból robusztus skálázás segítségével kialakított, ütközés esetén csak a lagged változókat tartalmazó adatállomány esetében



A könyökmódszer-vizualizáció eredménye a maláj, indonéz és szingapúri adatokat tartalmazó adatállományból min-max skálázás segítségével kialakított, lagged és nem lagged változókat is tartalmazó adatállomány esetében

A vizualizáció alapján megkerestem a könyökpontot, azaz a pontot, ahol a modell által megmagyarázott variancia növekedési üteme rohamosan csökkenni kezd. Ahol nem volt megfigyelhető könyökpont, ott olyan dimenziószámot választottam, mely megőrzi a megmagyarázott variancia 90%-át. Az egyes adatállományokhoz tartozó ideális dimenziószámok az alábbi táblázatban találhatóak meg.

	Thai adatokkal, csak lagged	Thai adatokkal, lagged és nem lagged	Thai adatok nélkül, csak lagged	Thai adatok nélkül, lagged és nem lagged
Min-Max Skálázás	25	55	21	55
Robusztus skálázás	8	8	8	8
Standardizálás	26	65	26	65

5.2.6.2 AutoEncoder-alapú dimenziócsökkentés

A könyökmódszer segítségével megtalált dimenziószámot a PCA melletti másik, neurális háló-alapú dimenziócsökkentő eljáráshoz is felhasználtam.

Az AutoEncoderekkel történő dimenziócsökkentés lényege, hogy az adatokat úgy kódolja, hogy azokat egy alacsonyabb dimenziós térbe tömöríti, majd azt dekódolva újjáépíti az eredeti inputot (Rajan, 2021). Utóbbi lépés az AutoEncoderek dimenziócsökkentésre való használata esetén a kódolás teszteléséhez szükséges (Riswanto, 2023): a dekódolás végén létrejött és az eredeti adathalmaz összehasonlításával (például az átlagos négyzetes eltérés alapján) javítható a dimenziócsökkentés folyamata.

Az általam használt kódoló modellnek 64 darab „dense”, azaz olyan rétege van, melynek összes neuronja kapcsolódik az előző réteg összes neuronjához (dátum nélk.). Aktivációs függvénye a LeakyReLu.

A LeakyReLu egy, a ReLuhoz hasonló aktivációs függvény. A ReLu pozitív input esetén azt teszi meg outputnak, negatív input esetén pedig 0-t. Ebből származó lehetséges probléma az ún. „haldokló ReLu”, ami akkor történik, amikor a ReLu csak negatív értékeket kap, az így kapott nullértékek pedig megnehezítik a hiba-visszaterjesztést. A LeakyReLu ezt úgy oldja meg, hogy amennyiben negatív értéket kap, 0 helyett azon értéket megszorozza egy alacsony számmal (az én kódom esetében ez 0.1), az így kapott értéket outputtá téve (Olamendy, 2023).

A kódolás regularizációja érdekében bevezettem a 0.1-es Dropout hiperparamétert, melynek lényege, hogy az adott réteg inputjainak 10%-át 0-ra állítja, megelőzve a túltanulást. A dekódolás szintén 0.1-es alfájú LeakyReLuval és 64 rétegű „dense” neurális hálóval történt.

A kódolás-dekódolás eredményeire támaszkodó AutoEncoder az „adam” módszert használja a gradiens ereszkedés optimalizálására. Az „adam” módszer két gradiens turbózó módszer keveréke (GeeksforGeeks, 2024): a „momentum” módszeré, mely a gradiensek exponenciálisan súlyozott átlagával számol, hogy az gyorsabban elérje a minimumot, illetve az RMSP-módszeré, mely a gradiensek exponenciális mozgóátlagát veszi.

Az AutoEncoder a veszteséget az átlagos négyzetes eltérés alapján számítja. A modell tanítása során az adatokat 80-20 arányban bontottam tanító-és tesztadatokra, mely felbontás a „shuffle”=True hiperparaméter következtében random történik, így megnehezítve a túltanulást.

Az epochok számát 50-ben maximáltam, azaz maximum ennyiszer futhat le a modell a tanítóadatokon. Mivel bevezettem a korai leállást (5 epochnyi türelmi idővel és a validációs veszteséggel, mint figyelendő tulajdonsággal), ez a szám alacsonyabb is lehet.

Végül az autoencoder eredményei által újra kódoltam az adatokat az ideális dimenziószámba.

A modellel kapcsolatban három nehézség vetődött fel: egyfelől a robosztus skálázással normalizált adatokon az autoencoder folyamatosan kizárólag nullértékeket tartalmazó adathalmazokat adott végeredményül. Ezt, bárhogy alakítottam át a hiperparamétereket, nem tudtam megoldani, szóval csak a másik két módszerrel normalizált adathalmazokon futtattam le az autoencodert.

Egy másik probléma az epochok számával volt: eredetileg az epochok száma 100 volt, ami a többi adathalmaz esetén is nullértékeket eredményezett. Végül ezt a szám 50-re való csökkentésével oldottam meg: azért feleztem meg, mert kétszer ekkora adathalmazon (egy adathalmaz, amelyben mind a lineáris, mind az időalapú módszer segítségével interpolált adatok megtalálhatóak voltak) az eopchokat 100-ban maximáló módszer jól teljesített.

Amikor lefuttattam a programot, egyes futások esetén a második probléma 50-ben maximált epoch-szám mellett is megmaradt: ezt egy kondicionális (a kapott adathalmaz és a kapott adathalmaz nullértékeket kiszűrt változatának oszlop-és sorszámainak összehasonlítása alapján) újrafuttatással oldottam meg.

5.2.7 Végső adatállományok

Az alábbiakban azon adatállományokat listázom, melyeken lefuttattam a 6. fejezetben felsorolásra kerülő modelleket. Összesen 32 darab ilyen adatállomány keletkezett, ezeket az alábbi táblázatban illusztrálom.

	Thai adatok nélkül		Thai adatokkal	
	Csak lagged	Nem csak lagged	Csak lagged	Nem csak lagged
PCA	3	3	3	3
AE	2	2	2	2
-	3	3	3	3

A végső adatállományok által tartalmazott első hónap minden esetben 2010 márciusa, az általa tartalmazott utolsó hónap pedig 2022 decembere. Ennek oka, hogy bár vannak adatok, amik 2024-ig tartanak, az adatállományok változóinak jelentős részének 2022 decemberére vonatkozik az utolsó értéke. Bár előrejelezhettem volna ARIMA segítségével (erre a 8. fejezetben lesz példa), a dolgozat 1. célját figyelembe véve csak az adatállományok eddigi előkészítési lépésein átment változatain futtattam le a modelleket.

6 Modellek

A továbbiakban az adatállományok elemzésére létrehozott modellek elméleti hátterét, működését és hiperparamétereit, illetve az azok optimalizációjára való módszereket fogom kifejteni. Ezután a modellek teljesítményét értékelem, majd a legjobban teljesítő modellek eredményeit fogom értelmezni.

6.1 Numerikus célváltozós modellek

A továbbiakban a numerikus célváltozóval rendelkező függvények működését fejtem ki. A célváltozó, azaz az előrejelezni kívánt változó minden esetben az incidensek havi alapon összesített száma volt, az annak előrejelzésére használt független változókat pedig minden esetben az adott adatállomány változói alkottak.

6.1.1 Lineáris regresszió

Az első általam használt modell egy lineáris regressziós modell volt, amit két tanító-teszt arány – 70-30 és 80-20 – mellett futtattam le. A lineáris regresszió a függő és független változók közt lineáris kapcsolatot feltételez (Kanade, 2023). A modellt a 5.2.6.1. fejezet ábrájában megtalálható összes adatállományon lefuttattam.

6.1.2 Lineáris regresszió RFE-vel

Az RFE (Recursive Feature Elimination, azaz rekurzív jellemzőeltávolítás) a dimenziócsökkentéshez hasonlóan egy, a nagy adathalmaz komplexitásának és méretének csökkentésére szolgáló módszer, mely egy másik modellel (esetemben lineáris regresszióval) együttműködve vethető be.

A teljes adathalmazból úgy választ adott számú független változót, hogy a modellt lefuttatja először az összes független változóval, majd újabb futtatásokkal fokozatosan eltávolítja a legkevésbé fontos független változókat (a fontosságot lineáris regressziós modell esetén az adott független változó és a függő változó közti regressziós együtthatóból, azaz az utóbbiban előbbi hatására bekövetkező változás mértékében becsléséből állapítja meg), míg adott számú nem marad belőlük (Analytics Vidhya, 2024).

A megmaradt független változók számát a felhasználó adja meg a modell futtatása előtt, én ezt az „egy a tízben” szabály (Chowdhury & Turin, 2020) alapján az adatállomány független változóinak 1/10-ében határoztam meg. A lineáris regresszió-alapú RFE-modellt a tanító-teszt adatok 70-30 és 80-20 arányú felosztása mellett futtattam le a 5.2.6.1. fejezet táblázatának

utolsó sorában megtalálható adatállományokon, azaz azokon, melyekre nem alkalmaztam semmiféle dimenziócsökkentő eljárást vagy jellemzőkiválasztást.

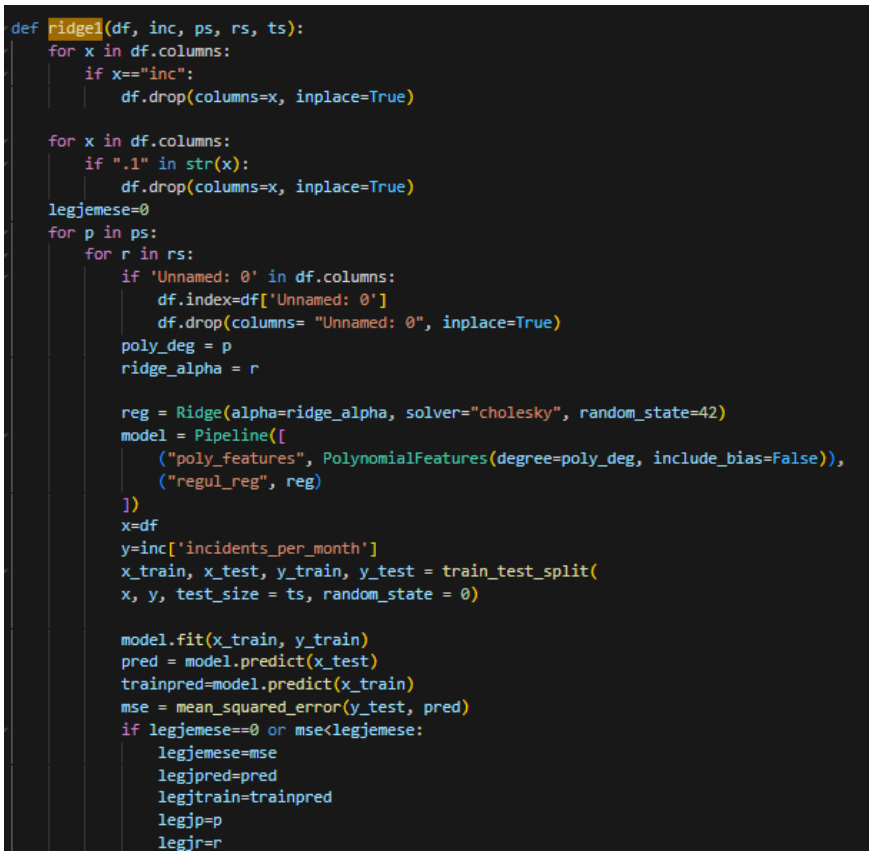
```
def linreg(df, inc, ts):
    for x in df.columns:
        if x=="inc":
            df.drop(columns=x, inplace=True)
    for x in df.columns:
        if ".1" in str(x):
            df.drop(columns=x, inplace=True)
    y=inc['incidents_per_month']
    df=df
    if 'Unnamed: 0' in df.columns:
        df.index=df['Unnamed: 0']
        df.drop(columns= "Unnamed: 0", inplace=True)
    x=df
    x_train, x_test, y_train, y_test = train_test_split(
            x, y, test_size = ts, random_state = 0)
    lm = LinearRegression()
    lm.fit(x_train, y_train)
    pred = lm.predict(x_test)
    mse = mean_squared_error(y_test, pred)
```

Forrás: A modell Python-béli megvalósulása a programomban (az adatvizualizációt és az eredmények gyűjtését szolgáló sorokat leszámítva).

6.1.3 Ridge regresszió

A ridge regresszió a regresszió egy regularizált változata, melynek célja a túltanulás csökkentése (Shelar, 2023) a súlyok alacsonyan tartásával, amit egy regularizációs büntetőkifejezéssel (a paraméterek négyzetének összege) ér el (Kuknyó, Üzleti Elemzések Módszertana 3. Gyakorlat: Regularizált Modellek, 2024, old.: 15). Két legfontosabb hiperparamétere a polinom foka és az alfa (az együttható, mely megadja a regularizáció mértékét). Ezen túl fontos hiperparaméter a „solver” (az én programom esetén Cholesky) (scikit-learn.org, dátum nélk.).

A modell optimalizációja az alfa és a polinom foka alapján, grid search módszerrel történt: különböző alfa és polinomfok-értékek mellett megnéztem a modell tesztadatokon mért MSE-jét, a legjobb értékhez tartozó modell adatait pedig elmentettem. A modellemben az alfa által felvett lehetséges értékek 0.01, 0.1 és 1 voltak, a polinom fokai pedig 2, 3, vagy 4. Ahogy az előző modelleket, úgy ezt is a tanító-teszt adatok 70-30 és 80-20 arányú felosztása mellett futtattam le. A modellt a 5.2.6.1. fejezet ábrájában megtalálható összes adatállományon lefuttattam.



```
def ridge1(df, inc, ps, rs, ts):
    for x in df.columns:
        if x=="inc":
            df.drop(columns=x, inplace=True)

    for x in df.columns:
        if ".1" in str(x):
            df.drop(columns=x, inplace=True)
    legjemese=0
    for p in ps:
        for r in rs:
            if 'Unnamed: 0' in df.columns:
                df.index=df['Unnamed: 0']
                df.drop(columns= "Unnamed: 0", inplace=True)
            poly_deg = p
            ridge_alpha = r

            reg = Ridge(alpha=ridge_alpha, solver="cholesky", random_state=42)
            model = Pipeline([
                ("poly_features", PolynomialFeatures(degree=poly_deg, include_bias=False)),
                ("regul_reg", reg)
            ])
            x=df
            y=inc['incidents_per_month']
            x_train, x_test, y_train, y_test = train_test_split(
            x, y, test_size = ts, random_state = 0)

            model.fit(x_train, y_train)
            pred = model.predict(x_test)
            trainpred=model.predict(x_train)
            mse = mean_squared_error(y_test, pred)
            if legjemese==0 or mse<legjemese:
                legjemese=mse
                legjpred=pred
                legjtrain=trainpred
                legjp=p
                legjr=r
```

Forrás: A modell Python-béli megvalósulása a programomban (az adatvizualizációt és az eredmények gyűjtését szolgáló sorokat leszámítva).

6.1.4 Ridge regresszió korai leállással

A ridge regresszió önmagában hajlamos volt a túltanulásra (mivel a legjobb modellt a tesztadatokon elért MSE alapján választottam ki, mely a pontosságot díjazta, nem pedig az általánosítóképességet), ezért a modellhez hozzáadtam a korai leállást: adott alfa és polinom fok mellett 100 epochnyi alkalmat adtam a modell lefutására, ahol 10 után, ha nem javult a tesztadatokon mért MSE, a futás leállt.

A modellt a tanító-és teszt adatok 70-30 és 80-20 arányú felosztása mellett futtattam le ugyanazokon az adatállományokon, mint a korai leállás nélküli ridge regressziót, ugyanazon okokból kifolyólag.

```python
def ridgerse(df, inc, ps, rs, ts):
    for x in df.columns:
        if x=="inc":
            df.drop(columns=x, inplace=True)

    for x in df.columns:
        if ".1" in str(x):
            df.drop(columns=x, inplace=True)
    legjemese="b"
    for p in ps:
        for r in rs:



            poly_deg = p
            ridge_alpha = r

            reg = Ridge(alpha=ridge_alpha, solver="cholesky", random_state=42)
            rfe = RFE(estimator=reg, n_features_to_select=int(len(df)/10))
            model = Pipeline([
                ("poly_features", PolynomialFeatures(degree=poly_deg, include_bias=False)),
                ("feature_selection", rfe),
                ("regul_reg", reg)
            ])
            x=df
            y=inc['incidents_per_month']
            print("x")
            x_train, x_test, y_train, y_test = train_test_split(
            x, y, test_size = ts, random_state = 0)
            print("y")
            model.fit(x_train, y_train)
            pred = model.predict(x_test)
            trainpred = model.predict(x_train)
            mse = mean_squared_error(y_test, pred)
            print(mse)
            if legjemese=="b" or mse<legjemese:
                legjemese=mse
                legjpred=pred
                legjtrain=trainpred
```

Forrás: A modell Python-béli megvalósulása a programomban (az adatvizualizációt és az eredmények gyűjtését szolgáló sorokat leszámítva).

6.1.5 LASSO regresszió

A LASSO (Least Absolute Shrinkage and Selection Operator) regresszió egy, a paraméterek abszolútértékeinek összegén alapuló büntetőkifejezés segítségével regularizálja a modellt (Kuknyó, Üzleti Elemzések Módszertana 3. Gyakorlat: Regularizált modellek, 2024, old.: 17). Két legfontosabb hiperparamétere a polinom foka és az alfa, a regularizáció mértékét megadó együttható.

A modell optimalizációja az alfa és a polinom foka alapján, grid search módszerrel történt. A modellemben az alfa által felvett lehetséges értékek 0.01, 0.1 és 1 voltak, a polinom fokai pedig 2, 3, vagy 4. A modell adott hiperparaméterek melletti minőségét a tesztadatokon elért MSE alapján határoztam meg. A modellt a tanító-és teszt adatok 70-30 és 80-20 arányú felosztása

mellett futtattam le. ugyanazokon az adatállományokon, mint a korai leállás nélküli ridge regressziót, ugyanazon okokból kifolyólag.

```
def indiana(df, inc, ps, rs, ts):
    for x in df.columns:
        if x=="inc":
            df.drop(columns=x, inplace=True)


    for x in df.columns:
        if ".1" in str(x):
            df.drop(columns=x, inplace=True)
    legjemese=0
    for p in ps:
        for r in rs:
            if 'Unnamed: 0' in df.columns:
                df.index=df['Unnamed: 0']
                df.drop(columns= "Unnamed: 0", inplace=True)

            poly_deg = p
            lasso_alpha = r

            reg = Lasso(alpha=lasso_alpha)
            model = Pipeline([
                ("poly_features", PolynomialFeatures(degree=poly_deg, include_bias=False)),
                ("regul_reg", reg)
            ])
            x=df
            y=inc['incidents_per_month']
            x_train, x_test, y_train, y_test = train_test_split(
            x, y, test_size = ts, random_state = 0)

            model.fit(x_train, y_train)
            pred = model.predict(x_test)
            mse = mean_squared_error(y_test, pred)
            if legjemese==0 or mse<legjemese:
                legjemese=mse
                legjpred=pred
                legjtrain=model.predict(x_train)
```

Forrás: A modell Python-béli megvalósulása a programomban (az adatvizualizációt és az eredmények gyűjtését szolgáló sorokat leszámítva).

6.1.6 LASSO regresszió korai leállással

A LASSO regresszióhoz általánosítóképességének javítása érdekében hozzáadtam a korai leállást: adott alfa és polinom fok mellett 1000 epochnyi alkalmat adtam a modell lefutására, ahol 10 után, ha nem javult a tesztadatokon mért MSE, a futás leállt. A modellt a tanító-és teszt adatok 70-30 és 80-20 arányú felosztása mellett futtattam le ugyanazokon az adatállományokon, mint a korai leállás nélküli ridge regressziót, ugyanazon okokból kifolyólag.

```python
def lassoes(df, inc, ps, rs, ts):
    for x in df.columns:
        if x=="inc":
            df.drop(columns=x, inplace=True)
    legjlegjemese=float('inf')
    for p in ps:
        for r in rs:
            if 'Unnamed: 0' in df.columns:
                df.index=df['Unnamed: 0']
                df.drop(columns= "Unnamed: 0", inplace=True)
            reg = Lasso(alpha=r, warm_start=True, max_iter=1, random_state=42)
            model = Pipeline([
                ("poly_features", PolynomialFeatures(degree=p, include_bias=False)),
                ("regul_reg", reg)
            ])
            x=df
            y=inc['incidents_per_month']
            x_train, x_test, y_train, y_test = train_test_split(
            x, y, test_size =ts, random_state = 0)
            legjemese = float('inf')
            patience=10
            noimp = 0
            for epoch in range(1000):
                model.fit(x_train, y_train)
                pred = model.predict(x_test)
                mse = mean_squared_error(y_test, pred)

                if mse < legjemese:
                    legjemese = mse
                    noimp = 0
                else:
                    noimp += 1
                if noimp >= patience:
                    break
            if legjemese < legjlegjemese:
                legjlegjemese = legjemese
                bestepoch=epoch
                bestmodel=clone(model)
                legjpred=pred
                legjtrain=model.predict(x_train)
                legjp=p
                legjr=r
```

Forrás: A modell Python-béli megvalósulása a programomban (az adatvizualizációt és az eredmények gyűjtését szolgáló sorokat leszámítva).

6.1.7 Regresszió gradiens turbózással

A gradiens turbózás egy együttes tanulási módszer, azaz több, gyengébb tanuló modell eredményeit iteratívan ötvözi egy erősebb tanuló modellbe (Verma, 2023). Ezt a gradiens ereszkedésre támaszkodva éri el: az algoritmus a rezidumok minimalizálásának érdekében javítja minden újabb modell paramétereit az előző eredményei alapján (Verma, 2023).

A modellt a Python scikit-learn.ensemble (scikit-learn.org, dátum nélk.) könyvtárában található GradientBoostingRegressor metódus segítségével építettem fel, annak alapértelmezett hiperparamétereit használva: 0.1-es tanulási sebesség, négyzetes hiba alapján optimalizált veszteségfüggvény és 100 darab szavazóosztály. A modellt a tanító-és teszt adatok 70-30 és 80-20 arányú felosztása mellett futtattam le.

```
def graddescent(df, inc, ts):
    for x in df.columns:
        if x=="inc":
            df.drop(columns=x, inplace=True)

    for x in df.columns:
        if ".1" in str(x):
            df.drop(columns=x, inplace=True)

    y=inc['incidents_per_month']
    df=df
    if 'Unnamed: 0' in df.columns:
        df.index=df['Unnamed: 0']
        df.drop(columns= "Unnamed: 0", inplace=True)
    x=df
    x_train, x_test, y_train, y_test = train_test_split(
            x, y, test_size = ts, random_state = 0)

    model = GradientBoostingRegressor(max_depth=2)
    model.fit(x_train,y_train)
    pred=model.predict(x_test)
    mse=mean_squared_error(y_test, pred)
    print(mse)
```

Forrás: A modell Python-béli megvalósulása a programomban (az adatvizualizációt és az eredmények gyűjtését szolgáló sorokat leszámítva).

6.2 Klasszifikációs modellek

6.2.1 Klasszifikációs modellek előkészítése

A klasszifikációs módszerekhez szükségesek kategóriák, amelyekbe az egyedek besorolhatóak. Az incidenseket darabszámát havi bontásban megadó oszlopokat annak átlagánál kettébontottam (azért annak átlagánál és nem mediánjánál, mert a békés időszakok nullértékei miatt a mediánon alapuló osztályozás nem reprezentálta volna megfelelően az adatokat). A további modellek az így létrejött két kategóriába sorolják be az egyedeket a 6.1. fejezet modelljeiben is megtalálható független változók alapján.

6.2.2 Logisztikus regresszió

A logisztikus regresszió egy, a bináris osztályozást valószínűségek becslésével elérő módszer: adott mintaaegyedre megbecsüli annak a pozitív osztályba tartozásának valószínűségét, ha ez

magasabb egy küszöbértéknél, a becslés szerint beletartozik (Kuknyó, Üzleti Elemzések Módszertana 2. Előadás: Osztályozás, 2024, old.: 26). A modell a valószínűség megbecslésére szigmoid függvényt használ: először egy lineáris predikciót állít elő, majd behelyettesíti azt a logisztikus függvénybe (Kuknyó, Üzleti Elemzések Módszertana 2. Előadás: Osztályozás, 2024, old.: 27).

A logisztikus regresszió használata során az elemzett adatállományt 80-20 arányban osztottam tanító-és tesztadatokra. Eredetileg terveztem 70-30 arányú felosztást is, azonban látva a 80-20 arányban felosztott adatok alapján végzett osztályozás pontosságát, úgy láttam, a 70-30-as arányból származó általánosítóképesség-növekedés nem érné meg a pontosság-beli veszteséget. A modellt mind a 32 adatállományon lefuttattam.

```
def logreg(df, clmet):
    if 'Unnamed: 0' in df.columns:
        df.index=df['Unnamed: 0']
        df.drop(columns= "Unnamed: 0", inplace=True)

    for x in df.columns:
        if ".1" in str(x):
            df.drop(columns=x, inplace=True)


    df["inc"]=clmet
    df.dropna(inplace=True)
    inc=df["inc"]
    df.drop(columns="inc", inplace=True)

    X_train, X_test, y_train, y_test = train_test_split(df, clmet, test_size=0.2, random_state=42)

    lr = LogisticRegression()
    model = lr.fit(X_train, np.array(y_train))
    pred = model.predict(X_test)
    og = np.array(y_test)
    og1 = np.array(y_train)
    trpred= model.predict(X_train)
```

Forrás: A modell Python-béli megvalósulása a programomban (az adatvizualizációt és az eredmények gyűjtését szolgáló sorokat leszámítva).

6.2.3 Ridge klasszifikáció

A Ridge klasszifikáció a Ridge regresszióhoz hasonlóan egy, a paraméterek négyzetének összegén és az alfa hiperparaméteren alapuló büntetőkifejezés segítségével éri el, hogy a modell ne váljon túltanulttá. Ahogy a Ridge regresszió esetében, úgy itt is a lehetséges alfa változókon (lehetséges értékei 0.01, 0.1 és 1) iteratívan keresztülhaladva igyekeztem a tesztadatokon legjobb pontosságot elérő modellt megtalálni.

Az alfa mellett fontos hiperparaméter a „max_iter”, a „solver” és a „tol”.

- max_iter: a maximális elvárható iterációszám, amit a solvernek el kell végeznie leállás előtt. A programom esetében 1000.
- solver: az én programom esetén „auto”, azaz automatikusan választ optimalizáló solvert (pl. cholesky).
- tol: konvergenciatolerancia, programom esetében értéke 1e-3. Amikor két iteráció közt a súlyok közti különbség alacsonyabb, mint ez az érték, a solver a lefutás végéhez ér.

A modellt 80-20 arányban tanító-és tesztadatokra bontott adatállományon futtattam le, mind a 32 adatállomány esetében.

```
def rdigecl(df, inc):
    if 'Unnamed: 0' in df.columns:
            df.index=df['Unnamed: 0']
            df.drop(columns= "Unnamed: 0", inplace=True)

    for x in df.columns:
        if ".1" in str(x):
            df.drop(columns=x, inplace=True)

    df["inc"]=inc
    df.dropna(inplace=True)
    inc=df["inc"]
    df.drop(columns="inc", inplace=True)

    X = df
    Y = inc

    x_train, x_val, y_train, y_val = train_test_split(X, Y, test_size=0.2, random_state=42)
    aa=[0.01, 0.1, 1]
    legjacc=0
    for x in aa:

        alpha = x
        max_iter = 1000
        solver = 'auto'
        tol = 1e-3
        rc = RidgeClassifier(
        alpha=alpha, max_iter=max_iter, solver=solver, tol=tol)
        rc.fit(x_train, y_train)
        pred=rc.predict(x_val)
        trpred=rc.predict(x_train)

        accuracy = accuracy_score(y_val, pred)
        if legjacc==0 or legjacc<accuracy:
            legjacc=accuracy
            legjpred=trpred
```

Forrás: A modell Python-béli megvalósulása a programomban (az adatvizualizációt és az eredmények gyűjtését szolgáló sorokat leszámítva).

6.2.4 Logisztikus regresszió RFE-vel

Ahogy a numerikus célváltozós modellek esetében, úgy itt is hasznos eszköznek bizonyult az RFE, melyet a logisztikus regresszióval kombinálva, a tanító-teszt adatok 80-20 arányú felbontásával vetettem be a dimenziócsökkentésnek és jellemzőkiválasztásnak alá nem vetett adatállományokon.

```
def rfelogreg(df, inc):
    if 'Unnamed: 0' in df.columns:
            df.index=df['Unnamed: 0']
            df.drop(columns= "Unnamed: 0", inplace=True)
    for x in df.columns:
        if ".1" in str(x):
            df.drop(columns=x, inplace=True)
    df["inc"]=inc
    df.dropna(inplace=True)
    inc=df["inc"]
    df.drop(columns="inc", inplace=True)
    x = df
    y = inc
    X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=0)
    lr = LogisticRegression()
    rfe = RFE(estimator=lr, n_features_to_select=int(len(df)/10))
    rfe.fit(X_train, y_train)
    sel= x.columns[rfe.support_]
    X_trainsel = X_train[sel]
    X_testsel= X_test[sel]
    lr.fit(X_trainsel, y_train)
    pred = lr.predict(X_testsel)
    trpred=lr.predict(X_trainsel)
    accuracy = accuracy_score(y_test, pred)
```

Forrás: A modell Python-béli megvalósulása a programomban (az adatvizualizációt és az eredmények gyűjtését szolgáló sorokat leszámítva).

6.2.5 SVM

Az SVM (Tartóvektor-gépek) egy adathalmaz két osztálya közti legszélesebb utat keresik, döntési határuk pedig ezen út közepén húzódik, melynek két oldalát margók szegélyezik (Kuknyó, Üzleti Elemzések Módszertana 6. Előadás: Tartó vektor gépek, 2024, old.: 4). Kernelük szerint lineáris, polinom (a döntési határ lineárisan megállapíthatóvá tétele érdekében a lineárisan nem transzformálható adatokat magasabb térbe transzformáló (Kuknyó, Üzleti Elemzések Módszertana 6. Előadás: Tartó vektor gépek, 2024, old.: 18)) és radiális bázis függvény-alapú (a döntési határt hasonlósági jellemzők alapján megállapító (Kuknyó, Üzleti Elemzések Módszertana 6. Előadás: Tartó vektor gépek, 2024, old.: 21)) modelleket különböztethetünk meg.

Az SVM-ek két legfontosabb hiperparamétere a gamma és a C. A gamma a haranggörbe-illesztés tágasságát határozza meg, egy adott pont befolyási tere ennek méretével fordítottan arányosan mozog (Kuknyó, Üzleti Elemzések Módszertana 6. Előadás: Tartó vektor gépek, 2024, old.:

21), helyes meghatározása fontos az alul-és túltanulás elkerülése érdekében (avicksaha, 2024). A gamma különösen az RBF kernelű tartóvektor-gépek esetén fontos (avicksaha, 2024).

A C a margó keménységét meghatározó hiperparaméter: beszélhetünk lágy-és keménymargós osztályozásról, előbbi eset tolerálja a margóra eső mintaegyedeket, utóbbi ezt nem tűri. A magas C-érték hátránya a rosszabb általánosítóképesség (Kuknyó, Üzleti Elemzések Módszertana 6. Előadás: Tartó vektor gépek, 2024, old.: 9).

Az optimális modellt a fentebbi két változón elvégzett „grid search" módszer segítségével igyekeztem megtalálni: a gamma által felvett lehetséges értékek 0.1, 1, 10, 100 és 1000 voltak, a C által felvett értékek pedig 1 és 10 között mozogtak, a legjobb modellt pedig a tesztadatokon elért pontosság alapján választottam ki. Mindezt mindhárom kernellel, az eddigiekhez hasonlóan az adatállományok 80-20% tanító-teszt felosztása mellett végeztem el a PCA és AE segítségével dimenziócsökkentett adatokon.

```python
def gridsearchsvm(df, inc, kern):
    hulk=[0.1, 1, 10, 100, 1000]
    legj=0
    for g in hulk:
        for c in range(1, 10):
            if 'Unnamed: 0' in df.columns:
                df.index=df['Unnamed: 0']
                df.drop(columns= "Unnamed: 0", inplace=True)
            for x in df.columns:
                if ".1" in str(x):
                    df.drop(columns=x, inplace=True)

            df["inc"]=inc
            df.dropna(inplace=True)
            inc=df["inc"]
            df.drop(columns="inc", inplace=True)
            x = df
            y = inc

            X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=0)
            svm = SVC(kernel=kern, C=c, gamma=g, random_state=0)
            svm.fit(X_train, y_train)
            sctrain=svm.score(X_train,y_train)
            sctest=svm.score(X_test,y_test)
            if sctrain>legj:
                legj=sctest
                legj0=sctrain
```

Forrás: A modell Python-béli megvalósulása a programomban (az adatvizualizációt és az eredmények gyűjtését szolgáló sorokat leszámítva).

6.2.6 Döntési fa-alapú klasszifikáció

A döntési fa-alapú klasszifikáció esetén az adathalmaz mintaegyedei változóikban felvett értékeik alapján, csomópontok kérdéseire válaszolva kerülnek osztályozásra (Kuknyó, Üzleti Elemzések Módszertana 4. Gyakorlat: Döntési fák, 2024, old.: 6). Háromféle csomópontot különböztetünk meg: a gyökeret (ennek csak outputja van), a belső csomópontot, melynek inputja és outputja is van, illetve a levelet, amely csak inputtal rendelkezik (Kuknyó, Üzleti Elemzések Módszertana 4. Gyakorlat: Döntési fák, 2024, old.: 7).

A csoportokra bontás történhet Gini-index és entrópia segítségével is: előbbi az adott csomópont tisztátlanságának (Thakar & Tahsildar, 2022), utóbbi annak bizonytalanságának megbecslésére hivatott.

Az entrópia egy adott halmaz osztályainak egymáshoz való aránya alapján számítható ki; minél egyenlőbb arányban vannak jelen a különféle osztályok, annál magasabb, ha pedig az egyik osztályba tartozó mintaegyedek aránylag többen vannak, az entrópia alacsonyabb. Az adott változó rosszul osztályozásának valószínűségére becslést adó Gini-index (Thakar & Tahsildar, 2022) értéke is hasonlóan alakul hasonló feltételek mellett.

Az adott csomóponton lévő ideális elágazás ezek segítségével lehetséges: attól függően, hogy melyik módszert választjuk, a döntési fa igyekszik a lehető legtisztább vagy legbiztosabb elválasztást eredményező elágazást létrehozni (Thakar & Tahsildar, 2022).

A döntési fáimat mind a Gini-index, mind entrópia alapján lefuttattam az adatok 80-20 arányú tanító-teszt felbontása mellett. A fáimon minden csomópontnak legalább öt egyedet kell tartalmaznia, hogy elágazás jöhessen rajta létre, illetve maximum ötszintes lehet a fa. Ezt a két hiperparamétert nem grid search segítségével határoztam meg, hanem egyszerű kísérletezéssel (egyéb értékek, amikkel lefuttattam a modellt: 1, 3, illetve 10).

A döntési fa-alapú modelleket mind a 32 adatállományon lefuttattam.

```python
def ginitree(df, inc):
    if 'Unnamed: 0' in df.columns:
        df.index=df['Unnamed: 0']
        df.drop(columns= "Unnamed: 0", inplace=True)
    for x in df.columns:
        if ".1" in str(x):
            df.drop(columns=x, inplace=True)
    df["inc"]=inc
    df.dropna(inplace=True)
    inc=df["inc"]
    df.drop(columns="inc", inplace=True)
    x = df
    y = inc

    X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=0)


    treegini= DecisionTreeClassifier(criterion="gini", random_state=42, max_depth=5, min_samples_leaf=5)


    treegini.fit(X_train, y_train)
    pred=treegini.predict(X_test)
    trpred=treegini.predict(X_train)
```

Forrás: A gini-alapú döntési fa Python-béli megvalósulása a programomban (az adatvizualizációt és az eredmények gyűjtését szolgáló sorokat leszámítva).

```python
def onodrim(df, inc):
    #entropy tree
    if 'Unnamed: 0' in df.columns:
        df.index=df['Unnamed: 0']
        df.drop(columns= "Unnamed: 0", inplace=True)
    for x in df.columns:
        if ".1" in str(x):
            df.drop(columns=x, inplace=True)
    df["inc"]=inc
    df.dropna(inplace=True)
    inc=df["inc"]
    df.drop(columns="inc", inplace=True)
    x = df
    y = inc

    X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=0)


    ent= DecisionTreeClassifier(criterion="entropy", random_state=42, max_depth=5, min_samples_leaf=5)


    ent.fit(X_train, y_train)
    pred=ent.predict(X_test)
    trpred=ent.predict(X_train)
```

Forrás: Az entrópia-alapú döntési fa Python-béli megvalósulása a programomban (az adatvizualizációt és az eredmények gyűjtését szolgáló sorokat leszámítva).

6.2.7 GBDT

A GBDT (Gradient-Boosted Decision Tree) egy együttes tanuló algoritmus, mely a 4.1.7.-ben említett modellhez hasonlóan működik, annyi különbséggel, hogy nem egy regressziós, hanem egy döntési fa-alapú modellt optimalizál a gradiens turbózás segítségével.

Az általam használt modell 200 szavazóosztályt használ, tanulási rátája 0.1, az előálló fa maximális mélysége pedig 3. A modellt 80-20 arányban tanító-és tesztadatokra bontott adatállományon futtattam le mind a 32 adatállomány esetében.

```
from sklearn.ensemble import GradientBoostingClassifier

def gbc(df, inc):
    if 'Unnamed: 0' in df.columns:
        df.index=df['Unnamed: 0']
        df.drop(columns= "Unnamed: 0", inplace=True)

    for x in df.columns:
        if ".1" in str(x):
            df.drop(columns=x, inplace=True)

    df["inc"]=inc
    df.dropna(inplace=True)
    inc=df["inc"]
    df.drop(columns="inc", inplace=True)
    x = df
    y = inc

    X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=42)

    gbclass=GradientBoostingClassifier(n_estimators=200, learning_rate=0.01,
    max_depth=3, random_state=0)
    gbclass.fit(X_train, y_train)
```

Forrás: A modell Python-béli megvalósulása a programomban (az adatvizualizációt és az eredmények gyűjtését szolgáló sorokat leszámítva).

6.2.8 Gaussi Naiv Bayes

A Gaussi Naiv Bayes egy valószínűségi alapú klasszifikációs módszer, mely azon feltételezésekből indul ki (Martins, 2023), hogy minden független változó a többitől függetlenül képes előrejelezni a függő változót, illetve abból, hogy minden osztály normális (Gaussi) eloszlást követ.

A modellt 80-20 arányban tanító-és tesztadatokra bontott adatállományon futtattam le mind a 32 adatállományon.

```python
from sklearn.naive_bayes import GaussianNB

def nb(df, inc):
    if 'Unnamed: 0' in df.columns:
        df.index=df['Unnamed: 0']
        df.drop(columns= "Unnamed: 0", inplace=True)
    for x in df.columns:
        if ".1" in str(x):
            df.drop(columns=x, inplace=True)
    df["inc"]=inc
    df.dropna(inplace=True)
    inc=df["inc"]
    df.drop(columns="inc", inplace=True)
    x = df
    y = inc

    X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=0)

    gauss = GaussianNB()
    gauss.fit(X_train, y_train)
    pred = gauss.predict(X_test)
    trpred=gauss.predict(X_train)
    print(accuracy_score(y_test, pred)*100)
```

Forrás: A modell Python-béli megvalósulása a programomban (az adatvizualizációt és az eredmények gyűjtését szolgáló sorokat leszámítva).

6.3 Adott hónapon belüli előrejelzések

Az eddig említett modellek egy adott hónapra kísérelték meg előrejelezni a kalóz-és tengeri fegyveres támadásokat. A továbbiakban adott hónapon belül igyekszem becslést adni a támadások valószínűségére. Mindezt a következőképpen teszem meg:

- A hónapok három időszakra való bontása: a hónapot 1.-10., 11.-20., illetve 21.-utolsó napjaik szerint szakaszokra bontom, az ezen szakaszok menti összesítés alapján pedig megpróbálok a hónapon belül pontosabb becslést adni a támadás valószínűségére.
- Héten belül: kimutatást készítek arról, a hét mely napján, illetve hétvégén vagy hétköznap milyen valószínűséggel történik támadás.
- Napon belül: kimutatást készítek arról, az adott napon belül reggel, délelőtt, délután és este milyen valószínűséggel történik támadás.
- Időjárási viszonyok alapján: az OpenMeteo API segítségével megszerzett időjárási adatokat azok átlaga alapján magas és alacsony kategóriákba sorolva kapcsolatot igyekszek találni azok, illetve az incidensek előfordulása között.
- Hajótípus alapján: Az adott hajótípus(ok) szerint csoportosítva is bemutatom a támadások gyakoriságát.

- Koordináták alapján: az incidensek koordinátáit klaszteranalízisnek vetem alá, így csoportosítva őket. Ez segítséget nyújthat az őrjáratok hatékonyabb megszervezésében.

7 Modellek értékelése és eredményei

7.1 Általános értékelési szempontok

A modellek értékelésénél vannak szempontok, amik nem specifikusak a modell típusára. Az elsődleges célom olyan modell választása, mely jó általánosítóképesség mellett megfelelő pontossággal, vagy a numerikus célváltozós modellek esetén alacsony MSE-vel képes előrejelzést készíteni. Mint ilyen, nem túl-vagy alultanult.

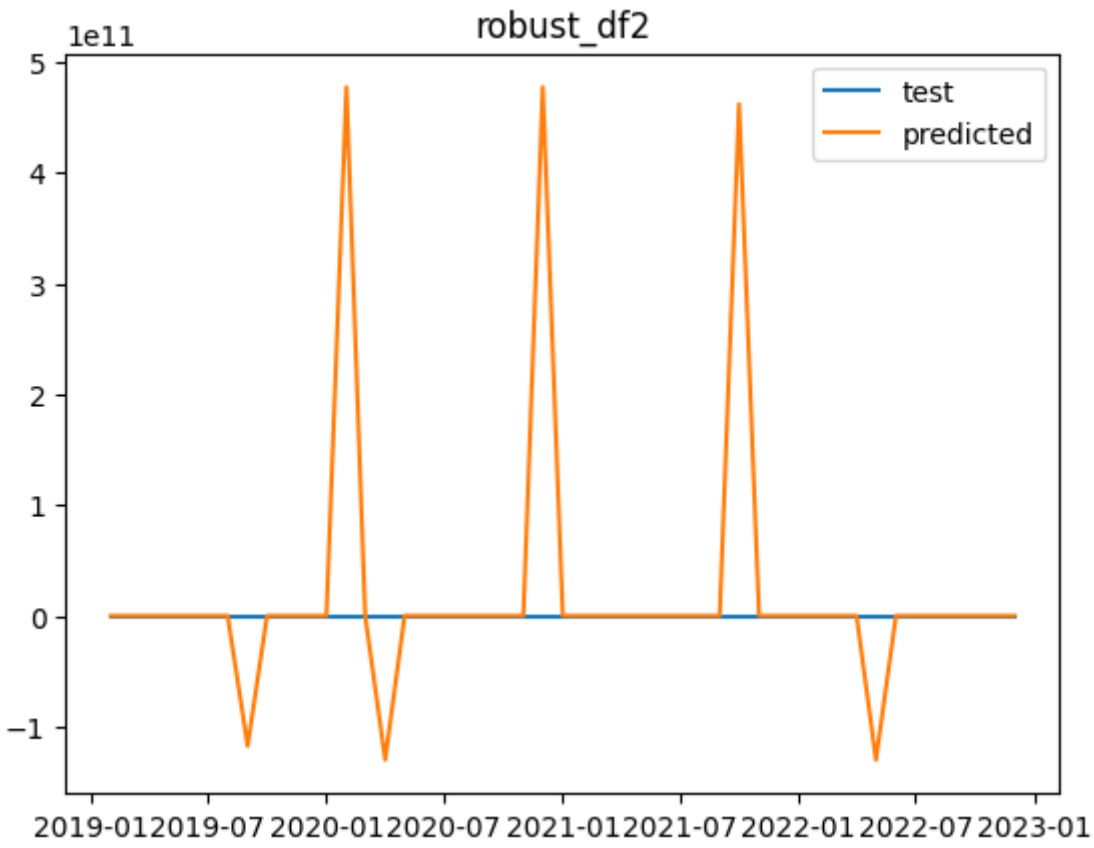
Szempont továbbá, hogy lehetőleg kevés, dimenziócsökkentés által nem érintett adatra támaszkodjon: ennek oka az, hogy a jó modell az idő múlásával könnyedén frissíthető legyen, ennek érdekében pedig ne kelljen legrosszabb esetben négy különböző ország (Thaiföld, Malajzia, Indonézia, Szingapúr) statisztikai hivatalának minden, a mostani elemzésembe bevont adatát megszerezni. Ez a szempont persze csak akkor érvényes, ha az első szempont is érvényesül (vagy legalábbis a legjobb modellnél nem sokkal rosszabb eredményeket tud felmutatni a második szempontnak megfelelő modell).

7.2 Numerikus célváltozós modellek értékelése

Az általános értékelési szempont fényében két fő modellt kerestem: egy olyat, amely dimenziócsökkentett adatokon futott le, valamennyivel nagyobb pontosságot elérve, mint amit a másik, RFE-nek alávetett adatállományon lefuttatott modell. Mindkettőnél szempont volt viszont a jó általánosítóképesség. Nyitva hagytam a lehetőséget azelőtt, hogy ez a két modell egy és ugyanaz, azonban a modellek kiértékelése során bebizonyosodott, hogy ez nem opció.

A modellek kiértékelése két lépésben zajlott: először kigyűjtöttem az adott modellek tesztadatokon nyújtott MSE-jét (itt figyelembe kellett venni a különböző skálázással előállított adatállományokat, adott MSE más minőséget képvisel egy robosztus, illetve egy min-max normalizálással előállított adatállományon lefuttatott modell esetében), majd az ígéretesek közül a tanító-és tesztadatokon nyújtott teljesítmény vizualizációja alapján kiválasztottam egy-egy olyan modellt, mely megfelelt a fentebb tárgyalt kritériumoknak.

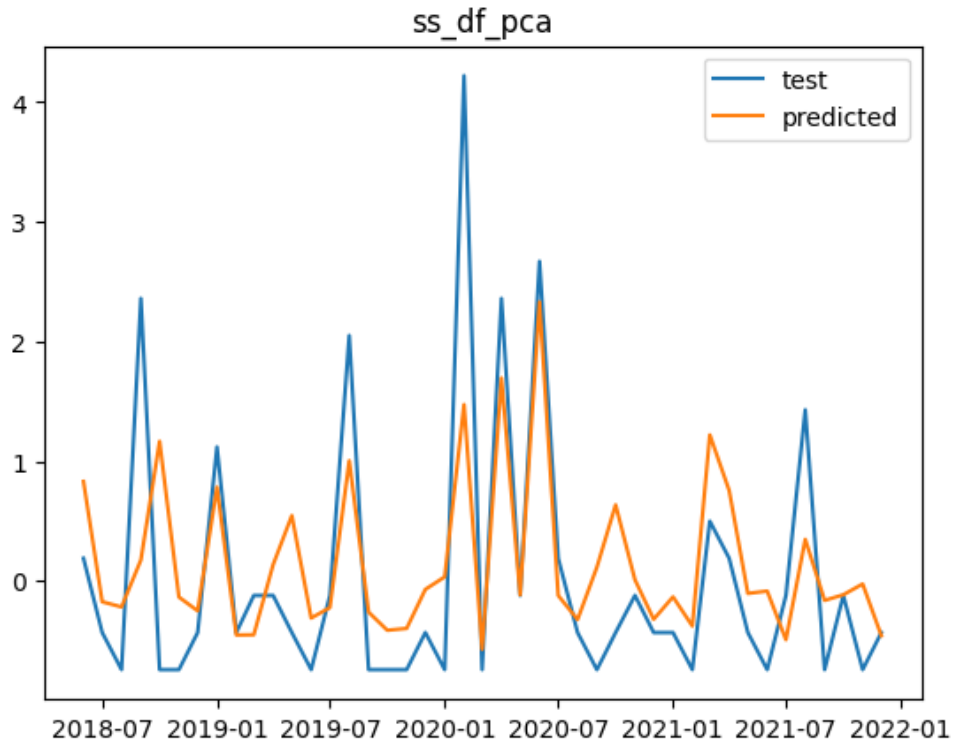
A keresés során a skálázásra különösen figyelnem kellett: az ígéretes MSE alapján jónak tűnő teljesítményt árnyalta, hogy a skálázás gyakran eltüntette a célváltozó kilengéseinek javát, így az azon adatokon elért teljesítmény csak látszólag volt megfelelő.



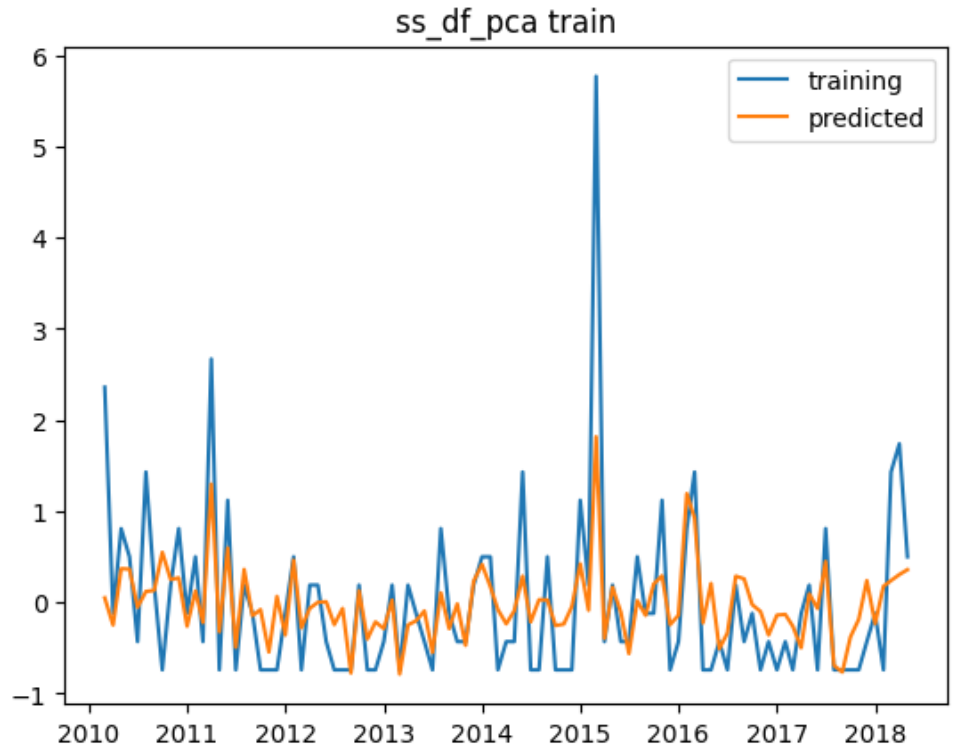
Forrás: saját szerkesztés az adatállományokon (lásd: 5.2.6.1. fejezet) lefuttatott modellek eredményei alapján

A fentebbi ábra a robosztus skálázással előállított adatállományon lefuttatott lineáris regressziós modell tesztadatokon nyújtott teljesítményét ábrázolja az előrejelzett és a valódi célváltozó összehasonlításával. Erre a problémára mindhárom skálázási módszer esetén akadt példa.

Mindennek fényében egy megfelelően teljesítő, skálázási problémák által nem érintett, megfelelő általánosítóképességgel bíró modell a mind a négy ország adatait tartalmazó, min-max skálázással előállított, adott változóból a lagged és nem lagged változó ütközése esetén mindkettőt megtartó, PCA-alapú dimenziócsökkentésnek alávetett, az adathalmaz 70-30 arányban tanító-és tesztadatokra bontott változatán lefuttatott, korai leállással kombinált LASSO regresszió volt, a polinom fok 2, az alfa pedig 1. A modell által a tesztadatokon elért MSE 0.607.



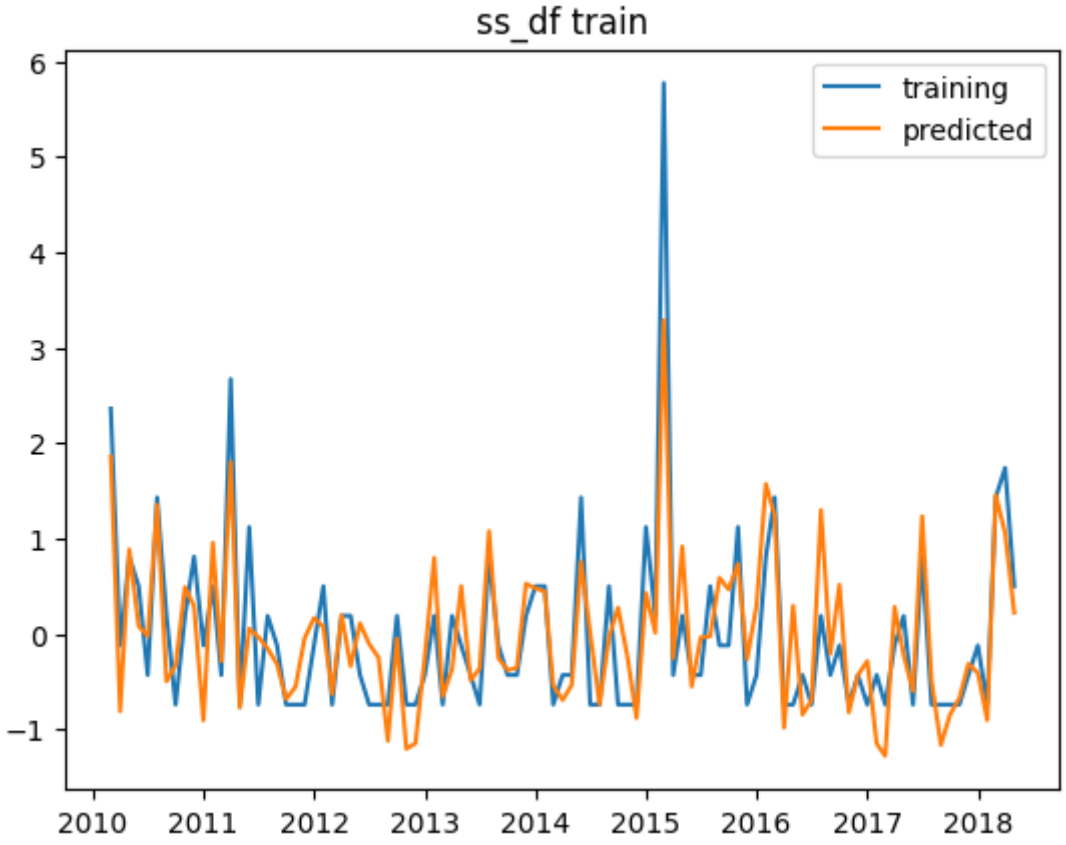
Forrás: saját szerkesztés az adatállományokon (lásd: 5.2.6.1. fejezet) lefuttatott modellek eredményei alapján



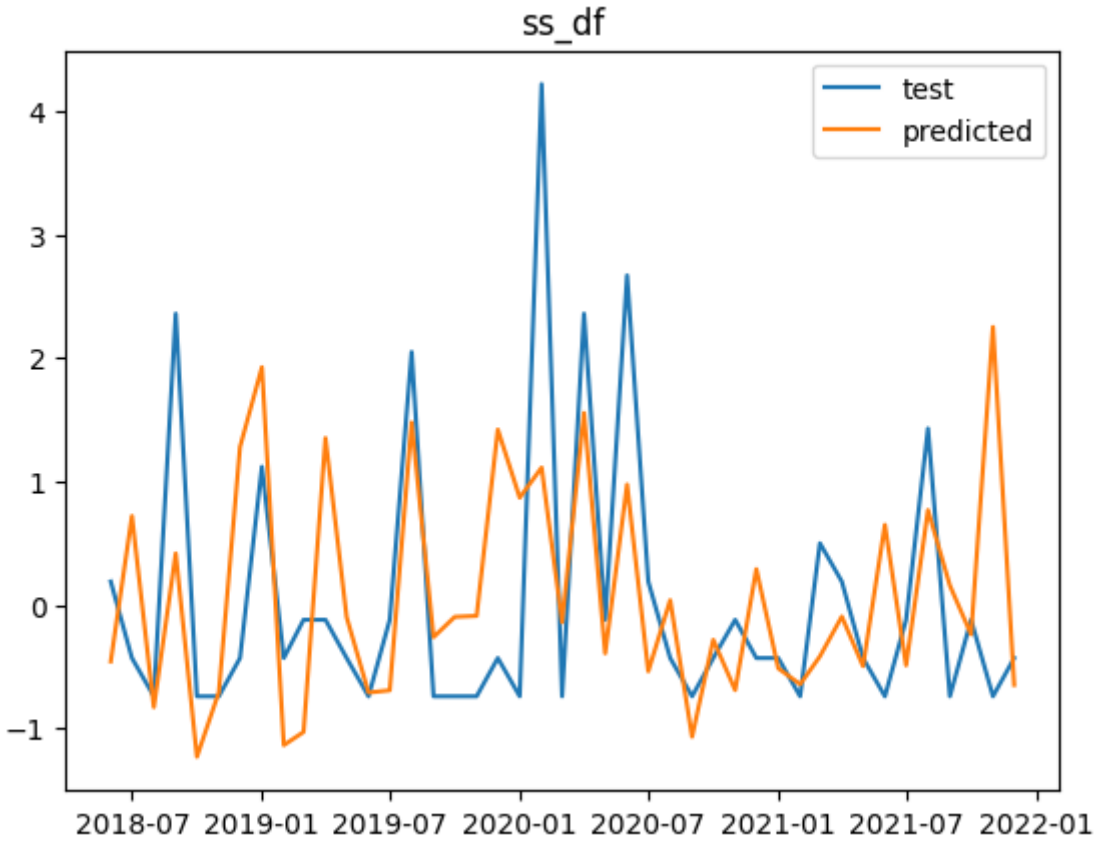
Forrás: saját szerkesztés az adatállományokon (lásd: 5.2.6.1. fejezet) lefuttatott modellek eredményei alapján

A fentebbi ábrákon a kiválasztott modell tanító-és tesztadatokon nyújtott teljesítménye figyelhető meg. Bár van példa arra, hogy a támadások számát túlbecsüli (bár inkább arra, hogy alul, ha nem is sokkal), a növekedés irányát kevés hibával, konzisztensen jól mutatja, ami kiemelten fontos, amennyiben a modell SOMS-béli ellenőrzőjáratok megtervezésére kerül felhasználásra.

A másik, kevesebb változó segítségével előállított, így a jövőben könnyebben frissíthető modell a standardizálással előállított, a lagged és nem lagged változók ütközése esetén mindkettőt megtartó, a Thaiföld adatait nem tartalmazó adathalmaz 70-30 arányban tanító-és tesztadatokra bontott változatán lefuttatott RFE-vel kombinált lineáris regresszió volt. Az alábbi ábrákon a modell tanító-és tesztadatokon nyújtott teljesítménye figyelhető meg. A modell által a tesztadatokon elért MSE 1.146 volt.



Forrás: saját szerkesztés az adatállományokon (lásd: 5.2.6.1. fejezet) lefuttatott modellek eredményei alapján



Forrás: saját szerkesztés az adatállományokon (lásd: 5.2.6.1. fejezet) lefuttatott modellek eredményei alapján

Bár ez a modell valamennyivel kevésbé pontos becslést ad, mint az első modell, a minőségbeli különbségért kárpótol, hogy kevés tényező alapján állította elő a modellt, így az könnyen frissíthető, illetve hogy 70-30-as tanító-teszt felbontás mellett is pontos becsléseket képes tenni.

A modell által felhasznált változók a következők:

- Bűnesetek száma az indonéz Juma, Nusa Tenggara Barat, Kalimantan Tengah és Kaminatan Selatan rendőri területi illetékességekben.
- Malajzia: leading diffusion (lásd: 3.1.3.) lagged változata.
- Szingapúri hajóforgalomra vonatkozó statisztikák (Általános rakomány ezer tonnában).
- Az indonéz Bali tartomány GRDP-jének mezőgazdasági, erdészeti és halászati szektorai által megtermelt része.
- Maláj infláció, GDP-deflátor (éves %)
- Szingapúri egészségügyre és nagykereskedelemre vonatkozó CPI.
- Lagged munkanélküliség az indonéz Bengkulu, Kaminatan Selatan, Di Yogyakarta és Sulawesi Utara tartományokban.

7.3 Klasszifikációs modellek értékelése

A klasszifikációs modellek értékelését is a 4.4.1.-ben említett szempontok alapján végeztem el. A mérőszám, ami alapján a legjobbnak ítélt klasszifikációs modellt választottam, a pontosság volt, ami azt mérte, a modell az adatok hány százalékát osztályozta helyesen. A megfelelő pontosságot elérő modellt annak F1-pontszáma, azaz a pontosság és a recall harmonikus átlaga (developers.google.com, 2021) alapján is értékeltem, így kezelve a lehetőséget, miszerint a két kategória kiegyensúlyozatlan aránya okozná a magas pontosságot.

Figyelembe vettem mind a tanító-, mind a tesztadatokon elért pontosságot, hogy kiszűrjem azon eseteket, ahol a korrekt tesztadatokon elért pontosság egy túltanult modell révén valósult meg (például ahol 100%-os volt), illetve azokat, ahol a tesztadatokon elért pontosság magasabb volt a tréningadatokon elértnél, ami a megbízhatatlanság jele.

A legjobban teljesítő, bár sok felhasznált változója miatt nehezebben karbantartható modell egy gaussi Naiv Bayes-modell volt, mely egy min-max módszerrel normalizált és PCA-val dimenziócsökkentésnek alávetett, ütközés esetén a nem lagged adatokat eldobó, Thaiföld adatait tartalmazó, 80-20 arányban tanító-és tesztadatokra bontott adatállományon futott le. A tesztadatokon 86,2%-os, a tanítóadatokon pedig 87,61%-os pontosságot ért el. Az előbbihez tartozó F1-pont 0.8 volt, ami megfelelő, mivel az adathalmaz nagyobb mintaegyed-számú kategóriájába annak 61%-a tartozik.

A második általam választott klasszifikációs modell egy, a logisztikus regressziót RFE-vel ötvöző modell volt, mely egy Thaiföld adatait nem tartalmazó, standardizált, dimenziócsökkentésnek alá nem vetett, ütközés esetén csak a lagged adatokat megtartó, 80-20 arányban tanító-és tesztadatokra bontott adatállományon futott le, annak tesztadatain 80,64, tanítóadatain 87,8%-os pontosságot elérve.

A modell által felhasznált változók a következők:

- A Kepulauan Bangka Belitung, Jawa Tengah, Di Yogyakarta, Nusa Tenggara Barat, Kalimantan Timur, Sulawesi Selatan, Maluku és Maluku Utara indonéz tartományok GRDP-jét mezőgazdaság, erdészet és halászat által kitevő része.
- A bűnesetek száma Lampung, Metro Jaya, Nusa Tenggara Barat, Sulawesi Utara és Maluku Utara rendőri területi illetékességekben, illetve Indonézia egészében.

A tanítóadatokon átlagosan legjobban az SVM-alapú modellek teljesítettek: a tesztadatokon elért átlagos pontosságuk 74,38% volt. Őket az RFE-alapú modellek követték, melyek átlagos

pontossága 71,43% volt. Fontos azonban megjegyezni, hogy az SVM-alapú modellek pontossága átlagosan 96,99% felett volt, így általánosítóképességük rosszabb. 71,12%-os tesztadatokon mért átlagos pontossággal a harmadik helyen a döntési fa-alapú modellek álltak.

A kategóriák nem átlag mentén, hanem – a csoportok tagjainak száma közti egyenlőtlenséget kezelendő – a növekvő sorba rendezett havi incidensek darabszámát kettéosztva „alacsony" és „magas" osztályokba való osztása révén kialakított osztályt célváltozóként használva is lefuttattam a modelleket. Az ennek eredményeképp kapott legjobban teljesítő modell a minmax normalizálásnak alávetett, ütközés esetén a nem lagged változókat eldobó, PCA-val dimenziócsökkentésnek alávetett, thai adatokat el nem dobó adatállományon lefuttatott lineáris kernelű SVM-modell volt, mely a tesztadatokon 78,79%-os, a tanítóadatokon 80%-os F1-pontot ért el, tesztadatokon mért pontossága 77,4%-os, a tanítóadatokon pedig 81,15%-os volt.

7.4 Modellek eredményeinek értelmezése

Az alábbiakban a modellek és egyéb statisztikai tesztek eredményeinek értelmezésére fog sor kerülni, a szakdolgozat céljait figyelembe véve.

A 2.1.2.-ben említett, a kalóztámadások prevalenciáját gazdasági okokkal magyarázó elméletek az általam lefuttatott modellek alapján helyesnek tűnnek: az általam választott második klasszifikációs modell a tesztadatokon 80%-os pontosságot ért el csupán egyes indonéz tartományok halászatból, erdészetből és mezőgazdaságból származó GRDP-je, illetve az egyes rendőri területi illetékességekben megesett bűnesetek alapján. Az említett területi illetékességek és provinciák kivétel nélkül vízpartiak (citypopulation.de, dátum nélk.), jelentős részük (Metro Jaya kivételével az összes) vidéki, melyek a halászat jelenlétének kedvező tényezők. Mindennek fényében a halászat gazdasági teljesítménye jó eséllyel magyarázó tényezője az incidensek számának.

A második kiválasztott numerikus célváltozós modell RFE-vel elvégzett változók közti szűrése is sok indonéz gazdasági és társadalmi adatot hagyott meg, köztük a fentebb említett klasszifikációs modellnél is meglévő bűnesetek számát, egyes tartományok halászat, mezőgazdaság és erdészet által kitermelt GRDP-jét, illetve egyes tartományok munkanélküliségi rátáját.

Az indonéz adatok mellett Malajziára vonatkozó gazdasági mutatók is megmaradtak. Fontos megjegyezni, hogy a 2010-es években Indonézia és Malajzia fontosabb gazdasági mutatói, úgymint az inflációs ráta (imf.org-2, 2024) és a GDP növekedés mértéke (imf.org-3, 2024) hasonlóképpen mozogtak, így az RFE által kiválasztott inflációs ráta a maláj helyzet mellett az

indonéziainak is indikátora. Tekintve azonban, hogy több Indonéziára vonatkozó gazdasági mutatónál jobbra értékelte a modell ezen változókat, azt a következtetést vonnám le, hogy a maláj gazdasági adatok is előjelezhetik, illetve okozhatják az incidensek előfordulását, ha nem is olyan mértékben, mint az indonéz adatok.

Ez is megerősíti a 2.1.2.-ben említett elméleteket.

Az indonéz adatok RFE általi kiválasztódása, így erős előrejelző léte nem magyarázható csupán az adatállományokon belüli súlyával: az RFE-modellek (a legjobb kettő mellettiek is) súlyához képest is sokszor választották őket a modellalkotásra leginkább alkalmas adatok közé.

A fenti adatokból látszólag kirajzolódik, hogy az indonéz gazdasági, politikai és társadalmi helyzet nagyobb mértékben határozza meg az incidensek számát, mint a többi országé: azonban jóval kevesebb Malajziára vonatkozó adatom volt (Indonéziában sok tartományra vonatkozó adattal rendelkeztem). Az RFE által kiválasztott változókba így is beférő maláj mutatók így nagyobb súlyt kapnak, továbbá megerősítik az elméletet, miszerint a maláj gazdasági helyzet is hatással van az incidensek előfordulására.

Fontos tisztázni Szingapúr helyzetét is: az ország gazdasági, társadalmi és szociális mutatói a szakirodalom alapján nincsenek nagy hatással az ARAS-incidensek előfordulására. Az RFE által mégis kiválasztott szingapúri változók is megerősíteni látszanak ezt: csupán három mutató került bel, melyek közül egy a hajóforgalomra vonatkozik. Ez – főleg az adatállományban lévő súlyához képest – kevés.

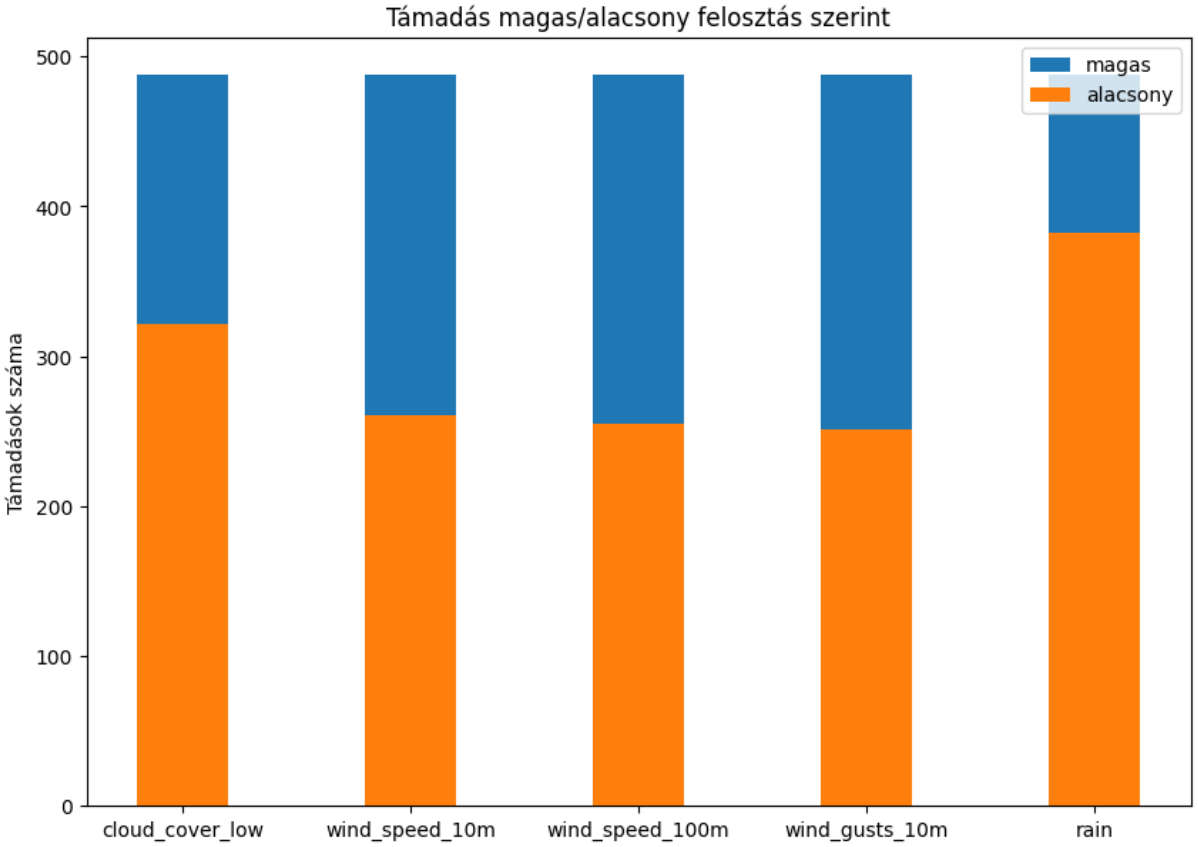
A modellek alapján Thaiföld is viszonylag kis hatással van az incidensek előfordulására: a thai adatokkal számoló RFE-alapú modellekbe kevés thai adat került be. Ez részben magyarázható azonban a többi adathoz képest kevés thai adattal. Az RFE által több esetben is kiválasztott, Thaiföldre vonatkozó adat az országra vonatkozó politikai stabilitási becslés, a GDP növekedés mértéke, illetve az általános iskolát megkezdők aránya. Mindebből feltételezhető, hogy – ha kisebb mértékben is, de a thaiföldi politikai, szociális és gazdasági mutatók befolyásolják az incidensek alakulását.

Összesítve tehát kijelenthető, hogy az indonéz és kisebb mértékben a maláj adatok (illetve még kisebb mértékben a thai adatok) alkalmasak az ARAS-incidensek előrejelzésére, ami bizonyítani látszik a kalóz-és tengeri fegyveres támadások gazdasági okairól szóló fennálló elméleteket.

7.5 Incidensekre vonatkozó adatokból levont következtetések

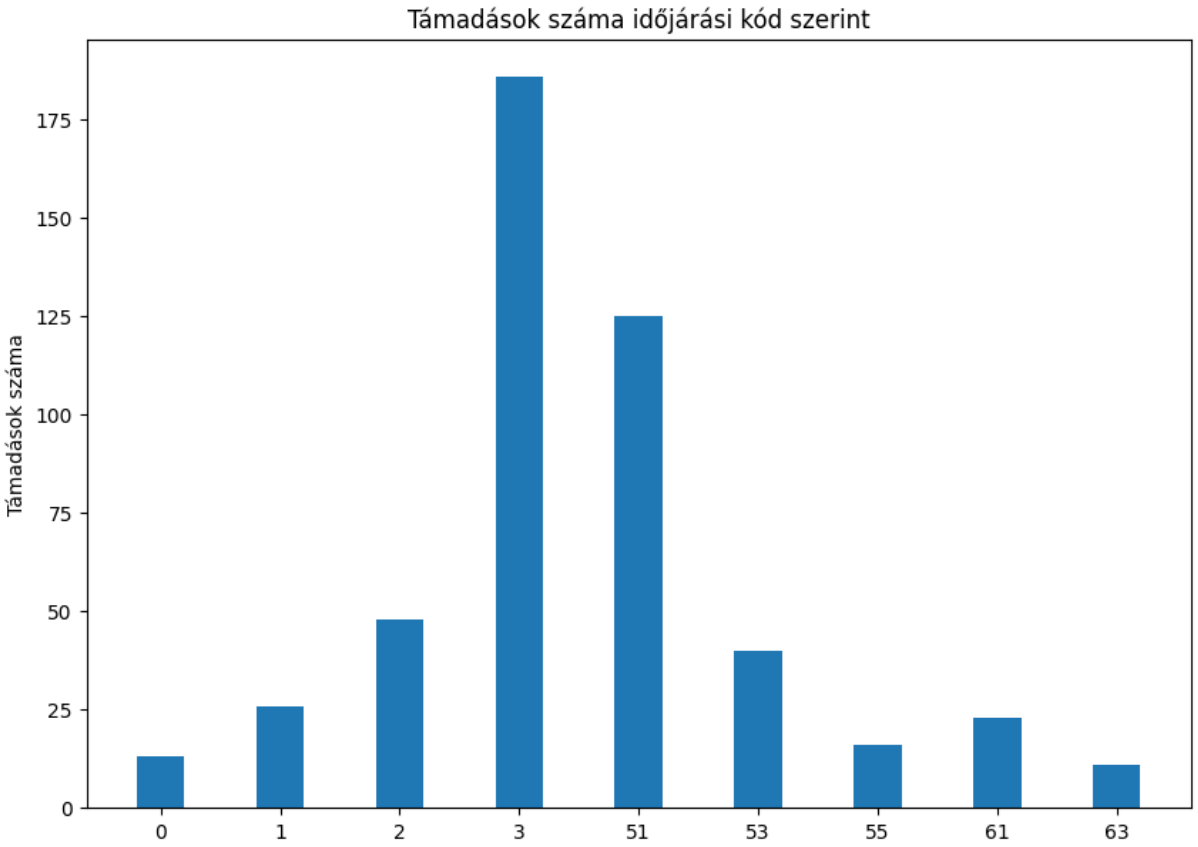
E fejezetben az adott hónapon belüli pontosabb előrejelzést szolgáló elemzéseimből született ábrák és az azokhoz tartozó magyarázat olvasható.

7.5.1 Támadások előfordulása az időjárási viszonyok függvényében



Forrás: saját szerkesztés (a ReCAAP ISC 2010-2024 közti éves adatait és az OpenMeteo API időjárási adatait felhasználva)

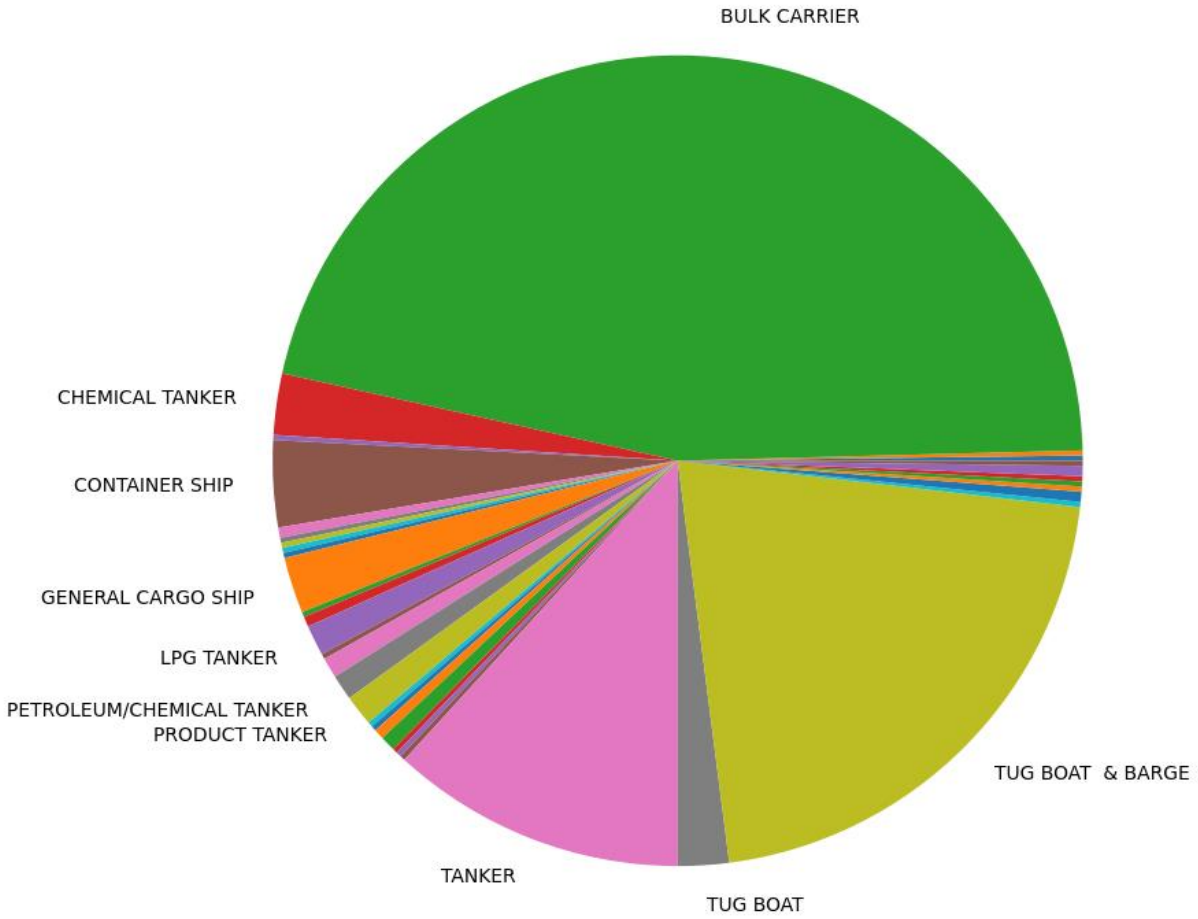
Az első ábrán az OpenMeteo API segítségével összegyűjtött időjárási adatok (az időjárási kódot leszámítva, melyre kategóriaváltozó lévén később kerül sor) átlagai mentén történő alacsony és magas kategóriák szerint összesített incidensek száma látható. Az öt tényező közül kitűnik az esőzés (rain): egyedül ennek mentén nem közel 50/50 arányban oszlanak el a támadások, bár a kapcsolat láthatólag nem erős. Nem magyarázható azonban csupán az esős napok gyakoriságával: a SOMS melletti Szingapúrban az adott év minden második napján esik (https://federicotartarini.github.io/, dátum nélk.). Egy lehetséges magyarázat a támadások esetében a nem esős időszakok felülreprezentáltságára az, hogy a más hajót megtámadni kívánók nem szívesen számolnak az eső jelentette rizikóval.



Forrás: saját szerkesztés (a ReCAAP ISC 2010-2024 közti éves adatait és az OpenMeteo API időjárási adatait felhasználva)

Az időjárási kód szerinti összesítés látszólag érdekesebb eredményeket hozott: a 3 (Felhőképződés) és az 51 (Folyamatos, szitáló, nem jeges, a megfigyelés idején kis mértékű eső) időjárási kóddal jelölt időszakokban jelentősen több támadás történik, mint a többiben. Ez azonban elsősorban azok gyakoriságával magyarázható: ez a kétféle időjárás gyakori a SOMS-ban, így ezen adatok alapján nem állapítható meg kapcsolat az időjárási kód és a kalóz-és tengeri fegyveres támadások gyakorisága közt.

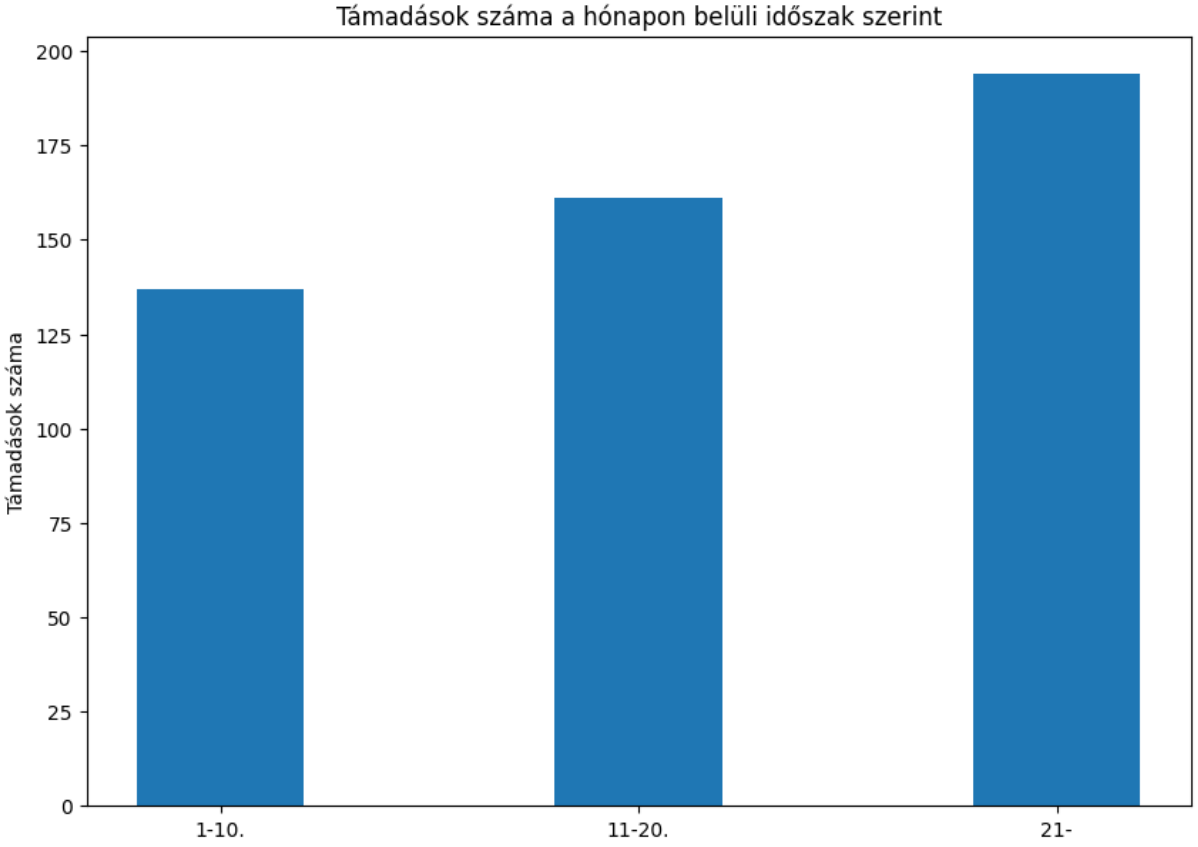
7.5.2 Támadások előfordulása hajótípus függvényében



Forrás: saját szerkesztés (a ReCAAP ISC 2010-2024 közti éves adatait felhasználva)

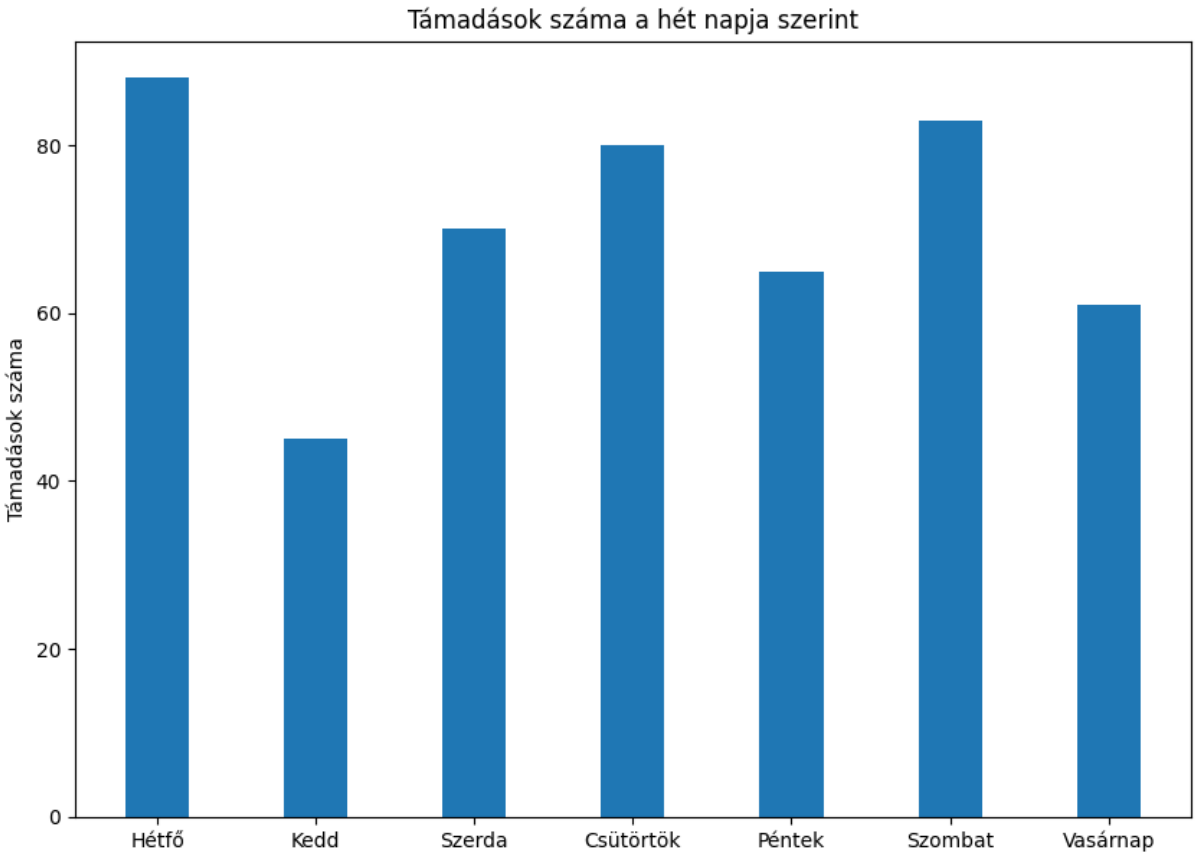
Bár a támadások adott hónapon belüli bekövetkeztének idejét nem segít megbecsülni, hasznos információt nyújt a támadások számának hajótípus szerinti ábrázolása: megállapítható, hogy a támadásoknak áldozatul eső hajók jelentős része három (azaz négy) hajótípusból áll: „bulk carrier” (ömlesztettáru-szállító hajó), „tanker” (tartályhajó, mely kapcsán megjegyzendő, hogy az „oil tanker”, azaz olajszállító hajó külön kategóriát képez), illetve az együtt mozgó „tug boat” és „barge” (vontatóhajó és uszály). Ezek közül is kimagaslóan sok támadás éri az ömlesztettáru-szállító hajókat.

7.5.3 Támadások előfordulása hónapon belüli időszakok függvényében



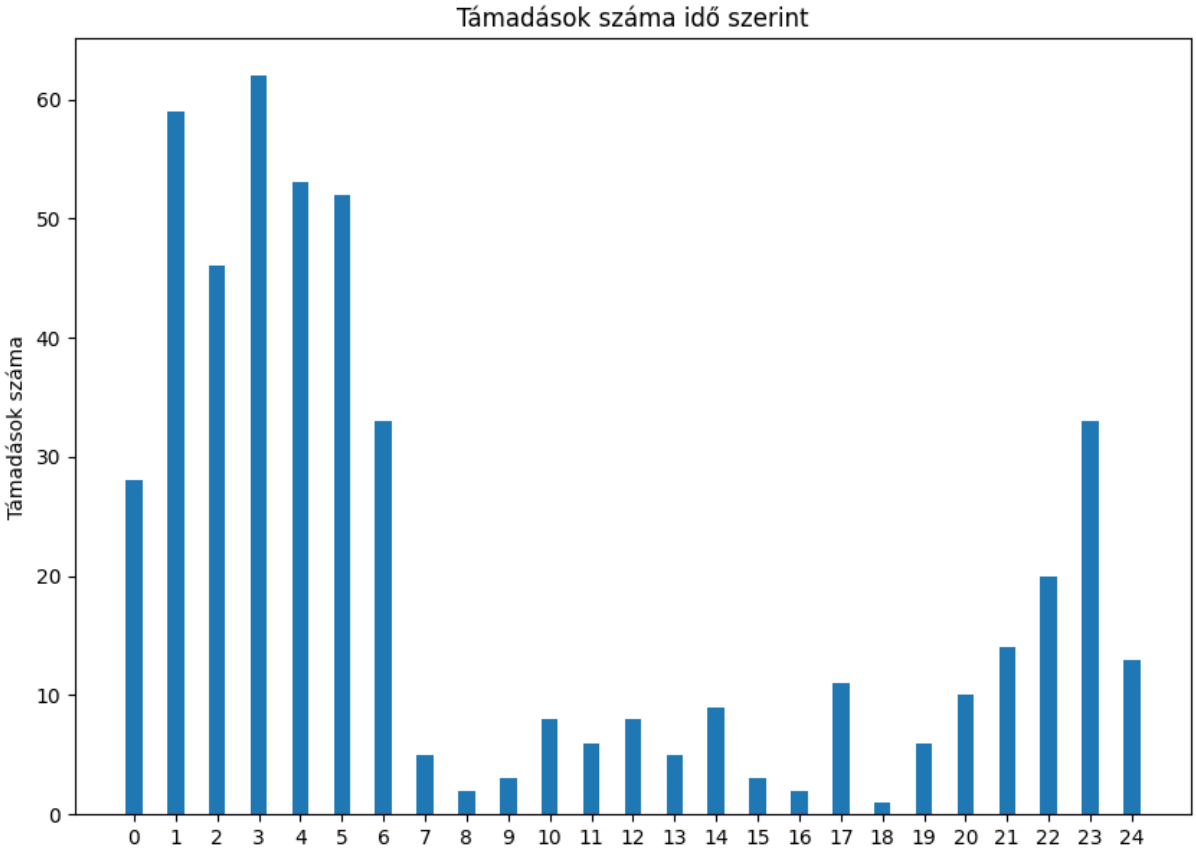
Forrás: saját szerkesztés (a ReCAAP ISC 2010-2024 közti éves adatait felhasználva)

A hónapok háromfelé való felosztása és a támadások gyakorisága között kapcsolat figyelhető meg: míg az 1-10. időszakban összesen 137 támadás történt, a 21-től hó végéig tartóban majdnem másfélszerese (1.416-szorosa), 194. A jelenségre magyarázatot nyújthat az indonéz és maláj fizetésnap: Indonéziában általában a hónap utolsó munkanapján, Indonéziában az adott hónap 7-ig történik a kifizetés. Mivel a támadásokat elsősorban a szegényebb rétegekből származók követik el, nem légbőlkapott feltételezés, hogy a legutóbbi fizetésnaptól távolodva egy, a támadáshoz hasonló rizikós döntést is nagyobb valószínűséggel hoznak meg. Ez a kapcsolat azonban nem mutatható ki egyértelműen az általam elemzett adatokból.



Forrás: saját szerkesztés (a ReCAAP ISC 2010-2024 közti éves adatait felhasználva)

A hét napja és a támadások száma közt nem mutatható ki erős kapcsolat.



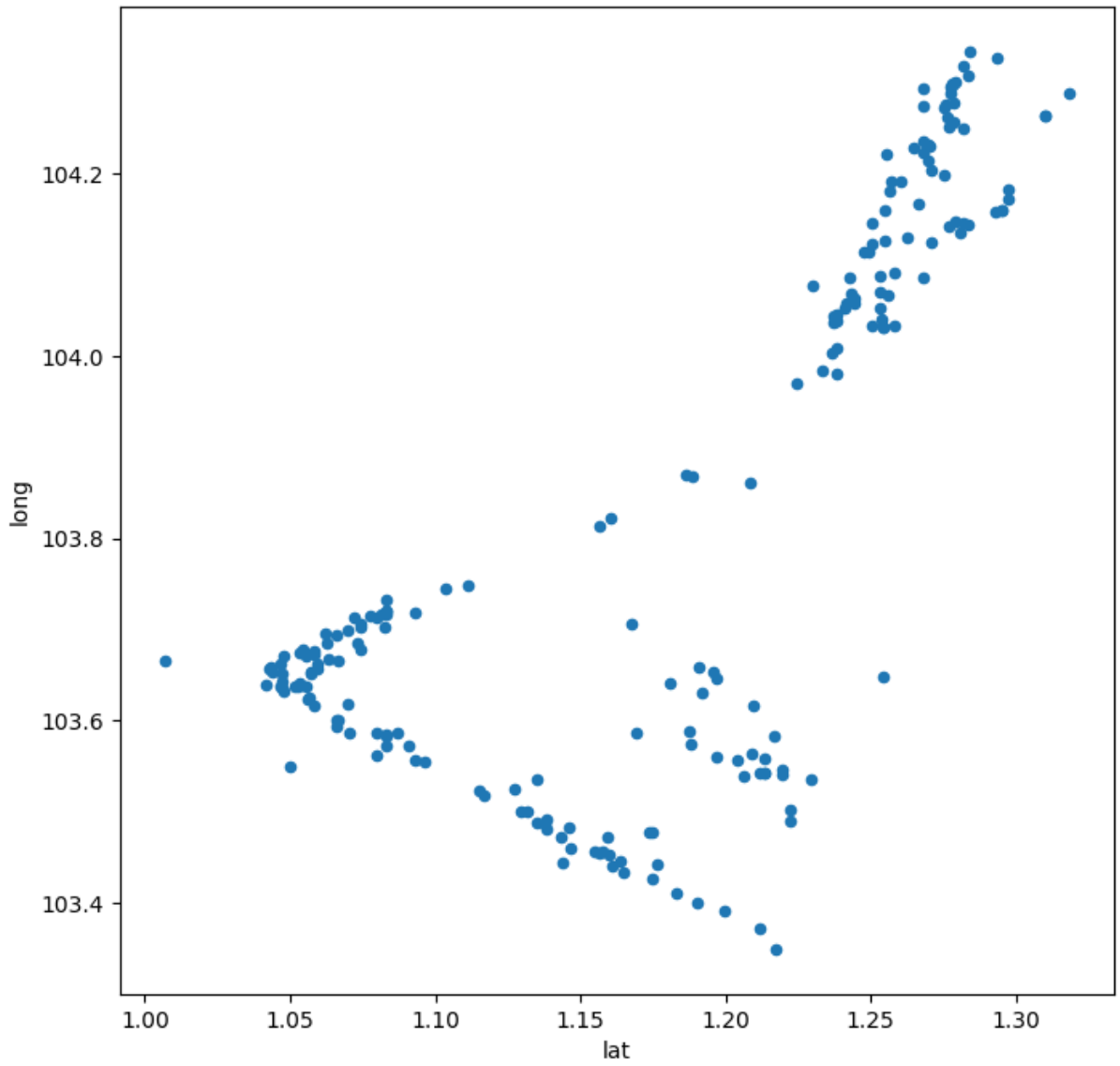
Forrás: saját szerkesztés (a ReCAAP ISC 2010-2024 közti éves adatait felhasználva)

Megfigyelhető, hogy este 9 és reggel 6 közt jelentősen több támadás történik, és ezen perióduson belül is kimagaslik az éjféltől kezdődő időszak. Ez a támadók számára kedvezőbb látási viszonyok jelenlétével magyarázható.

7.5.4 Támadások előfordulása koordináták szerint

A támadások helyszínét klaszterelemzésnek vetettem alá, azaz hosszúsági és szélességi pontjuk alapján csoportokba rendeztem. Az alábbiakban ennek folyamatát ismertetem.

Első lépésként a kiugró értékek kiszűrését végeztem el, a szakdolgozatban már említett z-score módszerrel: ennek eredményeképp a 220 incidensből hetet eldobtam, majd az azon incidenseket megjelenítő scatterplot alapján kézileg további szűréseket is elvégeztem. A szűrés eredményeképp az alábbi scatterplot-ot kaptam.

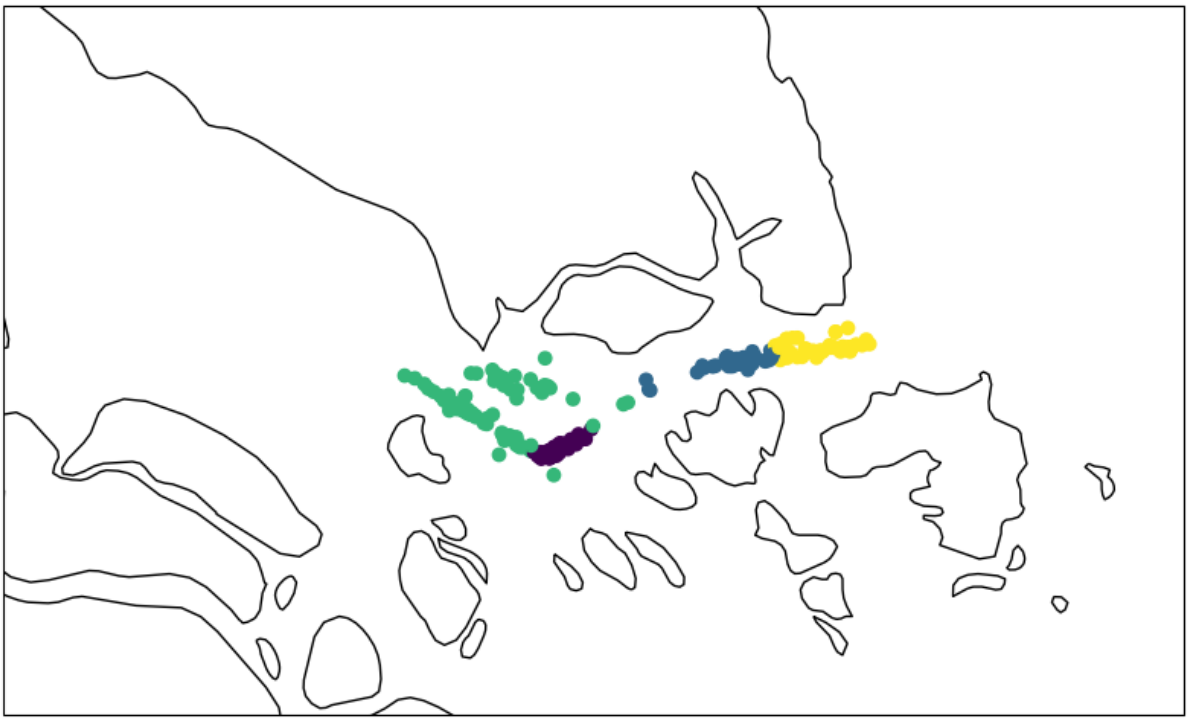


Forrás: saját szerkesztés (a ReCAAP ISC 2010-2024 közti éves adatait felhasználva)

A modell alkalmazása előtt megkerestem a könyökpontot, azaz a pontot, ahol a K-közép klaszterező algoritmus adott klaszterszám mellett megmagyarázott varianciája egy jelentős és gyors csökkenés után lassabban kezd csökkenni (ZalaRushirajsinh, 2023). Az adott klaszterszám (1 és 10 között) melletti inerciát az alábbi ábrán ábrázoltam. A könyökpont 2, azonban a fentebbi ábra fényében a két darab klasztert nem tartottam alkalmasnak a helyzet leírására és az útvonaltervezés megalapozására, így végül 4 darab klaszterrel futtattam le az algoritmust.

Forrás: saját szerkesztés (a ReCAAP ISC 2010-2024 közti éves adatait felhasználva)

A klaszterezés végeredménye a következő ábrán figyelhető meg. Az ábrán, melyen a klasztert, amelybe az adott pont tartozik, különböző színek jelöli, a cartopy Python-könyvtár segítségével térképen is megjelenítettem a koordinátákat.



Forrás: saját szerkesztés (a ReCAAP ISC 2010-2024 közti éves adatait felhasználva)

8 Előrejelzés 2025 decemberéig

8.1 Független változók előrejelzése

A független változókat ARIMA segítségével jeleztem előre 2025 decemberéig: egy havi alapú, négy ország adatait tartalmazó adatállományt a 3.2.4.-ben leírt módszerrel stacionáriussá tettem, majd a Python-béli statsmodels.tsa.arima.model könyvtár ARIMA modulja, és a pmdarima könyvtár auto_arima moduljai segítségével alkalmaztam rá az ARIMA-t.

8.2 Függő változók előrejelzése

A függő változókat, mivel diszkrétek, így ARIMA-t nem alkalmazhattam rájuk, mivel az folytonos változók előrejelzésére való (tibco.com, dátum nélk.). Ehelyett Poission-regresszió segítségével jeleztem őket előre, mely alkalmas az én célváltozómhoz hasonló (pl. darabszám) típusú adatok előrejelzésére: a Poisson-regresszió esetén a célváltozó, mivel darabszám, nem vehet fel 0 alatti értéket. Feltételezése továbbá, - mivel a Poisson-eloszláson alapul (itl.nist.gov, dátum nélk.) - hogy az alacsonyabb értékek gyakoribbak.

Az adatállományok alapján készített előrejelzéseket nem ezzel a modellel készítettem, ezt csupán az ARIMA-hoz hasonlóan a célváltozó kiterjesztésére használtam, hogy legyen mihez viszonyítani a modell eredményeit.

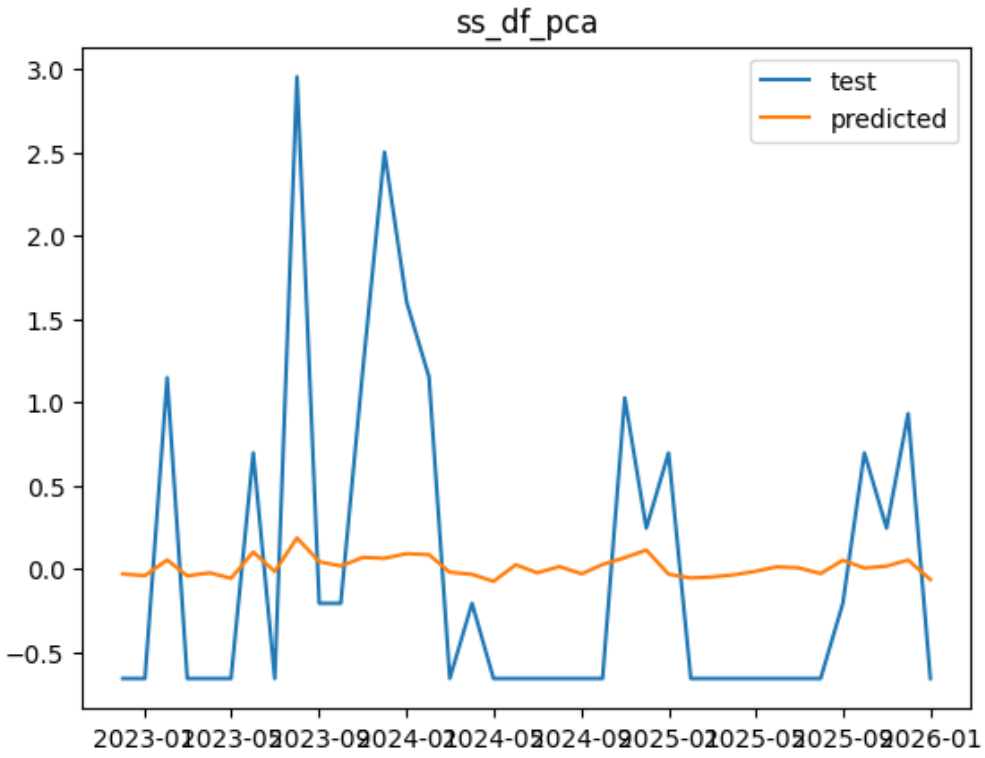
8.3 Adatelőkészítés

Az eredeti adatállományokon elvégzett összes előkészítési lépést elvégeztem: dimenziócsökkentés kétféleképpen, ismét a könyökszabály segítségével, normalizálás háromféleképpen. A lagged változók mentén ismét kétféle adatállomány jött létre: egy olyan, ahol ütközés esetén mind az eredeti, mind a lagged változat megmaradt, illetve egy olyan, ahol csak az utóbbi. A thai adatok mentén is kétfelé oszlottak az adatállományok: olyanokra, amik tartalmazták őket, illetve olyanokra, amik nem.

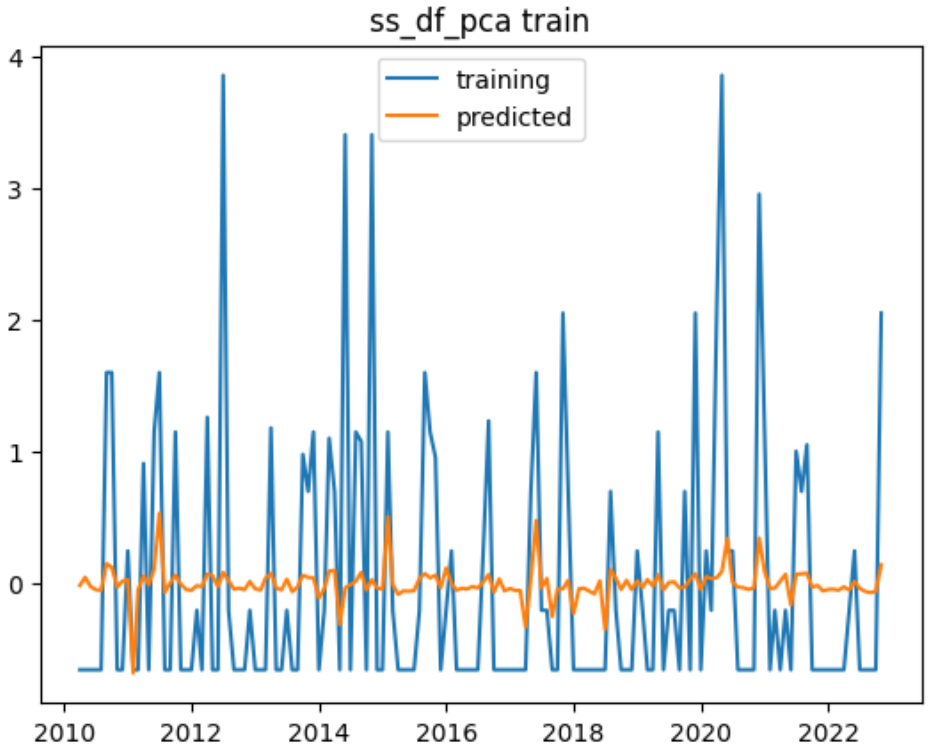
8.4 Kiválasztott modellek alkalmazása

8.4.1 Numerikus célváltozós modellek

Az eredeti, előrejelzés nélküli adatokon legjobban teljesítő numerikus célváltozós modellek egyike a 2025 végéig előrejelzett, hasonló arányban felosztott, skálázott és normalizált adatokon rosszabbul teljesített. Az alábbi ábrákon az eredeti adathalmazon legjobb eredményt hozó, korai leállással egybekötött Ridge regresszió a 2025-ig kiterjesztett adathalmazon nyújtott teljesítménye látható.

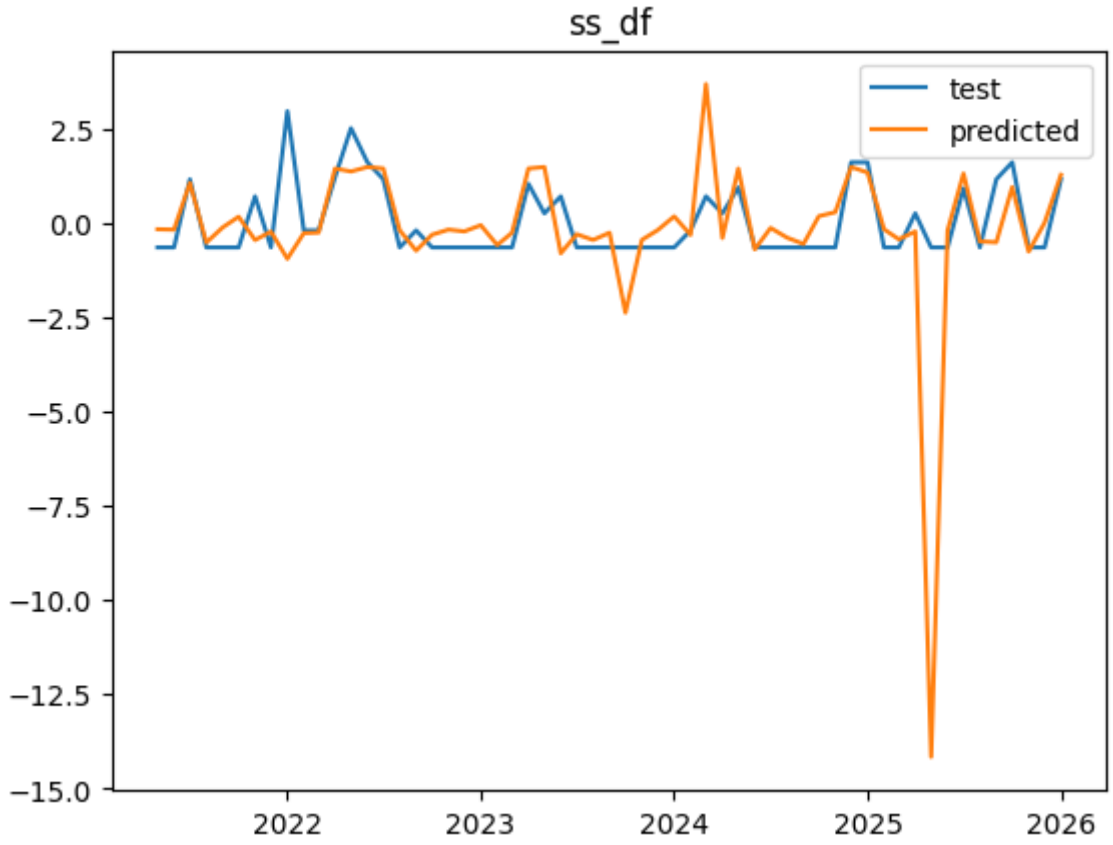


Forrás: saját szerkesztés az adatállományokon (lásd: 5.2.6.1. fejezet) lefuttatott modellek eredményei alapján

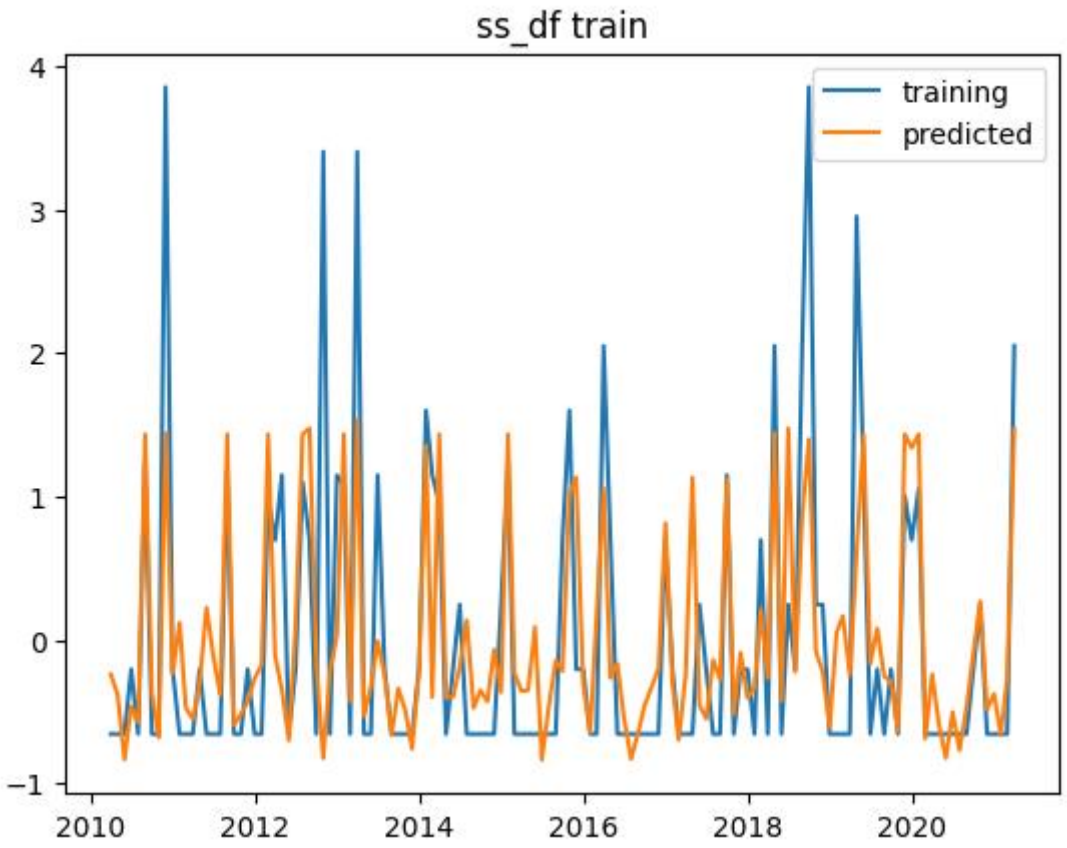


Forrás: saját szerkesztés az adatállományokon (lásd: 5.2.6.1. fejezet) lefuttatott modellek eredményei alapján

A másik, a lineáris regressziót RFE-vel összekapcsoló modell, melyet az eredetihez hasonlóan a standardizálással előállított, thai adatokat nem tartalmazó, lagged és nem változók ütközése esetén mindkettőt megtartó, 70-30 százalékban tanító-és tesztadatokra bontott adatállományon futtattam le, megfelelő eredményeket ért el:



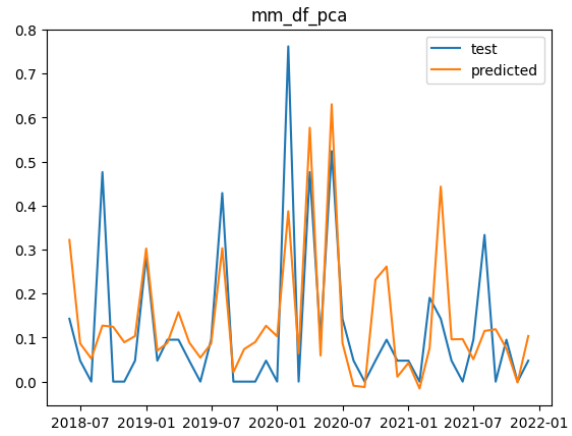
Forrás: saját szerkesztés az adatállományokon (lásd: 5.2.6.1. fejezet) lefuttatott modellek eredményei alapján



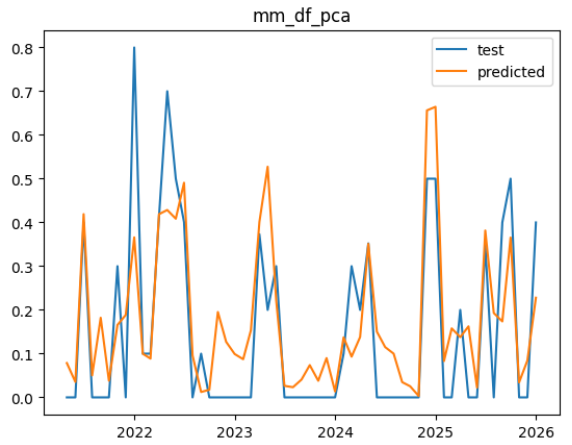
Forrás: saját szerkesztés az adatállományokon (lásd: 5.2.6.1. fejezet) lefuttatott modellek eredményei alapján

Ahogy a fentebbi ábrán megfigyelhető, a tesztadatokon futtatott modell elég (de nem túl) szorosan követi a valós (illetve 2024. decemberétől kezdve a Poission-eloszláson alapuló regresszióval előrejelzett) értékeket, két nagyobb kilengése pedig abból származik, hogy a regressziós modell nincs tisztában a jósolt változó darabszám-jellegével: ez könnyedén kikezdhető azzal, hogy a 0 alatti értékeket 0-vá alakítjuk futás után. Tesztadatokon elért MSE-je (jelenlegi formájában) 3.99, ami bár magasnak tűnik, a javát az említett egy darab, kiküszöbölhető kilengés adja.

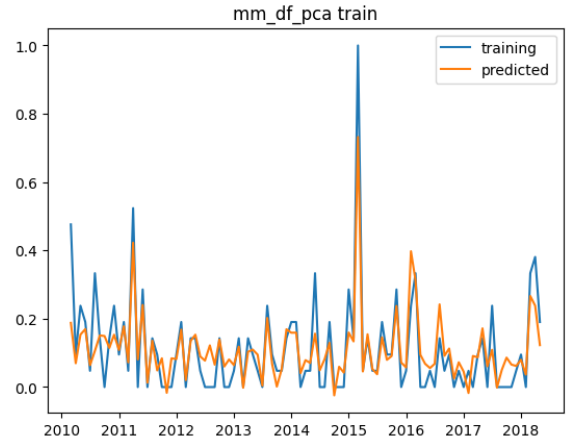
Egy modell, mely mind az eredeti, mind a 2025-ig kiterjesztett adatokon jól teljesít, a LASSO regresszió korai leállással kiterjesztett változata. A modell a min-max módszerrel normalizált, thai adatokat megtartó, PCA-val dimenziócsökkentésnek alávetett, ütközés esetén a nem lagged adatokat el nem dobó, 7-3 arányban tanító-és tesztadatokra bontott adatállományon lefuttatva a következő ábrákon illusztrált eredményeket érte el. A modellben az alfa mindkét esetben 0.1, a szabadságfok az eredeti adatok esetén 3, a kiterjesztettek esetén pedig 2 volt.



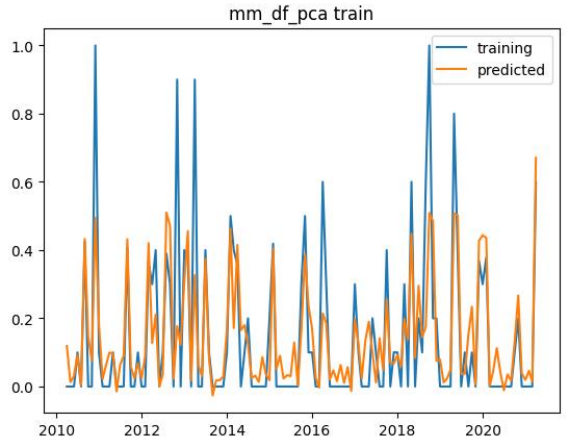
Eredeti tesztadatokon (MSE: 0.6399)



2025-ig kiterjesztett tesztadatokon (MSE: 0.2772)



Eredeti tanítóadatokon



2025-ig kiterjesztett tanítóadatokon

Forrás: saját szerkesztés az adatállományokon (lásd: 5.2.6.1. fejezet) lefuttatott modellek eredményei alapján

8.4.2 Klasszifikációs modellek

Az eredeti adatállományokon a legjobb eredményt elérő modellek itt is hasonló eredményeket értek el: a gaussi naiv Bayes-modell a min-max normalizálásnak és PCA-dimenziócsökkentésnek alávetett, thai adatokat tartalmazó és 8-2 arányban tanító-és tesztadatokra bontott adatállomány tesztadatain 86,84%-os, tanítóadatain pedig 89,47%-os pontosságot ért el. Az ezen eredményekhez tartozó F1-pontszám rendre 0,8, illetve 0,84, szóval az eredmény nem magyarázható az egyik kategória felülreprezentáltságával.

A másik, az eredeti adatállományokon jó eredményt nyújtó, mindezt RFE-vel elérő modell az új, eredetihez hasonlóan előkészített adatok esetében a tesztadatokon 92,11%-os, a tanítóadatokon 92,76%-os pontosságot ért el. Az eredményhez tartozó F1-pontok rendre 0,86, illetve 0,874 voltak.

Mivel ezen modellek mind az eredeti, mind a kibővített adatokon megfelelően teljesítettek, nem kerestem meg a kibővített adatokon legjobban teljesítő, átlag mentén kategorizált célváltozókkal dolgozó modellt.

A legjobbnak ítélt modell szerint két „magas” kategóriájú hónap lesz 2025-ben: május és november.

Az átlagon alapuló kategorizálás mellett egy, a havi incidensek számát a hármas szám mentén osztályozó (a háromnál nagyobb elemek az egyik, a többi a másik csoportba került) adatállományra is lefuttattam a modelleket, melyek az így szétválasztott modelleken is jól teljesítettek: egy, az adatokat 80-20 arányban tanító-tesztadatokra bontó, mind a négy ország adatait tartalmazó, standardizálással normalizált, dimenziócsökkentésnek alá nem vetett adatállományon a gaussi naiv Bayes-modell például 92%-os tesztadatokon és 93%-os tanítóadatokon mért pontosságot, illetve 0,84-es és 0,85-ös F1-pontot ért el. Ezt azért tartottam szükségesnek, hogy leteszteljem, a modelljeim képesek-e előjelezni a ritkább (190-ből 40 ilyen volt), de 3-nál több támadást számláló hónapokat is. Bebizonyosodott, hogy igen, bár az átlag mentén kategorizált célváltozón alapuló modelleket a térségen belüli járőrözések megszervezésében szignifikánsabbnak tartom.

8.4.3 Megjegyzések

A 2024 szeptembere utáni adatok a célváltozó esetében Poisson-regresszió, a független változók esetében pedig ARIMA segítségével álltak elő, így az adatok ezen részein (is) alapuló pontossági és egyéb metrikákat érdemes fenntartással kezelni.

9 Összefoglalás és a jövő lehetőségei, kihívásai

A továbbiakban a szakdolgozatom folyamatát és eredményeit fogom ismertetni.

A feldolgozott szakirodalom alapján összeállítottam egy, a szorosok szempontjából relevánsnak tűnő országokra – Thaiföld, Indonézia, Malajzia és Szingapúr – vonatkozó gazdasági, társadalmi és politikai adatokat tartalmazó, illetve egy, az adott incidensre vonatkozó adatokat tartalmazó adatállományt.

Az adatállományok létrejöttében fontos szerepe volt az adatok tisztításának (főleg a ReCAAP-től származó, incidensekre vonatkozó adatok esetében), szűrésének és a modellekhez való egyéb előkészületeknek, úgymint a lagged változók bevezetése, az interpoláció és a különféle dimenziócsökkentő eljárások.

Az adatállományokon alapvetően kétféle modellt futtattam le: klasszifikációsakat (logisztikus regresszió, ridge regresszió, LASSO regresszió, gradiens turbózás-alapú modell és döntési fák) és numerikusakat (lineáris, ridge, LASSO és gradiens turbózással kombinált regresszió). Ezen modelleket többféle módszerrel (jellemzőkiválasztás, hiperparaméter-optimalizáció, korai leállás) igyekeztem úgy kialakítani, hogy a legjobb eredményt érjék el, anélkül, hogy beleessenek a túltanulás csapdájába, ezzel elveszítve általánosítóképességüket.

A szakdolgozatom során, a különféle források alapján előállított adatállományokon lefuttatott modellek segítségével megállapítottam, hogy – ahogy a vonatkozó irodalomban is áll – kapcsolat mutatható ki a SOMS-béli tengeri fegyveres támadások gyakorisága és az azt körülölelő országok társadalmi (pl. indonéz bűnözési ráták) és gazdasági adatai (ezek közül a leghangsúlyosabbak az indonéz vízparti provinciák halászat, mezőgazdaság és erdészet által kitermelt GRDP-je volt) közt. A leghangsúlyosabban (az adatállományokon belüli arányukhoz képest is) az indonéz adatok jelentek meg, mint az incidensek előfordulását leginkább befolyásoló tényezők, de egyes malajziai és thaiföldi adatok is alkalmasnak bizonyultak arra, hogy részesei legyenek az incidensek előrejelzésének.

A különféle módon előállított adatállományokon több modellt is lefuttattam. Bár ezek közül több is remek eredményt ért el mind az eredeti adatállományokon, mind az ARIMA segítségével 2025 végéig tartó előrejelzéseket tartalmazókon, a mezőnyből kiemelkedtek az adatok átlagéra osztályhatárként támaszkodó klasszifikációs modellek, kiváltképp a gaussi naiv Bayes- és az RFE-vel kevert logisztikus regresszió-alapúak. Átlagosan az SVM-alapú modellek is jól

szerepeltek. Ezekkel a modellekkel jó előrejelzéseket tudtam tenni egy adott hónapra vonatkozóan.

A nem klasszifikációs modellek közt is volt olyan, mely jól teljesített: közülük kiemelendő a korai leállással kombinált LASSO-regresszió, mely mind az eredeti, mind az ARIMA és Poission-regresszió segítségével 2025 decemberéig kibővített adatállományon jól teljesített.

Az előrejelzések adott hónapon belüli pontosítására tett kísérleteim szerény sikerekkel zárultak: megállapítottam a hajótípus és a napon belüli idő erős, illetve a hónapon belüli időszak gyengébb, a támadások előfordulására gyakorolt hatását. Emellett sikerült klaszterekbe sorolnom a támadások helyszíneit, ami segíthet a hatékony járőr-útvonaltervezésben.

A szakdolgozat eredményeit látva ki merem jelenteni, hogy a gépi tanulás-alapú modelleknek –a szakdolgozatban használtaknak és általánosságban is – van szerepe a Szingapúri-és Malaka-szorosokon belüli tengeres fegyveres támadások előrejelzésében. Ennek persze vannak előfeltételei: a ReCAAP-nek többet kell tennie az adataik elemezhető formában való közzétételéért, illetve célszerű a regionális együttműködést és információmegosztást a környék országainak statisztikai hivatalaira is kiterjeszteni a modellek betaníthatóságának és frissíthetőségének érdekében.

10 Irodalomjegyzék

Analytics Vidhya. (2024. november 19). Recursive Feature Elimination (RFE): Working, Advantages & Examples. *Analytics Vidhya*. Forrás: https://www.analyticsvidhya.com/blog/2023/05/recursive-feature-elimination/

András, E. (2024. december 10). *github.com*. Letöltés dátuma: 2024. december 10, forrás: ecsand/szakdolgozat: https://github.com/ecsand/Szakdolgozat

avicksaha. (2024. június 10). Gamma Parameter in SVM. *GeeksforGeeks*. Forrás: https://www.geeksforgeeks.org/gamma-parameter-in-svm/

Bileta, V. (2022. november 27). The Seven Voyages of Zheng He: When China Ruled the Seas. *TheCollector*. Forrás: https://www.thecollector.com/zheng-he-seven-voyages/

Bobbitt, Z. (2021. augusztus 12). Z-Score Normalization: Definition & Examples. *Statology*. Forrás: https://www.statology.org/z-score-normalization/

byjus.com. (dátum nélk.). Letöltés dátuma: 2024. december 6, forrás: Interpolation | Definition, Formula, Methods & Uses: https://byjus.com/maths/interpolation/

Chowdhury, M. Z., & Turin, T. (2020. február 16). Variable selection strategies and its importance in clinical prediction modelling. *Fam Med Community Health*, 2. Forrás: https://pmc.ncbi.nlm.nih.gov/articles/PMC7032893/

citypopulation.de. (dátum nélk.). Letöltés dátuma: 2024. december 4, forrás: Indonesia: Administrative Division (Provinces, Regencies and Cities) with population statistics, charts and maps.: https://www.citypopulation.de/en/indonesia/admin/

data.gov.my-1. (2023. július 28). Letöltés dátuma: 2024. október 30, forrás: Poverty by State: https://data.gov.my/data-catalogue/hh_poverty_state

data.gov.my-1. (2023. július 28). Letöltés dátuma: 2024. 10 30, forrás: Poverty by State: https://data.gov.my/data-catalogue/hh_poverty_state

data.gov.my-2. (2024). Letöltés dátuma: 2024. november 6, forrás: Monthly Principal Labour Force Statistics: https://data.gov.my/data-catalogue/lfs_month

data.gov.my-2. (2024). Letöltés dátuma: 2024. 11 6, forrás: Monthly Principal Labour Force Statistics: https://data.gov.my/data-catalogue/lfs_month

data.gov.my-3. (2024). Letöltés dátuma: 2024. november 6, forrás: Malaysian Economic Indicator: https://data.gov.my/data-catalogue/economic_indicators

data.gov.my-4. (2024). Letöltés dátuma: 2024. november 6, forrás: Producer Price Index: https://data.gov.my/data-catalogue/ppi

databank.worldbank.org/-1. (2024). Letöltés dátuma: 2024. november 5, forrás: Indonesia Database for Policy and Economic Research: https://databank.worldbank.org/source/indonesia-database-for-policy-and-economic-research/preview/on#

databank.worldbank.org/-1. (2024). Forrás: Indonesia Database for Policy and Economic Research: https://databank.worldbank.org/source/indonesia-database-for-policy-and-economic-research/preview/on#

databank.worldbank.org-2. (2024). Letöltés dátuma: 2024. október 31, forrás: World Development Indicators | DataBank: https://databank.worldbank.org/reports.aspx?source=2&country=MYS#

databank.worldbank.org-3. (dátum nélk.). Letöltés dátuma: 2024. november 27, forrás: World Development Indicators | DataBank: https://databank.worldbank.org/reports.aspx?source=2&country=THA

data-explorer.oecd.org. (2024. március 8). Letöltés dátuma: 2024. december 3, forrás: OECD Data Explorer - Employment in fisheries, aquaculture and processing: https://data-explorer.oecd.org/vis?lc=en&df[ds]=dsDisseminateFinalDMZ&df[id]=DSD_FISH_EMP%40DF_FISH_EMPL&df[ag]=OECD.TAD.ARP&dq=.A...._T._T&pd=2010%2C&to[TIME_PERIOD]=false

data-explorer.oecd.org. (2024. március 8). Forrás: OECD Data Explorer - Employment in fisheries, aquaculture and processing: https://data-explorer.oecd.org/vis?lc=en&df[ds]=dsDisseminateFinalDMZ&df[id]=DSD_FISH_EMP%40DF_FISH_EMPL&df[ag]=OECD.TAD.ARP&dq=.A...._T._T&pd=2010%2C&to[TIME_PERIOD]=false

developers.google.com. (2021. Szeptember 8). Letöltés dátuma: 2024. december 4, forrás: Classification: Accuracy, recall, precision, and related metrics: https://developers.google.com/machine-learning/crash-course/classification/accuracy-precision-recall

Faridi, R. (dátum nélk.). Time Series Analysis. Letöltés dátuma: 2024. december 3, forrás: https://bookdown.org/rushad_16/TSA_Lectures_book/regression-models.html

fred.stlouisfed.org-1. (2024. április 15). Letöltés dátuma: 2024. november 2, forrás: Consumer Price Index: All Items: Total for Indonesia (CPALTT01IDM657N): https://fred.stlouisfed.org/series/CPALTT01IDM657N

fred.stlouisfed.org-10. (2024. október 15). Letöltés dátuma: 2024. november 7, forrás: World Uncertainty Index for Singapore (WUISGP): https://fred.stlouisfed.org/series/WUISGP

fred.stlouisfed.org-2. (2024. október 18). Letöltés dátuma: 2024. november 2, forrás: Nominal Gross Domestic Product for Indonesia (NGDPSAXDCIDQ): https://fred.stlouisfed.org/series/NGDPSAXDCIDQ

fred.stlouisfed.org-3. (2024. október 18). Letöltés dátuma: 2024. november 2, forrás: Nominal Gross Domestic Product for Indonesia (NGDPNSAXDCIDQ): https://fred.stlouisfed.org/series/NGDPNSAXDCIDQ

fred.stlouisfed.org-4. (2024. október 18). Letöltés dátuma: 2024. november 2, forrás: Real Gross Domestic Product for Indonesia (NGDPRSAXDCIDQ): https://fred.stlouisfed.org/series/NGDPRSAXDCIDQ

fred.stlouisfed.org-5. (2024. október 18). Letöltés dátuma: 2024. november 2, forrás: Real Gross Domestic Product for Indonesia (NGDPRNSAXDCIDQ): https://fred.stlouisfed.org/series/NGDPRNSAXDCIDQ

fred.stlouisfed.org-6. (2024. október 15). Letöltés dátuma: 2024. november 2, forrás: World Uncertainty Index for Indonesia (WUIIDN): https://fred.stlouisfed.org/series/WUIIDN

fred.stlouisfed.org-7. (2024. július 2). Letöltés dátuma: 2024. november 2, forrás: Gross Domestic Product for Malaysia (MKTGDPMYA646NWDB): https://fred.stlouisfed.org/series/MKTGDPMYA646NWDB

fred.stlouisfed.org-8. (2024. október 15). Letöltés dátuma: 2024. november 2, forrás: World Uncertainty Index for Malaysia (WUIMYS): https://fred.stlouisfed.org/series/WUIMYS

fred.stlouisfed.org-9. (2024. július 2). Letöltés dátuma: 2024. november 2, forrás: Inflation, consumer prices for Malaysia (FPCPITOTLZGMYS): https://fred.stlouisfed.org/series/FPCPITOTLZGMYS

GeeksforGeeks. (2024. március 20). What is Adam Optimizer? *GeeksforGeeks*. Forrás: https://www.geeksforgeeks.org/adam-optimizer/

GeeksForGeeks. (2024. szeptember 19). What is Lag in Time Series Forecasting. *GeeksForGeeks*. Letöltés dátuma: 2024. december 3, forrás: https://www.geeksforgeeks.org/what-is-lag-in-time-series-forecasting/

Gupta, L. (2024. március 29). Time Series Forecasting (Stationarity , Differencing , Transformations). *Medium*. Forrás: https://medium.com/@lahar091103/time-series-forecasting-stationarity-differencing-transformations-c4e2d52ddd47

Haire, P. (2021. január 25). The Currents in Singapore Strait are Extremely Complex. Here's Why. *Tidetech*. Forrás: https://www.tidetechmarinedata.com/news/the-complex-currents-in-singapore-strait

https://federicotartarini.github.io/. (dátum nélk.). Letöltés dátuma: 2024. december 5, forrás: Singapore's climate: https://federicotartarini.github.io/air-quality-weather-sg/climate-of-singapore/

https://www.recaap.org. (dátum nélk.). Letöltés dátuma: 2024. december 5, forrás: About ReCAAP Information Sharing Centre: https://www.recaap.org/about_ReCAAP-ISC

https://www.singstat.gov.sg/-1. (2024). Letöltés dátuma: 2024. november 7, forrás: Singapore Department of Statistics (DOS) | SingStat Table Builder: https://tablebuilder.singstat.gov.sg/table/TS/M651101

https://www.singstat.gov.sg/-2. (2024). Letöltés dátuma: 2024. november 7, forrás: Singapore Department of Statistics (DOS): https://tablebuilder.singstat.gov.sg/table/TS/M015651

https://www.singstat.gov.sg/-3. (2024. október 29). Letöltés dátuma: 2024. november 7, forrás: Singapore Department of Statistics (DOS): https://tablebuilder.singstat.gov.sg/table/TS/M182332

IBM. (2023. december 8). What is principal component analysis (PCA)? *IBM*. Letöltés dátuma: 2024. december 3, forrás: https://www.ibm.com/topics/principal-component-analysis

imf.org-2. (2024. október). Letöltés dátuma: 2024. december 4, forrás: World Economic Outlook (October 2024) - Inflation rate, average consumer prices: https://www.imf.org/external/datamapper/PCPIPCH@WEO/IDN/MYS

imf.org-3. (2024. október). Letöltés dátuma: 2024. december 4, forrás: World Economic Outlook (October 2024) - Real GDP growth: https://www.imf.org/external/datamapper/NGDP_RPCH@WEO/IDN/MYS

itl.nist.gov. (dátum nélk.). Letöltés dátuma: 2024. december 9, forrás: 1.3.6.6.19. Poisson Distribution: https://www.itl.nist.gov/div898/handbook/eda/section3/eda366j.htm

Kanade, V. (2023. április 3). What Is Linear Regression? Types, Equation, Examples, and Best Practices for 2022. *Spiceworks*. Forrás: https://www.spiceworks.com/tech/artificial-intelligence/articles/what-is-linear-regression/

keras.io. (dátum nélk.). Letöltés dátuma: 2024. december 3, forrás: Dense layer: https://keras.io/api/layers/core_layers/dense/

Kuknyó, D. (2024). Üzleti Elemzések Módszertana 3. Gyakorlat: Regularizált modellek. 17. Forrás: https://github.com/basictask/Elemzesmodszertan/blob/main/3_regularization/doc/3_regularization.pdf

Kuknyó, D. (2024). Üzleti Elemzések Módszertana 2. Előadás: Osztályozás. Letöltés dátuma: 2024. december 3, forrás: https://github.com/basictask/Elemzesmodszertan/blob/main/2_classification/doc/2_classification.pdf

Kuknyó, D. (2024). Üzleti Elemzések Módszertana 3. Gyakorlat: Regularizált Modellek. Forrás: https://github.com/basictask/Elemzesmodszertan/blob/main/3_regularization/doc/3_regularization.pdf

Kuknyó, D. (2024). Üzleti Elemzések Módszertana 3. Gyakorlat: Regularizált Modellek. 17. Letöltés dátuma: 2024. december 3, forrás: https://github.com/basictask/Elemzesmodszertan/blob/main/3_regularization/doc/3_regularization.pdf

Kuknyó, D. (2024). Üzleti Elemzések Módszertana 4. Gyakorlat: Döntési fák. Letöltés dátuma: 2024. december 3, forrás: https://github.com/basictask/Elemzesmodszertan/blob/main/4_decision_trees/doc/4_decision_trees.pdf

Kuknyó, D. (2024). Üzleti Elemzések Módszertana 6. Előadás: Tartó vektor gépek. Letöltés dátuma: 2024. december 3, forrás: https://github.com/basictask/Elemzesmodszertan/blob/main/6_svm/doc/6_svm.pdf

Mangale, S. (2020. augusztus 28). Scree Plot. *Medium*. Forrás: https://sanchitamangale12.medium.com/scree-plot-733ed72c8608

Martins, C. (2023. november 2). Gaussian Naive Bayes Explained With Scikit-Learn. *Built In*. Forrás: https://builtin.com/artificial-intelligence/gaussian-naive-bayes

Nalcin, S. (2022. október 11). StandardScaler vs. MinMaxScaler vs. RobustScaler: Which one to use for your next ML project? *Medium*. Forrás: https://medium.com/@onersarpnalcin/standardscaler-vs-minmaxscaler-vs-robustscaler-which-one-to-use-for-your-next-ml-project-ae5b44f571b9

Olamendy, J. C. (2023. december 4). Understanding ReLU, LeakyReLU, and PReLU: A Comprehensive Guide. *Medium*. Forrás: https://medium.com/@juanc.olamendy/understanding-relu-leakyrelu-and-prelu-a-comprehensive-guide-20f2775d3d64

open.dosm.gov.my. (dátum nélk.). Letöltés dátuma: 2024. 11 6, forrás: CPI by Strata & Division (2-digit): https://open.dosm.gov.my/data-catalogue/cpi_strata

OpenDOSM. (dátum nélk.). Letöltés dátuma: 2024. 11 6, forrás: CPI by Strata & Division (2-digit): https://open.dosm.gov.my/data-catalogue/cpi_strata

open-meteo.com. (dátum nélk.). Letöltés dátuma: 2024. december 5, forrás: Weather Forecast API: https://open-meteo.com/en/docs

pandas.pydata.org. (dátum nélk.). Letöltés dátuma: 2024. december 6, forrás: Pandas: https://pandas.pydata.org/

Panneerselvam, P., & Ramkumar, K. (2023. május 15). Piracy and Armed Robbery in Southeast Asia: The Need for a Fresh Approach. *The Diplomat*. Forrás: https://thediplomat.com/2023/05/piracy-and-armed-robbery-in-southeast-asia-the-need-for-a-fresh-approach/

Paterson, S. (2023. július 27.). Dire straits: Malacca, Singapore and the future of the global economy. *Asia Scotland Institute*. Forrás: https://asiascot.com/articles/straits-of-malacca

Rajan, S. (2021. október 26). Dimensionality Reduction using AutoEncoders in Python. *Analytics Vidhyia*. Forrás: https://www.analyticsvidhya.com/blog/2021/06/dimensionality-reduction-using-autoencoders-in-python/

Raymond, C. Z. (2009). PIRACY AND ARMED ROBBERY IN THE MALACCA STRAIT: A Problem Solved? *Naval War College Review*, 2. Letöltés dátuma: 2024. 12 3, forrás: https://www.jstor.org/stable/26397033

ReCAAP ISC. (2024). *ReCAAP ISC Annual Report 2023.* RecAAP ISC. Forrás: https://www.recaap.org/resources/ck/files/reports/annual/ReCAAP%20ISC%20Annual%20Report%202023.pdf: https://www.recaap.org/resources/ck/files/reports/annual/ReCAAP%20ISC%20Annual%20Report%202023.pdf

recaap.org. (dátum nélk.). Letöltés dátuma: 2024. december 5, forrás: ReCAAP ISC: https://www.recaap.org/resources/ck/files/Number%20of%20Incidents/2023/List%20of%20Incidents%20for%202023.pdf

Riswanto, U. (2023. május 18). What Is An Autoencoder? Why It's So Important For Dimensionality Reduction. *Medium*. Forrás: https://ujangriswanto08.medium.com/what-is-an-autoencoder-why-its-so-important-for-dimensionality-reduction-9ba723b34c2b

scikit-learn.org. (dátum nélk.). Letöltés dátuma: 2024. december 3, forrás: Ridge: https://scikit-learn.org/1.5/modules/generated/sklearn.linear_model.Ridge.html

scikit-learn.org. (dátum nélk.). Letöltés dátuma: 2024. december 6, forrás: sklearn.ensemble — scikit-learn 1.5.2 documentation: https://scikit-learn.org/1.5/api/sklearn.ensemble.html

Shelar, V. (2023. szeptember 10). *"Ridge Regression : Empowering Predictions with Ridge Regression"*. Forrás: medium.com: https://medium.com/@vishalshelar328/ridge-regression-empowering-predictions-with-ridge-regression-58f1075e548a

singstat.gov.sg-4. (2024). Letöltés dátuma: 2024. november 7, forrás: (DOS) | SingStat Table Builder – Consumer Price Index (CPI), 2019 As Base Year: https://tablebuilder.singstat.gov.sg/table/TS/M212881

Spiess, R. (2019. július 16). A pirate's paradise. *Southeast Asia Globe*. Letöltés dátuma: 2024. 12 3, forrás: https://southeastasiaglobe.com/how-corruption-is-fuelling-modern-day-piracy/

Storey, I. (2022. augusztus 15). Piracy and the Pandemic: Maritime Crime in Southeast Asia, 2020-2022. *Fulcrum*. Forrás: https://fulcrum.sg/piracy-and-the-pandemic-maritime-crime-in-southeast-asia-2020-2022/

Thakar, C., & Tahsildar, S. (2022. november 22). Gini Index: Decision Tree, Formula, Calculator, Gini Coefficient in Machine Learning. *Quantitative Finance & Algo Trading Blog by QuantInsti*. Forrás: https://blog.quantinsti.com/gini-index/

The Nippon Foundation. (dátum nélk.). *Safety in the Straits of Malacca and Singapore*. Letöltés dátuma: 2024. december 3, forrás: https://www.nippon-foundation.or.jp: https://www.nippon-foundation.or.jp/en/what/projects/safe_passage

tibco.com. (dátum nélk.). Letöltés dátuma: 2024. december 9, forrás: ARIMA Models and Forecasting: https://docs.tibco.com/pub/stat/14.0.0/doc/html/UsersGuide/GUID-BCF27023-F83B-4327-9271-9D760B67470D.html

Verma, N. (2023. november 6). *Gradient Boosting Regression Implementation in Python for Predictive Modeling.* Letöltés dátuma: 2024. december 3, forrás: medium.com: https://medium.com/@nandiniverma78988/gradient-boosting-regression-implementation-in-python-for-predictive-modeling-437e4ece8c9e

Watson, W. H. (2013. június 13). Singapore at Heart of Counter Piracy Worldwide. *MarineLink*. Forrás: https://www.marinelink.com/news/singapore-worldwide355616

Wintergalen, E. W., Oyanedel, R., Villaseñor-Derbez, J. C., Fulton, S., & Molina, R. (2022). Opportunities and challenges for livelihood resilience in urban and rural Mexican small-scale fisheries. *Ecology and Society*. Forrás: https://ecologyandsociety.org/vol27/iss3/art46/

WMO Code Table 4677. (dátum nélk.). Letöltés dátuma: 2024. december 4, forrás: https://www.nodc.noaa.gov: https://www.nodc.noaa.gov/archive/arc0021/0002199/1.1/data/0-data/HTML/WMO-CODE/WMO4677.HTM

www.adobe.com. (dátum nélk.). Letöltés dátuma: 2024. december 6, forrás: Convert PDF to Excel: https://www.adobe.com/uk/acrobat/online/pdf-to-excel.html

www.bps.go.id-1. (2024). Letöltés dátuma: 2024. november 5, forrás: [2010 Version] Quarterly GRDP At Current Market Price by Industrial Origin in Province All Over Indonesia (Billion Rupiah), 2010-2024 - Statistical Data: https://www.bps.go.id/en/statistics-table/1/MjIwNSMx/-2010-version--quarterly-grdp-at-current-market-price-by-industrial-origin-in-province-all-over-indonesia--billion-rupiah---2010-2024.html

www.bps.go.id-1. (2024). Forrás: [2010 Version] Quarterly GRDP At Current Market Price by Industrial Origin in Province All Over Indonesia (Billion Rupiah), 2010-2024 - Statistical Data: https://www.bps.go.id/en/statistics-table/1/MjIwNSMx/-2010-version--quarterly-grdp-at-current-market-price-by-industrial-origin-in-province-all-over-indonesia--billion-rupiah---2010-2024.html

www.bps.go.id-2. (2023). Letöltés dátuma: 2024. november 5, forrás: Number Of Crime According To Police Territorial Jurisdiction - Statistical Data: https://www.bps.go.id/en/statistics-table/2/MTAxIzI=/number-of-crime-according-to-police-territorial-jurisdiction.html

www.bps.go.id-2. (2023). Forrás: Number Of Crime According To Police Territorial Jurisdiction - Statistical Data: https://www.bps.go.id/en/statistics-table/2/MTAxIzI=/number-of-crime-according-to-police-territorial-jurisdiction.html

www.bps.go.id-3. (dátum nélk.). Letöltés dátuma: 2024. december 5, forrás: Unemployment Rate by Province (Percent), 2024: https://www.bps.go.id/en/statistics-table/2/NTQzIzI=/unemployment-rate--february-2024.html

www.bps.go.id-3. (dátum nélk.). Forrás: Unemployment Rate by Province (Percent), 2024: https://www.bps.go.id/en/statistics-table/2/NTQzIzI=/unemployment-rate--february-2024.html

www.imf.org. (2024). Letöltés dátuma: 2024. november 6, forrás: World Economic Outlook (October 2024) - GDP per capita, current prices: https://www.imf.org/external/datamapper/PPPPC@WEO/OEMDC/ADVEC/WEOWORLD

www.imf.org. (2024. november 6). Forrás: World Economic Outlook (October 2024) - GDP per capita, current prices: https://www.imf.org/external/datamapper/PPPPC@WEO/OEMDC/ADVEC/WEOWORLD

www.macrotrends.net. (dátum nélk.). Letöltés dátuma: 2024. november 9, forrás: Singapore Crime Rate & Statistics 1990-2024: https://www.macrotrends.net/global-metrics/countries/sgp/singapore/crime-rate-statistics

www.recaap.org. (dátum nélk.). Letöltés dátuma: 2024. december 5, forrás: Reports: https://www.recaap.org/reports

www.recaap.org. (dátum nélk.). Letöltés dátuma: 2024. szeptember 22, forrás: ReCAAP ISC: https://www.recaap.org/resources/ck/files/Number%20of%20Incidents/2022/List%20of%20Incidents%20for%202022.pdf

www.recaap.org. (dátum nélk.). Letöltés dátuma: 2024. szeptember 24, forrás: https://www.recaap.org/resources/ck/files/Number%20of%20Incidents/2024/List%20of%20Incidents%20for%202024.pdf

ZalaRushirajsinh. (2023. november 4). The Elbow Method: Finding the Optimal Number of Clusters. *Medium*. Forrás: https://medium.com/@zalarushirajsinh07/the-elbow-method-finding-the-optimal-number-of-clusters-d297f5aeb189

11 Melléklet: országokra vonatkozó adatok ismertetése

11.1.1 Indonéziára vonatkozó adatok

A továbbiakban az Indonéziára vonatkozó adatokat fogom ismertetni. Ezek négy fő forrásból származnak: a BPS-től (Indonéz Statisztikai Hivatal), a St. Louis-i Központi Bank által fenntartott FRED (Federal Reserve Economic Data) adatbázisból, a Világbank DataBank oldaláról, illetve az OECD Data Explorer oldaláról.

Az országra vonatkozó adatállományba bekerült adatok a következők:

- 2010 első, és 2024 harmadik negyedéve közt mért negyedéves és éves GRDP (Gross Regional Domestic Product) provinciára, azon belül pedig iparra lebontva, milliárd indonéz rúpiában megadva (www.bps.go.id-1, 2024): ezen belül rászűrtem a provinciánként összesített GRDP-re, illetve az "Agriculture, Forestry and Fishing"-hez („Mezőgazdaság, erdőgazdálkodás, halászat”) tartozó provinciánként mért GRDP-re, majd eldobtam az éves GRDP-adatokat.
- Évente elkövetett bűnesetek darabszáma rendőri területi illetékességekre lebontva 2008 és 2022 közt (www.bps.go.id-2, 2023): mivel a hivatal oldaláról két-és hároméves periódusokra (pl. 2021-22 vagy 2003-5) vonatkozó adatokat lehet egyszerre letölteni, ezeket először dataframe-ekbe szedtem, majd azokat konkatenáltam.
- Félévente mért munkanélküliség provinciára lebontva, százalékos formában, 2006 és 2024 februárja közt (www.bps.go.id-3, dátum nélk.): az oldalról egyéves periódusokra (pl. 2024 februárja és augusztusa) lebontva lehetett egyszerre letölteni az adatokat, így ezeket először dataframe-ekbe szedtem, majd azokat konkatenáltam.
- Havonta mért fogyasztói árindex: az előző periódus növekedési értéke. A FRED-ről letöltött idősor 1968 februárja és 2024 márciusa közti adatokat tartalmaz (fred.stlouisfed.org-1, 2024).
- Negyedévente mért nominális GDP, szezonalitással, millió indonézi rúpiában megadva, 1990 januárja és 2024 márciusa közt (fred.stlouisfed.org-2, 2024).
- Negyedévente mért nominális GDP, szezonalitás nélkül, millió indonézi rúpiában megadva, 2008 januárja és 2024 márciusa közt (fred.stlouisfed.org-3, 2024).
- Negyedévente mért reál GDP, szezonalitással, millió indonézi rúpiában megadva, 2000 januárja és 2024 márciusa közt (fred.stlouisfed.org-4, 2024).
- Negyedévente mért reál GDP, szezonalitás nélkül, millió indonézi rúpiában megadva, 2008 januárja és 2024 márciusa közt (fred.stlouisfed.org-5, 2024).

- Bizonytalansági index (fred.stlouisfed.org-6, 2024): A World Uncertainty Index Indonéziára vonatkozó negyedéves adatai 1952 januárja és 2024 júliusa közt: A „Bizonytalanság” szó előfordulásait számolja össze a negyedéves Economist Intelligence Unit országos jelentéseiben.
- A Világbank DataBankjából (databank.worldbank.org/-1, 2024) kinyert (éves) adatok a következők:
    - Population, total: Teljes népesség.
    - GNI, Atlas method (current US$): Atlas-módszerrel (a módszer lényege, hogy hároméves mozgóátlag segítségével elsimítja a valuatárfolyam-fluktuációkat) mért, aktuális US$ árfolyamon megadott GNI.
    - GDP (current US$): GDP aktuális US$ árfolyamon.
    - Inflation, GDP deflator (annual %): Infláció, GDP-deflátor (éves %)
    - Agriculture, forestry, and fishing, value added (% of GDP): Mezőgazdaság, erdészet és halászat hozzáadott értéke (GDP %-a).
    - Adjusted net national income (current US$): Módosított nettó nemzeti jövedelem aktuális US$ árfolyamon.
    - Adjusted net national income per capita (current US$): Egy főre jutó módosított nettó nemzeti jövedelem aktuális US$ árfolyamon.
    - Adjusted net national income per capita (constant 2015 US$): Egy főre jutó módosított nettó nemzeti jövedelem 2015-ös US$ árfolyamon.
    - Adjusted net national income per capita (annual % growth): Egy főre jutó módosított nettó nemzeti jövedelem (éves növekedés százalékban megadva).
    - Poverty headcount ratio at $2.15 a day (2017 PPP) (% of population): a 2017-es vásárlóerő-paritás alapján $2.15-ös napi jövedelemnél meghúzott szegénységi vonal alattiak aránya (össznépesség százalékában megadva).
- Halászatban, akvakultúrában, halfeldolgozásban dolgozók száma (data-explorer.oecd.org, 2024).

A fentebbi adatokat interpolációnak vetettem alá (erről bővebben a 3.2.2. fejezetben írok), majd egy havi alapú adatállománnyá fűztem őket össze.

11.1.2 Malajziára vonatkozó adatok

A továbbiakban a Malajziára vonatkozó adatokat fogom ismertetni. Ezek a következő forrásokból származnak: a data.gov.my oldalról, azaz Malajzia hivatalos nyílt adatportáljáról, az OpenDOSM-ről, azaz a maláj statisztikai hivatal adatportáljáról, a St. Louis-i Központi Bank

által fenntartott FRED (Federal Reserve Economic Data) adatbázisból, a Világbank DataBank oldaláról, az OECD Data Explorer oldaláról, illetve az IMF (International Monetary Fund) DataMapper platformjáról.

Az országra vonatkozó adatállományba bekerült adatok a következők:

- Éves GDP aktuális USD-árfolyamon, szezonalitás beleszámítása nélkül 1960 és 2023 közt (fred.stlouisfed.org-7, 2024).
- Bizonytalansági index: A World Uncertainty Index Malajziára vonatkozó negyedéves adatai 1952 januárja és 2024 júliusa közt (fred.stlouisfed.org-8, 2024): A „Bizonytalanság" szó előfordulásait számolja össze a negyedéves Economist Intelligence Unit országos jelentéseiben.
- Éves infláció, százalékosan, 1960 és 2023 közt (fred.stlouisfed.org-9, 2024): a fogyasztói árindex százalékos változása alapján, szezonalitással számított éves infláció.
- Egy főre jutó GDP vásárlóerő-paritás alapján (www.imf.org, 2024).
- Államonkénti szegénységi ráta 1972 és 2022 közt, két-háromévente (data.gov.my-1, 2023): három változója az abszolút szegénység (a szegénységi vonal alatti havi jövedelmű háztartások százalékban megadott aránya), a durva szegénység (az ételszegénységi vonal alatti havi jövedelmű háztartások százalékban megadott aránya) és a relatív szegénység (azon háztartások százalékban megadott aránya, melyek havi jövedelme nem éri el az adott állam medián jövedelmének felét).
- Havi lebontású munkaerő-piaci statisztikák 2010 januárja és 2024 szeptembere közt (data.gov.my-2, 2024): minden hónapra megadja a munkaerő mértékét (ezer személyben), a foglalkoztatottak számát (ezer személyben), a munkanélküliek számát (ezer személyben), a munkaerőpiacon kívül esők számát (ezer személyben), a munkanélküliségi rátát (százalékban), a munkaerő-piaci részvételi arányt (százalékban), illetve a foglalkoztatottság arányát a népességhez (százalékban).
- Maláj gazdasági indikátorok (data.gov.my-3, 2024): havi alapú gazdasági indikátorok 1990 januárja és 2024 szeptembere közt. Változói a Leading index (a következő hónapok gazdasági teljesítményére vonatkozó várakozások mérőszáma), a Coincident Index (az adott hónapra vonatkozó gazdasági teljesítmény mérőszáma), a Lagging Index (A Leading Index és a Coincident Index validálására szolgáló index), valamint a Leading Index (Diffusion)és a Coincident Index (Diffusion). Utóbbi kettő a Leading Index és a Coincident Index komplementereiként szolgálnak, 100 és 0 közti értéket

vehetnek fel. 100-as értékük azt mutatja, hogy minden összetevő növekszik, a 0 azt, hogy mindegyik csökken.

- Termelői árindex (data.gov.my-4, 2024): 2010 januárja és 2024 szeptembere közti termelői árindex öt szektorra (mezőgazdaság, bányászat, elektromosság, vízellátás, gyártás) vonatkozóan és összesítve. A változást a 2010-es értékekhez viszonyítja.
- Fogyasztói árindex város/vidék felosztás és 13 termék-és szolgáltatáscsoport szerint 2010 januárja és 2024 szeptembere közt: a (OpenDOSM, dátum nélk.) változást a 2010-es értékekhez viszonyítja. Ezen adatokból rászűrtem a vidékiekre, mivel ott nagyobb a halászat súlya. (Wintergalen, Oyanedel, Villaseñor-Derbez, Fulton, & Molina, 2022)
- A Világbank DataBankjából kinyert (éves) adatok a következők (databank.worldbank.org-2, 2024):
    - Population, total: Teljes népesség.
    - GNI, Atlas method (current US$): Atlas-módszerrel (a módszer lényege, hogy hároméves mozgóátlag segítségével elsimítja a valuatárfolyam-fluktuációkat) mért, aktuális US$ árfolyamon megadott GNI.
    - GDP (current US$): GDP aktuális US$ árfolyamon.
    - Inflation, GDP deflator (annual %): Infláció, GDP-deflátor (%)
    - Agriculture, forestry, and fishing, value added (% of GDP): Mezőgazdaság, erdészet és halászat hozzáadott értéke (GDP %-a).
- Halászatban, akvakultúrában, halfeldolgozásban dolgozók száma (data-explorer.oecd.org, 2024).

Az indonéz adatokhoz hasonlóan a fentebbi adatokat interpolációnak vetettem alá, majd egy havi alapú adatállománnyá fűztem őket össze.

11.1.3 Szingapúrra vonatkozó adatok

A továbbiakban a Szingapúrra vonatkozó adatokat fogom ismertetni. Ezek a következő forrásokból származnak: a SingStatról, azaz a Department of Statistics Singapore oldaláról, a St. Louis-i Központi Bank által fenntartott FRED (Federal Reserve Economic Data) adatbázisból és a MacroTrends oldalról.

Az országra vonatkozó adatállományba bekerült adatok a következők:

- Szingapúri bűnözési ráta 1990 és 2021 közt (www.macrotrends.net, dátum nélk.): bár az adatforrás nevében 2024 áll, mint a legutóbbi év, amire vonatkozó adatot tartalmaz a forrás, valójában 2021 az.

- Bizonytalansági index (fred.stlouisfed.org-10, 2024): A World Uncertainty Index Szingapúrra vonatkozó negyedéves adatai 1952 januárja és 2024 júliusa közt: A „Bizonytalanság” szó előfordulásait számolja össze a negyedéves Economist Intelligence Unit országos jelentéseiben. Az idősor adataiból rászűrtem a 2008-től kezdődőekre.
- Tengeri rakományokra és szállítmányozásra vonatkozó statisztikák (https://www.singstat.gov.sg/-1, 2024): 1995 januárja és 2024 szeptembere közti adatokat tartalmazó idősor. A következő változókat tartalmazza:
    - Érkező hajók száma
    - Érkező hajók bruttó regisztertonnában (a hajók teljes belső térfogata)
    - Rakomány (ezer tonnában)
    - Általános rakomány (ezer tonnában)
    - Teljes konténerforgalom (ezer TEU-ban megadva)
    - Bunkereladások (ezer tonnában)
    - Periódus végén regisztrált hajók száma (darab)
    - Periódus végén regisztrált hajók (bruttó regisztertonnában)
- Negyedéves GDP aktuális árakon, összesítve és szektorokra bontva (https://www.singstat.gov.sg/-2, 2024): 1975 első, és 2024 második negyedéve közti idősor.
- 14 év felettiek munkanélküliségi rátája (https://www.singstat.gov.sg/-3, 2024): 1992 és 2024 közti éves, a szezonalitás figyelembevételével készített statisztika, az adott év júniusi állapotát mutatja.
- Fogyasztói árindex (singstat.gov.sg-4, 2024): 2019-et bázisévként használó, havi alapú kimutatás 1961 januárja és 2024 szeptembere közt. Összesített és termékekre (pl. kenyér) vonatkozó adatokat is tartalmaz.

Az eddigi országok adataihoz hasonlóan a fentebbi adatokat interpolációnak vetettem alá, majd egy havi alapú adatállománnyá fűztem őket össze.

11.1.4 Thaiföldre vonatkozó adatok

A Thaiföldre vonatkozó adatokat a Világbank DataBankjából (databank.worldbank.org-3) töltöttem le.

- Population, total: teljes népesség.
- Population growth (annual%): Népességnövekedés (éves, százalékban kifejezve)

- Population density (people per sq. km of land area): Népsűrűség (egy négyzetkilométerre jutó emberek száma)
- Poverty headcount ratio at national poverty lines (% of population): Nemzeti szegénységi szint alattiak rátája (népességhez mérve százalékban)
- Poverty headcount ratio at $2.15 a day (2017 PPP) (% of population): a 2017-es vásárlóerő-paritás alapján $2.15-ös napi jövedelemnél meghúzott szegénységi vonal alattiak aránya (össznépesség százalékában megadva).
- GNI, Atlas method (current US$): Atlas-módszerrel mért, aktuális US$ árfolyamon megadott GNI.
- GNI per capita, Atlas method (current US$): Atlas-módszerrel mért, aktuális US$ árfolyamon megadott egy főre jutó GNI.
- GNI, PPP (current international $): vásárlóerő-paritás alapján számított GNI.
- GNI per capita, PPP (current international $): vásárlóerő-paritás alapján számított egy főre jutó GNI.
- Income share held by lowest 20%: az összjövedelem alsó 20%-a által birtokolt aránya.
- Life expectancy at birth, total (years): várható élettartam (év).
- Mortality rate, under-5 (per 1,000 live births): 5 év alattiak halálozási rátája (1000 születésenként)
- School enrollment, primary (% gross): általános iskolába beíratottak aránya (százalékban megadva)
- School enrollment, secondary (% gross): középiskolába beíratottak aránya (százalékban megadva).
- School enrollment, primary and secondary (gross), gender parity index (GPI)
- Annual freshwater withdrawals, total (% of internal resources): Éves frissvíz-felhasználás (belső erőforrásokhoz mérten, százalékban).
- Urban population growth (annual %): Városi népességnövekedés (éves, %-ban megadva).
- GDP (current US$): GDP aktuális US$ árfolyamon.
- GDP growth (annual %): GDP növekedés (éves, százalékban kifejezve).
- Exports of goods and services (% of GDP): termékek és szolgáltatások exportja (GDP-hez mérten százalékban kifejezve).
- Imports of goods and services (% of GDP): termékek és szolgáltatások importja (GDP-hez mérten százalékban kifejezve).

- Revenue, excluding grants (% of GDP): jövedelem a segélyek figyelembe nem vételével (GDP-hez mérten, százalékban kifejezve).
- Domestic credit provided by financial sector (% of GDP): a pénzügyi szektor által biztosított hazai hitel (GDP-hez mérten, százalékban kifejezve).
- Merchandise trade (% of GDP): Árukereskedelem (GDP-hez mérten, százalékban kifejezve).
- Net barter terms of trade index (2015 = 100): Külkereskedelmi cserearány-index (2015=100)
- External debt stocks, total (DOD, current US$): Teljes, külföldiek felé szóló adótartozás (DOD, aktuális US$ árfolyamon)
- Total debt service (% of exports of goods, services and primary income)
- Political Stability and Absence of Violence/Terrorism: Estimate (Becslés a politikai stabilitásra és az erőszak/terrorizmus hiányára)

Az eddigi országok adataihoz hasonlóan a fentebbi adatokat interpolációnak vetettem alá, majd egy havi alapú adatállománnyá fűztem őket össze.