



BME Matematika Intézet

**Dimenziócsökkentési Eljárások
Eredményeinek Értelmezése**

DIPLOMAMUNKA

Ragács Attila

Témavezető: Dr. Kovács Edith Alice

2022

Tartalomjegyzék

1. Bevezetés	3
1.1. Előzmények	3
1.2. A Shapley-érték	5
2. Regressziós modellek értelmezése, dimenziócsökkentés	7
2.1. A Kernel Shapley eljárás elméleti háttére	7
2.2. Transzformációk	12
3. Klaszterezés értelmezése Shapley megvilágításban	16
3.1. Silhoutte	17
3.2. Klaszter Shapley eljárás elméleti háttere	21
3.2.1. Az osztályozás értelmezése Shapley-érték segítségével .	21
3.2.2. A klaszterezés értelmezése Shapley-érték segítségével .	25
4. Szocio- és demográfiai területen való alkalmazás	29
4.1. Adatbemutatás	29
4.2. Regresszió	31
4.3. Klaszterezés	36
4.3.1. Silhoutte mutató alapú elemzés	39
4.3.2. A klaszterekbe sorolás hátterének felderítése	45
4.3.3. Anomáliadektálás	50
5. Összefoglalás	51
6. Köszönetnyilvánítás	53

A dolgozatban a dimenziócsökkentést fogjuk megközelíteni több szempontból, felügyelet melletti és felügyelet nélküli gépitanulásban, és bemutatjuk ezekben való szerepét, a játékelméletből származó Shapley-értéknek. Az 1. fejezetben egy rövid irodalom áttekintés után ismetetjük a Shapley-értékhez köthető legfontosabb definíciókat és tulajdonságokat. Az 1.1. alfejezetben képet adunk a korábbi munkákról, amelyek felhasználták ezt a fogalmat. Ezek javarészt felügyelt regressziós illetve osztályozási feladatokhoz kapcsolódnak, illetve azok eredményeinek értelmezése. Az 1.2. alfejezetben pedig olvashatók a pontos játékelméleti definíciók, tételek, melyek ismerete szükséges az alkalmazott modellek jó megértéséhez.

A 2. fejezetben ismertetjük először az alkalmazott algoritmusok lelkét adó Kernel Shapley eljárást először a 2.1. alfejezetben. Mind a regressziós, mind a klaszterezési alkalmazásainkban kulcsfontosságú ez az eljárás, amely lényegében egy hatékony közelítése az eredeti költségesen kiszámolható Shapley-értékeknek. Ugyanebben a fejezet 2.2. alfejezetben vizgáltuk, hogy a kapott Shapley-értékek hogyan alakulnak, amennyiben az eredeti adathalmazt leíró változókat transzformáljuk.

Majd a 3. fejezetben az egyik leggyakrabban használt felügyelet nélküli gépi tanulási feladatra, a klaszterezésre mutatunk be egy új eljárást, a Klaszter Shapley módszert. Ennek erőssége, hogy a kapott klaszterekhez képesek vagyunk jelentést kapcsolni. A módszert automatizálásához felhasznált Silhoutte algoritmust is bemutatjuk (3.1. alfejezet), ez a paraméterek optimális beállítását teszi lehetővé, azáltal hogy mérőszámot ad a klaszterezés jóságára. Így ki tudjuk választani az adatunkon legjobban teljesítő algoritmust, illetve ebben a klaszterek számát. A 3.2. alfejezetben írjuk le azt a módszert, amellyel az egyes klaszterekben megkapható az adatpontok Shapley-értéke és ezáltal a változók segítségével karakterizálhatók ezek a klaszterek.

A 4. fejezetben szerepelnek a 2. és 3. fejezetben bemutatott eljárások alkalmazásai, a felhasznált adathalmaz bemutatását követően. Végül az 5. fejezetben a dolgozat összefoglalója található, ahol kitérünk újra a saját

hozzájárulásainkra és számításba vesszük a jövőbeli lehetséges folytatásokat.

A dolgozat új eredményei a a 2. fejezetben a transzformációk vizsgálatához kapcsolódnak, a 3. fejezetben a 2-nél magasabb dimenzióba való vetítés esetén az automatizált klaszterezés. A 4. fejezetben bemutatjuk a módszerek alkalmazását egy szocio- és demográfiai adaton, ahol további új alkalmazhatóságokat is tárgyalunk, mint például az anomáliák megmagyarázását, illetve a csoport-Shapley által kínált lehetőségeket.

1. Bevezetés

A diplomamunka fő motivációja bonyolult a sok változóval, sok adattal működő úgy nevezett "black-box" modellek értelmezése, intelligens adatcsökkenése.

Erre a célra a matematika különböző területeiről származó módszereket, fogalmakat alkalmaztunk. A dolgozatban központi szerepet játszik a Shapley-érték, amit a játékelmélettől kölcsönöztött az utóbbi években a gépi tanulással foglalkozó társadalom. A Shapley-érték segítségével sikerült végrehagyani, hogyan hatnak a magyarázó változók a célváltozóra, akkor is, ha ez nem egy lineáris regresszió vagy egy logisztikus regresszió, ahova a képletüknek hálá betekintést nyerünk a modellbe, hanem egy "black-box" modell, mint például a véletlen erdő vagy a neurális hálók. Ezidáig tipikusan felügyelet melletti tanulásra használták szélesebb körben. [RPB20] [SN20] Felügyelet nélküli tanulásban csak a közelmúltban kezdték el használni. [MJE21]

1.1. Előzmények

Sundararajan és Najmi cikkében [SN20] egy tömör bevezetés után a Shapley-értékről, az axiómákra építve annak kibővítéseit tárgyalják. Gépi tanulási modellek bemeneti változóihoz rendelnek fontossági értékeket különböző módon. A cikk elején kapunk egy jó áttekintést a 2020 előtt megjelent mun-

kákról.

Leginkább a regressziós és klasszifikációs eljárásokban használták eddig a Shapley-értéket. Ehhez kapcsolódóan megemlítjük Rodríguez-Pérez és Barjorath 2020-as cikkét [RPB20], amiben klasszikus osztályozási és regressziós modellek predikcióira adnak magyarázatot. Az újításuk egy olyan módszer, ami neurális hálók esetén is használható, amelyek működéséről a legtöbb esetben igen keveset vagy semmit sem tudunk a felépítésükből adódóan.

A dolgozat, ami egy egészen új hozzáállást vezetett be az Marcílio-Jr és Eler 2021-es [MJE21] cikke, amely továbblép a felügyelet nélküli tanulás felé. A dolgozatban azt a kérdést járjuk körül, hogy hogyan lehet megmagyarázni egy klaszterezési eljárást, azt, hogy bizonyos adatpontok miért kerülnek bele bizonyos klaszterekbe. Marcílio-Jr és Eler a klaszter-orientált munkájukban a dimenziócsökkentési eredmények értelmezéséhez dolgoztak ki egy új módszert, amit Klaszter Shapley eljárásnak neveztek el és alkalmazzák sikeresen. Szemléletes vizualizációkkal adnak karakterizációt a kapott klasztereknek. Ez az új eljárás inspirálta a dolgozatunk egyik fejezetét. Az eljárást, illetve az eljárás egyes elemeinek továbbgondolását, fejlesztését tartalmazza a dolgozat 3. fejezete.

Az előző említett cikkben szereplő algoritmus működéséhez alapot biztosít Lundberg és Lee fontos és sokat hivatkozott munkája [LL17], amelyben egy additív függvényosztály megadásával bemutatnak több módszert a Shapley-érték hatékony becslésére. Ennek köszönhetően a Klaszter Shapley algoritmus is hatékonyan fut.

Mase, Owen és Seiler 2020-ban bemutatott módszerében az adathalmaz szétválogatott részein alkalmazták a Shapley-érték által elősegített regressziót. [MOS20] Egy t tesztalanyhoz változók tetszőleges részhalmazain hasonló adatpontokon számították ki az értékeket, majd ezeket átlagolták. Az eljárás előnyei közé tartozik, hogy csak a valóságban is előforduló magyarázóváltozó-kombinációkat használ, illetve változók egyforma realizációihoz ténylegesen egyforma fontossági értéket rendel, szemben számos korábbi megoldással.

Egy korábbi cikkben Štrumbelj és Kononenko [ŠK14] érzékenység elemzéssel adott magyarázatot regressziós és gépi tanulási modellek kimeneteire. A korábban elterjedt módszerek kevésbe tudták figyelembe venni a változók közötti összefüggéseket, erre is megoldást nyújtottak. Továbbá szokatlan módon emberekkel tesztelték az eljárás hatékonyságát és azt találták, hogy valóban segíti a jobb megértést.

A kontrasztív főkomponens analízis egy továbbfejlesztett változatával adott klasztermagyarázatokat Fujiwara, Kwon, és Ma. [FKM19] A szerzők interaktív rendszert építettek, amelyben megadják, hogy az egyes változók mennyiben járulnak hozzá a klaszterek közötti eltérésekhez.

1.2. A Shapley-érték

A Shapley-érték eredetileg a kooperatív játékelméletből származó fogalom, ami számszerűsíti, hogy, ha adott egy N főből álló koalíció, amely elvégez egy közös feladatot és ebből származik egy összesített érték, akkor abból az egyes játékosok milyen arányban részesülnek. [Sha53]

A gépi tanulási regressziós modellek körében ez úgy fogalmazható meg, hogy az értéket kiosztó függvény maga a regressziós modell kimenete, míg a bemeneti változók az egyes játékosok.

Tekintsük először át, mit tudunk a Shapley-értékről, hogyan definiáljuk, mik a tulajdonságai.

Definíció. Egy koalíciós játék egy $\langle N, v \rangle$ pár, ahol $N = \{1, 2, \dots, n\}$ egy véges halmaz amely n játékosból áll és egy $v : 2^N \rightarrow \mathbb{R}$ értékelő-függvény azzal a tulajdonsággal, hogy $v(\emptyset) = 0$.

A v függvényt az N összes részhalmazán definiáljuk. A cél olyan megoldást találni, amely segítségével a teljes kifizetést $v(N)$ -t elosztjuk a igazságosan az n játékos között.

A φ Shapley-érték egy ilyen eloszlást fog definiálni.

Definíció. Egy $\langle N, v \rangle$ játékban az i játékos Shapley-értéke a következő módon értelmezzük:

$$\varphi_i(v) = \sum_{S \subseteq N \setminus i} \frac{|S|!(|N| - |S| - 1)!}{N!} (v(S \cup i) - v(S)) \quad (1)$$

A $v : 2^N \rightarrow \mathbb{R}$ értékelő-függvény a koalíció minden lehetséges részhalmazához hozzárendeli annak értékét. $v(N)$ pedig a teljes koalíció értéke.

Értelmezzük a fenti képletet, végigvesszük a játékosok összes lehetséges S halmazát, ami nem tartalmazza i -t, majd nézzük ennek a játékosnak a határhozzájárulását (jelölje ezt $v'_i(S) = v(S \cup i) - v(S)$) az S halmazhoz. Az együttható szerepe pedig a határhozzájárulások átlagolása, így kapjuk meg végül összegzés után az i játékos φ_i értékét.

A Shapley-érték egy egyértelmű igazságos elosztása a $v(N)$ kifizetésnek az n játékos között, amely a következő tulajdonságokkal rendelkezik:

- **hatékony**, ha az egyes játékosok egyéni értékeinek összege a teljes koalíció összege:

$$\sum_{i \in N} \varphi_i(v) = v(N).$$

- **nulljátékos tulajdonságú**, ha egy játékos nem járul hozzá egyetlen koalícióhoz sem, legyen 0 az értéke

$$\forall S \subseteq N \setminus \{i\}, v(S \cup \{i\}) = v(S) \Rightarrow \varphi_i(v) = 0$$

- **egyenlően kezelő (szimmetria)**, ha két játékos határhozzájárulásai teljesen megegyeznek tetszőleges játékoshalmazon, az értékük is meg kell, hogy egyezen

$$\forall i, j \in N, S \subset N \setminus \{i, j\}, v(S \cup \{i\}) = v(S \cup \{j\}) \Rightarrow \varphi_i(v) = \varphi_j(v)$$

- **additív**, ha $\forall \langle N, v \rangle, \langle N, w \rangle, \forall S \subseteq N, (v + w)(S) = v(S) + w(S)$:

$$\varphi(v + w) = \varphi(v) + \varphi(w)$$

Ezek a tulajdonságok jól ellenőrizhetőek, ami előnyössé teszi a fogalom felhasználását.

1.2.1. Tétel. A φ függvény pontosan akkor Shapley-érték, ha hatékony, null-játékos tulajdonságú, egyenlően kezelő és additív.

2. Regressziós modellek értelmezése, dimenziócsökkentés

Ebben a fejezetben a regressziós eljárásokhoz kapcsolódó értelmezésről és dimenziócsökkentésről lesz szó. Az első részben bemutatjuk a Kernel Shapley eljárást, illetve a legfontosabb magyarázóváltozók kiválasztását Shapley-értékre támaszkodó módszertanát. A fejezet második részében a magyarázóváltozók transzformációi által kapott eredményeket vizgáljuk, az eredményeket értelmezzük.

2.1. A Kernel Shapley eljárás elméleti háttére

Gyakori alkalmazása volt a Shapley-értéknek az elmúlt években a regressziós gépi tanulási modellek kimeneteinek értelmezése. Ilyenkor a kezünkben van egy olyan modell, amely tetszőleges számú változót kap bemenetként és ebből számítja ki a folytonos értékű kimenetet. Ebben a modellben célunk megmagyarázni, hogy az egyes bemeneti változók milyen mértékben járultak hozzá a kimenethez.

Tekintsünk egy regressziós feladatot és jelölje $\{\mathbf{x}^i y^i\}_{i=1,\dots,M}$ az M nagyságú tanuló mintát, legyen d a magyarázóváltozók száma. Jelölje (x_1, \dots, x_d, y)

az (X_1, \dots, X_d, Y) valószínűségi vektor egy realizációját, ahol Y -t egy folytonos valószínűségi változónak tekintjük, amit az \mathbf{x} vektor alapján meg szeretnénk becsülni egy $f(\mathbf{x})$ regressziós modellel.

Legyen \mathbf{x}^* egy adott realizáció, szeretnénk megmagyarázni az $f(\mathbf{x}^*)$ becslést az egyedi magyarázó változók értékeinek alapján.

Strumbelj és Kononenko ajánlotta először erre a célra a Shapley-értéket.[SK10]

Az $f(\mathbf{x}^*)$ játssza a kifizetés szerepét, a magyarázó változók pedig a játékosokét.

Ennek alapján $f(\mathbf{x}^*)$ a következőképpen írható fel:

$$f(\mathbf{x}^*) = \varphi_0 + \sum_{i=1}^d \varphi_i^* \quad (2)$$

ahol $\varphi_0 = E(f(\mathbf{x}))$, φ_i^* pedig az φ_i értéke az $\mathbf{x} = \mathbf{x}^*$ esetben.

Ha a fenti (2) képletet felírjuk, mint a becsült érték és a becsült értékek átlaga közötti különbséget

$$f(\mathbf{x}^*) - E(f(\mathbf{x})) = \sum_{i=1}^d \varphi_i^*$$

láthatjuk hogy a Shapley-értékek megmagyarázzák, hogy mely magyarázó változók okozzák adott \mathbf{x}^* esetén a becsült $f(\mathbf{x}^*)$ értékének az eltérését a becslések átlagától.

Ahhoz, hogy kiszámolhassuk a Shapley-értékeket, szükségünk van a $v(S)$ függvény definiálására tetszőleges S változóhalmaz esetén:

$$v(S) = E [f(\mathbf{x}) | \mathbf{x}_S = \mathbf{x}_S^*]$$

Gyakran a feltételes várható érték helyett szokták használni a feltételes mediánt vagy móduszt.

Megjegyzés: Amennyiben az $f(\mathbf{x}^*)$ egy lineáris modell:

$$f(\mathbf{x}) = \beta_0 + \sum_{i=1}^d \beta_i x_i$$

és az összes magyarázó változó független egymástól, akkor a Shapley-értékek a következők:

$$\begin{aligned}\varphi_0 &= \beta_0 + \sum_{i=1}^d \beta_i E(X_i) \\ \varphi_j &= \beta_j (x_j^* - E(X_j)), \quad j = 1, \dots, d.\end{aligned}$$

Két komoly probléma merül föl,

- a változók számával exponenciálisan növekszik az S halmazok száma,
- $v(S) = E[f(\mathbf{x})|\mathbf{x}_S = \mathbf{x}_S^*]$ értékekre nehéz becslést adni minden \mathbf{x}_S^* reálizációra.

Kernel Shapley módszer

Lundberg és Lee bemutatta a Kernel Shapley módszert a [LL17] cikkben, de sajnos a módszer leírása nem volt jól követhető, ezért később megjelent egy újabb cikk, amiben igyekeznek letisztázni matematikailag a módszer leírását. [CL20] Ezt is figyelembe véve a következőkben leírjuk az eljárás lényegét.

A cikk két problémára ad megoldást. Az egyik a Shapley-értékek leegyszerűsített kiszámolása, a másik pedig a $v(S)$ érték becslése.

Tekintsük a következő $\varphi_0, \dots, \varphi_d$ szerinti optimalizálási feladatot:

$$\sum_{S \in \mathcal{M}} \left(v(S) - \left(\varphi_0 + \sum_{j \in S} \varphi_j \right) \right)^2 k(d, S) \rightarrow \min, \quad (3)$$

ahol

$$k(d, S) = \frac{d-1}{\binom{d}{|S|} \cdot |S| \cdot (d-|S|)}$$

jelöli a Shapley kernel súlyokat.

Tekintsünk egy $\mathbf{Z} \in \mathbb{R}^{2^d \times (d+1)}$ mátrixot, amely elemei 0 és 1 értékeket vehetnek föl. Az első oszlopa a mátrixnak csak 1 értéket tartalmaz, mivel a φ_0 együtthatóinak felel meg. A mátrix minden sora meghatároz egy S halmazt, vagyis, amennyiben egy X_j változó benne van az S halmazban, úgy azon a j helyen 1 lesz, ha egy X_k nincsen benne az S halmazban, úgy a k helyen 0 lesz.

Legyen $\mathbf{W} \in \mathbb{R}^{2^d \times 2^d}$ diagonális mátrix mely tartalmazza a $k(d, S)$ elemeket. Jelölje továbbá $\varphi = [\varphi_1, \dots, \varphi_d]$ a Shapley-értékek vektorát, illetve $\mathbf{v} = [v(S_1), \dots, v(S_{2^d})]$.

A fenti (3) optimalizálási problémát mátrixosan a következőképpen is felírhatjuk:

$$(\mathbf{v} - \mathbf{Z}\varphi)^T \mathbf{W} (\mathbf{v} - \mathbf{Z}\varphi) \rightarrow \min.$$

A feladat optimális megoldása:

$$\varphi = (\mathbf{Z}^T \mathbf{W} \mathbf{Z})^{-1} \mathbf{Z}^T \mathbf{W} \mathbf{v}. \quad (4)$$

Az alkalmazásokban a $k(d, d) = k(d, 0) = \infty$ értéket egy c nagyon nagy számmal helyettesítjük.

Amennyiben a modell néhány változónál többet tartalmaz, úgy φ -t nehéz kiszámolni.

A Kernel Shapley trükk abban áll, hogy a súlyozott Shapley eltérések összegét fogjuk approximálni. Ezek a súlyok különböznek, lehetnek nagyon kicsik is, ilyenkor az adott S részhalmaza a magyarázó változóknak kis mértékben járul hozzá a Shapley-értékhez.

Az alapötlet az volt, hogy az \mathcal{M} halmazból visszatevéssel veszünk egy

mintát. A mintát \mathcal{D} -vel jelöljük. A minta a kernel súlyozást fogja követni. A továbbiakban csak a \mathcal{D} -hez tartozó sorokat vesszük figyelembe a \mathbf{Z} mátrixból, illetve csak az ezeknek megfelelő értékeket kell majd kiszámolni. Ennek alapján a φ -t meghatározó képlet a következőképpen néz majd ki:

$$\varphi = \left[\left(\mathbf{Z}_{\mathcal{D}}^T \mathbf{W}_{\mathcal{D}} \mathbf{Z}_{\mathcal{D}} \right)^{-1} \mathbf{Z}_{\mathcal{D}}^T \mathbf{W}_{\mathcal{D}} \right] \mathbf{v}_{\mathcal{D}}. \quad (5)$$

Az utolsó képlet (5) nagy előnye, hogy a szögletes zárójelben lévő részt, ami egy $(d+1) \times |\mathcal{D}|$ mátrix, csak egyszer kell kiszámolni. minden egyes becsléshez meghatározhatunk egy φ vektort (5). A φ kiszámolásához csak a $\mathbf{v}_{\mathcal{D}}$ vektor elemeit kell kiszámolni, a szögletes zárójelben lévő mátrix változatlan marad.

Lássuk most hogyan számoljuk ki, illetve közelítjük a $\mathbf{v}_{\mathcal{D}}$ -ben lévő értékeket, vagyis a $v(S) = E[f(\mathbf{x})|\mathbf{x}_S = \mathbf{x}_S^*]$ értékeket, minden S halmazra, ami megfelel egy egy sornak a \mathbf{Z} -ből.

Jelöljük \bar{S} -el az S komplementerét.

$$E[f(\mathbf{x})|\mathbf{x}_S = \mathbf{x}_S^*] = E[f(\mathbf{x}_{\bar{S}}, \mathbf{x}_S)|\mathbf{x}_S = \mathbf{x}_S^*] = \int f(\mathbf{x}_{\bar{S}}, \mathbf{x}_S) p(\mathbf{x}_{\bar{S}}|\mathbf{x}_S = \mathbf{x}_S^*) d\mathbf{x}_S$$

ahol $p(\mathbf{x}_{\bar{S}}|\mathbf{x}_S = \mathbf{x}_S^*)$ a feltételes valószínűséget jelöli.

Általában $p(\mathbf{x}_{\bar{S}}|\mathbf{x}_S = \mathbf{x}_S^*)$ feltételes valószínűségek nem ismeretesek. Lundberg és Lee [LL17] azzal a feltételezéssel számol, hogy a változók függetlenek egymástól, így $p(\mathbf{x}_{\bar{S}}|\mathbf{x}_S = \mathbf{x}_S^*)$ -et $p(\mathbf{x}_{\bar{S}})$ -el közelítik és így

$$v(S) = \int f(\mathbf{x}_{\bar{S}}, \mathbf{x}_S) p(\mathbf{x}_{\bar{S}}) d\mathbf{x}_S. \quad (6)$$

(6) integrálja Monte Carlo intergrálással approximálható:

$$v_{KerSHAP}(S) = \frac{1}{M'} \sum_{k=1}^{M'} f(\mathbf{x}_{\bar{S}}^k, \mathbf{x}_S^*)$$

ahol $\mathbf{x}_{\bar{S}}^k$, $k = 1, \dots, M'$ minták az eredeti tanulómintából, és mivel a függetlenség lett feltételezve, ezért ezek függetlenül lettek mintavételezve \mathbf{x}_S -

től.

Fontos megjegyezni, hogy a Shapley-értékek minden adatpont (mintaelem) esetén számolandók, és rátmutat arra, hogy az adott bemenet esetén mely változók, milyen mértékben járulnak hozzá a regresszió végeredményéhez.

Ha ezeket a Shapley-értékeket átlagoljuk minden változó esetén az összes mintaelemre, akkor minden változóhoz kapunk egy átlag Shapley-értéket, amely szerinti csökkenő rendezés egy fontossági rendet definiál (elől vannak a legfontosabbak). A sorrend alapot szolgáltat a változók számának csökkenéséhez. Így érhetünk el dimenziócsökkentést Shapley-érték alapján a regressziós problémák esetében.

Amióta Lundberg cikke megjelent több cikk foglalkozott a Shapley-értékek felhasználásával. Ezek bizonyos szempontból javítani igyekeznek azokon a pontokon, ahol túlságosan le lett egyszerűsítve a feltételezés.[LL17]

Számos gyakorlati haszna van egy ilyen módszernek. Ha páldául a lakásárakra szeretnénk becslést kapni, akkor egy jó képet kaphatunk arról, hogy mely változók milyen mértékben növelik vagy csökkentik az árat. Amennyiben kapunk egy 30 millió forintos kimenetet, a Shapley-értékek nélkül még nem tudjuk, hogy ehhez hogyan járult hozzá a szobák száma, az, hogy megengedett-e háziállatot hozni a lakásba, mennyire veszélyes az adott kerület vagy éppen a lakás mérete négyzetméterben. A következő részben teszteljük néhány általunk generált transzformációval a változókhöz rendelt Shapley-értékeket.

2.2. Transzformációk

Ebben a részben vizsgálni fogjuk, hogyan viselkednek a Shapley-értékek a változók bizonyos transzformációk esetén. A Shapley-értékek axomatikarendszerének megfelelően tudunk néhány tulajdonságot, de itt az is kérdés, hogy a beépített approximációs módszer hogyan reagál ezekre.

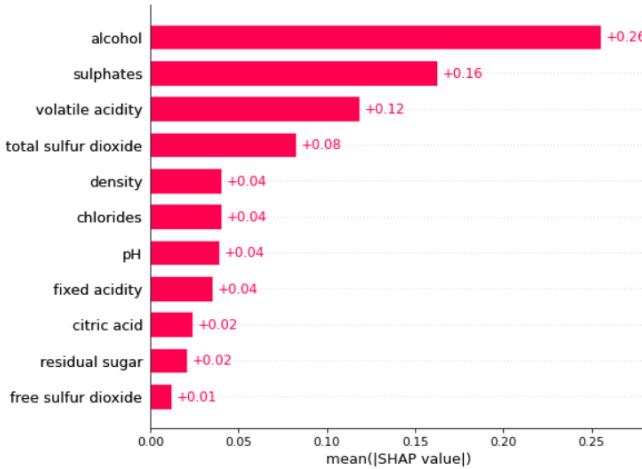
Ehhez a széleskörben alkalmazott vörösbor adatokat tartalmazó adathal-

mazt használtuk fel, ennek az első 5 sora látható az 1. ábrán. Az ebben szereplő oszlopok minden magyarázó változók az utolsó, 'quality' oszlopot leszámítva, ez a célváltozó, amire becslést szeretnénk adni. A célváltozó 6 különböző értéket vesz fel 3 és 8 között, így ezt a feladatot tekinthetnénk osztályozási feladatnak is, de úgy találtuk ennyi lehetséges érték elég, hogy regresszióval próbáljuk megoldani.

fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5
7.8	0.88	0.00	2.6	0.098	25.0	67.0	0.9968	3.20	0.68	9.8	5
7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.9970	3.26	0.65	9.8	5
11.2	0.28	0.56	1.9	0.075	17.0	60.0	0.9980	3.16	0.58	9.8	6
7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5

1. ábra. A vörösboros adathalmazból vett minta

A tanításnál egy Gradient Boosting technikát alkalmazó modellt használtunk, az XGBoost-ot.[CG16] Az adatpontok 80%-án tanított modell Shapley-értékeit átlagoltuk az összes ponton, majd ezeket ábrázoltuk a 2. ábrán. Egyértelműen látszik, hogy az 'alcohol' és 'sulphates' változók vannak a legnagyobb befolyással a kapott predikcióra.

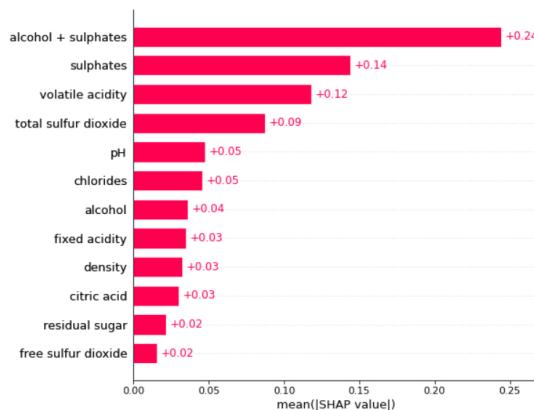


2. ábra. A vörösboros adathalmaz változóinak átlagos Shapley-értékei

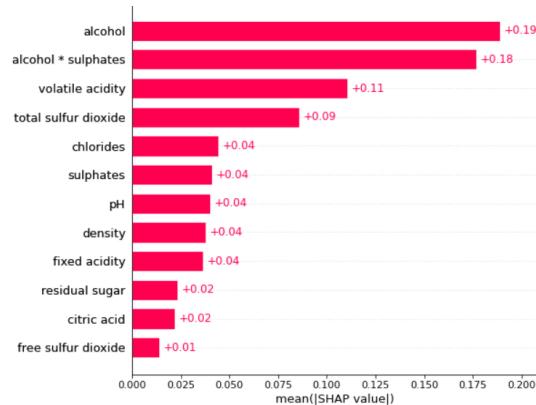
Most azonban arra szeretnénk választ kapni, hogy amennyiben egyes változókat módosítunk, vagy valamilyen transzformáltjukat új változóként a modellhez adjuk, az milyen hatást kelt a Shapley-értékeket illetően.

Először újból hozzáadtuk új néven az 'alcohol' változót, ez nem változtatta a sorrenden, 0 Shapley-értéket kapott az új változó. Ez egyrészről előnyös, hiszen egy redundanciát sikeresen kiszűr, azonban az egyenlően kezelő tulajdonságnak ellent mond, ami rávilágít a módszer egyik hiányosságára. Akkor sem volt változás, amikor a jelenlévő 'alcohol' változót szoroztuk 2-vel és nem vettünk be új változót, ugyanannyi maradt ennek is a Shapley-értéke. Továbbá gyakori előfeldolgozó lépés a változók skálázása, most a MinMax skálázást próbáltuk ki az 'alcohol' változón, azonban ez sem módosította az értékeket.

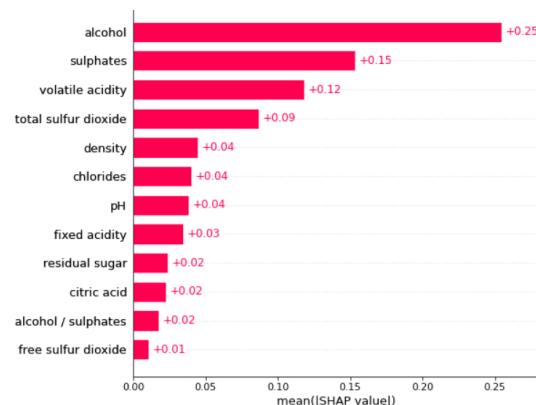
Azonban láttunk már változást, amikor új változóként megadtuk a két legfontosabb meglévő változó összegét, szorzatát, illetve ezek hányadosát. Mindhárom esetben másfajta változást tapasztaltunk, ezek Shapley-értékei a 3-5. ábrákon szerepelnek. A 3. ábrán az összegnél az 'alcohol' vesztett az értékéből és az új változó lett a legfontosabb, míg a 4. ábrán a szorzat esetén a 'sulphates' gyengült le és az új változó szintén fontos szerepet tölt be. Ellenben az 5. ábrán az új hányados változó minimális hatást kelt.



3. ábra. A vörösboros adathalmaz változóinak átlagos Shapley-értékei összeg hozzáadása után



4. ábra. A vörösboros adathalmaz változóinak átlagos Shapley-értékei szorzat hozzáadása után



5. ábra. A vörösboros adathalmaz változóinak átlagos Shapley-értékei hányados hozzáadása után

Ezek az eredmények azért fontosak, mert számos területen bevett gyakorlat a meglévő változók valamelyen kombinációját teljesen új változókként kezelni és fontos szerepet adni ezeknek a modellben. Ezen az egyszerű példán megfigyelhettük azonban azt, hogy különböző a transzformációk mennyire elterő magyarázó erővel lehetnek a modell eredményére.

Ebben az alfejezetben rávilágítottunk arra, hogy a Shapley-érték jól használható újabb, származtatott változók modellbeépítési hatékonyságának a

feltérképezésére.

3. Klaszterezés értelmezése Shapley megvilágításban

Az eddigiekben a Shapley-értéket kizárolag felügyelet melletti tanulásnál használtuk. Ebben a fejezetben a klaszterezéshez tartozó problémákkal foglalkozunk. Nevezetesen azzal, hogy a Shapley-érték segítségével magyarázatot adjunk arra, hogy egy adatpont, miért tartozik egy adott klaszterbe, vagyis mely változók hatására került oda. Egy másik fontos kérdés, ami az előzőhöz szorosan kapcsolódik, hogy amennyiben a klaszterek száma nem a szakértők által van meghatározva, hogyan lehet ezt a kérdést automatizálva megválaszolni.

A klaszterezési eljárás a felügyelet nélküli gépi tanulási eljárások közé tartozik. Az adatpontok klaszterezése során azt szeretnénk elérni, hogy a hasonló elemek egy halmazba kerüljenek. Erre a cérla számos módszert fejlesztettek ki az elmúlt évtizedekben, melyek közül a legelterjedtebb a k -közép algoritmus. [M⁺67] Ez egy partícionáló eljárás, ahol a név arra utal, hogy k darab klaszterbe soroljuk az adatpontokat egyesével, aszerint, hogy melyik klaszterátlaghoz vannak a legközelebb. További nagyobb kategóriákat alkotnak a hierarchikus vagy a sűrűség alapú klaszterezők. Amennyiben alacsony, 2 vagy 3 dimenzióba transzformáljuk az adatokat, a klaszterezés jól használható vizualizációra, és ennek alapján háttérinformációk felderítésére. Alkalmazási területei közé tartozik:

- Ajánlórendszerek
- Orvosi adatok elemzése
- Piackutatás, ügyfélelemzés
- Közösségi háló analízis

A dolgozatban a Shapley-érték segítségével fogunk rávilágítani arra, hogy a klaszterezési eljárás során kialakult klaszterekbe való sorolásra mely valószínűségi változók, attribútumok hatnak leginkább. A fejezetben alkalma-zott eljárás a [MJE21] cikkre támaszkodik. Ebben a cikkben a szerzők a klasztereket vizuális alapon határozzák meg, alacsony, tipikusan 2 dimenzióban. Ha az adatpontokat nem látjuk (3-nál magasabb dimenzóba vetítettük őket), akkor erre a feladatra szükséges automatizált eljárást felhasználni. Ez a lépés az általunk ajánlott általánosított eljárás egyik főeleme. A 3.1 alfejezetben mutatjuk be az automatizált klaszterezési eljárás egy lehetőségét, aminek eredménye a klaszterszám meghatározása, illetve a klaszterezési eljárás kiválasztásában is segíthet. A 3.2 alfejezetben a kapott klaszterek karakterizációját a Shapley-érték segítségével adjuk majd meg.

3.1. Silhoutte

Ebben a részben először egy klaszterezést elősegítő módszert fogunk bemutatni. A Silhoutte algoritmus [Rou87] a meglévő partíció minden eleméhez rendeli hozzá az úgynevezett Silhoutte mutatókat. Ezek az eredményül kapott klaszterek szeparációja alapján kerülnek kiszámításra és egy jó képet adnak arról, hogy egyes elemek a megfelelő klaszterbe lettek-e besorolva, klaszterek határán helyezkednek el vagy esetleg teljesen rossz csoportba kerültek. A módszer továbbá nagy segítséget nyújt a megfelelő algoritmus és a klaszterszám kiválasztásában.

Az algoritmusnak mindössze 2 dologra van szüksége a hatékony működéshez: a klaszterezés eredményeként kapott partícióra és a klaszterezett objektumok páronkénti távolságára. Itt fontos megjegyezni néhány kiegészítést, egyrészt a módszer működik bármilyen klaszterező algoritmussal, mivel csak az annotált adathalmazra van szükség. Másrészt a páronkénti távolság lehet az euklideszi-távolság, bár itt ügyelni kell a dimenzió nagyságára, hiszen ha ez túl nagy, könnyen értelmét veszítheti a távolság fogalma (dimenzionalitás átnézés). Továbbá a távolság alatt érhetünk páronkénti hasonlóságot

vagy különbséget is, ebben a dolgozatban azonban euklideszi-távolságokkal dolgozunk.

A következőkben a Silhouette mutatók kiszámítását írjuk le, amit az i objektum esetében jelöljön $s(i)$ (illusztráció a 6. ábrán). Továbbá jelölje A azt a klasztert amelyikbe az i került a particionálás során és legyen

$$a(i) = i \text{ átlagos távolsága az összes többi objektumtól az } A \text{ klaszterben.}$$

Ez itt most egyszerűen a teljes egészében az A halmazon belüli pontozott vonalak átlagos hossza. Továbbá legyen egy A -tól különböző C klaszterre

$$d(i, C) = i \text{ átlagos távolsága az összes } C\text{-beli objektumtól.}$$

Miután az összes ilyen C klaszterre kiszámoltuk ezeket az átlagtávolságokat, jelölje

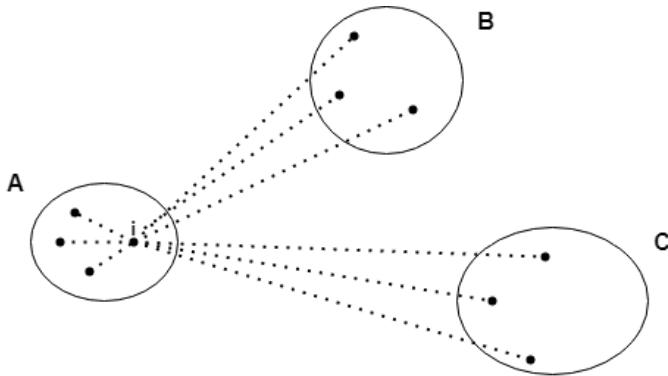
$$b(i) = \min_{A \neq C} d(i, C)$$

ezek közül a minimálisat, az ehhez a távolsághoz tartozó B klasztert A szomszédjának nevezzük. Úgy is gondolhatunk erre a klaszterre, hogy ha i -t nem tudnánk az A klaszterbe illeszteni, akkor ez lenne a következő legjobb választás. Persze ezen a ponton érdemes felenni azt is, hogy több, mint 1 klaszterrel dolgozunk.

Ha birtokában vagyunk a fenti mennyiségeknek, kiszámítható már az $s(i)$ értéke:

$$s(i) = \begin{cases} 1 - a(i)/b(i), & \text{ha } a(i) < b(i) \\ 0, & \text{ha } a(i) = b(i) \\ b(i)/a(i) - 1, & \text{ha } a(i) > b(i) \end{cases}$$

Szemügyre véve ezt a mennyiséget, láthatjuk, hogy egy -1 és 1 közötti értéket kapunk minden esetben. Egyszerűen, ha az objektum klaszterbe sorolása szinte tökéletes, azaz ha $a(i)$ alacsony és $b(i)$ magas, 1-hez közelíti az értéket



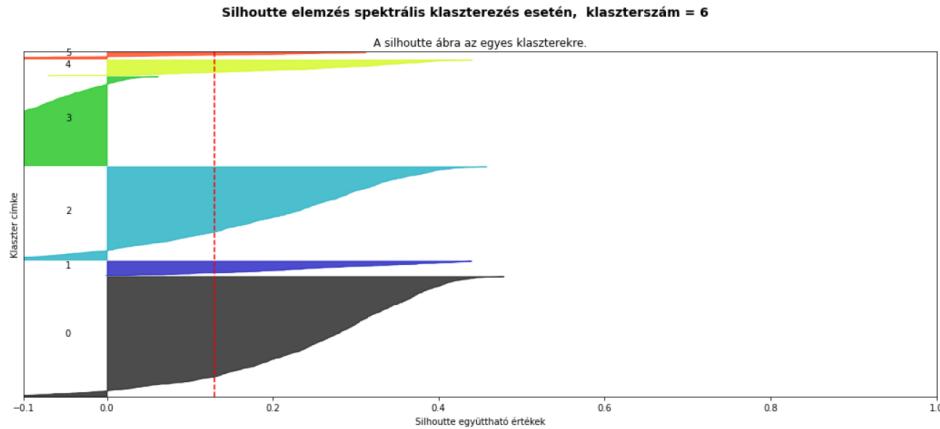
6. ábra. Elemek illusztrációja a Silhouette konstrukció során

kapunk. Azonban, ha épp fordítva, egy másik klaszter lényegesen közelebb van átlagosan, mint a partícióban kapott klaszter ($a(i) > b(i)$), akkor -1-hez lesz közel a Silhouette együttható. Továbbá, ha valamelyik két klaszter határán helyezkedik el, akkor 0 az együttható.

Néhány további megjegyzés a Silhouette együtthatóhoz, hogy abban az esetben, ha a klaszter csak egy j elemet tartalmaz, akkor $s(j) = 0$. Fel kell tenni azt is, hogy olyan távolság vagy hasonlósági mértéket használunk, ami méretarányos (ratio scale), azaz valóban elvárhatjuk, hogy a 40 kétszer annyi, mint a 20. Emiatt az együttható invariáns marad, ha az eredeti távolságokat egy pozitív konstanssal szorozzuk.

Egy konkrét példán keresztül nézzi is meg, vizuálisan hogyan lehet ábrázolni ezt az együtthatót. A 7. ábrán látható a Silhouette, amikor 10 dimenzióban lettek klaszterezve a magyar települések, 6 darab klaszterbe és az algoritmus a spektrális klaszterezés volt. (erről részletesebben a 3.2. alfejezetben). Ezt egyfajta eloszlásként is felfoghatjuk az egyes klasztereken belül, ugyanis itt egy szín egy klaszterre vonatkozik. A két dimenzióból a magasság egyszerűen az elemszámra utal, a szélesség pedig maga a Silhouette együttható. Ideális esetben körülbelül egyforma magasságú részeket szeretnénk látni, amik a pozitív irányban minél szélesebbek is.

Megfigyelhető ebben az esetben, hogy a klaszterek fele az átlagosnál



7. ábra. 6 klaszter esetén a Silhouette ábra

alacsonyabb elemszámú, ráadásul előfordulnak bennük negatív együtthatós adatpontok is. Továbbá, a 3-as számú klaszter szinte teljes egészében negatív együtthatós, amiből arra lehet következtetni, hogy egy természetes klasztert szétvágunk azzal, hogy 6-nak választottuk a klaszterek számát. Így minden bizonnal jobban járunk egy alacsonyabb klaszterszámmal a felhasznált adatok mellett és emellett megéri több, különböző algoritmust is kipróbálni. A függőleges piros vonal a globális átlagot mutatja, ami a különböző klaszterszám értékekre egy jó mutató, ha azt szeretnénk eldönteni, hány klaszterrel érdemes futtatni az algoritmust.

Összefoglalva a leírtakat, azt lehet elmondani a Silhouette együtthatóról, hogy egy nagy előnye az, hogy csupán a partíciótól függ, amit tetszőleges algoritmussal megadhatunk. Így aztán használhatjuk ugyanazon az adathalmazon több különböző algoritmus összehasonlítására, vagy egy adott algoritmus paramétereire. Szintén hasznos lehet az optimális klaszterszám megállapítására, hiszen ha tegyük fel túl alacsonyra állítjuk, akkor természetes klaszterekeket kell összevonjon az algoritmus, ami magas belső távolságokat eredményez (magas $a(i)$ -k), így alacsony $s(i)$ értékeket kapunk. Hasonló eredménnyel járhat az is, ha a klaszterszámot túl magasra állítjuk, ekkor ugyanis szétvágja az algoritmus a természetes klaszterekeket (amennyiben ezek

léteznek), ekkor viszont a köztes távolságok ($b(i)$) lesznek alacsonyak.

3.2. Klaszter Shapley eljárás elméleti háttere

Ezidáig a Shapley-értéket főleg regressziós feladatok megmagyarázására, illetve klasszifikációs eljárásoknál használták. A [MJE21] cikk mutat be először egy olyan módszert, ahol felügyelet nélküli tanulásban, pontosabban klaszterezés esetén, a Shapley-értékeket használják fel az egyes klaszterekbe tartozás megmagyarázására a bemeneti változók függvényében. Az eljárás eredménye nagy fontosággal bír az adatelemzők számára, mivel a magyarázatok mentén jobban megérthetjük magát a folyamatot is, amelyet modellezünk.

Ebben az alfejezetben leírjuk a módszer működési elvét. Mivel a módszer egy annotálással kezdődik, ezért az adatpontok ennek alapján már osztályokba sorolhatók. Az olvasó érzése az lehet, hogy ily módon tulajdonképpen egy osztályozási feladatot kell elvégezni. Ezért fontosnak tartjuk ezt az alfejezetet ketté bontani. Az elsőben bemutatjuk az osztályozási feladat megmagyarázását a Shapley-érték segítségével. A második alfejezetben bemutatjuk [MJE21] -ban bevezetett magyarázó módszert és rávilágítunk arra, miben különbözik az osztályozó eljárástól, ha kiindulunk egy annotált adathalmazból.

3.2.1. Az osztályozás értelmezése Shapley-érték segítségével

Ez a rész arra ad választ, hogy a magyarázóváltozók, hogyan hatnak az osztályozásra. A probléma megértésére egy példát hozunk Strumbelj és Kononenko 2010-es cikkéből.[SK10] Tegyük fel, hogy például a Titanic hajóról való menekülés ténye az osztályozó célváltozó (túlélte vagy sem). Szeretnénk meghatározni, hogy mely változók voltak hatással a túlélésre egy adott személy esetében. Erre a célra Strumbelj és Kononenko talált egy mutatót, amiről bebizonyítottak, hogy eleget tesz a Shapley-érték feltételeinek. Fontos

megjegyezni, hogy ez volt az első cikk, amelyben a Shapley-értéket alkalmazzák egy gépi tanulási eljárás megmagyarázására. Azt gondoljuk, ez a cikk megnyitott egy fontos kaput a értelmezhető gépi tanulási modellek felé.

A [SK10] cikkben kiindulnak egy klasszifikációs eljárásból, majd bevezetnek néhány az osztályozáshoz kapcsolódó fogalmat, amiről bebizonyítják, hogy megfelelnek a játékelméletből ismert Shapley-értéknek.

Legyen d magyarázó váltózónk, jelölje $X = \{X_1, \dots, X_d\}$ a magyarázó változók halmazát, és legyen Y az osztályozó célváltozónk, amely K diszkrét értéket vehet fel. Továbbá $\mathcal{A} = \otimes \mathcal{A}_i$, ahol \mathcal{A}_i az X_i diszkrét értékkészlete és $V = \{1, \dots, d\}$

Definíció. Osztályozónak nevezzük a következő f leképezést :

$$f : \mathcal{A} \rightarrow \{0, 1\}^K.$$

Legyen $\mathbf{x} = (x_1, \dots, x_d)$ egy realizáció, amelyhez az osztályozó hozzárendel egy k osztályt, $k \in \{1, \dots, K\}$. Jelöljük $f_k(\mathbf{x})$ -el a klasszifikálót, amely a k osztályba sorol. Értékei 0 és 1 annak alapján, hogy besorolja az adott mintaelementet a k osztályba vagy sem.

Az alapötlet a következő: meg szeretnénk mondani mely magyarázóváltózók járulnak hozzá ahhoz, hogy megmagyarázzák a különbséget aközött, hogy egyes adatpontok milyen osztályba lettek besorolva, ahhoz képest, hogy hova soroltuk volna őket, hogyha egyik magyarázó változó értékét sem ismertük volna. Ezt a különbséget fogjuk definiálni és ezt a magyarázóváltózók halmazának részhalmazához is lehet kötni.

Definíció. Egy $S \subseteq V$ részhalmazhoz tartozó osztályozási különbséget a következő módon értelmezzük:

$$\Delta_k(S) = \frac{1}{|\mathcal{A}_{V \setminus S}|} \sum_{\mathbf{x} \in \mathcal{A}_{V \setminus S}} f_k(\tau(\mathbf{x}^*, \mathbf{x}, S)) - \frac{1}{|\mathcal{A}|} \sum_{\mathbf{x} \in \mathcal{A}} f_k(\tau(\mathbf{x})),$$

$$\text{ahol } \tau(\mathbf{x}^*, \mathbf{x}, S) = (z_1, \dots, z_d), z_i = \begin{cases} x_i^* & , i \in S \\ x_i & , i \notin S \end{cases}.$$

A fenti képletben lévő osztályozási különbség $\Delta_k(S)$ megadja a különbsséget az osztályozás várható értéke között, ha egy $S \subseteq V$ indexű részhalmaz a magyarázó változók között ismert, azzal szemben, ha nem ismert egyik magyarázó változó értéke sem. A továbbiakban egy rögzített k mellett a $\Delta(S)$ jelölést fogjuk használni. Fontos megjegyezni, hogy itt semmilyen előzetes feltételt nem teszünk fel a releváns magyarázó változók tekintetében.

Az eddig használt magyarázó modellek egyike sem veszi figyelembe a változók közötti kölcsönhatást. Ennek fényében a [SK10] cikkben definiálják a kölcsönhatást a következő módon:

$$\Delta(S) = \sum_{W \subseteq S} I(W), \quad S \subseteq V$$

ahol $I(\emptyset) = 0$.

A következő rekurzív definícióhoz jutunk az interakciókat leíró I függvényre, ami adott V és $\Delta(S)$ függvény esetén minden létezik és egyértelmű:

$$I(S) = \Delta(S) - \sum_{W \subset S} I(W), \quad S \subseteq V. \quad (7)$$

A következő lépés, hogy elosszuk az interakciók hozzájárulását a d magyarázó változó között:

$$\varphi_i(\Delta) = \sum_{W \subseteq S} \frac{I(W \cup \{i\})}{|W \cup \{i\}|}, \quad i = 1, \dots, d. \quad (8)$$

Ennek segítségével megadható az a téTEL, amely először köti össze a játekelméletet a gépi tanulási eljárások értelmezésével:

3.2.1. Tétel. $\langle V = \{1, \dots, d\}, \Delta \rangle$ egy koalíciós játék, melynek $\varphi(\Delta) = (\varphi_1, \dots, \varphi_d)$ vektora megfelel a játék Shapley-értékeinek.

Bizonyítás. A bizonyítás első lépése hogy a (7) rekurzív formulából kiszámolható a következő formula:

$$I(S) = \sum_{W \subseteq S} \left((-1)^{|S|-|W|} \Delta(S) \right),$$

ami indukcióval bizonyítható.

Ennek alapján (8) felhasználásával φ_i -re is adható egy nemrekurzív formula:

$$\varphi_i(\Delta) = \sum_{W \subseteq V \setminus \{i\}} \frac{\sum_{Q \subseteq W \cup \{i\}} \left((-1)^{|W \cup \{i\}| - |Q|} \Delta(Q) \right)}{|W \cup \{i\}|}$$

Összeszámolva, hogy hányszor jelenik meg a $\Delta(S \cup \{i\})$ ($S \subseteq V$) a fenti képletben, eljutunk a

$$\varphi_i(\Delta) = \sum_{S \subseteq V \setminus \{i\}} \frac{(d! - |S| - 1)! |S|!}{d!} (\Delta(S \cup \{i\}) - \Delta(S))$$

képlethez, amivel bebizonyítottuk, hogy a $\varphi_i(\Delta)$ értékek megfelelnek a Shapley-értékeknek. \square

Érdemes itt megjegyeznünk, hogy hogyan értelmezhető ebben a környezetben a három Shapley-értékhez tartozó axióma:

- Az első axióma, a $\Delta(S) = \sum_{W \subseteq S} I(W)$ ($S \subseteq V$) dekompozíciónak felel meg.
- A második axióma azt mondja ki, hogyha egy magyarázóváltozó nincsen hatással az osztályozásra, akkor a Shapley-értéke nulla.
- A harmadik pedig azt állítja, hogyha két magyarázóváltozónak ugyanolyan a hatása van az osztályozásra, akkor a hozzájuk rendelt Shapley-értékeknek is azonosnak kell lennie.

A három axiómából látszik, hogy a változók, amik hatással vannak az osztályozásra, azok hatása a Shapley-értékben tükröződni fog.

Sajnos az elméleti eljárás nagyon számolásigényes lenne, ezért a [SK10] cikkben bevezettek egy approximációs eljárást erre a célra, amely egy véletlen visszatevéses mintavételezésre épül. Erre most nem térünk ki ebben a dolgozatban.

Az alfejezetet azzal zárjuk be, hogy felhívjuk a figyelmet, hogy itt arra kapunk választ, hogy egy adott osztályba való besorolási probléma esetén (0-1), mely változik, milyen hatással bírnak. Ebben az esetben az osztályozó függvény csupán 0 vagy 1 értéket vehetett fel.

3.2.2. A klaszterezés értelmezése Shapley-érték segítségével

Most áttérünk a [MJE21] cikkre, amely a jelen dolgozatunkban a klasztereket magyarázó módszer kiterjeszésének kiinduló alapja lesz.

A klaszterezés egy felügyelet nélküli módszer, amely eredménye egy dimenziócsökkenés, olyan értelemben, hogy a mintaelemeket K darab klaszterbe soroljuk be (K tipikusan alacsony, < 10). Nagyon ritka, hogy megsejtjük, hogy pontosan mi okozza a klaszterek kialakulását és ezt egy színezéssel vizualizálni is tudjuk. Ha a klaszterbe sorolás elve nem világos, akkor hasznos, ha a szakértők kezében egy olyan módszer is van, amellyel erre a kérdésre fényt deríthatnak. A [MJE21] cikkben a szerzők a klaszterezést egy csökkengett 2 dimenziós térben végezik, ahol a felhasználó vizualizáció alapján dönt az adatpontok (mintaelemek) annotálásáról.

A dolgozatban a két dimenziós térben való klaszterezésből kilépve az eljárás alkalmazhatóságát magasabb térré is kiterjesztjük a Silhouette mutató (3.1. alfejezet) segítségével.

A Silhouette mutatót két fő lépésben is alkalmazzuk:

- segítségével kiválasztjuk a vizsgált adathalmazon melyik klaszterezési algoritmus produkálja a legjobb mutatókat,
- ezt követően a kiválasztott eljárás alapján megválasztjuk a klaszterek megfelelő számát.

Az ilyen módon meghatározott klasztereket értelmezzük és magyarázzuk a Klaszter Shapley módszerrel. A 1.2. alfejezetben bemutatott Shapley-érték szolgáltatja az eljárás alapját, amit a bevezető fejezetben az (1) egyenlettel definiáltunk.

A következőkben a [MJE21] cikkben bemutatott eljárást fogjuk részletesen bemutatni. A cikkben az algoritmus néhány lépése van leírva, viszont az eljárás matematikai hátterét nem részletezik. Az eljárás célja, hogy Shapley-értékkal kapunk arra választ, hogy mely változók felelősek azért, hogy egy adatpont (mintaelem) egy adott klaszterbe kerüljön.

Tekintsük most az adatpontok vetületét egy kisebb dimenziós térben (a cikkben 2 dimmenzós teret vesznek).

A cikkben bemutatott eljárás két algoritmusból jön létre.

- Az adatpontok annotálása.
- A Shapley-értékek meghatározása.

Az adatpontok annotálására két módszert is adnak. Az egyik, amikor a szakértők kézzel kijelölik a klasztereket (vizualítás alapján), a másik amikor egy klaszterezési eljárás eredményeként történik a klaszterekbe való tartozás. A Shapley-értékek meghatározására a következő módszert fejlesztették ki:

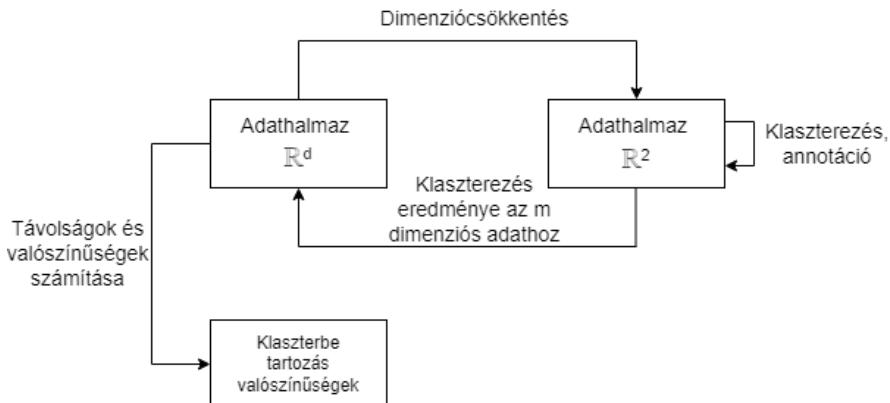
Kiválasztották az adat 20%-át. Ezeknek a pontoknak vették az euklideszi távolságát a centroidokhoz. Ennek alapján mindenhez valószínűségeket rendeltek a következő képlettel:

$$p_k = \frac{d_k}{\sum_{i=1}^K d_i} \tag{9}$$

ahol k a klasztert jelöli, d_k pedig a k . klaszter centrumához vett euklideszi távolság. Könnyen észrevehető, hogy minél nagyobb a távolság, annál nagyobb ez az arány. Tehát minél közelebb van egy adatpont a klaszter centrumához, a p_k arány annál kisebb. Ebből következik, hogy azok a változók

járulnak hozzá az egyes centrumokhoz való "közelséghöz", amelyek Shapley-értékei negatívak. A becsült Shapley-értékek megmagyarázzák a változók szerepét a klasztereződésben. Azok amelyek nagy abszolút értékekkel bírnak, erősebben befolyásolják a klasztereződést. A (9) képlet minden egyes adatponthoz egy valószínűségi eloszlást rendel, amely megmondja milyen valószínűsséggel tartozik az egyes klaszterekhez.

A Shapley-értékek kiszámolásához a cikk a Kernel Shapley becsléseket használja fel [LL17], aminek a működését a 2.1. alfejezetben írtuk le, ahol az f az egyes klaszterekhez tartozó valószínűségeket rendeli az adatpontokhoz. Ez egy fontos megjegyzés a későbbi elemzések értelmezéséhez, hiszen nem teljesen intuitív az a gondolat, hogy azt szeretnénk, hogy $f_{S \cup \{i\}}$ értéke alacsonyabb legyen, mint f_S .



8. ábra. Klasztervalószínűségek kiszámításának folyamatábrája

Tehát a (4) képlettel minden egyes adatpontra kiszámolhatók a attributumokhoz tartozó Shapley-értékek, aminek eredménye minden X_i változóra ($i \in \{1, \dots, d\}$, ha d dimenziós az eredeti adathalmaz): $\varphi_i = (\varphi_{i1}, \dots, \varphi_{im})^T$, ahol s_{ij} az i változó Shapley-értéke a j adatpont esetén ($j \in \{1, \dots, m\}$, ha M adatpont van a tanulóhalmazban, m pedig a redukált minta, a teszthalmaz mérete). Ismét hangsúlyozzuk, hogy a negatív Shapley-értékű változókat ke-

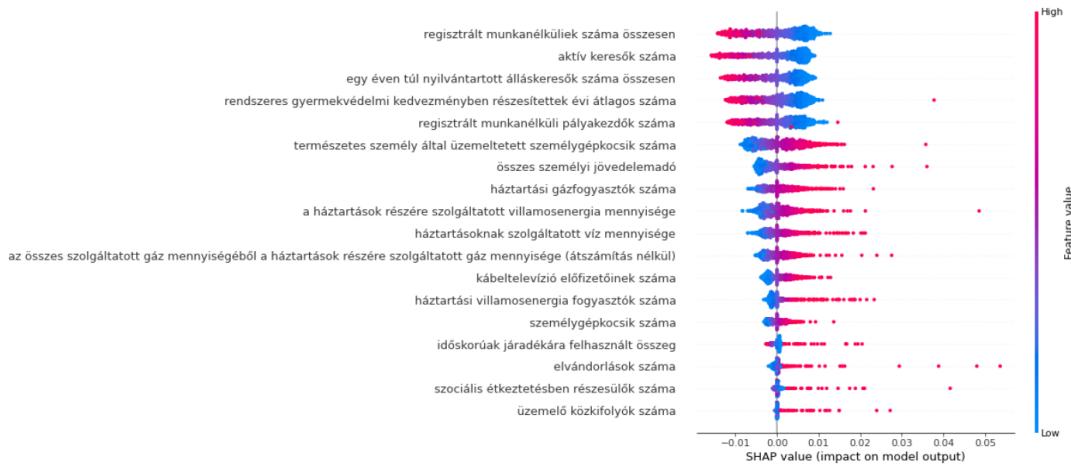
ressük, egy adott klaszter formálásának megmagyarázására.

Abban az esetben ha általánosságban akarunk konkluzókat levonni az egyes attribútumok Shapley-értékeiről, akkor ezeket átlagoljuk az összes adatpont fölött egy adott klaszterre nézve.

Vegyük észre, hogy ebben az esetben nem egy klasszifikációs feladatot akarunk megmagyarázni, hanem, hogy egy adott klaszterbe való tartozás, milyen attributumok által van megmagyarázva. Ez a fő különbség az új módszer és a osztályozás esetében használt módszer között.

Visszakanyarodva az eredeti klaszterezési feladatunkhoz, a Kernel Shapley módszer előtt kiszámoltuk egy új adatpont esetén a klaszterközéppontoktól vett távolságokat, ezeket L1-normalizációval valószínűségekké alakítottuk. Egy nagyobb adathalmaz esetén úgy járunk el, hogy azt tanító- és teszthalmazokra bontjuk, majd a tanítóhalmaz pontjain tanítjuk a Kernel Shapley eljárás lineáris regresszióját, amit a valószínűségekre alkalmazunk. Majd a tanított algoritmus segítségével a teszthalmaz pontjaira megkapjuk a Shapley-approximációkat. Ennek eredménye minden klaszterre egy $m \times d$ -s mátrix, ahol m a teszthalmaz nagysága (az alapbeállítás az adatpontok 20%-a), d pedig a változók száma. Tulajdonképpen adott klaszter esetén minden egyes adatpont minden változójára kapunk egy Shapley-értéket, aminek jelentése, hogy negatív érték esetén az a változó alakítja az aktuális klasztert, a pozitív érték pedig szeparálja, és persze magasabb abszolút érték erősebb hatást vált ki. Éppen ezt is szerettük volna, hiszen így minden klaszter esetén meg tudjuk állapítani, hogy melyik változók alakítják, magyarázzák azt.

A 9. ábrán az értékek egy eloszlása látható változók szerinti fontossági sorrendben, amiről rögtön leolvashatók maguk a változók is és a hatásuk erőssége is, ezekről az eredményekről részlesebben írunk a 4.3.2. fejezetben.



9. ábra. Egy településeket tartalmazó adathalmazban az egyik klaszter legfontosabb változói

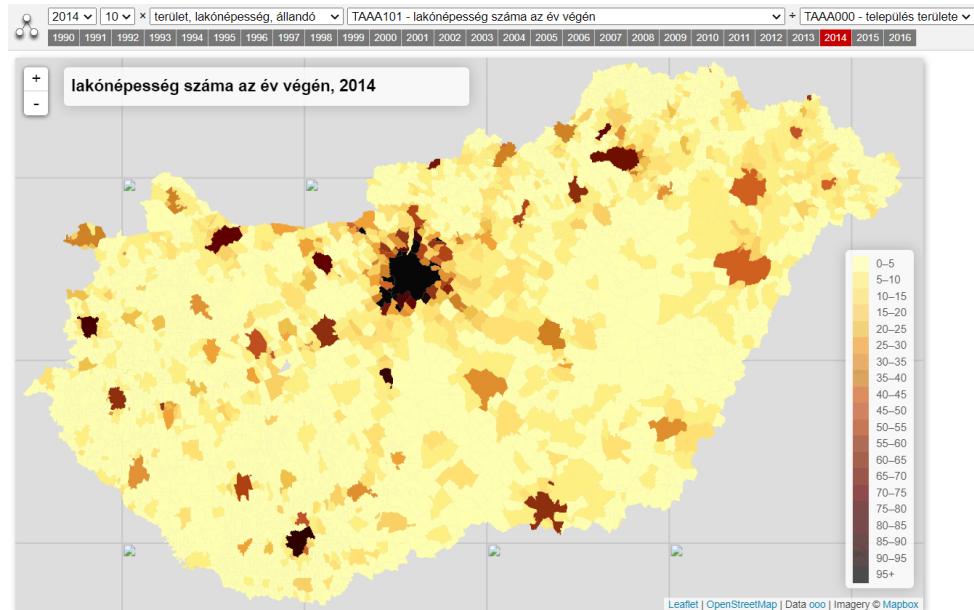
4. Szocio- és demográfiai területen való alkalmazás

Ebben a fejezetben a korábban bemutatott eljárások gyakorlati alkalmazásait mutatjuk be. A programozási feladatokat Pythonban végeztük, a Google Colab felületen, a készített ábrák is innen származnak. Először a 4.1. alfejezetben a felhasznált adathalmazt írjuk le, majd a 4.2. és 4.3. alfejezetekben a regressziós és klaszterezési alkalmazások jelennek meg. A klaszterezés egyfajta hozadékaként anomáliadetektálás is végezhető, erről a 4.3.3. alfejezetben ejtünk szót.

4.1. Adatbemutatás

Egy gazdaságföldrajzi adathalmazzal dolgoztunk a dolgozat keretein belül, ami Dr. Szakadát István tanár úrtól származik, aki az o-o-o.hu weboldal és egy rendkívül gazdag adatbázis üzemeltetője. Az elmúlt 30 év adatai érhetők el településekre lebontva éves szinten, így lehetőség nyílik a több, mint 3000 magyar település együttes vizsgálatára. A weboldalon szemléletes il-

lusztrációk tekintetében meg terképeken (10. ábra), az adatbázisba gyűjtött adatok fő eredeti lelőhelye pedig a Központi Statisztikai Hivatal volt.



10. ábra. Egy az o-o-o.hu weboldalon található interaktív térképek közül

A többszáz elérhető statisztika közül éppen az aktuális feladatnak megfelelően válogattunk, ezek a választások a megfelelő fejezetekben szerepelnek. Általánosságban jellemző ezekre az adatsorokra, hogy rendelkeznek adathibákkel, igaz igen alacsony számban. Továbbá nem minden évre érhető el a legtöbb statisztika, ezért az alkalmazásaink során csak a 2010-es adatokra támaszkodtunk, amelyik az egyik legtöbb statisztikával rendelkezik. Ezért elég volt automatizált előfeldolgozó módszerekhez fordulnunk, amelyek ki töltötték a hiányzó adatokat. Továbbá egy adathalmaz-specifikus lépés volt minden esetben a teljes adattábla leosztása az év végi lakosságszámmal normálás céljából, hiszen így a kisebb-nagyobb lakosságszámkból adódó egyenetlenségeket is ki tudtuk küszöbölni. A 11. ábrán látható egy Pandas adattábla részlet, amelyen a fenti lépéseket elvégeztük.

label	villamosenergia mennyisége	az összes szolgáltatott gáz		regisztrált munkanélküliek	személygépkocsik száma
		a háztartások részéről szolgáltatott szolgáltatott mennyisége	a háztartások részére szolgáltatott mennyisége		
ind					
(Aba, 1512, 47.0343556977497, 18.524754962635)	0.926401	0.360291	0.026744	0.056055	0.226786
(Abaliget, 1035, 46.1438461930372, 18.1181108415909)	1.316781	0.000000	0.034247	0.099315	0.330479
(Abasár, 2209, 47.8006871447644, 20.0083983387857)	1.375349	0.415636	0.032708	0.060630	0.378141

11. ábra. Az előfeldolgozást követően kapott adattábla részlete

4.2. Regresszió

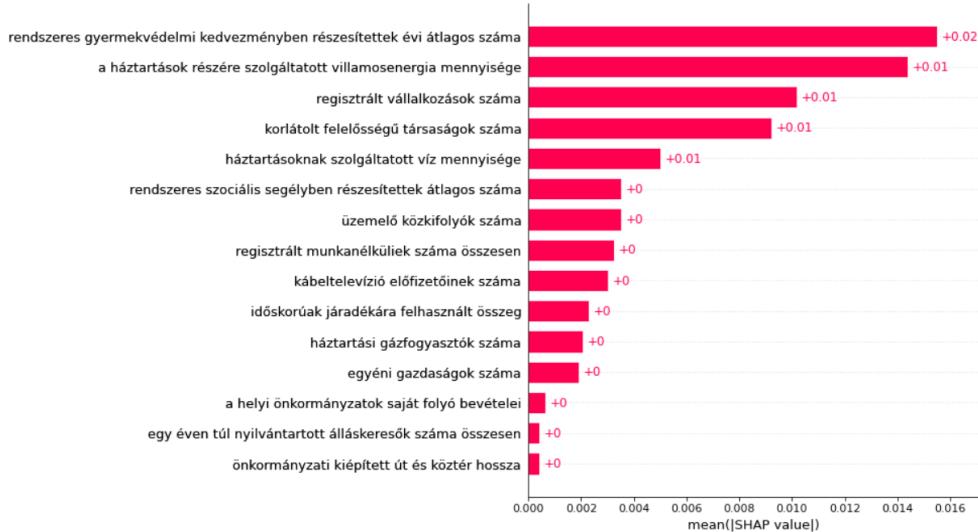
Ezen munka során a regressziós célváltozó a személygépkocsiszám volt, amelyet a statisztikák elérhetőségétől függően 10-15 magyarázóváltozóval bocsültünk. A gépkocsiszám egy szinte tökeletes helyettesítője a személyi jövedelemadónak (a két változó közötti korreláció 1 közel), amelyről kevesebb adat állt rendelkezésre, viszont egymagában is jól jellemzi egy település gazdasági helyzetét. A következő magyarázóváltozókat használtuk:

- a helyi önkormányzatok saját folyó bevételei
- a háztartások részére szolgáltatott villamosenergia mennyisége
- egy éven túl nyilvántartott álláskeresők száma összesen
- egyéni gazdaságok száma
- háztartási gázfogyasztók száma
- háztartásoknak szolgáltatott víz mennyisége
- időskorúak járadékára felhasznált összeg
- korlátolt felelősséggű társaságok száma

- kábeltelevízió előfizetőinek száma
- regisztrált munkanélküliek száma összesen
- regisztrált vállalkozások száma
- rendszeres gyermekvédelmi kedvezményben részesítettek évi átlagos száma
- rendszeres szociális segélyben részesítettek átlagos száma
- önkormányzati kiépített út és köztér hossza
- üzemelő közkifolyók száma

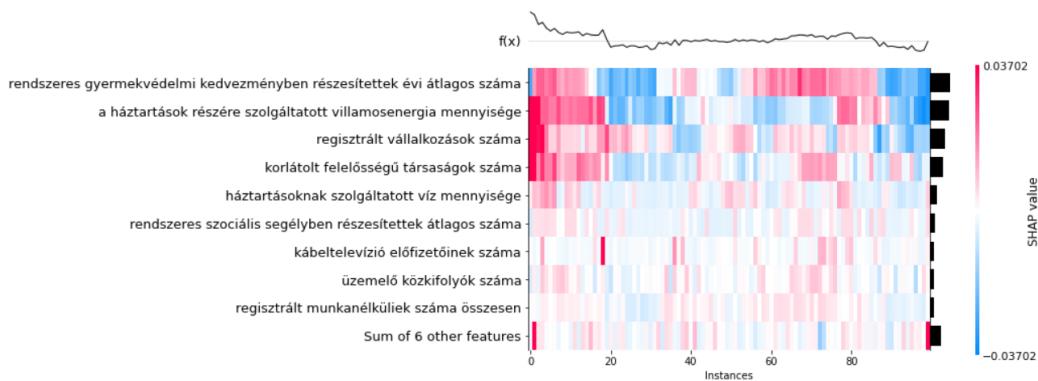
A fenti változók manuális kiválasztása után itt is hozzáartozott az előfeldolgozáshoz a lakosságszámmal való normálás. Az XG Boost modellt az adat véletlenszerűen kiválasztott 80%-án tanítottuk be, a maradék 20%-on pedig becsültük a Shapley-értékeket.[CG16] Ezek a becslések az egyes települések esetén is magyarázhatók, először azonban egy globális mutató sor a 12. ábrán, ami tulajdonképpen a 20%-on kapott értékek átlaga. Erről az ábráról egy fontossági sorrend olvasható le, mivel abszolútértéket nézünk nem derül ki egyelőre, hogy negatív vagy pozitív irányba hatnak a változók, ezt majd az egyéni esetekben fogjuk látni.

Az 12. ábrán látjuk, hogy a legfontosabb változók a gyermekvédelmi támogatásban részesülők száma, áramfogyasztás és a vállalkozások, kft-k száma az egyes településekben. Bár sejtjük, hogy általában ezek milyen hatást gyakorlnak a gazdasági erőre, még szeretnénk ténylegesen megbizonyosodni erről. Erre a célra jól megfelel a 13. ábra, amelyen jelen esetben a teszthalmaz 100 adatpontjának Shapley-értékeit szemléltetjük egyszerre. A vékony oszlopok tartoznak egy településhez, ezek kék színe változónként negatív Shapley-értéket jelöl, a piros pozitívat. Az egyes változókhöz tartozó sorok végén ismét megjelenik az abszolútértékek átlaga, illetve felül a kapott predikciók értéke, ami nem más, mint a célváltozó várható értékének és az egy oszlopban



12. ábra. A teszthalmazon kapott Shapley-értékek átlaga

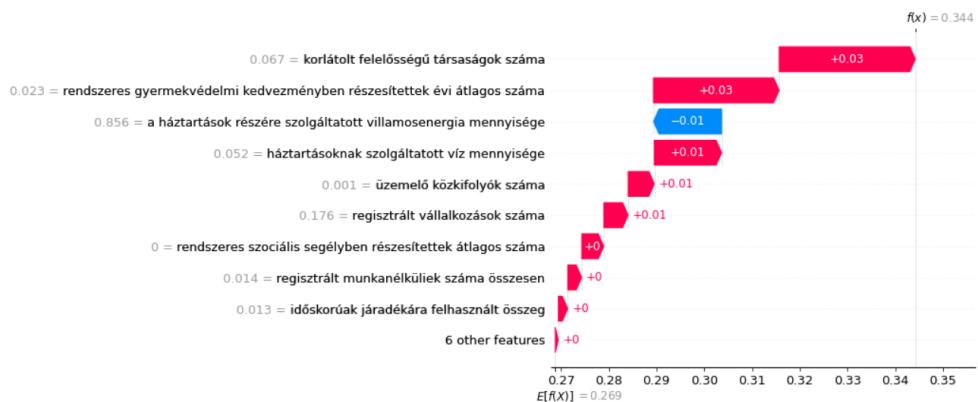
szereplő értékeknek az összege. Látható is, hogy azon oszlopok felett, ahol a kék, negatív értékek dominálnak, az $f(x)$ becslés is alacsony. Továbbá innen már az is jól látszik, hogy az egyes változók többnyire negatív vagy pozitív hatással vannak a gazdasági erőre.



13. ábra. 100 adatpont Shapley-érték vizualizációja

A Shapley-értékek globális vizsgálata után tekintsünk most egyéni eseteket is, melyek bár hasonlóak a 13. ábrán látottakhoz, ahol egyszerre 100

ilyen szerepel, de így szemléletesebb magyarázatot tudunk adni az egyes eredményekre, ami hitelképesség megítélésénél vagy orvostudományi alkalmazásoknál kifejezetten el is várt a megrendelők részéről. A 14. ábrán Üröm adatai szerepelnek, lényegében a célváltozó tanítóhalmazon számított várható értékéből kiindulva jutunk el a tényleges becsléshez a Shapley-értékek hozzáadásával.



14. ábra. Az Ürömre vonatkozó predikció magyarázata

A következőkben egy finomítást végeünk, amely további elemzői kérdésekre nyújthat választ. Adott változó értékkészletét több részre vágjuk és újra elvégezzük a Shapley-érték számítást, úgy, hogy a különböző részhalmazokra saját modellekkel tanítunk. Ezt a módszert "cohort" (csoport) eljárásnak nevezi a programcsomag eredeti szerzője. Mivel a dolgozat készítésekor a eljárás nem műköött ezért aját magunk implementáltuk eztaz eljárást, így esetleges eltérések előfordulhatnak. Az értékkészlet vágását vagy szakértők végzik, valamilyen háttér információ alapján, vagy úgy kell megválasztani, hogy a célváltozóra való hatásnak megfelelően lehetőleg két homogén csoportba sorolja az adatpontokat. A vágást döntési fával végeztük, amellyel elérhető, hogy homogén részhalmazokat kapjunk eredményül.

A jelen alkalmazásban egy vágást végeztünk a település adathalmazon, ami a munkanélküliségre vonatkozott, így kaptunk egy alacsony és egy ma-

gas munkanélküliségű részhalmazt. A 15. ábrán először az alacsony rész Shapley-értékei, itt lényeges különbséget nem tapasztalunk, ugyanazok a fontos változók, csak a sorrend változik.



15. ábra. Alacsony munkanélküliségű települések Shapley-értékei

Magas munkanélküliség mellett már megfigyelhető eltérés a nagyságrendekben, illetve még érdekes lehet, hogy a kábeltelevízió előfizetők számának fontossága teljesen visszaesett. Ennyiből is érezhetjük, hogy amennyiben az adatpontok egy olyan részhalmazára vagyunk kiváncsiak, melyek rendelkeznek egy jól elkülöníthető tulajdonsággal, már érdemes ehhez egy külön, új modellt tanítani, hiszen eltérhetnek az eredmények a teljes és a szűkebb halmazon.

Egy másik fontos megjegyzés a módszer ezen adathalmazon való alkalmazásához, hogy a transzformációkra az osztást leszámítva nem érzékeny. Így annak még volt hatása, hogy normáltunk a lakosságszámmal és ezzel egy korrekt adatsorhoz jutottunk, az összes többi vizsgált transzformáció nem módosította a végső Shapley-értékeket (ugyanazon az adatsoron). Ilyenek voltak az adathalmaz skalárral szorzása, több változó egymással szorzása, összadása, standard skálázás. Ilyen módon új, mesterséges változókat hoztunk létre, amelyek jelenléte nem módosította az eredeti Shapley-értékeket, illetve maga az új változó 0 értéket kapott. Egyedül akkor nem volt ez így, amikor



16. ábra. Magas munkanélküliségű települések Shapley-értékei

a változókat egymással osztottuk. Tehát más eredményeket láttunk, mint a 2.2. alfejezetben, így kijelenthető, hogy minden különböző adathalmazon szükséges megvizsgálni az egyes transzformációk hatását Shapley-értékekre, mivel várhatóan különböző eredményeket kapunk.

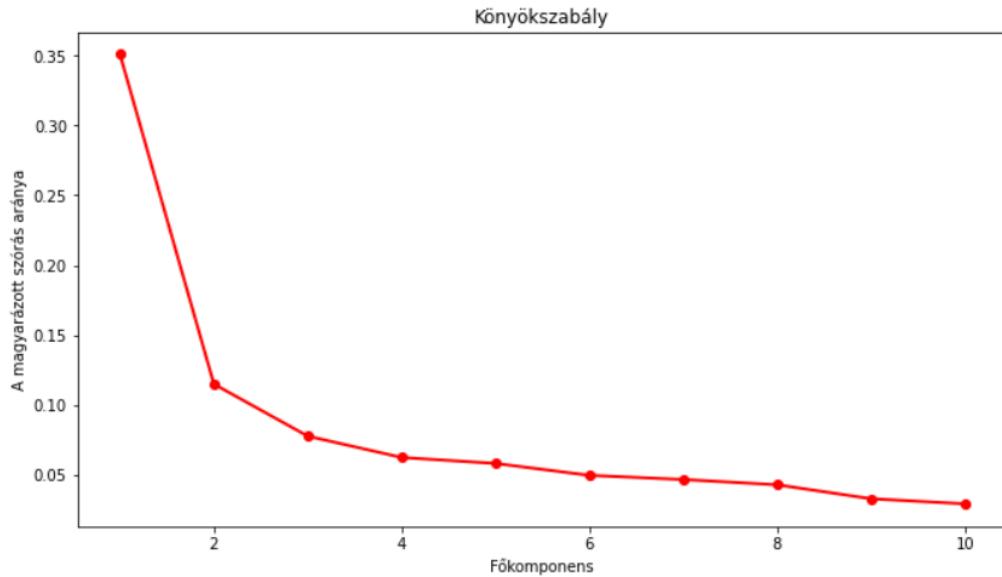
4.3. Klaszterezés

Az adatbázisban több évtizeden átívelően megtalálható több száz statisztika településszinten, amelyek közül kézzel választottunk ki olyanokat, amelyek véleményünk szerint jól jellemzik a települések gazdasági helyzetét (felügyelet nélküli feladatról lévén szó, itt már nem célváltozó a gépkocsiszám). Ehhez segítségünkre volt egy korábbi közös projekt, amit Dr. Szakadát István tanár úrral végeztünk és szintén klaszterezési feladatot hajtottunk végre, de nem magyaráztuk az eredményeket a Shapley-értékkel. Így végül adódott 18 változó, amelyekről úgy gondoltuk, hogy jó magyarázóerővel bírnak és manuálisan már nem tudtuk tovább szűrni őket. Ugyanakkor szándékosan bevettünk néhány, már szemmel is redundánsnak tűnő statisztikát, amelyekkel tulajdonképpen a módszert is próbára szerettük volna tenni, hogy ezeket megtalálja-e magától. A változók a következők:

- a háztartások részére szolgáltatott villamosenergia mennyisége
- aktív (állás)keresők száma
- az összes szolgáltatott gáz mennyiségéből a háztartások részére szolgáltatott gáz mennyisége
- egy éven túl nyilvántartott álláskeresők száma összesen
- elvándorlások száma
- háztartási gázfogyasztók száma
- háztartási villamosenergia fogyasztók száma
- háztartásoknak szolgáltatott víz mennyisége
- időskorúak járadékára felhasznált összeg
- kábeltelevízió előfizetőinek száma
- regisztrált munkanélküli pályakezdők száma
- regisztrált munkanélküliek száma összesen
- rendszeres gyermekvédelmi kedvezményben részesítettek évi átlagos száma
- személygépkocsik száma
- szociális étkeztetésben részesülők száma
- természetes személy által üzemeltetett személygépkocsik száma
- összes személyi jövedelemadó
- üzemelő közkifolyók száma

Ezen ponton alkalmaztuk a Klaszter Shapley módszert, ami magas eredményi dimenziószám mellett is képes kiválasztani egy adott klaszterezés esetén, hogy mely változók definiálják az egyes klasztereket. Első lépésben ehhez szükség volt egy klaszterezésre, amit az eredeti dimenzióinál lényegesen alacsonyabb dimenzióban célszerű végezni. A Klaszter Shapley által meghatározott fontos változók azonban csak a folyamat végén állnak rendelkezésre, így ekkor még hagyományosabb dimenziócsökkentő eljárásokhoz lehet fordulni.

Egy lényegi eltérés az eredeti munkához képest, hogy ők mindig 2 dimenzióba csökkentettek, azonban a mi kísérleteinkből is kiderül, hogy a módszer nem veszít a hatékonyságából akkor sem, ha 2-nél magasabb dimenzióba csökkentünk. [MJE21] Ebben a munkában a Főkomponens-analízis volt a használt eljárás, azaz Principal Component Analysis (továbbiakban PCA). Ennek alapgondolata, hogy a dimenziót úgy csökkentse, hogy az újonnan kapott változók az eredetiekben jelen lévő szórást minél nagyobb mértékben magyarázzák, a kapott változókat nevezük főkomponenseknek. [JC16]



17. ábra. Az egyes főkomponensek által magyarázott szórás

Azt, hogy pontosan hány változóra csökkentsünk, a magyarázott variancia arányával tudjuk eldönten. Erre segítséget nyújt a Könyökszabály is (Scree plot), amelyen az egyes faktorok által magyarázott varianciát ábrázolják csökkenő sorrendben. A mi adatunkhoz készült verzió a 17. ábrán látható. Erről leolvasható, hogy az első 2-3 főkomponens hordozza a magyarázott szórás javát, azonban mi összesen 85%-ot tüztünk ki célul ebben a dolgozatban. Ehhez szükség volt az első 10 főkomponensre, így végül az eredeti 18 dimenziós adathalmazból főkomponens-analízissel 10 dimenziósat készítettünk. Fontos még megjegyezni, hogy ezek a változók nem bírnak hasonló jelentéssel, mint az eredeti változók, de mindegyik tartalmaz információt azokról.

4.3.1. Silhouette mutató alapú elemzés

Az elemzés következő lépéseként a kapott 10 dimenziós adathalmazon végrehajtunk egy klaszterezést. Ehhez segítséget nyújt a 3.1. fejezetben bemutatott Silhouette módszer, amivel kiválaszthatjuk az adatunkhoz leginkább illő algoritmust és klaszterszámot. Az eljárás jellegéből fakadóan olyan algoritmusok jöhettek szóba, ahol állítható bemeneti paraméter a klaszterszám, így végül 3 opciót vizsgáltunk meg:

- K-közép klaszterezés
- Spektrális klaszterezés
- Hierarchikus klaszterezés

A k -közép klaszterezés során az adatpontokat k darab diszjunkt klaszterbe soroljuk, amelyekben a szórás közel egyforma. Jól skálázódik sok minta esetén is, azonban igazán jól a gömbszerű klasztereket találja meg, nem reagál jól az elnyújtott sokaságokra. Lusta tanuló algoritmus, ami annyit tesz, hogy akkor dolgozik, ha új adatpontot kell osztályozni. Ezt pedig úgy teszi,

hogy az eddigi klaszterek pontjaiból számít egy átlagos pontot, amit centroidnak neveznek. Ez nem feltétlenül egy létező adatpont, de ugyanabban a térben létezik és az új adatpont azt a címkét kapja, amelyik centroidhez a legközelebb helyezkedik el. [M⁺67]

Amikor spektrális klaszterezést hajtunk végre, akkor lényegében szintén egy k-közép algoritmust futtatunk, azonban ezt megelőzi néhány előkészítő lépés, amelyek következtében ez a változat más szerkezetű adaton fog a legjobban működni. Valójában a Laplace mátrix legnagyobb sajátértékeihez tartozó sajátvektorain futtatjuk a k-közepet, ezt a mátrixot egy fokszám és egy hasonlósági mátrix különbségeként kapjuk. A hasonlósági mátrix gráfok esetén lehet egyszerűen egy szomszédossági mátrix, általánosan az alábbi $n \times n$ -es, n adatpont esetén:

$$A_{ij} = e^{-(\epsilon \|X_i, X_j\|)^2}$$

ahol ϵ egy skála paraméter. A fokszám mátrix pedig ebből a következő A sorösszegeiből (vagy oszlopösszeg) képzett diagonális mátrix:

$$D_{ii} = \sum_{j=1}^n A_{ij}$$

A fenti mátrixok birtokában a Laplace mátrix megkapható:

$$L = D - A$$

Ezzel a módszerrel, ha a hasonlósági mátrix a szomszédossági mátrix, normalizált gráf vágásokat is találhatunk, továbbá a klaszterezést tekintve, legjobban olyan alapszerkezetű adaton működik, aminek a 2-dimenziós képe egymásba ágyazott körökkel mutat.

A hierarchikus klaszterezés általánosan klaszterekeket épít fel, vagy úgy, hogy szétvágja őket lépéseként vagy összeilleszti. Végül egy úgynevezett dendrogramhoz jutunk, ami egy faábrázolása a kapott klasztereknek, amely-

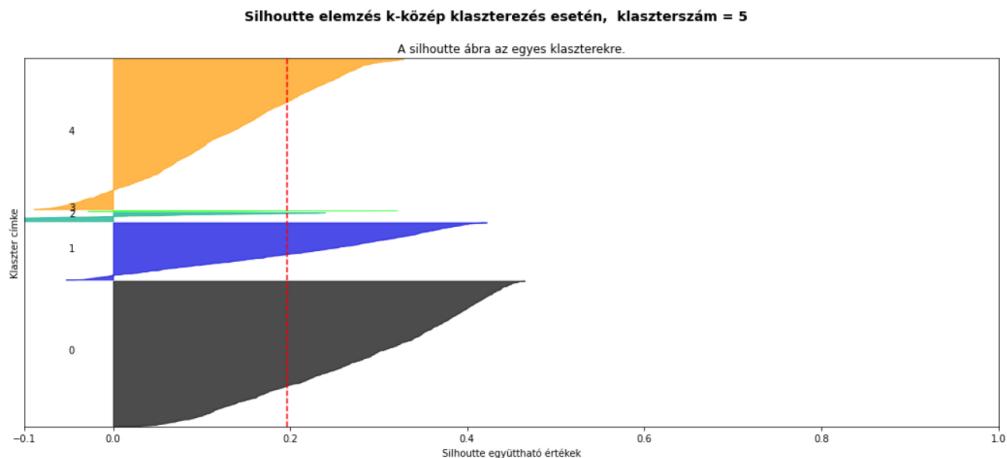
ben a csúcs az összes adatpontot tartalmazó klaszter, a levelek pedig az egyetlen pontot magába foglaló klasztereket reprezentálják. Az agglomeratív klaszterezés az a változat, amikor egymásnak klaszterekből fokozatosan épülnek fel a nagyobb elemszámlásúak. Különböző kritériumok léteznek arra, hogy mi alapján történjen az egyesítés, mi a Ward-féle kritériummal dolgoztunk, ami a négyzetes eltérés-összegek minimalizálásával vonja össze a klasztereket, egyszerre minden egy párt. A k-középhez hasonlóan ez a módszer is jól skálázódik a minták számával, azonban erőforrást tekintve előnyt jelent, ha van valamilyen megkötés a pontok közötti kapcsolatokra, hiszen ebben az esetben nem kell minden egyes párra végigszámolni a négyzetes eltéréseket.

A tényleges számításokat Pythonban, a Scikit-learn csomag beépített algoritmusaival végeztük. [PVG⁺¹¹] A használt paraméterek a klaszterszám kivételével mindenhol az alapbeállítás szerintiek. Az összes kipróbált algoritmusnál és klaszterszámánál két dolgot vettünk szemügyre: a globális Silhouette együttható átlagot és a kapott ábrát. A 3 kipróbált algoritmus a fentebb említettek, ezeknek pedig 2-től 10-ig adtunk meg bemenetként klaszterszámot.

Általánosan igaz, hogy a klaszterszám növekedésével csökken az átlagos Silhouette együttható is, bár itt személyes preferenciák is közbeszólhatnak, hiszen az adott feladathoz szerettünk volna több, mint 2-3 klasztert meghatározni, hiába ezekben az esetekben a legjobb az átlagos mutató. A további értelmezés előtt felsorolásszerűen a kapott Silhouette átlagok:

- K-közép: k=2: 0.2994, k=3: 0.1847, k=4: 0.1944, k=5: 0.1966, k=6: 0.1778, k=7: 0.1526, k=8: 0.1409, k=9: 0.1394, k=10: 0.1438
- Spektrális klaszterezés: k=2: 0.2965, k=3: 0.1805, k=4: 0.1520, k=5: 0.1284, k=6: 0.1303, k=7: 0.1083, k=8: 0.1059, k=9: 0.0929, k=10: 0.0892
- Hierarchikus: k=2: 0.2884, k=3: 0.2936, k=4: 0.1215, k=5: 0.1267, k=6: 0.1189, k=7: 0.1275, k=8: 0.1286, k=9: 0.1299, k=10: 0.1322

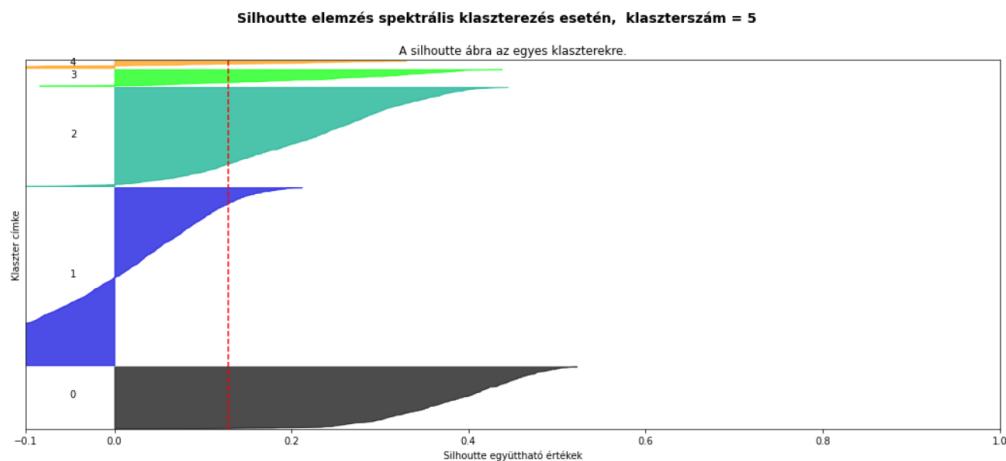
Ebből is látszik, hogy a legjobb pontszámok 2-3 klaszter esetén adódnak és itt is kevéssel a k -közép vezet, viszont ha több klasztert szeretnénk, akkor már egyértelmű, hogy rá esik a választásunk. A hierarchikus klaszterezés nem teljesít rosszul magas számú, 9-10 klaszter esetén sem, ami betudható annak is, hogy maga a módszer a 3 közül a leghatékonyabb magas klaszterszám esetén. A k -közép listájából kiindulva nekünk optimális a $k = 5$ választás (hiszen szeretnénk több, mint 2 klasztert), lentebb látható, hogy az ábrák mit mutatnak ebben az esetben a különböző algoritmusoknál.



18. ábra. 10 főkomponens klaszterezése után a k -közép algoritmus Silhouette ábrái

Először a 18. ábrán a k -közép esete, a 3 vizsgált algoritmus közül ennélfoglalt volt a legnagyobb az együtthatók átlaga (0.1966). Kaptunk 3 jó méretű és pozitív együtthatós klasztert, azonban a legnagyobbnál adódott számos negatív együtthatós adatpont is, illetve itt az adatpontok többsége az átlag alatt található. Nem túl meggyőző a másik két klaszter, mivel minden kettő igen alacsony elemszámú, ráadásul a 2-es számú klaszterben látszólag több a negatív együtthatós minta, mint a pozitív. A 3-as számú klaszterben feltéhetőleg olyan adatpontok szerepelnek, amelyek egy vagy több változójuk miatt semmilyen másik klaszterbe nem illeszkednek, azaz anomáliák, a Silho-

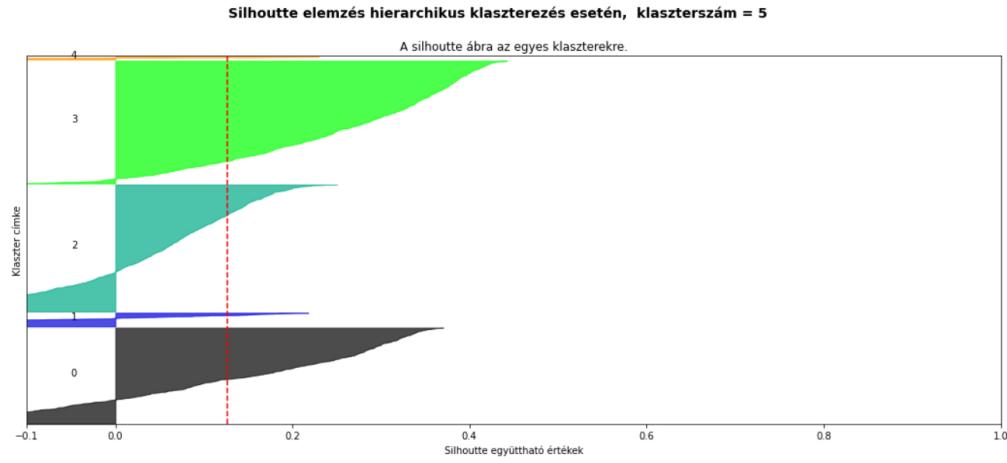
utte módszer ezen alkalmazásáról részletesebben a 4.3.3. fejezetben még lesz szó. Annak ellenére, hogy találtunk egy ilyen klasztert, hasznos megjegyezni, hogy még mindig magasabb a globális együttható-átlag, mint a $k = 4$ esetben (ahol 0.1944 volt), így nem érdemes összevonni a klasztereket azáltal, hogy kevesebbnek választjuk a klaszterek számát.



19. ábra. 10 főkomponensnél a spektrális klaszterezés Silhouette ábrái (eltolni a skálát hogy látszódjon a negatív oldal is teljesen)

Következzen a spektrális klaszterezés, ez a 19. ábrán tekinthető meg. A k -középpel szemben itt már csak 2 olyan klasztert kaptunk, amelyek méretere is megfelelnek és az együtthatók többnyire pozitívak, sőt az átlagot is meghaladják. A globális átlag is jelentősen elmarad a k -középtől, itt ez 0.1284. Most a két alacsony elemszámú klaszter is többnyire pozitív együtthatókat hoz, viszont még mindig nem mondhatjuk, hogy ezek jó klaszterek lennének további értelmezés nélkül. A problémát az 1-es klaszter okozza ennél az algoritmusnál, ami a legnagyobb elemszámú, de többségében ezek a mintapontok negatív együtthatóval rendelkeznek és alig van olyan, ami meghaladja az átlagot, így ez a klaszter egyedül felelőssé tehető a k -középtől való elmaradásért.

Végül a hierarchikus klaszterezés, ami minimálisan még a spektrális klasz-



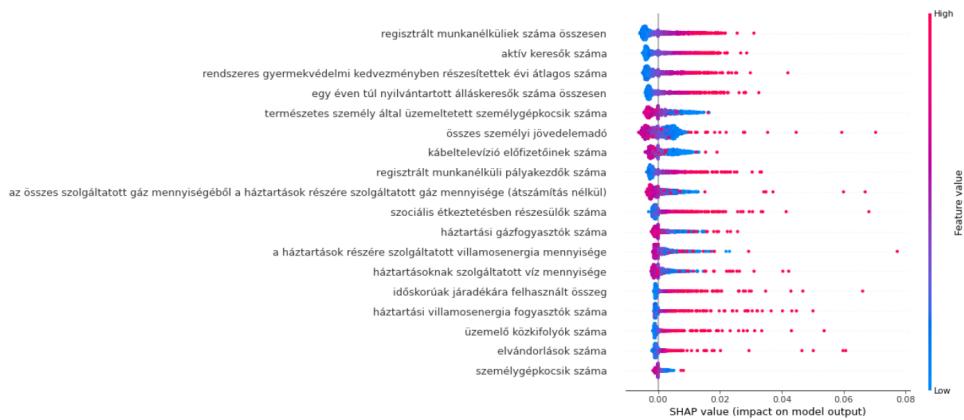
20. ábra. 10 főkomponensnél a hierarchikus klaszterezés Silhouette ábráí

terezésnél is rosszabb együttható átlaggal rendelkezik (0.1267). A 20. ábrát figyelve látjuk, hogy megint 3 nagyobb elemszámú klasztert kaptunk, de ezek közül már csak az egyikről mondható el, hogy megfelel az igényeinknek. A másik kettő nagy klaszter inkább lefelé húzza az átlagot, mint ahogyan a két kisebb klaszter is negatívan járul hozzá a klaszterezés minőségéhez. Bár itt újra meg lehet jegyezni, hogy a 4-es klaszter annyira kevés pontot tartalmaz, hogy azt anomáliaként is karakterizálhatjuk.

Az eddigieket összefoglalva, döntöttünk amellett, hogy az eredeti 18 dimenziót kevesebben végezzük el a klaszterezést. Erre a célra főkomponens analízzissel 10 dimenzióra csökkentettük az adathalmazt, amivel ugyan az egyes változóink elvezették magyarázóerejüket, de együttesen a szórás kellő részét még tudják magyarázni. Az eredeti módszertől eltértünk ezzel, ugyanis ott pontosan 2 dimenzióba csökkentettek minden esetben, hogy lehetőséget adjanak a kézi klaszterezésre is. Ezután a Silhouette elemzés segítségével kiválasztottuk a k-közép klaszterezést $k = 5$ klaszterrel, hiszen a legjobb együttható-átlagot is ez hozta, illetve az ábrák is alátámasztják ezt a választást.

4.3.2. A klaszterekbe sorolás hátterének felderítése

Az előző alfejezetekben előkészítettük az adatot és kiválasztottuk az ehhez optimális klaszterező algoritmust, így már minden készen áll a tényleges klaszterezés futtatására. Ebben az alfejezetben az eredményül kapott klaszterezés értelmezése olvasható, minden egyes klaszternek adunk egy karakterizációt a Shapley-értékek segítségével, aminek részletes háttere a 3.2. alfejezetben megtalálható.

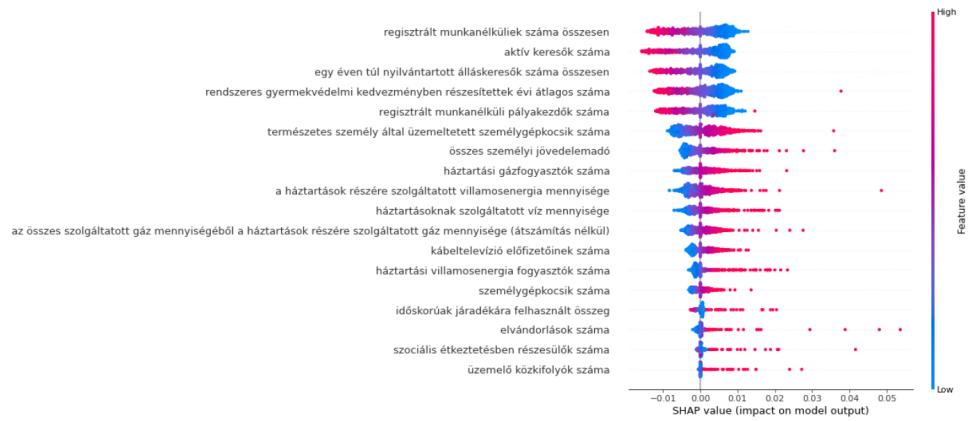


21. ábra. A 0-ás klaszter legfontosabb változói

Az első, 0 indexű klaszter esetén a 21. ábrán láthatjuk a Klaszter Shapley elemzés útján kapott sorrendet. Ez a klaszter az egyik legnagyobb méretű, 1245 települést foglal magába. Emlékeztetőül, a negatív Shapley-értékek segítik elő a klaszterek megformálódását, míg a pozitívak széthúzó erővel bírnak. Ezért találhatóak a munkanélküliséghoz köthető statisztikák az ábra tetején, egészen pontosan azt jelenthetjük ki, hogy alacsony munkanélküliség jellemzi azokat a településeket, amelyek ebbe a klaszterbe kerültek.

A pontok színezése a statisztika konkrét értékére vonatkozik (kék az alacsony, piros a magas értékeket jelenti) és egy sorban az adott változó Shapley-értékeinek eloszlása látható. minden sorban ugyanannyi pont van, ami jelen esetben az összes település 30%-a, ami 945 véletlenszerűen választott adatpontot jelent. Ez azért van, mert ezen 945 pont mindegyikénél kiszámolta

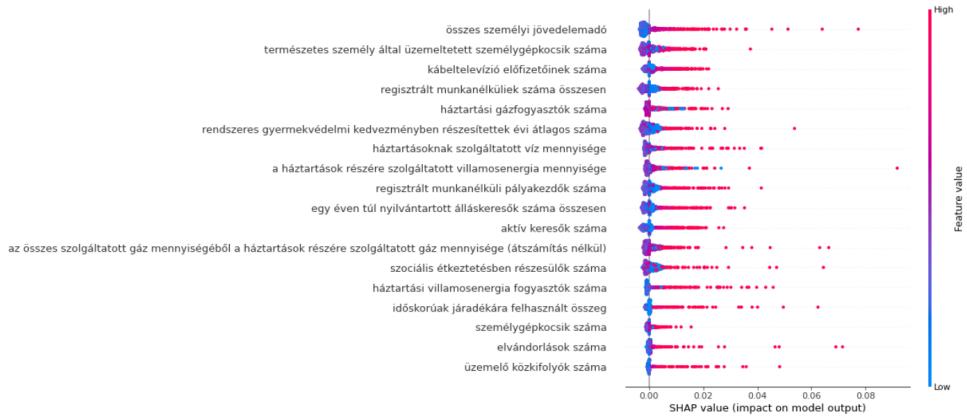
a Klaszter Shapley algoritmus a Shapley-értékeket az összes változóra. Ez a számítás az egész elemzés lelke és igaz, hogy csak egy közelítésről van szó, így is egy erőforrásigényes folyamat. Azonban a módszer jóságát emeli ki, hogy ha nem is lineárisan, de igen jól skálázódik időben az algoritmus, ha nagyobbnak vagy kisebbnek vesszük a teszthalmazt, ami a mi esetünkben 30% volt. Ebben az esetben 17 perces futási időt kaptunk, 20%-nál 9 perces, míg 10%-nál 4 percet kellett várni a Shapley-értékek közelítésének kiszámítására.



22. ábra. Az 1-es klaszter legfontosabb változói

Visszakanyarodva az eredmények értelmezéséhez, a 22. ábrán az 1-es klaszter változólistája látható. Ideális esetben a lista tetején olyan változókat szeretnénk látni, amelyeknél a negatív irányban minél messzebb és minél több pont található. Ebből a szempontból ez egy jobban definiált klaszter, mint a 0-ás, hiszen itt szintén a munkanélküliség kerül előtérbe, de még alacsonyabb Shapley-értékeket kapunk. Ezúttal a magas értékek formálják a klasztert, ami egy 525 elemű halmaz, ezek tehát a magas munkanélküliségű települések.

A 2-es klaszter esetében ismét egy nagy elemszámú klaszterrel állunk szemben, ez a legnagyobb, egy 1296 elemű halmaz. Hasonlóan 0-ás klaszterhez, ez sem túl jól definiált, de lista alapján két igen erősen korreláló változó határozza meg a leginkább. Alacsony személyi jövedelemadó és személygép-



23. ábra. A 2-es klaszter legfontosabb változói

kocsiszám jellemző, azaz a Shapley-féle értelmezésben ezek a szegény települések. Érdekesség még, hogy itt megjelenik két hasonló változó közül az egyik a lista élmezőnyében (természetes személy által üzemeltetett személygépkocsik száma), azonban a másik a lista végén szerepel csak (személygépkocsik száma). Ez egy pozitívumként fogható fel a módszerrel kapcsolatban, ugyanis nem használ többszörösen redundáns változókat.



24. ábra. A 3-as klaszter legfontosabb változói

A következő a 3-as klaszter, ami egy lényegesen kisebb, 78 elemű halmaz. Viszont ahogyan azt a 24. ábrán is láthatjuk, annál karakteresebb klasz-

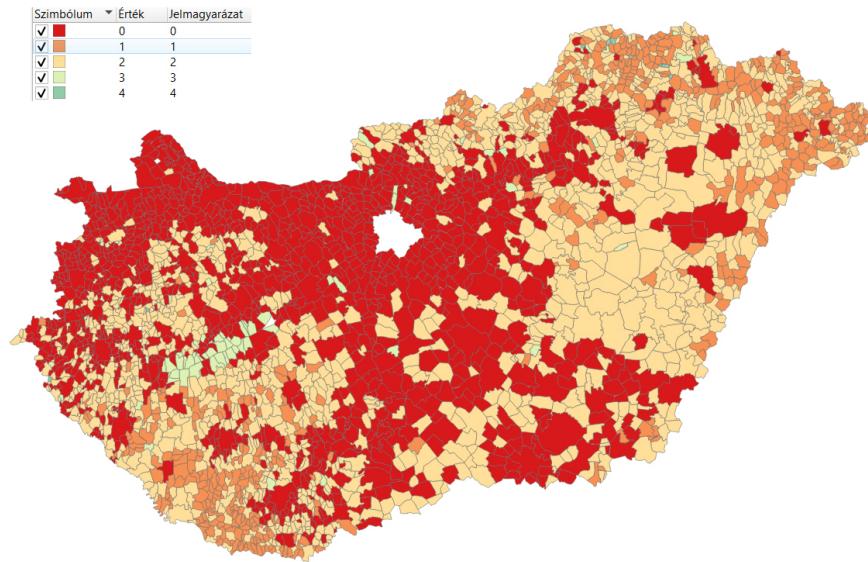
terről van szó, ezek a magas villany, gáz és vizfogyasztó települések. Mivel már lakosságszámarányosan szerepelnek az adatok (a kezdeti lépések között szerepelt ez a normálás), olyan turizmus szempontjából erős települések szerepelnek, ahol kevés lakosra is magas rezsifogyasztás jut. Ennek oka, hogy ezeken a helyeken a be nem jelentett lakosok (turisták) viszik fel ezeket a számokat. Ahogyan majd a térképen is látni fogjuk (26. ábra), ezek tulajdonképpen a Balaton körüli települések és még néhány falu, város elszórva az országban.



25. ábra. A 4-es klaszter legfontosabb változói

Az utolsó, 4-es klaszter a legkisebb, mindössze 8 település tartozik bele. Ezekről részletesebben a 4.3.3. alfejezetben írunk, itt csak megjegyezzük, hogy a rendkívül alacsony elemszámból kiindulva ezek lehetnek anomáliák is. Ha megnézzük a 25. ábrán, hogy melyek a befolyásoló változók, szintén látjuk, hogy a Shapley-értékek is kiugróak néhány speciális esetben. Ezen változók (kábeltelevízió előfizetők, gázfogyasztók, elvándorlások száma) alapján nem is lehetséges egy egységes karakterizációt megfogalmazni, így ezek olyan települések lehetnek, amelyek annyira nem illettek egyik klaszterbe sem, hogy az algoritmus létrehozott számukra egy teljesen újat.

Legvégül pedig a 26. ábrán egy helyen láthatjuk a fent kivesézett klasztereket Magyarország térképén a hovatartozás szerint színezve. Az adattípusok



26. ábra. A települések a fenti klaszterek szerint színezve

miatti eltérés miatt Budapest adatai nem szerepelnek a térképen, egyébként hiánytalanul színezve vannak a települések. Meglehetősen erős regionalitás figyelhető meg az egyes klasztereket illetően, például a 0-ás klaszter Közép- és Északnyugat-Magyarországra jellemző kiegészülve a kelet-magyarországi nagyvárosokkal. Míg az 1-es klaszter délnyugaton és északkeleten jelenik meg, vagy a 3-as turista-klaszter főleg a Balaton környékét foglalja magába.

Kijelenthetjük, hogy az általunk választott változóhalmazt sikerült jobban megismernünk a Shapley magyarázatok segítségével. Ezen változók használatával Magyarország településeit egészen jól elkülöníthető szigetekbe osztottuk szét, amely a szemléletes vizualizáció mentén szakértők által is eredményesen elemzhető. Mindezen túl maguk a klasztermagyarázatok is adnak már egy elemzést, ennek jósága függhet a szakértői véleménytől. Megjegyezzük, hogy ez az elemzés tetszőleges adathalmazon tetszőleges változókkal hasonlóan elvégezhető, ez jól jelzi a módszer hasznosságát.

4.3.3. Anomáliadetektálás

Az előző, 4.3. alfejezetben röviden bemutatott 4-es klasztert elemezzük részletesebben ebben a részben. Ez a klaszter 8 települést tartalmaz, amelyek feltevéseink szerint a változók olyan realizációival bírnak, melyek alapján nem lehetett semelyik másik klaszterbe sem besorolni őket. Megnéztük pontosan mely települések voltak ezek:

- Arka, Megyer, Gagyapáti, Sima, Tornabarakony, Debréte, Tornakápolna, Lendvajakabfa

A következőképpen vizsgáltuk ezen klaszter kialakulásának az okát: megvizsgáltuk a Shapley-értékes elemzésben kapott ábrán a kiugró értékekkel rendelkező változókat, illetve azokat, ahol az egyes települések szemmel láthatóan magas statisztikával rendelkeznek. A vizsgálat itt annyit jelentett, hogy néztük az adott statisztikában 10 legnagyobb értékkel rendelkező települést (továbbra is az egy főre jutó adatokkal számolunk), arra figyelve, hogy a fent felsorolt 8 település valamelyike szerepel-e köztük. Azt nem vizsgáltuk, hogy a 10 legalacsonyabb között ott vannak-e, hiszen előfordulhattak adathiányok, ahol 0-val lett helyettesítve valamelyik statisztika, vagy egyszerűen olyan kicsi a település, hogy valóban 0 némelyik mutató. Felsorolva a megtalált változók, utánuk a települések, amelyek a 4-es klaszterből az első 10-ben vannak:

- elvándorlások száma: Arka, Megyer, Gagyapáti, Sima, Tornabarakony, Debréte, Tornakápolna
- összes személyi jövedelemadó: Gagyapáti, Tornakápolna
- regisztrált munkanélküliek száma összesen: Gagyapáti, Megyer
- személygépkocsik száma: Gagyapáti, Megyer
- üzemelő közkifolyók száma: Gagyapáti, Tornabarakony, Debréte, Tornakápolna, Sima

Azonnal megfigyelhetjük, hogy Gagyapáti egy olyan település, amely minden szempontból kiemelkedik, de Tornakápolna vagy Megyer is 3-3 változónál fordul elő. Ezekről a településekéről külön-külön még nem állíthatjuk, hogy anomáliák lennének, azonban több statisztikát együttesen figyelve már szokatlanul jó mutatókkal rendelkeznek. Fontos megjegyezni még, hogy ez már nem egy automatizált eljárás, teljesen manuálisan működik, így például arra sem kaptunk pontos választ, hogy Lendvajakabfa miért nem szerepel sehol és így mi okból kerülhetett ebbe a klaszterbe. Számos további lehetőséget végigvizsgálva kaphatnánk választ, ez a módszernek egy jövőbeli továbbfejlesztése lehet.

5. Összefoglalás

Munkánk összegzéseként elmondható, hogy sikeresen megismertünk és alkalmaztunk egy még viszonylag újnak számító módszert egy régóta fennálló probléma megoldására, ami a black-box gépi tanulási modellek kimeneleinek megmagyarázását képezi. Az alapvető Shapley-értékkel kapcsolatos játékelmeli fogalmak és tételek bevezetését követően megadtuk az eljárás egyik legfontosabb részének, a Kernel Shapley módszernek a leírását, amely a Shapley-értékek közelítéséért felel. Ezt követően alkalmaztuk a Shapley-féle elemzést egy regressziós és klaszterezési feladaton, miközben az eredeti folyamatokon különféle módosításokat végeztünk, azokat kiegészítettük, illetve további alkalmazhatóságokra világítottunk rá.

Amikor a regressziós modellt vizsgáltuk, külön figyelmet fordítottunk a tanítást megelező transzformációk hatására, amiről korábban keveset szolt az irodalom. Azt találtuk, hogy különböző adathalmazokon meglehetősen más eredményeket produkálnak ugyanazon transzformációk, és ezeket a Shapley-értékek tükrözni tudják. Ezért a bevezetésük előtt hasznos lehet egy előtanulmányt készíteni a Shapley-értékek segítségével.

Miután meghatároztuk a globális értelemben fontos változókat, az átl-

gos Shapley-értékek alapján, lehetőség nyílik a dimenziócsökkentésre, hiszen kiindulva egy magas dimenziós térben megkaptuk a változók fontossági sorrendjét, és ez lehetőséget ad arra, hogy elvégezzük az elemzést alacsonyabb dimenzióban is a kapott változókkal, így egy egyszerűbb modellt kaphatunk.

A Shapley-értékek lehetőséget biztosítanak egyéni esetek felkutatására is, ami rendkívül széles körben felhasználhatóvá teszi.

A klaszterezés esetében bemutattuk a Klaszter Shapley módszert, ami szintén támaszkodva a Kernel Shapley közelítésekre megadta, hogy az egyes klaszterekbe mely változóik miatt kerülnek az adatpontok. Az eredeti ötletgazdákhoz [MJE21] képest, mi egy segédeljárás, a Silhouette mutatók segítségével határoztuk meg a klaszterezéshez használt algoritmust és ebben a klaszterek számát. Ezzel elértük, hogy a vizsgált lehetőségekhez mérten valóban a legjobb modellt elemezzük, továbbá 2-nél magasabb dimenzióban is elvégezhetővé válik a módszer. További újdonság, hogy rámutattunk, hogy a módszer anomália vizsgálatra is alkalmas.

Az eljárásokat egy valós adatokon alkalmaztuk, ahol eddig ezeket a módsereket nem használták. Azt kaptuk, hogy a magyar települések klaszterezéskor, milyen gazdasági és társadalmi mutatók alakítják a kapott klasztereket, ezáltal szintén egyfajta dimenziócsökkentést végeztünk, hiszen elhagyható minden olyan változó, ami nem befolyásolja egyik klaszter kialakulását sem. Végül pedig a módszer hozadékaként anomáliákat is megfigyeltünk, amelyek nem kerültek be egyetlen jól karakterizálható osztályba sem, majd külön-külön megvizsgálva ezeket tapasztaltunk összefüggést egyes kiugró mutatókkal.

A módszer jövőbeli lehetséges fejlesztései között szerepelhet a paramétertér bővítése. Egyszerűen a mostani megoldásunkban csak olyan klaszterező eljárásokat vizsgáltunk, amelyekben állítható paraméter a klaszterszám, ezen a téren lehetne változásokat eszközölni. A regressziót pedig egyfajta, javasolt gépi tanulási modellt alkalmaztunk, de itt is végezhető volna egy hasonló elemzés, amivel kiválasztunk egy pontosabb modellt az adathalmazunkhoz.

Általánosságban az is elmondható, hogy az alapbeállításokkal alkalmaztuk a modelleket, amin hiperparaméteroptimalizációval még javíthatnánk. Nagyobb figyelmet fordíthatunk a jövőben a csoport módszerre, ahol akár jó döntési fák segítségével többszörösen vághatjuk az adatot, mélyebb elemzésekhez jutva ezáltal. Végül pedig az anomáliadetektálás terén nem egy automatizált megoldást adtunk meg, további munka eredményeként egy kevésbé manuális módszerhez jutnánk, ami általánosabb körben is felhasználhatóvá tenné.

6. Köszönetnyilvánítás

Először is szeretném megköszönni a témavezetőmnek, Dr. Kovács Edith Alicenak a rendkívül lelkismeretes munkáját, amely nélkül nem íródhatott volna meg ez a dolgozat. Betekintést nyerhettem egy nagyon érdekes témaiba, ezeknek az ismereteknek kétség kívül rengeteg hasznát veszem még a jövőben. Bármilyen kérdés, elakadás esetén rövid időn belül elérhető volt és a felmerülő problémákat rendszerint gyorsan sikerült orvosolni, minden egybevetve élveztem a közel egy éves közös munkát.

Köszönettel tartozom még Dr. Szakadát Istvánnak is, aki eredetileg vezetett a társadalom- és gazdaságföldrajzi adatok elemzésének világába. Mindemellett természetesen az általa gondozott adathalmaz nélkül nem születhettek volna meg a dolgozat gyakorlati elemzései.

Hivatkozások

- [CG16] Tianqi Chen and Carlos Guestrin. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 785–794, New York, NY, USA, 2016. ACM.

- [CL20] Ian Covert and Su-In Lee. Improving kernelshap: Practical shapley value estimation via linear regression. *arXiv preprint arXiv:2012.01536*, 2020.
- [FKM19] Takanori Fujiwara, Oh-Hyun Kwon, and Kwan-Liu Ma. Supporting analysis of dimensionality reduction results with contrastive learning. *IEEE transactions on visualization and computer graphics*, 26(1):45–55, 2019.
- [JC16] Ian T Jolliffe and Jorge Cadima. Principal component analysis: a review and recent developments. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 374(2065):20150202, 2016.
- [LL17] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017.
- [M⁺67] James MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA, 1967.
- [MJE21] Wilson E Marcílio-Jr and Danilo M Eler. Explaining dimensionality reduction results using shapley values. *Expert Systems with Applications*, 178:115020, 2021.
- [MOS20] Masayoshi Mase, Art B Owen, and Benjamin Seiler. Explaining black box decisions by shapley cohort refinement. *arXiv preprint arXiv:1911.00467*, 2020.
- [PVG⁺11] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot,

- and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- [Rou87] Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.
- [RPB20] Raquel Rodríguez-Pérez and Jürgen Bajorath. Interpretation of machine learning models using shapley values: application to compound potency and multi-target activity predictions. *Journal of computer-aided molecular design*, 34(10):1013–1026, 2020.
- [Sha53] L. S. Shapley. *17. A Value for n-Person Games*, pages 307–318. Princeton University Press, 1953.
- [SK10] Erik Strumbelj and Igor Kononenko. An efficient explanation of individual classifications using game theory. *The Journal of Machine Learning Research*, 11:1–18, 2010.
- [ŠK14] Erik Štrumbelj and Igor Kononenko. Explaining prediction models and individual predictions with feature contributions. *Knowledge and information systems*, 41(3):647–665, 2014.
- [SN20] Mukund Sundararajan and Amir Najmi. The many shapley values for model explanation. In *International Conference on Machine Learning*, pages 9269–9278. PMLR, 2020.