

A magyar programozók béregyenletének becslése és bérkülönbségeinek elemzése dekompozíciós modellezéssel a Stack Overflow kérdőíve alapján

Tudományos Diákköri dolgozat

Vastag Emese
Gazdaságinformatikus Bsc.
2023

Témavezető: Kovács László

Tartalomjegyzék

1. BEVEZETÉS	4
2. A MAGYARORSZÁGI IT SZEKTOR JELLEMZÉSE 2011 ÉS 2022 KÖZÖTT	5
2.1. A szoftverfejlesztés kereslete hazánkban	5
2.2. A szoftverfejlesztők bérezését meghatározó tényezők	6
3. AZ ELEMZÉSBEN FELHASZNÁLT STATISZTIKAI – ÖKONOMETRIAI MÓDSZERTANOK BEMUTATÁSA.....	14
3.1. A nem-metrikus többdimenziós skálázás módszertana	14
3.2. A Blinder – Oaxaca dekompozíció.....	17
4. A KUTATÁSBAN FELHASZNÁLT ADATOK, ADATFORRÁS ÉS ELEMZÉSI KERETRENDSZER BEMUTATÁSA.....	19
4.1. A kutatásban felhasznált adatbázis bemutatása	20
4.2. Az elemzéshez és vizualizációhoz használt keretrendszer és csomagok bemutatása 20	
5. AZ ADATOK ELŐKÉSZÍTÉSE, ADATTISZTÍTÁS	21
5.1. A becsült eredményváltozó megtisztítása.....	21
5.2. Döntési fa modell az eredményváltozóra	23
5.3. Magyarázóváltozók átalakítása.....	25
6. KIEGÉSZÍTŐ MEGJEGYZÉSEK AZ ADATBÁZISRÓL ÉS A MINTAVÉTEL MÓDJÁRÓL	26
6.1. Adatbázis	26
6.2. Mintavétel módja	27
7. A KIINDULÓ ADATBÁZIS LEÍRÓ STATISZTIKÁJA	28
8. AZ ELEMZÉS EREDMÉNYEI ÉS AZOK ÉRTÉKELÉSE	31
8.1. Modellszelekció.....	31
8.2. Nem – metrikus többdimenziós skálázás.....	39
8.3. Specifikáció, nemlineáris tagok bevonása	41
8.4. Blinder – Oaxaca dekompozíció.....	43
9. AZ ELEMZÉSEK ALAPJÁN TETT VÉGSŐ MEGÁLLAPÍTÁSOK	54
9.1. Munkatapasztalat, betöltött pozíció	54
9.2. Iskolai végzettség, technológiai ismeretek	55
9.3. Demográfiai tényezők.....	55
10. TOVÁBBI FELMERÜLŐ KUTATÁSI KÉRDÉSEK.....	56

10.1.	A senior informatikai szakértelem fejlesztése és megtartása egy munkaerőhiány sújtotta piacon.....	56
10.2.	A magyar informatikai munkaerő szerepköre és helyzete a KKV és nagyvállalati szektorban	57
10.3.	Az informatikai képzést biztosító magyar felsőoktatási intézmények jövőképe ...	57
11.	KÖSZÖNETNYILVÁNÍTÁS	59
12.	IRODALOMJEGYZÉK	60
13.	ÁBRAJEGYZÉK	67
14.	TÁBLÁZATOK JEGYZÉKE	68
15.	MELLÉKLETEK	69
15.1.	Módszertani mellékletek	69
15.2.	Mellékelt ábrák	84

1. BEVEZETÉS

Az informatika ma már a mindennapjaink részét képezi és az utóbbi években rendkívül felgyorsult az iparágban az innovációk megjelenése. A terület ezért egyértelműen a K + F szektorba sorolható és egy ország informatikai fejlettségét többek között az ebben az ágazatban dolgozók bérezése is jól jellemzi. Éppen ezért dolgozatomban azokat a tényezőket vizsgálom, amelyek hozzájárulnak egy magyar szoftverfejlesztő fizetéséhez. Ehhez az évente megjelenő kérdőív, a *Stack Overflow Annual Developer Survey* által összegyűjtött adatokat fogom használni a magyar válaszadókra nézve. Dolgozatomban programozónak vagy informatikusnak tekintek minden kitöltőt, akik a Stack Overflow felhasználói közösségének tagjai. Ezek a szakemberek zömében szoftver – illetve webfejlesztéssel foglalkozó vagy rendszerüzemeltető szakemberek.

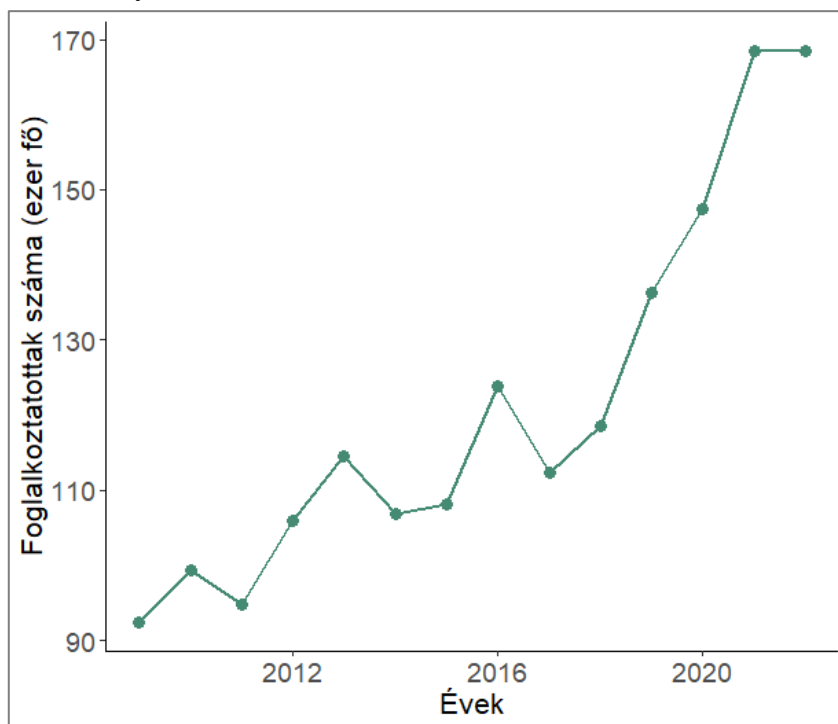
A bérezés vizsgálata nemcsak a magyar informatikai szektor fejlettségére ad válaszokat, hanem a leendő munkavállalók és munkáltatók számára is hasznos lehet. Munkavállalói oldalról megközelítve az itt feltárt információk a jelenleg és potenciálisan informatikával foglalkozók számára is segítséget tudnak nyújtani abban, hogy a szektor mely részterületein érdemes ma Magyarországon elhelyezkedni és ahhoz milyen kvalitások szükségesek. A munkáltatók szemszögéből nézve a kutatás eredményei toborzási szempontból is alkalmazhatók, mert rávilágítanak arra, hogy milyen elvárásokat szükséges felállítaniuk leendő munkavállalóikkal szemben, illetve, hogy a bérezés kialakításánál a munkatársak mely tulajdonságait szükséges figyelembe venni fő szempontként.

A dolgozatban tárgyalt kérdéskör továbbá az oktatási intézményekhez is szorosan kapcsolódik, amelyek a képzési paletták és lehetőségek frissen és változatosan tartásával képesek leendő hallgatóikat megnyerni és a munkaerőpiacot kiszolgálni. A diploma szükségességének vizsgálata a hazai informatika szektorban az újonnan megjelenő népszerű tanulási platformok (pl.: Udemy, DataCamp stb.) térhódításával rendkívül időszerűvé vált. Kutatásom eredményei a felsőoktatási intézmények számára is hasznosak lehetnek egy a munkaerőpiaci, valamint a hallgatói igényekhez is igazodó, korszerű képzési portfólió összeállításában.

2. A MAGYARORSZÁGI IT SEKTOR JELLEMZÉSE 2011 ÉS 2022 KÖZÖTT

2.1. A szoftverfejlesztés kereslete hazánkban

Az informatika, mint iparág – ahogy azt már korábban is említettem – rendkívül gyorsan fejlődik, az újszerű technológiák elterjedése pedig a keresletet is serkenti a munkaerőpiacon a szoftverfejlesztők irányában.



1. ábra: Az információ, kommunikáció szektorban foglalkoztatottak száma (ezer fő).
Forrás: KSH (2023a). Saját készítés.

Hazánkban a Tevékenységek Egységes Ágazati Osztályozási Rendszere (a továbbiakban TEÁOR '08) a szoftvertechnológiai tevékenységeket az információ és kommunikáció nemzetgazdasági szektorba sorolja, ezen belül is a 62-es számú, információ – technológia szolgáltatások közé. Az információ, kommunikáció szektorban kb. 168,5 ezer főt foglalkoztattak 2022 – ben, ez a 2011 - es adathoz majdnem 78%-kal több (KSH, 2023a). Mindemellett fontos megjegyezni azt is, hogy ebbe az ágazatba a telekommunikációs vállalatok is beletartoznak, illetve vélhetően más nemzetgazdasági területeken is dolgoznak szoftverfejlesztők, például a pénzügyi, biztosítási, ipari tevékenységeket magába ölelő szektorokban, így a szám inkább csak a nagyságrend szemléltetésére szolgál. A betöltött állásokkal párhuzamban a betöltetlenek is folyamatosan növekedtek. Míg az információ, kommunikáció szegmensben 2011 -ben 1257 betöltetlen állás volt, addig 2022 -ben ugyanez több mint négyezerre, azaz több, mint háromszorosára nőtt (KSH, 2023b). Mindezek alapján elmondható, hogy az informatikai területen korántsem történt a munkaerőpiacon telítődés, ez

szintén alátámasztja azt, hogy a kereslet az informatikusokra (és ezen belül szoftverfejlesztőkre) egyre csak növekszik hazánkban.

Az ugrásszerű növekedés egyik legkézenfekvőbb magyarázata lehet, hogy Magyarországon a kétezres évektől kezdve egyre több multinacionális cég leányvállalata telepedett le, rengeteg munkahelyet teremtve. Ez a tendencia főként az ipari (és azon belül az autóipari) külföldi nagyvállalatokra jellemző (Lux, 2017). Emellett megjelentek az SSC-k, azaz Shared Service Centerek, melyek az egyszerű kiszervezéssel ellentétben elkülöníthető, elszámoltatható egységek és főként adminisztratív, informatikai, pénzügyi szolgáltatásokat nyújtanak a vállalat többi (akár más országban található) területeinek, valamint vállalaton kívüli, külső ügyfeleknek is (Janssen és Joha, 2006). A KSH (2023c) szerint 2019 -ben 26, míg egy évvel később már 39 szolgáltató központ működött Magyarországon az információ, kommunikáció szektorban, ezzel utolérve a feldolgozóipart és megelőzve a kereskedelem, gépjárműjavítást. 2022 – ben ugyanezen területen már 51 darab SSC működött, amelyet egyedül a szakmai, műszaki, tudományos tevékenységeket magába foglaló nemzetgazdasági ág előzött meg. Az IT – iparágat ezek az események közvetett módon szintén érinthették, hiszen egy olyan nemzetközi vállalat esetében, amelynek több országban is működnek központjai, mindenképp fontos egy megfelelő szerverpark, felhős architektúra és a különböző, belsőleg használt szoftverek fejlesztése. Mindezen feladatok ellátásához pedig szükség van a szakképzett informatikai munkaerőre. A Hays Salary Guide (2023) szerint például 2023 – ra az IT szektorban növekedett az igény arra, hogy a leendő munkavállalók legalább egy cloud környezetet ismerjenek, különösképp a szerződéses, külsős szakértők esetében.

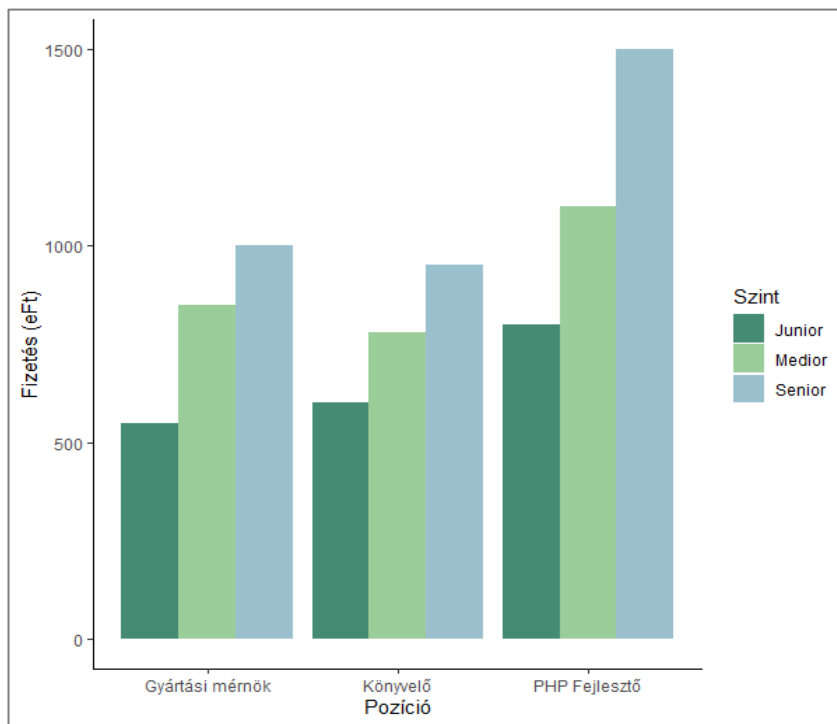
2.2. A szoftverfejlesztők bérezését meghatározó tényezők

A következő elemzési aspektusokat a Stack Overflow kérdőívének logikáját alapul véve határoztam meg, mivel ezek a jellemzők fogják képezni elemzésem tárgyát is.

2.2.1. Munkatapasztalat

A 2.1 fejezetben bemutatott, informatikai szektorban megvalósuló munkaerőhiány miatt a munkáltatók jellemzően magasabb fizetéssel honorálják az ilyen jellegű állásokat. A Hays Hungary Salary Guide (2023) szerint Magyarországon 2022-ben a szoftverfejlesztésben már a junior (0-3 év munkatapasztalattal rendelkező) szinten is havi 800 – 850 ezer Ft között mozgott az átlagos bruttó bér. Az adat nem tartalmazza a béren kívüli juttatásokat és prémiumokat, valamint csak a teljes munkaidőben dolgozó szoftverfejlesztőket vették figyelembe. Egy medior (3-5 év munkatapasztalatot megszerzett) programozó esetében pedig ugyanez a szám

már az egymillió forintot is meghaladja. Ahhoz kétség sem férhet, hogy természetes a tendencia, hogy a bérek az idő múlásával emelkednek az infláció és a tapasztalat hatásai miatt. Ezen a ponton inkább azt a volument emelném ki, amennyivel az informatikusok bére az eltelt évek függvényében növekszik. Ehhez szintén a Hays Hungary Salary Guide adatait használom fel.



2. ábra: Egyéb, elsősorban nem informatikai tudást igénylő állások bruttó átlagbérei (juttatások és bónusz nélkül) 2023-ban tapasztalat szerint (eFt). Forrás: Hays Hungary Salary Guide (2023). Saját készítés.

Mivel az általam feldolgozott Stack Overflow adatbázisban a webfejlesztők vannak nagyobb többségben (melyet a 7. fejezetben részletesebben kifejték), egy PHP fejlesztő karrierszint szerinti bérnövekedését viszonyítottam más, elsősorban nem informatikai végzettséget igénylő állásokhoz. A 2. ábrán megjelenő junior – medior – senior szint évenkénti tagolása a következő: a junior szint 0-3, a medior 3-5, míg a senior 5 évnél több munkatapasztalatot jelent. Az ábrán is látható, hogy egy PHP webfejlesztő bruttó átlagfizetése 5 évnél több munkatapasztalat után jelentősen megugrik, arányaiban a junior szint körülbelül 200%-ára változik. A junior – medior szint közti különbséget megnézve 3 év munkatapasztalat után a bruttó bér átlagosan 37,5%-kal nő bruttó 800 ezer forintról 1,1 millió forintra. Ezt a változást a gyártási mérnök pozíció előzi meg, átlagosan másfélszeres növekedési volumennel.

Kutatásomban többek között azt a hipotézisem szeretném igazolni, hogy bár a junior – medior – senior karrierút szerint nézve a fizetések drasztikusan nőnek, senior szint felett a programozási – fejlesztési tapasztalatnak kevesebb hozzáadott értéke van a bérhez. Úgy gondolom, az

informatika szektorban lévő gyors változások következtében a fiatalabb generáció tudása sok esetben naprakészebb tud lenni, míg az idősebbek inkább projektmenedzsment - vezetési feladatokat és kevesebb operatív fejlesztési munkát látnak el. Ebből adódóan a seniorabb korosztály magasabb bérezésében már nagyobb szerepe lehet magának a vezetői beosztásnak és a projektek eredménytermékekért való felelősségvállalásnak, mint a programozói tudásnak. Elemzésem során ezen tendencia kimutatására szeretnék valamilyen mintázatot találni az adatokban.

2.2.2. Ismert technológiák, iskolai végzettség

2.2.2.1. Ismert technológiák

A Hays Hungary Salary Guide (2023) a tapasztalat mellett a használt technológiák és programozási nyelvek szerint is megkülönbözteti a különböző szoftverfejlesztői állásokat és látható, hogy az átlagbérek eszerint is szóródnak, de nagyjából egységesek.

Emellett a kiadvány azt is megállapítja, hogy az IT szektorban 2022-ben is folytatódott a növekedés, Magyarország kedvező választás a befektetők számára, mert a munkaerő továbbra is megfizethető és a kompetencia színvonala is megfelel a nemzetközi elvárásoknak. Azt a Hays Salary Guide is kijelentette, ami a KSH adataiból is szembetűnő, hogy a piacon alapvetően még mindig a túlkereslet jellemző, amely a tavalyi évben csak tovább fokozódott. Emellett ugyanakkor 2022-ben trenddé vált a digitális elvándorlás, amelynek oka valószínűleg a kecsegtető nemzetközi bérekből adódó jobb megélhetés reménye, valamint a forint nagymértékű gyengülése (Hays Hungary Salary Guide, 2023). Mivel a munkaerőhiány miatt az IT szektorban dolgozók a bérek tekintetében jobb alkupozícióban vannak, a fizetések is átlagosan nézve követték az inflációt. Ez már korábban a 2021-es évre is hasonlóan jellemző volt, akkor az IT terület minden szegmensében legalább 10%-kal megemelkedtek a bérek az előző évhez képest, de a szoftverfejlesztés részterületen ez a növekedési volumen a 20%-ot is meghaladta, a Cloud DevOps és Cyber Security területekkel versenyezve (Hays Hungary Salary Guide, 2022). 2022 – ben továbbá az idej kiadvány szerint legkeresettebb tudás a backend fejlesztésben a Java és C#, míg a frontendben az Angular és React. Ebből az okból kifolyólag a későbbiekben ezen eszközök ismeretének bérre gyakorolt hatásait is megvizsgáltam külön – külön.

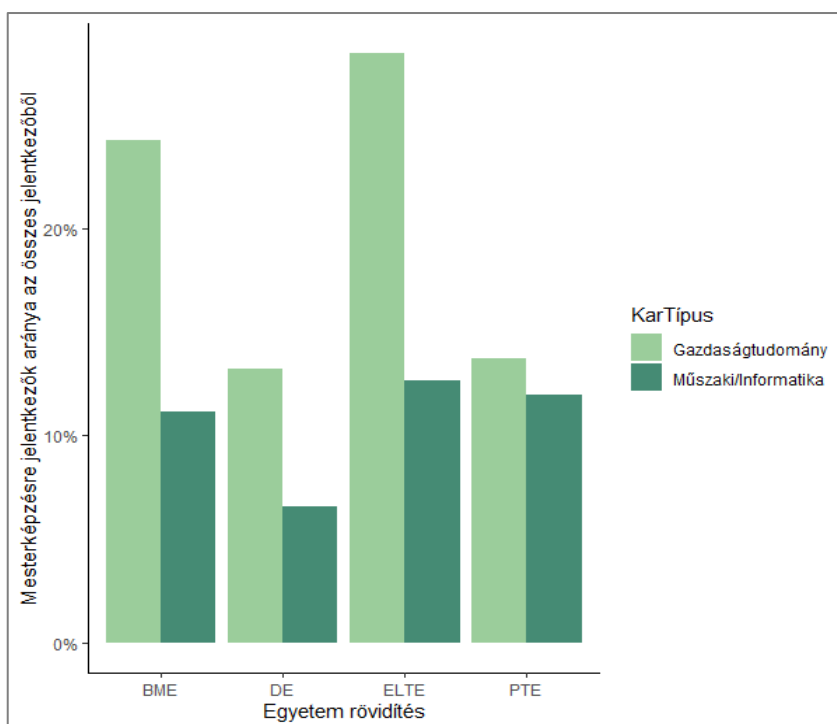
Összességében magabiztosan megállapítható a tény, hogy bár önmagában az informatikusokra a kereslet továbbra is gyarapszik, nem elhanyagolható, hogy pontosan milyen pozíciót töltenek be, ugyanis bérezésük ez alapján akár jelentősen is eltérhet. Ahogy azt korábban említettem, a

Hays Salary Guide pozíciók és technológiák szerint is elkülöníti a fejlesztőket, a bérekben pedig látható szóródás a részcsoportok között. Nagyon látványos például a tény, hogy a Data Scientistek és Machine Learning specialisták átlagos bruttó bére már junior szinten is megütheti az egymillió forintot. Maga az iparág Magyarországon még igencsak gyerekcipőben jár, ezért a téma szakértői kifejezetten előnyös alkupozícióban vannak a bérezésüket illetőleg (Mayer – Schönberger és Cukier, 2014). Elemzésemben az egyes technológiák fontossága mellett azt is vizsgálom, hogy vajon a fentebb említett pozíciók, mint részcsoportok szerinti szóródás szignifikáns – e és hogy van - e olyan részcsoport, amely kimutathatóan eltér a többitől (pozitív vagy negatív irányban).

2.2.2.2. Iskolai végzettség

Az ismert technológiákkal szoros viszonyban áll az iskolai végzettség, illetve különböző képzettségek kérdésköre, amely már egészen a közoktatási szintig visszanyúlik. A digitális kompetenciák fontosságáról már 2016 – ban is értekezett Árgilán és Kelemen (2016), akik kutatása szerint akkor a magyarországi technológiaoktatás inkább az irodai eszközök használatára készítette fel a tanulókat és kevésbé a kreatív, algoritmikus gondolkodást igénylő problémák megoldására. Emellett megállapították, hogy a középiskolában tanult informatika tananyag az ilyen irányban továbbtanuló érettségizőket nem készítik fel megfelelően az egyetemi elvárásokra. A 2020 -as Nemzeti Alaptantervben már történt frissítés a 2016 -os állapothoz képest, például a programozás már középszinten is helyet kap (5/2020. (I. 31.) Korm. rendelet).

A közoktatáson túllépve az egyetemi informatikatanítás kérdéskörének is hatalmas szerepe van az IT szektor kiszolgálásában. Összességében az informatikai alapszakok országszerte népszerűek még önköltséges formában is. A 2023 szeptemberi képzések általános felvételi eljárásában az ELTE Informatikai Karára összesen 3066 ember jelentkezett, míg a Budapesti Műszaki és Gazdaságtudományi Egyetem Villamosmérnöki és Informatikai Karára 3427. A népszerűbb vidéki egyetemek közül a Debreceni Egyetem Informatikai Karának 1690, míg a Pécsi Tudományegyetem Műszaki és Informatikai Karának 2586 jelentkezője van (Felvi.hu, 2023).



3. ábra: Mesterszakra jelentkezők arányának összehasonlítása a gazdasági és informatikai területek közt 2 - 2 népszerűbb fővárosi és vidéki egyetemen a 2023 szeptemberben induló képzésekre. Forrás: Felvi.hu (2023). Saját készítés.

A mesterszakokra történő jelentkezések száma viszont elmarad a népszerűnek számító gazdaságtudományi területhez képest, amelyet a 3. ábrán jelenítettem meg a korábban említett 2 – 2 közismert fővárosi és vidéki egyetem esetén. Az ELTE Informatikai Karának mesterképzéseire például összesen 388 fő jelentkezett, míg a Gazdaságtudományi Karra 2022 (Felvi.hu, 2023). A tendencia a többi, korábban felsorolt egyetem esetén is hasonló. Fontos megjegyezni, hogy az egyes egyetemek képzési palettája némileg eltér, például a Debreceni Egyetemen különválnak a Műszaki és Informatikai Kar, míg a Budapesti Műszaki és Gazdaságtudományi Egyetemen az informatikai szakok mellett mérnöki területet lefedő képzések is vannak a Villamosmérnöki és Informatikai Karon. Ebből kifolyólag vélhetően a jelentkezések számát befolyásolja még az adott karon indítani tervezett szakok, illetve az azokra meghirdetett helyek száma, ugyanakkor az adatok alapján mindenképp felmerül a kérdés, hogy a magyar hallgatók milyen mértékben tartják hasznosnak az informatikával foglalkozó mesterszakokat a teljes állásban történő munkavégzéshez képest.

A Magyar Állam reakciója a koronavírus járvány után emelkedett szinten megjelenő IT munkaerőhiányra az Újratervezés Program elindítása volt, melynek keretein belül a 18. életévüket betöltő magyar állampolgárok állami támogatással jelentkezhetnek intenzív IT képzésekre. A programot az állam időszakosan hirdeti meg, 2023 februárjában például adatbázis üzemeltető specialista, illetve junior automata tesztelő kurzusokra volt lehetőség

jelentkezni. A Kormány célja a kezdeményezéssel olyan informatikai szakemberek gyors ki – illetve átképzése, akik képesek feltölteni a munkaerőpiaci űrt az IT szektorban. A kurzusok lebonyolításával az állam több független, technológiai képzéseket biztosító szervezetet bíz meg, amelyek az államilag támogatott oktatásokon kívül is számos önköltséges tanulási lehetőséget kínálnak (Nemzeti Szakképzési és Felnőttképzési Hivatal, 2023).

Az egyetemek hiányosságairól a nemzetközi szakirodalomban már Stroustrup (2010) is értekezett, aki megállapította, hogy egyre inkább fellelhető egy „rés” aközött, amit az informatikát tanuló hallgatók az egyetemen tanulnak és amit a munkaerőpiac elvárna a leendő informatikusoktól. Sok esetben a probléma gyökere az, hogy maga az egyetemi képzés nem elég gyakorlatias, a hallgatók kevés valós problémával találkoznak vagy az egyetemi oktatók nem feltétlenül töltnek be az egyetemi oktatáson kívül szoftverfejlesztői pozíciót. A rengeteg elméleti alapotárgy elveszi a kellő hangsúlyt és időt a gyakorlatorientált oktatástól és arra kényszeríti a tanulókat, hogy a programozást is, mint elméleti tárgyat sajátítsák el. Ennek eredményeként lehet, hogy jó eredményekkel zárják az egyetemi éveket, de önállóan csak nagyon kevés kódot írtak meg.

A diploma fontosságát dolgozatomban két szempontból is vizsgálom. Egyfelől dekompozíciós modellezéssel a bérezés oldaláról mutatom ki, hogy összességében mennyivel több a mesterszakot végzett informatikusok átlagbére az alapszakokkal rendelkezőknél, illetve, hogy ezen két részcsoporthoz belül milyen összetétel – és részhatásbeli különbségek vannak. A másik aspektus, amelyet körbejárok, tágabb értelemben véve a diploma fontossága, ezt is hasonlóképpen dekompozíciós eljárással teszem meg.

2.2.3. Demográfiai tényezők

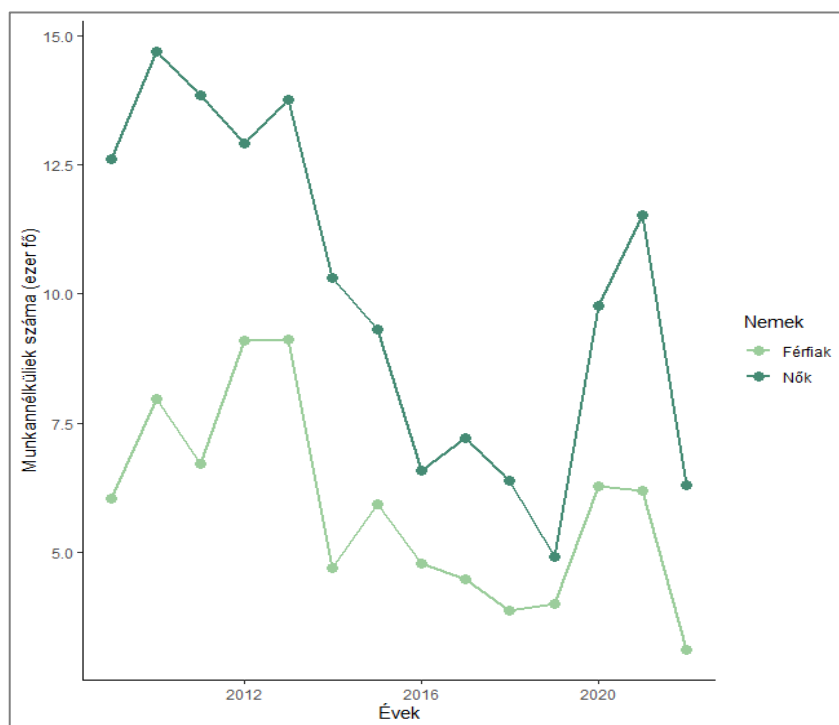
Fontos faktor még nemcsak a szoftverfejlesztők, hanem tágabb értelemben véve a munkavállalók béreiben a nemek közötti differencia. Az OECD ennek mérésére alkotta meg a „gender wage gap” mutatót, amely a női és férfi munkavállalók béreinek mediánjainak különbségét hasonlítja a férfiak béreinek mediánjához (OECD, 2022). A felmérésbe nemcsak a teljes állásban dolgozók, hanem a részmunkaidős és egyéni vállalkozók adatai is bekerültek. Minél kisebb ez a mutatószám, várhatóan annál kisebb a bérkülönbség a nemek között, ugyanakkor természetesen ezt több szempont is befolyásolja. Magyarország ebben a felmérésben 2020-ban egyébként elmaradt az OECD átlagtól, amely 11,6% volt, míg hazánk eredménye 12,4% lett. Egy másik jellegű, de hasonló tartalommal bíró mutató a World Economic Forum által számított Global Gender Gap Index, amely négy aspektus szerint méri a

nemek közti bérkülönbségeket: gazdasági aktivitás és lehetőségek, iskolai végzettség, egészségügyi állapot és politikai aktivitás. Ez az index éppen fordítva közelíti meg a bérezések közti differenciákat: azt mutatja meg, hogy hány %-ban sikerült már bezárni a rést a férfiak és nők között. Globálisan ez a szám 2022 – ben 68,1% volt. A világ országait rangsorba állítva hazánk itt globális tekintetben nézve elmaradottabb, a 88-ik helyet szerezte meg 2022-ben 69,9 % -kal, viszont 2021-hez képest 3 helyet javított. (World Economic Forum, 2022).

A koronavírus járvány hatásai szintén nem elhanyagolhatóak a témában. Összességében a nők munkaerőpiaci helyzetére a World Economic Forum (2022) szerint a COVID járvány negatív hatást gyakorolt. A tanulmány azt állítja, hogy a nők nagyobb arányban dolgoznak olyan állásokban, amelyek az oktatással, egészségüggyel kapcsolatosak vagy a szolgáltatás és vendéglátás szektorba tartoznak, ezeket pedig a lezárások érzékenyebben érintették, hiszen jellegükből adódóan kevésbé tudják az otthonról végezhető munkavégzést kivitelezhetővé tenni. Ezzel a ténnyel a magyar adatok is összhangban vannak. Hazai viszonylatban nézve a KSH (2023a) szerint a következő TEÁOR '08 szerinti nemzetgazdasági szektorokban volt 2009 – től kezdve minden évben nagyobb, mint 60% a nők aránya a munkavállalókon belül: oktatás, humán – egészségügyi, szociális ellátás és kiskereskedelem. A turisztikai szektorban is többségben vannak a nők, a 2009 – 2022-es évek átlaga 57,6%. A 2021 – ben, közvetlenül a COVID időszaka után kiadott World Economic Forum (2021) tanulmányban globálisan nézve 2020 április – májusában az előző év ugyanezen időszakához viszonyítva minden szektort tekintve csökkent a nők foglalkoztatása, ugyanakkor a leglátványosabb recesszió (8-10%-os csökkenés) a turizmus és szolgáltatások szektoraiban következett be, ezek pedig éppen azok a területek, ahol arányaiban több női munkavállaló van.

A legkevésbé érintett iparág az információs technológia és IT szolgáltatások voltak, ahol pedig alapvetően a férfi munkaerő van többségben. Bár a nők foglalkoztatottsága itt is visszaesett fél százalékkal az előző évhez képest 2020 tavaszán, nyárra már éppen, hogy növekedett az arány 2019 nyarához képest (World Economic Forum, 2021). Ennek egyik oka lehet az, hogy vélhetően rengeteg munkahelyen vezették be a vírus hatására a home office munkarendet, viszont számos vállalat nem volt technológiailag érett, hogy ilyen megoldásokat a hirtelen kialakult helyzetben hatékonyan implementáljon. Sok helyen még stabil VPN hálózatok sem voltak megfelelően kiépülve és a távmunka beköszöntével ezek a cégek egyre nagyobb kibebiztonsági kockázatnak lettek kitéve, illetve a mindennapi munka is akadozott a kis szerverkapacitás miatt (Pranggono és Arabo, 2021). Valószínűsíthető, hogy ezek a problémák piaci igényeket is generálhattak az informatikai szektor felé, így az itt tevékenykedő

vállalatoknak megnőtt az igénye az új munkaerőre. Mindemellett az informatikai tevékenységet folytató cégek számára értelemszerűen kevesebb bonyodalmat okoz a home office munkavégzés, így elképzelhető, hogy a koronavírus járvány következtében állásukat elvesztő nők közül sokan karriert váltva próbáltak a jellemzően több távmunka lehetőséget kínáló üzleti – informatikai szegmensekben elhelyezkedni, például adminisztratív vagy egyéb támogató tevékenységekben (marketing, HR, IT support, stb.) vagy adott esetben akár éppen a fejlesztésben. A home office munkának ezen felül feltételezésem szerint más pozitív aspektusai is vannak a nőkre nézve: a szülési szabadság után otthonról dolgozva könnyebben tudhatnak visszaállni a munkába, így a karrierváltásnak akár ilyen szempontból is lehetnek pozitív hozadékai. Számos szerző, köztük Billioniere és munkatársai (2020) is kiemelik, hogy a STEM – szektor (Science, Technology, Engineering and Mathematics) egyes iparágainak hatalmas munkaerőhiányára egy lehetséges válasz lehetne, ha a vállalatok jobban támogatnák a (munkába visszaálló) nőket az ilyen területeken történő elhelyezkedésben. Ezen komplex kérdéskör vizsgálata további kutatási kérdéseket vet fel, amelyek jelenleg nem állnak dolgozatom fókuszában.



4. ábra: Munkanélküliek száma nemenkénti bontásban, ha az adott személy előző munkahelye szálláshely – szolgáltatás, vendéglátás szektorba esett, 2009 és 2022 között (ezer fő). Forrás: KSH (2023d). Saját készítés.

A fentebb ismertetett trendeket egyébként a magyar változások is követik, amelyeket a 4. ábrán is szemléltettem. A TEÁOR '08 besorolás szerint 2019 - ben nagyjából 4900 olyan nő volt munkanélküli, akinek előző munkahelyének nemzetgazdasági ága a szálláshely – szolgáltatás,

vendéglátás szektor volt, ez a szám pedig 2020 – ra 9800-ra, tehát a duplájára nőtt. Az érték 2021 – ben tetőzött 11500 fővel, majd 2022 – re konszolidálódott 6300 emberre. A férfiakat megnézve, ugyanezen gazdasági szektorban 2019 -ben 4000, míg 2020 -ban 6300 férfi volt munkanélküli, amely kb. 57% - os növekedés. 2021 – ben a munkátlan férfiak száma a területen 6200 – ra, majd 2022 – ben 3100 főre csökkent. A KSH a 8 éven belüli munkatapasztalattal rendelkező embereket vette figyelembe az összesítésben (KSH, 2023d). Ugyanakkor a foglalkoztatottak száma az információ, kommunikáció szegmensben mindkét nem esetében nőtt a pandémia első két évében, ez pedig szintén egybecseng az előző bekezdésben tett megállapításokkal (KSH, 2023a). Ezen információk alapján kutatásomban arra a kérdésre fogom keresni a választ, hogy a magyar fejlesztői piacon van - e szignifikáns eltérés a bérezést tekintve a nemek között, vagy ezen a területen nem értelmezhető a férfi – női bérdifferentia vizsgálata. A kérdés relevanciája, hogy ezáltal jobban megérthetjük, hogy a programozással foglalkozó szakemberek körében a nemek közti bérezési különbségek hogyan viszonyulnak az itt felvázolt hazai munkaerőpiaci trendekhez.

3. AZ ELEMZÉSBEN FELHASZNÁLT STATISZTIKAI – ÖKONOMETRIAI MÓDSZERTANOK BEMUTATÁSA

Elemzésem során az alapvető ökonometriai és adatbányászati megközelítésen alapuló módszertanokat alkalmaztam a modellépítési folyamat során, amelyeket részletesen a 15.1 fejezetben a módszertani mellékletben mutatok be. Ebben a fejezetben azon két további technikát emelem ki, amelyek elemzésem fókuszában álltak és az eredmények jelentős részét szolgáltatják.

3.1. A nem-metrikus többdimenziós skálázás módszertana

3.1.1. A multikollinearitás torzító hatása

A non - metric multidimensional scaling, azaz nem – metrikus (vagy ordinális) többdimenziós skálázás a főkomponenselemzéshez hasonlóan dimenziócsökkentés egyik eszköze. A dimenziócsökkentés lényege, hogy az egymással szoros korrelációban álló változókat egy új, látens változóba sűrítjük. Ezzel csökkenteni tudjuk a multikollinearitást a modellben. A multikollinearitás jelenségének következménye, hogy a magyarázóváltozók lineáris kapcsolatban állnak egymással. Természetesen az a tény, hogy a modellbéli változók magyarázzák egymást, még nem jelent káros mértékű multikollinearitást (például a kutatásban használt adatbázisban vélhetően korrelál egymással az életkor és a szakmai tapasztalat). Amikor viszont két változó közt erős korreláció áll fent, akkor hatásuk összemosódik és csak egyben

becsülhetők jól, külön - külön nem. A multikollinearitást a VIF (variancia - infláló) mutatóval tudjuk jellemezni (Helms és Mansfield, 1982).

$$VIF_j = \frac{1}{1-R_j^2} \quad (1.)$$

A mutató jelentése az (1.) egyenlet szerint adott j változóra nézve, hogy annak hányszorosára nőtt a standard hibájának négyzete, azaz becslési varianciája csak a multikollinearitás miatt. Amennyiben ez a szám 2 és 5 közé esik, gyenge multikollinearitásról beszélünk, ha viszont ötnél nagyobb, akkor az már jelentősen torzítja az adott változó parciális t – próbájának standard hibáját, amely a próbafüggvény nevezőjében található, így az eredményül adott p – értéket irreálisan lecsökkenti. Ennek következményeképp hibásan szignifikánsnak ítéltünk olyan változókat, amelyeket a modellben más változók szinte teljesen megmagyaráznak, viszont kihagyni sem tudjuk őket, mert akkor elvesztenénk a többi, velük korreláló változó információtartalmát is (Acharya, Lavery, Sivo és Xu, 2019).

Amennyiben jelentéstartalomban hasonló változóink vannak, amelyek káros multikollinearitásban állnak egymással, érdemes dimenziócsökkentést végrehajtani, ezeket a változókat egy közös látens változóba sűrítve. Ezt megtehetjük például főkomponenselemzéssel vagy a nem – metrikus többdimenziós skálázással is. Én elemzésemben az utóbbit alkalmaztam, ugyanis a kiinduló magyarázóváltozók mindegyike kategorikus ismérv volt, a főkomponenselemzés pedig numerikus adatokon alkalmazható, mivel az ahhoz használt Pearson - féle korrelációmátrix sajátérték - sajátvektor felbontásából indul ki (Henrion, 1994). Általánosabban a többdimenziós skálázás módszertanát Torgerson (1952) dolgozta ki, valamint Shepard (1962), Kruskal (1964) és Guttman (1968) is továbbfejlesztették és kiegészítették. A társadalomtudományokon túl számos más alkalmazási területe van például a biológiában és egyéb természettudományokban. A nem – metrikus többdimenziós skálázást Kruskal (1964) fejlesztette ki. A módszertan azért nem – metrikus, mert ordinális mérési skálán mérhető adatok egymástól mért távolságával dolgozik. Ezáltal nem feltétel, hogy az adatunk numerikus legyen (a legalább intervallum mérési szintű numerikus adatokon végzett skálázást hívjuk klasszikus metrikus skálázásnak).

Általánosságban véve a többdimenziós skálázás alapvető célja az egymással korreláló változók korrelálatlan látens változó(k)ba csökkentése úgy, hogy az általunk választott k dimenziószám mellett megfigyelések távolsága a lehető legkevésbé térjen el az eredeti p dimenzióban mért távolságoktól. A főkomponens – elemzéshez képest a módszernek hátránya, hogy $k - t$ nekünk kell előre megadnunk, tehát szükséges az előfeltételezés, változóink között hány látens

változóba tömöríthetők az egymással korreláló változók. Az eredeti p dimenziójú adatokból D távolságmátrixot állíthatjuk elő, a megoldandó feladat pedig D^* távolságmátrix megtalálása adott k dimenzió mellett, ahol $k < p$. Ezt úgy kell megtennünk, hogy a D^* távolságmátrix és D közt kalkulált négyzetes különbség (Stress: Standardized Residual Sum of Squares) minimális legyen (Mead, 1992).

3.1.2. A távolságmátrix előállítása

A nem – metrikus többdimenziós skálázás első lépése a távolságmátrix meghatározása. Ehhez ki kell választanunk kettő vagy több ismérvet. A távolságmátrix előállításának számos módja van, például az Eukleidészi távolság vagy például a Manhattan távolság. A kutatásomban használt nem – metrikus többdimenziós skálázás során viszont ordinális skálán mérhető kategorikus ismérvek távolságmátrixára volt szükségem, ezért a Gower – féle távolságszámítással dolgoztam, amely alkalmas erre a célra. A Gower – metrika további előnye még, hogy nem követeli meg, hogy az általunk választott ismérvek azonos mérési skálán mérhetők legyenek (Gower, 1971). A Gower – féle hasonlósági index képlete a következő:

$$d_{ij} = \frac{\sum_{k=1}^n \delta_{ijk} d_{ijk}}{\sum_{k=1}^n \delta_{ijk}} \quad (2.)$$

Ahol $\delta_{ijk} = 0$, amennyiben az adott mátrixok megfelelő értékeinek bármelyike üres érték és d_{ijk} pedig 3 alesetre bontható:

1. Ha a k -adik változó legalább *intervallum mérési szintű*: $d_{ijk} = \frac{|x_{ik} - x_{jk}|}{R_k}$ (ahol R_k a k -adik változó terjedelme).
2. Ha a k -adik változó *ordinális mérési szintű*: $z_{ik} = \frac{r_{ik} - 1}{\max r_{ik} - 1}$, (ahol r_{ik} az i -edik megfigyelés sorszáma a k -adik változó szerint). Ebben az esetben z_{ik} kiszámolása után ez az aleset visszavezethető az 1. pontra.
3. Ha a k -adik változó *nominális mérési szintű*: $d_{ijk} = 0$, ha $x_{ik} = x_{jk}$, különben 1.

3.1.3. Skálázás a távolságmátrix alapján

A távolságmátrix kiszámítása alapján a nem – metrikus (ordinális) skálázás elvégzésekor szem előtt kell tartanunk, hogy az eredeti és származtatott távolságok között az ordinális kapcsolat továbbra is fenn fog állni. A skálázás célja a Stress minimalizálása $d_{ij} \leq d_{kl} \rightarrow d_{ij}^* \leq d_{kl}^*$ feltétel mellett, ahol d az eredeti és d^* a származtatott távolságok (Kovács, 2014).

$$Stress = \sqrt{\frac{\sum_{ij} (d_{ij} - d_{ij}^*)^2}{\sum_{ij} d_{ij}^2}} \quad (3.)$$

A Stress (3.) egyenlet alapján kiszámolt értékét Kurskal (1964) szerint az alábbi feltételrendszerrel tudjuk értelmezni:

- $< 0,05$: kiváló
- $0,05 - 0,1$: jó
- $0,1 - 0,15$: közepes
- $0,15 - 0,2$: gyenge
- $0,2 \leq$: nem elfogadható

Az eredeti és újonnan számolt távolságmátrix hasonlósága továbbá a determinációs együtthatóval (R^2) is leírható (Peres – Neto és Legendre, 2006).

3.2. A Blinder – Oaxaca dekompozíció

3.2.1. Háromtagú dekompozíció

A Blinder – Oaxaca dekompozíció csoportok közti, egy konkrét változó átlagában mért különbség felbontására alkalmas eszköz, amelyet a társadalomtudományokban kifejezetten a bérdifferentiák szemléltetésére használnak, Oaxaca (1973) szintén ezen a példán mutatta be a módszertant. A kutatásomban azért alkalmaztam, hogy az előzetes kérdésfeltevéseim alapján a minta különböző részcsoportjai közti átlagos bérkülönbségeket mélyebben elemezhessem.

Egy általános lineáris regressziós analízis során C csoport várható értékének becslése az alábbi:

$$\bar{Y}_C = \beta_C \bar{X}'_C \quad (4.)$$

Ahol $\bar{X}'_C$ a magyarázóváltozók átlagos értékeit tartalmazza, β_C pedig a regressziós együtthatókat jelöli, valamint $\bar{Y} = E(\hat{Y})$ legjobb becslése a megfigyelt mintában. A hibatag várható értéke a 0 a 6 standard modellfeltevés értelmében (amelyeket a 15.1. módszertani mellékletben részletesebben kifejték). Amennyiben A és B csoport átlagos béreinek különbségét szeretnénk megtudni, a képleteket hasonlóan az előzőhöz, felírhatjuk mindkét csoportra. Ezután a két csoport becslült átlagainak különbsége a következő:

$$\Delta \bar{Y} = \beta_A \bar{X}'_A - \beta_B \bar{X}'_B \quad (5.)$$

$$\Delta \bar{Y} = \underbrace{(\bar{X}_A - \bar{X}_B)' \beta_B}_{\text{összetételhatás}} - \underbrace{\bar{X}'_B (\beta_A - \beta_B)}_{\text{részhatás}} + \underbrace{(\bar{X}_A - \bar{X}_B)' (\beta_A - \beta_B)}_{\text{interakció}} \quad (6.)$$

A (6.) egyenlet a háromtagú dekompozíció, amelynek első tagját a szakirodalom *endowmentnek*, azaz adottságnak hívja, ez jeleníti meg a két csoport közti különbség azon részét, amely magukból a magyarázóváltozókból adódik, ez az összetételhatás (Blinder, 1973). A bérezések példáját folytatva amennyiben például a férfiak és nők közti bérdifferentiákat vizsgáljuk, a kor szerinti összetétel hatása lehet például az, ha a mintában több pályakezdő – és emiatt kevesebbet kereső - nő van, mint férfi, amely a nők bérátlagát csökkenti. A második komponens a koefficiensek közti különbség, amely konkrétan azokat a csoportkülönbségeket jeleníti meg, amelyeket nem az összetétel magyaráz. Ezt más szóval részhatáskülönbségnek is nevezhetjük: tehát az eredményváltozó átlagának különbsége a magyarázóváltozók értékei alapján adott csoportokon belül (pl. egy adott korcsoporton belüli átlagos bérkülönbség). A dekompozíció harmadik tagja az előző kettő interakciója, amely azt implikálja, hogy a rész – és összetételhatás különbségek egyszerre is fennállhatnak a két csoport közt, egymást erősítve (Jann, 2008).

3.2.2. Kéttagú dekompozíció

A (6.) egyenlet átrendezésével megkapjuk a kéttagú dekompozíciót, amelyet saját kutatásomban is alkalmaztam:

$$\Delta \bar{Y} = \underbrace{(\bar{X}_A - \bar{X}_B)' \beta_R}_{\text{összetételhatás}} + \underbrace{\bar{X}_A'(\beta_A - \beta_R) + \bar{X}_B'(\beta_R - \beta_B)}_{\text{részhatás és nem megfigyelt hatások}} \quad (7.)$$

A kéttagú dekompozíció egy adott R referenciaesetre épített regresszió becsült paramétereirehöz viszonyítja az összehasonlított csoportok regresszióbéli változóinak koefficienseit, ezt a referenciaesetet a munkaerőpiacot elemző szakirodalom a „diszkrimináció hiányának” nevezi, tehát ez az az eset, ha a két csoport közt semmilyen megkülönböztetés nem lenne. A dekompozíció azt vizsgálja, hogy a két csoport közti különbség milyen mértékben tér el ettől az elméleti értéktől és hogy az eltérés mennyire magyarázható a csoportok összetételével és a csoportokon belüli különbségekkel (Blinder, 1973).

A kéttagú dekompozíció első tagja a megmagyarázott különbség, amely jelentéstartalomban ekvivalens a fentebb tárgyalt összetételhatással. Ez az a differencia, amelyet a mintában található két csoport összetételkülönbségei okoznak. A második tag a megmagyarázatlan komponens, amelyet a szakirodalom „diszkriminációnak” hív, ugyanakkor ez magába foglalja a nem megfigyelt hatásokat is. Jelentést tekintve ez a kéttagú felbontás azon része, amely a részhatáskülönbséggel megegyező. A háromtagú felbontás interakciós tagja az egyenletrendezés során gyakorlatilag ebbe a komponensbe került, így ezen tényező megfelelő

értelmezése nem mindig egyértelmű, számos nyitott kérdést hagyhat, különösen azért, mert a modellbéli változókat több modellen kívüli hatás is befolyásolja, amelyek felvetik a kauzalitás módszertani mellékletben tárgyalt problémáját (Jann, 2008).

A referenciaeset koeficiensének megválasztására többfajta lehetőség is adódik. Választhatjuk például konkrétan az egyik csoport regressziós együtthatóit, amennyiben megvan az előfeltételezésünk, hogy egyedül az a csoport tapasztal megkülönböztetést (például a férfi – női bérkülönbségek esetén feltételezhetjük, hogy csak a nőket diszkriminálják) (Oaxaca, 1973). Ugyanakkor vannak matematikai megközelítések is, Cotton (1988) például a csoportok együtthatóinak mintabeli megoszlással súlyozott átlagát javasolja. Dolgozatomban én a Neumark – féle megközelítést alkalmaztam, amely olyan regressziót épít, mely egyáltalán nem foglalja magába a csoportképző ismérvet, mint regresszort (Neumark, 1988).

4. A KUTATÁSBAN FELHASZNÁLT ADATOK, ADATFORRÁS ÉS ELEMZÉSI KERETRENDSZER BEMUTATÁSA

A Stack Overflow egy olyan weboldal, amely fórumot és közösséget biztosít világszerte a szoftverfejlesztőknek, illetve mindenkinek, aki a programozással hobbi vagy professzionális szinten foglalkozik. A honlapot 2008 -ban alapították, mára pedig egy világszerte ismert tudásmegosztási platformmá nőtte ki magát. A Stack Overflow fórumon a legtöbb esetben a felhasználó egy olyan konkrét informatikai problémát fogalmaz meg, amelyet ő valamilyen programnyelvben próbált megoldani, de akadályokba ütközött, például hibára futott a kódja vagy nem tudta elérni a kívánt eredményt. A kérdés szerkesztőjében van lehetőség beágyazott kódrészleteket és a hibaüzenetet is bemásolni. A többi felhasználó a komment szekcióban ad megoldási javaslatokat, legtöbbször a saját kódjaikkal és azok magyarázataival vagy a hivatalos dokumentációval egybekötve. Ezeket a bejegyzéseket a nem regisztrált látogatók is elérhetik, de kommentet nem tudnak hozzáfűzni a kérdéshez, illetve értékelni sem tudják a válaszok minőségét (stackoverflow.co, n.d.).

A weblap készítői minden évben közzétesznek egy kérdőívet, a *Stack Overflow Annual Developer Survey*-t, amelyben a kitöltő szoftverfejlesztők demográfiai adatain túl a bérezésükről, munkahelyükről, tapasztalataikról kérdezik őket. A kérdőív eredményeit és a begyűjtött anonim adatokat minden évben ingyenesen elérhető formában megosztják (stackoverflow.com, n.d.). Mivel a felmérésben magyar szoftverfejlesztők is részt vettek, így az ő adataikra szűkítem az elemzést, azon belül is kifejezetten a 2018 – 2022 közti időszakra, ugyanis 2018 – tól kezdődően áll rendelkezésre megfelelő számú (évente átlagosan 450 – 500)

magyar kitöltő adata, illetve a gyors technológiai változások miatt 5 évnél korábbi adatokat nem tartottam érdemesnek vizsgálni.

A *Stack Overflow Annual Developer Survey* adatait több kutatásban is felhasználták már. Arbidāne és Silickis (2018) kifejezetten a litván IT munkaerőpiac kompetenciáira szűkítve vetették össze a 2018 -as Stack Overflow kérdőív litván eredményeit a saját, vállalatoknak szóló kérdőívük kitöltéseivel, ezáltal a munkavállalói és munkáltatói aspektusokat ütköztetve. Emellett Silveira és munkatársai (2019) szintén a 2018 – ban publikált kérdőív adatai alapján globális szemszögből nézve elemezték a STEM – szektorban működő vállalatoknál dolgozó munkaerő diverzitását. Mindkét tanulmány elsősorban leíró statisztikai elemzéseken alapult.

4.1. A kutatásban felhasznált adatbázis bemutatása

Ahogy azt már korábban is említettem, a kérdőív eredményeit tartalmazó felhasznált adatbázis évről évre ingyenesen letölthető a *Stack Overflow Annual Developer Survey* honlapjáról .xlsx és .csv kiterjesztésben is. Emellett a bejövő adatokat maga a Stack Overflow is minden évben feldolgozza és az eredményeket globális szinten publikálja. Ezek az elemzések elsősorban leíró jellegűek, statisztikai becsléseket, előrejelzéseket kevésbé végeznek bennük, legfeljebb az előző évek eredményeivel vetik össze az adott évben kinyert információkat (survey.stackoverflow.co, 2022).

A kiindulási adatbázisban az egyes, sorokban található egyedek maguk a válaszadók voltak, míg az oszlopok azok tulajdonságai. Egy válaszadó jellemzői az alábbi nagyobb csoportokra voltak tagolhatók: demográfiai adatok, képzettséggel és munkahellyel kapcsolatos adatok, technológiai ismeretek és Stack Overflow használati szokások.

4.2. Az elemzéshez és vizualizációhoz használt keretrendszer és csomagok bemutatása

Az adattisztítás nagy részét és az elemzést R-ben készítettem el. Az R egy nyílt forráskóddal rendelkező, kifejezetten statisztikai elemzésekre és azok grafikus ábrázolására specializálódott programnyelv, amelyben az adattisztítást, adatmanipulációt és vizualizációt is könnyen elvégezhetjük. Az R számos további függvényeket, számításokat tartalmazó csomaggal bővíthető, amelyet a közösség tagjai is fejlesztenek. Ezek a csomagok a CRAN (Comprehensive Archive Network) FTP szerveréről letölthetők és frissíthetők (r-project.org, n.d).

Az elemzés elkészítésekor a döntési fához az *rpart* és *rpart.plot* csomagokat vettem igénybe. A nem-metrikus többdimenziós skálázás kiszámításához több importált csomagot is felhasználtam. A két egybevont változó Gower - féle távolságmátrixának kalkulálásában a

cluster, a konkrét skálázás elvégzésében pedig a *vegan* csomag állt rendelkezésemre. A Blinder – Oaxaca dekompozícióhoz az *oaxaca* csomag függvényeit használtam fel.

A kiinduló adatbázis, és az adattisztítást, valamint a teljes elemzést megvalósító R kód az alábbi mappában érhető el:

https://unicorvinus-my.sharepoint.com/:f/g/personal/emese_vastag_stud_uni-corvinus_hu/Eh7vfR9q5rhAkJGucQLyrM8BuHAo3ARrQhEpzkJZh9oo-Q?e=NrXg5G

Az elemzés eredményei szintén ide kerültek felmentésre *VastagEmese_TDK_vegso_eredmenyek.RData* néven. Ez az R Workspace fájl tartalmaz minden, a kutatás során felmerült (köztes) változót, valamint a modelleket is.

5. AZ ADATOK ELŐKÉSZÍTÉSE, ADATTISZTÍTÁS

Miután a 15.1. fejezet módszertani mellékletében ismertetett feltevéseket alapul véve megállapítottam, hogy az adatbázis nem használható fix egyedhatásokat is kezelő panelökonometriai modellezésre, a pooled OLS becsléshez az évek keresztmetszeti adatainak egymás alá másolásával kezdtem. Ahhoz, hogy az egybemácsolást elvégezhessem, az első nagyobb feladat az évenkénti adattáblák oszlopainak konzisztens állapotba hozása volt. Javítást igényelt például, hogy több kategorikus ismérvnél, ahol a felhasználók néhány előre adott válasz közül választhattak, az egyes ismérvkimenetek évről évre eltértek, bár a jelentéstartalmuk egyezett. Szükség volt továbbá a nagyon kis gyakoriságú ismérvkategóriák egybevonására is (például az Etnikum ismérvnél, ugyanis a magyar adatokban csak néhány tucat nem európai származású volt együttesen).

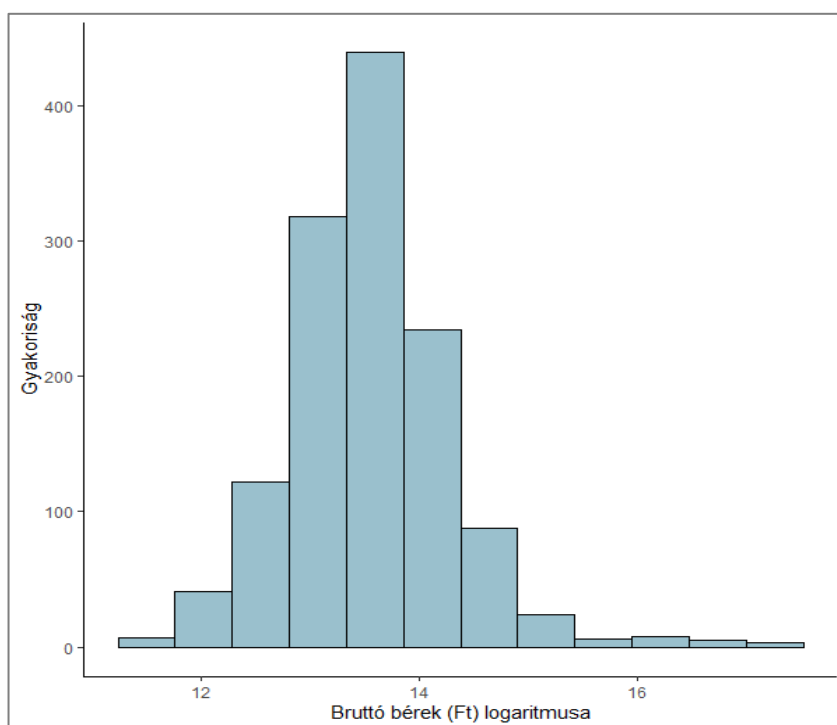
A kérdőívben szereplő kérdésekben évről évre történtek kisebb változások, éppen ezért csak azon oszlopok kerültek bele a végső adatbázisba, amelyek minden évre rendelkezésre álltak. Egy évnyi adatban átlagosan 60 - 70 féle ismért állt rendelkezésre, az összefűzött adatbázisba pedig végül 27 oszlop került, ekkora volt a 2018 – 2022 közötti adatok közös metszete. Ez a kiinduló adatbázis összesen 2228 válaszadó adatait tartalmazta.

5.1. A becsült eredményváltozó megtisztítása

Az évenkénti adatok konzisztens összemácsolása után az egyes oszlopok megtisztítása következett. A Fizetés oszlopból ki kellett, hogy szűrjem azokat a kitöltőket, akik nem adták meg a fizetésüket. Ezután a szűrés után 1295 ember maradt az adatbázisban, évenként átlagosan 250 – 300. Az ismeretlen fizetéseket azért nem lehetett semmilyen gépi tanuló módszerrel visszapótolni, mert éppen maga a Fizetés a becsült eredményváltozó. Azzal, ha a többi

magyarázóváltozó alapján akarnánk a hiányzó értékeket feltölteni, eltorzítanánk a későbbi becslések eredményeit, hibásan nagyítanánk fel egyes változók hatásait. Ebből az okból kifolyólag az egyetlen megoldás azon sorok kiszűrése volt, ahol a Fizetés nem volt adott (Van Buuren, 2018). Ez nagyjából 42%-os adatvesztéssel járt, ugyanakkor az évek megoszlása egyenletes maradt. Ez azt is jelenti, hogy évente átlagosan az emberek közel fele nem adta meg a bérét.

A szűrés után a bent maradó emberek nagyjából 75%-a jelölte, hogy forintban kapja a fizetését, ugyanakkor 102 ember az eurót, 42 az amerikai dollárt, 11 az angol fontot és 1 ember pedig a svájci frankot jelölte a bére devizájának. Az átváltásokhoz minden devizának megkerestem az adott évi átlagos árfolyamát a Magyar Nemzeti Bank honlapján és évenként elkülönítve váltottam át az értékeket. A kiinduló adatbázisba bekerülő bérek tehát a forintban értendő bruttó havi bérek voltak, amelyeket a válaszadóknak a bónuszok és egyéb béren kívüli juttatások nélkül kellett megadni. A legnagyobb gondot ezen túl az a 172 ember okozta, akik bár megadták a bérüket, de a deviza mezőjét nem töltötték ki. Mivel a két leggyakoribb, forinttól különböző deviza az euró és az amerikai dollár volt, így ezekben az esetekben, ahol 3 vagy 4 számjegy hosszú volt a fizetés, ott az adott évi USD és EUR átlagárfolyamok átlagával szoroztam fel a béreket. Végezetül a fizetést logaritmáltam, ugyanis – a bér adatokra jellemző módon – rendkívül jobbra elnyúló volt. A logaritmikus skálára váltás után az eloszlás közel normális lett, ezzel eleget téve a módszertani mellékletben említett 6 modellfeltevés egyikének.



5. ábra: A bruttó bérek (Ft) eloszlása logaritmálás után.

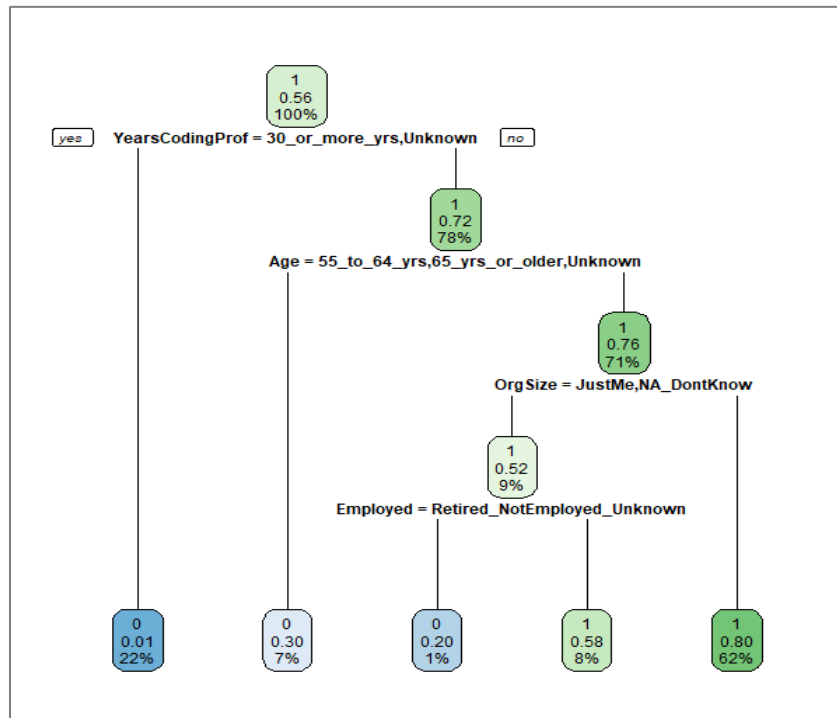
5.2. Döntési fa modell az eredményváltozóra

Mivel azok kiszűrése, akik nem vagy hibásan adták meg a fizetésüket kb. 42%-os adatvesztéssel járt, úgy gondoltam, azzal is értékes információkat nyerhetek ki, ha egy döntési fa modellel megvizsgálom, van -e valamilyen jellegzetes mintázat abban, hogy kik nem adják meg a fizetésüket. A döntési fát az R *rpart* csomagjának *rpart* függvényével építettem fel. Ehhez az adaton még egy olyan módosítást kellett végezni, hogy egy dummy változóba kódoltam, hogy az illető megosztotta -e vagy sem a fizetését, a döntési fa ezt a változót becsülte (0 esetén nem adta meg, 1 esetén igen). A komplexitás paraméterét 0,009-re állítottam, tehát ha egy levél nem segítette legalább ennyivel az illeszkedést, nem is jött létre. A másik paraméter, amelyet beállítottam, a maximális mélység volt, ameddig a fa lenyúlhatott, ez a végső modellben 20 lett.

A célom a döntési fával nem az előrejelzés volt, hanem a mintán belül minél homogénebb csoportok képzése, így nem is alkalmaztam tanító – teszt felbontást az adathalmazon. Ezt a módszertant azért használtam, hogy jobban megértsem, jellemzően milyen tulajdonsága van azoknak, akik nem adták meg a fizetésüket, hiszen ez is egy további információtartalmat hordozó tényező. Ebből kifolyólag azt tartottam szem előtt, hogy a fa egyszerre legyen kellően komplex, de még vizuálisan is átlátható, mely tulajdonságok a fentebb említett 0,009 és 20 komplexitás és mélység paraméterek mellett valósultak meg legjobban.

A döntési fa egyik legjellemzőbb levele a 6. ábrán a legalsó szinten látható, ebbe a részcsoportha a minta 62% -a esik és azok a kitöltők, akik idetartoznak, 80% valószínűséggel megadják a fizetésüket. Ezen részcsoportha foglalja magába azokat az informatikusokat, akik kevesebb mint 30 év professzionális kódolási tapasztalattal rendelkeznek, 55 évnél fiatalabbak és nem egyéni vállalkozók. Azon szakemberek, akiket a fa idesorolt, jellemzően megadták a fizetésüket. A fa második legjellemzőbb levele 22% - os relatív gyakoriságú, ide azok a kitöltők tartoznak, akik több, mint 30 év professzionális kódolási tapasztalattal rendelkeznek, vagy nem adták meg a

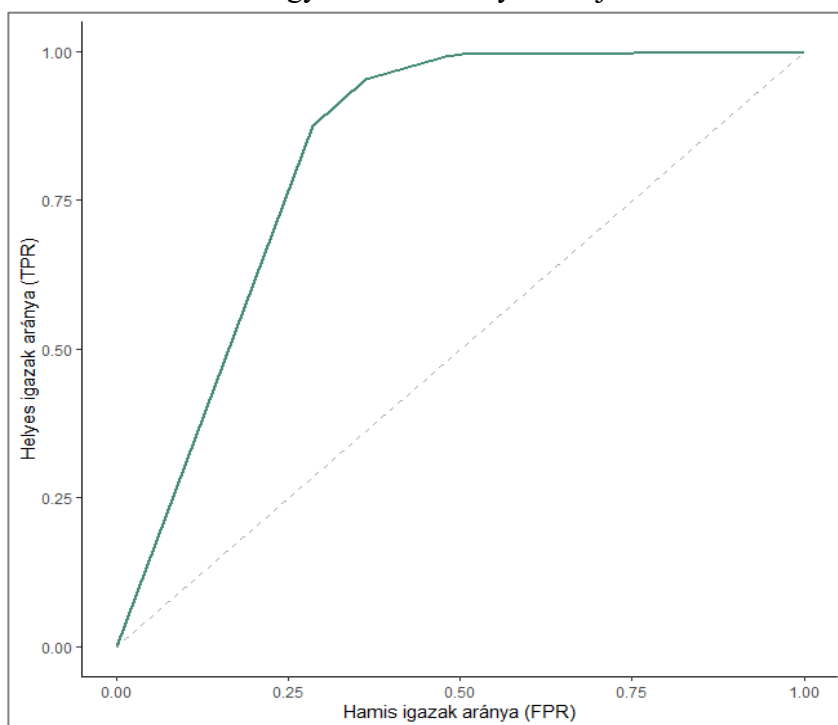
korukat. Azon fejlesztők, akik ezekkel a paraméterekkel rendelkeznek, várhatóan mindössze 1% valószínűséggel adják meg a jövedelmüket.



6. ábra: Döntési fa modell a jövedelem megosztásával kapcsolatban.

A döntési fa eredményeiről a külön - külön nem részletezett leveleket is figyelembe véve, általánosságban elmondható, hogy jellemzően az idősebb, valószínűleg nyugdíjas korosztály nem adta meg a jövedelmét. Ugyanakkor minden magyarázóváltozóban előfordul az „Ismeretlen” részkategória is, így összességében megállapítható a tény, hogy nagy valószínűséggel, akik nem adták meg a fizetésüket, azok más, életkorra, munkahelyre és tapasztalatokra vonatkozó információkat sem szívesen osztottak meg a kérdőívben. Ezt az is alátámasztja, hogy például a korcsoportok és nemek szerint megbontva az tapasztalható, hogy mindkét ismerv esetén az „Ismeretlen” részkategóriák átlagosan 83% -a bérét sem adta meg.

A döntési fa kiértékelése során a ROC görbe alatti terület közel 0,83, amely közelebb áll 1 -hez mint 0,5 -höz, ezért ilyen szempontból értékelve a modell becslése megfelelően sikerült, a hiányzó fizetések mintázatának nagyobb részét helyesen írja le.



7. ábra: A döntési fa modellt kiértékelő ROC görbe.

5.3. Magyarázóváltozók átalakítása

A modellszelekciót megelőzően a bér változóján kívül minden egyéb ismérv kategorikus ismérv volt, így ezeket R faktorváltozókká alakítottam. Az iskolai végzettség esetén a Bsc. vagy azzal egyenértékű diplomával rendelkező volt a referenciakategória, ugyanis az 1295 emberből 432 tartozott ebbe a kategóriába, így ez volt a leggyakoribb. Az életkoroknál a legfiatalabb, de összevont kategóriát, a 18 év alatti/18 – 24 év közötti korcsoportot választottam referenciának. Az összevonás oka, hogy a fizetések kiszűrése után 1 olyan ember maradt az adatbázisban, aki 18 év alatti és rendelkezik rendszeres jövedelemmel.

Mivel az adatbázisban szereplő változók zöme olyan kérdésekből eredt, ahol a kitöltő több válaszlehetőséget is megjelölhetett, az alapvető kiindulópont az ismérvkimenetek képzésében az volt, hogy a leggyakoribbakat válogattam ki vagy az egymáshoz logikailag közel állókat egy változóba soroltam, a többi, kis gyakoriságú kimenetet pedig egyéb kategóriába mentettem. A kiválogatott ismérvkategóriákból ezután külön változókat képeztem és 0 – 1 dummykká kódoltam őket (ahol 0: az egyén nem tartozik az adott csoportba vagy nem válaszolt a kérdésre 1: az egyén megjelölte az adott válaszlehetőséget). Minden ilyen jellegű változó esetében a 0 lett a referencia kategória.

A *Stack Overflow Annual Developer Survey* kérdőívében minden évben vannak kérdések az egyén által ismert és munkában használt különböző technológiákról, amelyek a következő kategóriákba sorolhatók (mindhez 1-1 kérdés tartozott a kérdőívben): munkában használt programnyelvek, adatbázistechnológiák, platformok és keretrendszerek. Mindegyik kategóriában a már elsajátított és használt technológiákon túl volt egy – egy olyan kérdés is, amely arra irányult, hogy melyek azok az adott kategórián belül, amelyeket a kitöltő a későbbiekben szeretné elsajátítani. Ezeknél a kérdéseknél a kitöltő bármennyi válaszlehetőséget megjelölhetett. Ebből fakadóan a technológiai magyarázóváltozók esetén én is az említett négy kategória szerint képeztem meg az ismérveket. A használt logika itt is az volt, hogy csak azokat legismertebb technológiákat – például programnyelvek esetén a TIOBE index¹ által 2022 szeptemberében 20 legnépszerűbbnek tartott nyelv – válogattam ki külön dummy változókba, amelyek az adatbázisban is kellően nagy számossággal (nagyobb, mint 100) rendelkeztek. Emellett itt is vontam össze változókat, amennyiben logikailag közel álló technológiákról volt szó és a kitöltők jellemzően együtt jelölték meg őket (pl. jellemzően front - end webfejlesztéshez használt HTML, CSS, JavaScript, PHP és TypeScript nyelvek). Azon technológiákat, amelyek nem kaptak különálló dummy változót, egy „*Egyéb*” kategóriába soroltam, melynek jelentése, hogy az illető megjelölt egy, a külön dummy változókba sorolt technológiáktól különböző technológiát, vagy nem adott választ a kérdésre. Ezt az előzetes válogatási módszert a programnyelvek, adatbázistechnológiák, platformok és frameworkök esetén is külön – külön elvégeztem. Továbbá minden, külön dummy változóba sorolt technológia kétféle változóval rendelkezett: a kitöltő dolgozott -e már az adott technológiával („*WorkedWith*”) vagy szeretné -e azt a későbbiekben ismerni, megtanulni („*Desire*”). A technológiákkal kapcsolatos magyarázóváltozók bemutatása, valamint a válogatás részletezése a 15.1 -es számú módszertani mellékletben található meg.

6. KIEGÉSZÍTŐ MEGJEGYZÉSEK AZ ADATBÁZISRÓL ÉS A MINTAVÉTEL MÓDJÁRÓL

Az elemzések elvégzésekor, valamint az eredmények értelmezésekor fontos szem előtt tartani az adatbázisból és a mintavétel jellegéből adódó torzító tényezőket.

6.1. Adatbázis

Az adatbázis – bár számos lényeges kérdéssel kapcsolatban összesít válaszokat – nem terjed ki például a fejlesztők egyéni attitűdjére. Véleményem szerint a pszichológiai – személyiségi

¹ A TIOBE index bemutatása a módszertani mellékletben olvasható a 15.1. fejezetben.

tényezők legalább annyira befolyásolják egy ember karrierútját, mint a demográfiával, munkatapasztalattal és kompetenciákkal kapcsolatos ismérvek. Az egyes évek eredeti kérdőíveiben szerepeltek ilyen jellegű kérdések (pl. munkahelyi elégedettség, szokások), ugyanakkor ezek a vizsgált évek között nem voltak egységesek, így nem volt lehetőségem őket bevonni a végső, kiinduló adatbázisba. Az ilyen és ehhez hasonló tényezők okozhatnak olyan torzításokat, amelyeket a kauzalitás problémájának kapcsán a módszertani mellékletekben ismertettem.

6.2. Mintavétel módja

A kérdőív kitöltéséhez nem szükséges Stack Overflow felhasználónak lenni, ugyanakkor várhatóan azok töltötték ki, akik – még ha nem is felhasználók – naponta látogatják az oldalt. Ennélfogva a mintában felmerülhet egy torzítás ezen csoport felé, amely például abban mutatkozhat meg, hogy elsősorban a fiatalabb korosztály van túlsúlyban a kitöltők között.

Továbbá, mivel a kitöltés önbevallásos alapon történt, fennáll annak veszélye, hogy egyes kitöltők nem a valós információkat adták meg magukról. Ez történhetett szándékos és véletlen módon is, például, ha félreértették a kérdést. A véletlen félreértések okozta torzítások egyike lehet, ha az ismert technológiák esetén a kitöltők nemcsak azokat a jelölték meg, amelyekkel munkahelyen már dolgoztak, hanem minden olyan továbbit, amelyet csak felületesen is, de ismernek (például egyetemi órákról), ám ténylegesen nem használnak. Ez különösen a fiatalabb, egyetemista vagy pályakezdő informatikusok esetén tud zavaró befolyással bírni az eredményekre.

7. A KIINDULÓ ADATBÁZIS LEÍRÓ STATISZTIKÁJA

Az elemzéshez használt kiinduló adatbázisról a következő áttekintést készítettem el:

	2018	2019	2020	2021	2022
Gyakoriság	230	328	234	261	242
Nők (fő)	9	19	4	5	4
Férfiak (fő)	203	303	212	249	232
Leggyakoribb életkor kat.	25 - 34 év	25 - 34 év	25 - 34 év	25 - 34 év	25 - 34 év
Fizetések átlaga (Ft)	2786156,62	2930089,27	3040425,80	3004724,51	3843516,99
Fizetések mediánja (Ft)	954000	813000,00	1100000,00	1300000,00	1500000,00
Leggyakoribb pozíció	Full-Stack fejlesztő	Full-Stack fejlesztő	Full-Stack fejlesztő	Full-Stack fejlesztő	Full-Stack fejlesztő
Leggyakoribb iskolai végzettség	Msc. vagy ezzel egyenértékű szint	Bsc. Vagy ezzel egyenértékű	Bsc. Vagy ezzel egyenértékű	Bsc. Vagy ezzel egyenértékű	Bsc. Vagy ezzel egyenértékű
Leggyakoribb munkahely típus	10 - 100 fős	11 - 100 fős	12 - 100 fős	13 - 100 fős	14 - 100 fős
Leggyakoribb ismert programnyelv	Webfejlesztéshez használt nyelvek	Webfejlesztéshez használt nyelvek	Webfejlesztéshez használt nyelvek	Webfejlesztéshez használt nyelvek	Webfejlesztéshez használt nyelvek
Leggyakoribb kódolási tapasztalat kat.	9-11 év	6-8 év	9-11 év	9-11 év	6-8 év
Leggyakoribb professzionális kódolási tapasztalat kat.	3-5 év	3-5 év	3-5 év	3-5 év	3-5 év

1. táblázat: A kiinduló adatbázis leíró statisztikájának összesítése.

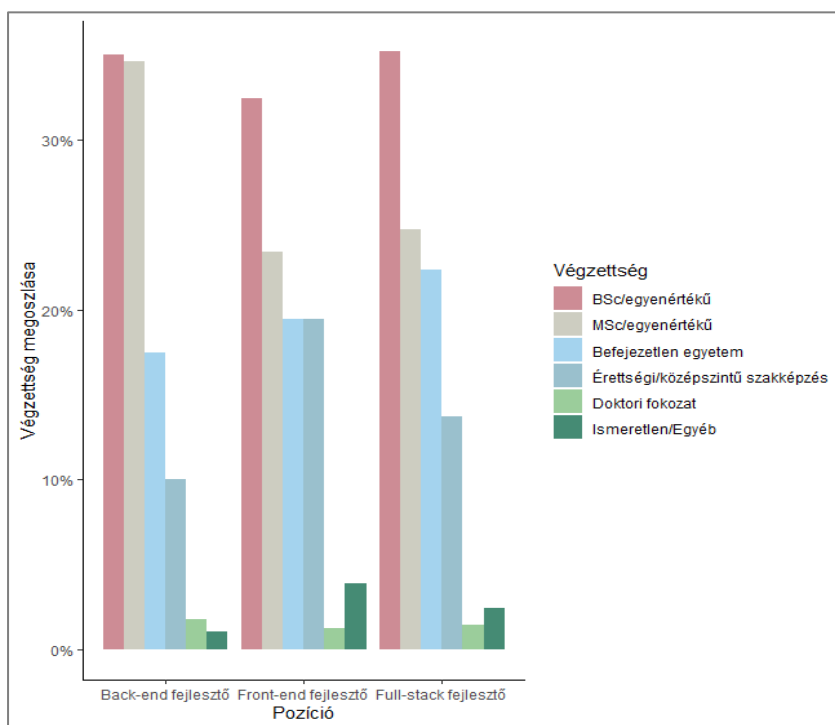
Az adatbázisban szereplő kitöltők gyakorisága az évek során stagnált, átlagosan 250 fő körül mozgott. A nők számottevően kevesebben voltak minden évben, ugyanakkor voltak olyan szoftverfejlesztők, akik nem adták meg a nemüket. Az életkorokat tekintve relatíve fiatalabb munkavállalók voltak a kitöltők, minden évben a 25 – 34 éves korosztály volt a leggyakoribb, őket pedig rendre a 35 – 44 évesek követték. Ez egybe is vág azzal a ténnyel, hogy a kódolási tapasztalatokat tekintve például professzionális szinten a 3 – 5 évig terjedő kategória volt a leggyakoribb.

Az összes évre jellemző az, hogy a legtöbb válaszadó full - stack fejlesztő volt, amely magyarázza azt, hogy mért a webfejlesztéshez használt nyelvek fordultak elő legtöbbször az adatbázisban. Ezek azok a nyelvek, amelyeket az adatbázis tisztítása során egy közös kategóriába soroltam: HTML, CSS, JavaScript, PHP és TypeScript. Hozzá kell tenni viszont, hogy mivel ez a kategória öt nyelvet is magába foglal, a magas számú gyakoriság az összevonásból is adódhat. A második leggyakrabban használt nyelv az SQL volt (évenként átlagosan 136 felhasználó adta meg), ezt pedig a Python és Bash/Shell/PowerShell követték évenként átlagosan 100 és 98 kitöltővel. Ezen a ponton fontos megjegyezni, hogy arról nincs információ, hogy az adatbázisban található kitöltők milyen mértékben reprezentatívak a magyar szoftverfejlesztői piacot tekintve. A későbbi eredmények kapcsán annak, hogy a webfejlesztésben dolgozók a leggyakoribbak, lehet olyan torzító hatása, hogy a többi, kisebb

létszámban előforduló pozíciók és az ezekhez köthető technológiák változói kevésbé tűnhetnek szignifikánsnak a valóságnál.

A leggyakoribb legmagasabb iskolai végzettség kategória az első vizsgált évet leszámítva az alapképzés vagy azzal egyenértékű szint volt és a mesterfokozat gyakorisága is minden évben csak kevéssel lemaradva követte ezt. Érdekes információ továbbá, hogy minden évben a kitöltők átlagosan 20%-át tették ki olyan emberek, akik elkezdtek valamilyen főiskolai vagy egyetemi képzést, de azt nem fejezték be. Az egyetemi képzés fontosságának kérdését az informatikusok bérezését illetően később bővebben részletezem.

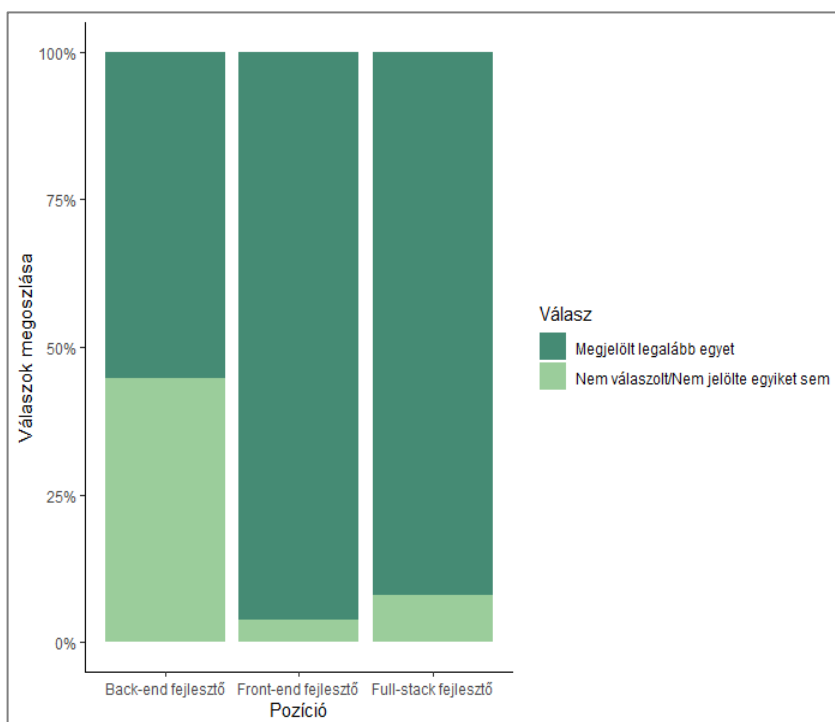
Ami a fizetéseket illeti, az adatbázisban alapvetően a bruttó fizetések adottak, illetve az átlagot a kiugró értékek könnyen el tudják mozdítani, így az elsőre irreálisan magasnak tűnő bérek ebből adódhatnak. A mediánok már valósabb képet mutatnak, ebből látszik, hogy a fizetések évről évre növekvő tendenciát mutattak. Ez egyfelől lehet az infláció, illetve az árfolyamok hatása forinttól különböző devizák esetén. Ugyanakkor, alátámasztja a 2.2.2.1. fejezetben megállapítottakat, hogy a magyar IT szektor munkavállalói az erős munkaerőhiány miatt kedvező alkupozícióban vannak a munkáltatókkal szemben. A táblázatban feltüntetett fizetések a Hays Salary Guide (2023) szerinti átlagos median – senior fizetést tükrözik. Az átlag és medián elhelyezkedése minden évben jobbra elnyúló eloszlásra utalt.



8. ábra: A webfejlesztés területén dolgozók megoszlása iskolai végzettség szerint a Stack Overflow adatbázisban.

Mivel az adatbázisban a leggyakoribbak a webfejlesztéssel foglalkozó informatikusok voltak, így ezt a részcsoportot mélyebben is megvizsgáltam, amelyet a 8. és 9. ábrákon szemléltettem.

Iskolai végzettségeket tekintve az mondható el, hogy az adatbázisban található webfejlesztők közül arányaiban az egyetemi végzettséget megszerzettek fordulnak elő a leginkább. A legtöbb mesterfokozatot elért informatikus aránylag a back - end fejlesztők körében van. Érdekes információ továbbá, hogy egyik csoport körében sem elhanyagolható az egyetemet otthagynók és középiskolai – középszintű szakképesítéssel rendelkezők aránya, amelyek a front – end fejlesztők körében például szinte megegyeznek.



9. ábra: A front - end fejlesztéshez használt nyelvek (HTML, CSS, PHP, JavaScript és TypeScript) megoszlása a webfejlesztéssel foglalkozók között a Stack Overflow adatbázisban.

A fenti ábrán annak a változónak a megoszlását vizsgáltam a csoportok között, amely a HTML, CSS, PHP, JavaScript és TypeScript nyelveket tartalmazta egybevonva (az egybevonás mikéntjét a módszertani mellékletében részletezem). Mivel ezeket a nyelveket főként a front-end fejlesztéshez használják, így érthető a tendencia, hogy a back - end fejlesztők jelölték meg őket arányaiban nézve a legkevésbé. A munkarendet megfigyelve mindhárom webfejlesztői állásnál a legjellemzőbb a teljes munkaidőben foglalkoztatott kategória, átlagosan 65% körül, ezt pedig rendre mindegyik csoportban az egyéni vállalkozó – szerződéses munkavállalói státusz követi, amely mindhárom fejlesztői részcsoportban körülbelül 31%-os arányban fordul elő.

8. AZ ELEMZÉS EREDMÉNYEI ÉS AZOK ÉRTÉKELÉSE

A bérek becslésében ökonometria megközelítést alkalmaztam, melynek során olyan modellt építettem, amely a logaritmált béreket becsülte a legkisebb négyzetek módszerén alapulva. A logaritmálás következtében egy adott változó modellben szereplő együtthatóját e hatványára emelés után tudjuk visszakapni. A kiinduló modellből a Wald - tesz és információs kritériumok segítségével szelektáltam a nem szignifikáns változókat, majd további dimenziócsökkentést hajtottam a nem - metrikus többdimenziós skálázás módszerével. A végső modellen több szempontból is végeztem dekompozíciós modellezést a Blinder – Oaxaca dekompozíció segítségével.

8.1. Modellszelekció

A kiinduló modellemben a logaritmált béreket becsültem az adatbázisban található összes egyéb változóval. Ezután először azokat a nominális változókat szelektáltam ki, amelyeknek egyik ismérvkimeneteléhez tartozó dummy változó sem volt szignifikáns a semmilyen szokásos szignifikanciaszinten a parciális t - próbák p – értékei szerint. Ebben az esetben a szelekciót megelőzően minden változóra külön – külön ellenőriztem az AIC, BIC és Wald – teszt által adott eredményeket. Amennyiben mindhárom mutató szerint elhagyható volt a változó, kivettem a modellből. A technológiai változók esetén úgy végeztem el a szelekciót, hogy az adott technológiai dummy változót a hozzá tartozó egyéb kategóriába soroltam, majd kivettem a modellből. A teljes szelekciós fázist megelőzően ellenőriztem minden változó VIF mutatóját, amely csak 4 darab esetén jelzett káros multikollinearitást. Ezek a *YearsCoding*, *YearsCodingPro*, *SOParticipate* és *SOHasAccount* változók voltak. Az eredmény azért nem meglepő, mivel feltételezhető az erős kapcsolat aközött, hogy valaki hány éve kódol és hány éve kódol professzionális szinten. Hasonlóan, akinek van Stack Overflow felhasználója, az várhatóan nagyobb eséllyel vesz részt a fórumbeszélgetésekben, így a szoros összefüggés itt is érthető. A szelekció során ebből a négy változóból csak a *SOHasAccount* -ot hagytam el, ennek oka, hogy a *SOParticipate* rendelkezett olyan ismérvkimenetekkel, amelyek minden szokásos szignifikancia - szinten szignifikáns dummy változóval bírnak. Illetve ez a tényező magában hordozza a *SOHasAccount* információtartalmát olyan szempontból, hogy csak az tud hozzászólni a fórumon feltett kérdésekhez, akinek van felhasználói fiókja. A kódolási tapasztalatokra vonatkozó 2 változót nem hagytam el, hanem a nem – metrikus többdimenziós skálázás módszerét felhasználva sűrítettem egybe hatásaikat egy látens „tapasztalati” változóba, amelyet a 8.2. fejezetben részletesebben is kifejtek.

A szelekció főbb fázisait az alábbi táblázat foglalja magába (a teljes táblázat a kihagyott változók pontos felsorolásával a mellékletben található):

Állapot	Modell változójának neve az R kódban	Leírás	Változók száma (tengelymetszet nélkül)	AIC	BIC	Wald - teszt p-érték
1	model_full_log	Kiinduló modell, amely az összes változóval (kivétele a deviza) becsülte a logaritmált béreket	110	2391	2970	-
2	model_szelektalt_3	Első szűkített modell, amelyben csak a p - érték alapján nem szignifikáns változókat szelektáltam	93	2387	2878	4,77%
3	vegso_szelektalt	Végso szűkített modell	70	2364	2736	55,62%

2. táblázat: A modellszelekció főbb lépései.

A deviza változóját abból az okból kifolyólag volt szükséges már a kiindulási modellből kihagyni, mivel a béreket ezen alapulva váltottam vagy alakítottam át, ezáltal a kapcsolat a két változó közt szinte lineáris. A Wald – teszt p – értékei a táblázatban mindig az adott állapotot hasonlítják az előző állapothoz, tehát a 2. táblázat harmadik sorában az 55,62% a *model_szelektalt_3* modell és *vegso_szelektalt* modell közti 23 változó elhagyására kalkulált p – érték, ugyanakkor minden változót külön – külön is teszteltem, mielőtt elhagytam volna őket.

8.1.1. Demográfiai jellemzők

A kiinduló adatbázisban a következő demográfiai jellemzők szerepeltek: életkor, nem és szexuális orientáció. Ezek közül egyedül az életkor volt az, amely kellően szignifikánsnak bizonyult. Utóbbi kettőnél nem voltak matematikailag megfelelően mérhetőek az egyes részcsoportok közti különbségek a nagyon kis elemszám miatt. Például a nemek esetében – ahogy azt már korábban a 7. fejezetben leíró statisztika kapcsán említettem – elenyésző volt a nők száma a férfiakhoz képest, emiatt a modellben a standard hiba számítása a szignifikanciaszinthez nem adott volna pontos értéket.

Összességében tehát valóban olyan eredményt kaptam, amelyet vártam, hogy az IT szektorban dolgozók esetében nem szignifikáns a nemek közti bérdifferentia. Ezzel szemben viszont fontos megjegyezni, hogy önmagában a nők ilyen kis létszáma is hordoz magában olyan mondanivalót, hogy valószínűleg eleve alulreprezentáltak a magyar piacon, mivel véleményem szerint a társadalom a programozói – informatikai állásokat inkább maszkulinnak tartja. Ezt a nemzetközi elemzések is alátámasztják, O’Dea, Lagisz, Jennions és Nakagawa (2018) szerint a STEM – szektorban található munkavállalók túlnyomó része többek között azért férfi, mert a lányok karrierválasztási és bérrel kapcsolatos preferenciáit már az iskola legelső szintjétől kezdve negatív irányba formálja a társadalom és a közvetlen környezet sztereotipikus gondolkodásmódja a nemi szerepekről.

A szexualitás esetén hasonlóan a heteroszexuálisok voltak túlnyomó többségben a mintában, ugyanakkor ezen ismerv kapcsán azt gondolom, hogy olyan magánéleti kérdéskörbe tartozik, amelyet vélhetően a munkavállaló nem oszt meg a felettesével, tehát nem hiszem, hogy ez alapján jellemző lenne a bérdifferentiálás.

8.1.2. Iskolai végzettség

Az iskolai végzettségek dummy változói közül az alapszakos vagy azzal egyenértékű végzettséggel rendelkező kitöltők csoportja volt a referencia kategória. A mesterszakot végzett informatikusok együtthatója szignifikáns volt, ugyanakkor csak 10% -os szignifikanciaszinten. A végső modell regressziós becslése szerint minden egyéb tényező változatlansága mellett a magyar informatikusok közt átlagosan kb. 8,8% - al keres többet egy mesterdiplomával rendelkező szakember egy alapszakot végzettnél.

5% -os szignifikanciaszint mellett továbbá az eredmények alapján az állítható, hogy azon kitöltők, akik nem adták meg iskolai végzettségüket, nem végezték el a közoktatást, vagy csak általános iskolát végeztek, várhatóan ~24% -kal keresnek kevesebbet az alapszakkal rendelkezőknél, feltételezve, hogy minden egyéb tényezőt konstans értéken tartunk.

Az összes többi, iskolázottság szerinti részcsoport (gimnáziumi – középfokú szakképesítés, egyetemet otthagyó, doktori fokozat) dummy együtthatója a végső modellben negatív volt, tehát ezek a kitöltők várhatóan kevesebbet keresnek az alapszakos diplomával rendelkező informatikus szakembernél. Bár ezen változók parciális t – próba szerinti p – értékei nem jeleztek szignifikanciát, érdekes megfigyelés, hogy az egyetem – otthagyók várhatóan kevesebb bérre számíthatnak egy Bsc. diplomával rendelkező informatikushoz képest (várhatóan kb. 6% -kal).

Ez az eredmény ebben az értelemben igazolja azt, hogy a magyar IT szektorban továbbra is van relevanciája az alapfokú egyetemi végzettségnek, a szakképzés és egyéb nem egyetemi kurzusok erőteljesebb megjelenése ellenére is. Emellett megállapítható a tény, hogy bár a mesterfokozatot elért informatikusok átlagosan magasabb fizetéssel rendelkeznek, ez a doktori fokozatú szakemberekre már nem igaz. Feltételezhetően a doktori végzettséggel rendelkező kitöltők zöme egyetemen is oktató és kutató tanár, számukra pedig a tudományos kutatásokhoz valószínűleg a szoftverfejlesztés egy eszköz, ezért ezek a tevékenységek nem igényelnek olyan mélységű fejlesztési tudást, mint egy szoftvert, mint értékesített terméket fejlesztő informatikus esetében. Az eredmények egybevágóak a 2.2.2.2. fejezetben tisztázottakkal, hogy alapvetően jellemző a tendencia, hogy az egyetemeken programozást tanító oktatók jelentős része

ténylegesen nem szoftverfejlesztő, így, ha tanítanak is programozást, az számos esetben kevésbé komplex, mint egy gyakorlatban megoldandó probléma. Emellett további magyarázat lehet a jelenségre, hogy véleményem szerint az egyetemi oktatás alacsonyabb jövedelmet kínál egy szoftverfejlesztői pozíciónál.

8.1.3. Programnyelvek

Összességében elmondható, hogy a programnyelvekkel kapcsolatos magyarázóváltozók közül szinte egyik sem volt szignifikáns és elhagyásuk minden tekintetben javított a modellen. Ugyanakkor, hogy elkerüljem azt az esetet, hogy a programnyelvi ismereteket egyáltalán nem veszem figyelembe, a konkrét nyelvek helyett két olyan változót vettem be, amelyek azt foglalják magukba, hogy az illető hány darab nyelvet ismer vagy hány darab nyelvet szeretne a későbbiekben megismerni. Ezek közül érdekes módon az ismert programnyelvek száma negatív előjellel jelent meg a végső modellben, míg a megismerni vágyott nyelvek száma pozitívvá. Az eredményre magyarázat lehet, hogy számos, ma már elavultabbnak tekinthető programnyelv is létezik, melyek ismerete nem feltétlenül növeli a bért, azt viszont befolyásolhatják, hogy egy fejlesztő később más nyelvek szintaxisát és logikáját hogyan sajátítja el. Ugyanakkor feltételezhetően a megtanulni vágyott nyelvek olyan technológiák, amelyek a piacon is nagy népszerűségnek örvendenek.

8.1.4. Platformok

A platformok tekintetében is hasonlóan jártam el, mint a programnyelvek esetén, ugyanis külön – külön az ezekre az ismérvekre létrehozott változók sem voltak szignifikánsak semmilyen szokásos szignifikanciaszinten. Később azonban bebizonyosodott, hogy a platformok esetén a „Hány darab platformot ismer?” és „Hány darab platformot szeretne megismerni?” változók szintén nem rendelkeztek elegendő szignifikanciával, így ezeket végleg elhagytam a modellből. Tehát az elemzések alapján azt az eredményt kaptam, hogy a platformok ismerete nem jelent szignifikáns különbséget a bérezésben és a programnyelvek tekintetében is inkább az számít, hogy az illető hány darabot ismer -e.

Az eredményeknek olyan értelmezése lehetséges, hogy összességében az, hogy egy bizonyos programnyelvet vagy platformot ismerjen a fejlesztő, nem megkülönböztető tulajdonság, hanem eleve belépési kritérium az adott vállalathoz, ahol ez elvárás, így a magasabb kompenzációt a fejlesztők az ezen kompetenciákon felül felmutatott teljesítményükért kaphatják. Emellett a munkaerőhiányból adódóan előfordulhat az is, hogy az adott vállalat olyan munkavállalókat is felvesz, akik nem ismerik az ott használt programnyelveket vagy

platformokat, de már van munkatapasztalatuk más felületekkel vagy más nyelvekkel, így gyorsan átképezhetők. Általánosan a szoftverfejlesztői állások más területek pozícióival összehasonlítva valóban átlagosan magasabb bért kínálnak (ahogy azt az 2.2.1 fejezetben említettem), ugyanakkor mivel az adatbázisban csak informatikusok vagy informatikai területen dolgozók voltak, így egymás között megfigyelve vélhetően nem akkora a szóródás, mint ahogyan arra előzetesen számítottam.

8.1.5. Adatbázistechnológiák

Az adatbáziskezeléssel kapcsolatos ismeretek esetén a végső modellbe az Elasticsearch ismerettel kapcsolatos „*WorkedWith*” és „*Desire*” változók, valamint az SQL alapú adatbáziskezelők hasonló jelentéstartalmú „*WorkedWith*” és „*Desire*” változói maradtak bent. A többi, ebbe a részcsoporthoz tartozó technológia a szignifikancia hiánya miatt az egyéb NoSQL nyelveket magába tömörítő változóba került. Az SQL alapú adatbázis rendszerekkel kapcsolatos változók önmagukban („*WorkedWith*” és „*Desire*”) szintén nem voltak szignifikánsak, ugyanakkor bent maradtak a modellben az Elasticsearch mellett, mivel jelentésüket tekintve nem voltak az egyéb NoSQL alapú keresőmotorok változóiba sorolhatók. Ennélfogva a végső modellben az adatbázistechnológiák változónak jelentése a következő:

- *ElasticsearchWorkedWith*: Az illető dolgozott – e már az Elasticsearch keresőmotorral.
- *SQLWorkedWith*: Az illető dolgozott – e már SQL alapú keresőmotorral (pl. MSSQL, MySQL ... stb).
- *OtherNoSQLWorkedWith*: Az illető ismer -e egyéb, nem SQL alapú adatbáziskezelő technológiát, amely nem az Elasticsearch, vagy egyáltalán nem adott választ a kérdésre.

Az SQL alapú adatbázistechnológiák változója egyébként vélhetően hasonló okból nem lett szignifikáns, mint a programnyelvek: a legtöbb munkahelyen nagy valószínűséggel már eleve bemeneti követelmény az SQL – ismeret. A vágyott adatbázis – ismeretek változói kapcsán („*Desire*”) a bent maradt változók ugyanezt a rendszert követik. Ez alapján ezen változók közül például az *ElasticsearchWorkedWith* végleges (specifikált, nemlineáris tagokkal kiegészített) modellben található együttthatója ~ 1,219 volt, amely azt jelenti, hogy egy olyan fejlesztő, aki munkája során dolgozott már az Elasticsearchel, átlagosan kb. 22% -al keres többet azoknál, akik még nem, minden egyéb tényező változatlansága mellett.

8.1.6. Ismert keretrendszerek

Az ismert keretrendszereket tekintve már több dummy változó is szignifikánsnak mutatkozott: az Angularra, Djangora és jQueryre vonatkozó „*WorkedWith*” ismérvek (ezek közül az első kettő pozitív, míg az utolsó negatív előjelű együtthatóval). Ugyanezen technológiák „*Desire*” párjait tekintve csak a *DjangoDesire* volt szignifikáns, viszont a konzisztencia miatt a másik kettő keretrendszer „*Desire*” dummy változói is bent maradtak a modellben.

A jQuery ismeretének bércsökkenő hatását az magyarázhatja, hogy ez már egy alapvetően elavultabbnak tekinthető technológia. Az elemzés alapján ezen keretrendszerrel dolgozó informatikusok a végső modell szerint minden más tényező változatlanságát feltételezve várhatóan kb. 11% -kal keresnek kevesebbet az IT szektor már frameworköket használó szakembereinél. Hasonló feltételezések mellett, azon szakemberek, akik ezt a technológiát szeretnék használni vagy tanulni a későbbiekben, átlagosan kb. 6,7% -al kevesebb jövedelemmel rendelkeznek, mint azok, akik nem (10% -os szignifikanciaszint mellett). Ezt a technológiát még 2006 - ban fejlesztettek ki azzal a céllal, hogy a különböző weboldalak közti korábban fennálló kompatibilitási problémákat áthidalja, például a DOM (Document Object Model) objektumok lekérdezését, illetve az egyes elemek különböző böngészőkben történő vizualizációját. Ennek oka az volt, hogy egyes JavaScript motorok máshogyan működtek például az Internet Explorerben vagy a Safariban. Ugyanakkor a JavaScript könyvtárak fejlesztésével és a webböngészők egységesítésével ma már kevésbé van szükség egy jQuery-hez hasonló áthidaló megoldásra. Ennek okáért például 2018 – ban az egyik legnagyobb fejlesztési verziókövető és kollaborációs felület, a GitHub törölte a jQuery -től való függőségeit a weboldalainak frontend felületeinek kódjaiból és más, dinamikusabb és modernebb megoldásokkal váltotta azt ki (github.blog, 2018).

8.1.7. Operációs rendszerek

Az operációs rendszerek közül egyedül a macOS -re vonatkozó változó volt rendkívül szignifikáns minden szokásos szignifikanciaszinten, így a Windows és Linux változóit elhagytam. Mivel a Mac használatával kapcsolatos ismerv pozitív modellbéli együtthatóval igen kiugróan szignifikáns volt, így a későbbiekben a dekompozíció során is megvizsgáltam, hogy a Macet használók és nem használók körében milyen további, bérre gyakorolt hatáskülönbségek lehetnek. Összességében a Macet használó fejlesztők bruttó jövedelme minden más tényező változatlansága mellett átlagosan kb. 32% -al volt több azoknál, akik nem

Mac-et használnak. A Mac jelentősége a szoftverfejlesztésben véleményem szerint kisebb, mint például a MacSources (2022) szerint a grafikában vagy multimédiás anyagok készítésében, mivel úgy gondolom, hogy az egyes fejlesztői környezetek tulajdonosai megoldásaikat minden operációs rendszerrel kompatibilis formában értékesítik. A témát a dekompozíciós elemzés során a 10.4. fejezetben részletesebben is kifejtem.

8.1.8. Betöltött pozíció, munkahely mérete

A betöltött pozíciók közül szinte az összeset szelektálnom kellett a szignifikancia hiánya miatt. Egyedül az adatbázis - és rendszeradminisztrátori pozíciót betöltők dummy változója volt szignifikáns, ezen részcsoporthoz modellbeli együtthatója $\sim 0,87$ volt, tehát várhatóan kb. 13% -kal keres kevesebbet egy ilyen állást betöltő informatikus Magyarországon a más pozíciójú IT szakemberekhez képest, feltételezve a többi tényező változatlanságát. Az eredmény úgy is értelmezhető, hogy a magyar informatikai munkaerőpiacon végső soron mégsem akkora jelentőségű a jövedelem szerinti szóródás, mint ahogy azt előzetesen vártam. Eszerint bár vannak eltérések (ahogy azt a Hays Salary Guide-ban is láthattuk az 2.2.2.1. fejezetben), ugyanakkor ezeknek összességében nézve nincs jelentősége bérezés szempontjából, mert minden pozíció egységesen magas béreket kap. Egyedül az adatbázis – rendszeradminisztrátori állásokat van értelme megkülönböztetni, ugyanis az ilyen munkakört betöltő informatikusok lényegesen kevesebbet keresnek a többiekénél. Az eredmény olyan szempontból nem meglepő, hogy feltehetőleg egy ilyen állás betöltéséhez kevesebb programozási vagy egyéb speciális technológiai ismeretek szükségesek. A tapasztalt jelenség, hogy a konkrét betöltött pozíciók kevésbé jelentősek bérezés szempontjából továbbá hasonlóan is magyarázható, mint a programnyelvek esete: hogy mivel az adatbázisban csak informatikusok voltak, így a szóródásuk egymáshoz képest kevésbé szignifikáns, mintha egyéb szektorok munkavállalóit hasonlítottuk volna hozzájuk, ahogyan azt korábban a 2. ábrán is tettem.

A munkahely méretét (fő) tekintve a referencia kategória a több mint 10 ezer főt számláló multinacionális cégek voltak. A végső modellben az ismeretlen, illetve kisebb, mint 10 és 10 – 100 fő közötti vállalatok dummy változóinak regressziós paraméterei bírtak magas szignifikanciaszinttel. Az eredmények szerint a kevesebb mint 10 főt számláló (véltetően tech start – up) cégek informatikusai kb. 22% -kal keresnek kevesebbet a több mint 10 ezer embert foglalkoztató multinacionális cégek dolgozóitól, míg a 10 – 100 munkavállalót számláló KKV-kban dolgozó fejlesztők kb 18% -kal kapnak kisebb bért, mint a referencia kategória (minden egyéb tényező változatlansága mellett).

Az eredményekre több magyarázat is létezik. Véleményem szerint a 6.2. fejezetben taglaltak szerint előfordulhat egy olyan mintavétel módja miatti torzítás az adatokban, hogy egyes figyelmetlen kitöltők a juttatásaikat, bónuszait is beleszámolták a havi bérükbe (a kérdőívben csak a bruttó bért volt szükséges megadni). Úgy gondolom, a multinacionális vállalatok a fluktuációk csökkentése érdekében számos béren felüli juttatást, bónuszt kínálnak a munkavállalóknak, összességében többet, mint egy – sokszor kisebb költségvetésből gazdálkodó – KKV vagy start – up. További indok lehet még, hogy az életciklusuk elején járó kisebb start – upok vagy kisvállalkozások vevőköre még nem kiforrott ezért az elkönyvelt bevételek így még nagymértékben ingadoznak és bizonytalanabbak, amelyek a beosztottak bérére is negatív hatással lehetnek. Emellett viszont érdekes lehet a kérdéskör további vizsgálata, hogy a magyar piacon pontosan miben tér el egy start – upban vagy KKV – ban dolgozó informatikus munkája és az azért kapott juttatások a nagyvállalatoknál alkalmazottakétól.

8.1.9. Munkaviszony típusa

A munkaviszony esetén a teljes állásban (heti 40 óra) foglalkoztatottak voltak a referencia kategória. Nem meglepő módon a részmunkaidőben dolgozók minden további tényező változatlansága mellett átlagosan kb. 32% -al kevesebb bért kapnak a teljesidőben foglalkoztatottaknál. Érdekes eredmény viszont, hogy a szerződéses munkaviszonyban dolgozók ugyanezek feltételezések mellett várhatóan kb. 31% -al magasabb jövedelemre számíthatnak egy teljesidejű munkavállalónál. A szerződéses keretek közt dolgozó kitöltők zöme valószínűleg a KATA (Kisadózó Vállalkozók Tételes Adója) konstrukcióban dolgozó egyéni vállalkozónak bejelentett informatikusok voltak, akik sok esetben a saját munkahelyüknek számláztak. A KATA adónem ezen szakembereknek nagyon kedvező feltételeket biztosított, mivel havonta fix 50 ezer Ft adófizetésre kötelezte az adóalanyokat. A szabályozó törvényt ugyanakkor 2022 júliusában módosították, így a 2022. évi, a kisadózó vállalkozók tételes adójáról szóló XIII. törvény értelmében a vállalatoknak való számlázás lehetősége gyakorlatilag megszűnt. Ez azonban a 2022 -es évi Stack Overflow kérdőívet még nem befolyásolta, mert a mintavétel az év első felében történt ([survey.stackoverflow.co, 2022](https://survey.stackoverflow.co/2022)).

8.1.10. Stack Overflow használati szokások

A Stack Overflow használati szokásokra vonatkozó ismérvek közül egyedül a *SOParticipate* változó maradt bent a modellben, azaz, hogy az illető milyen gyakorisággal vesz részt a weboldal fórumbeszélgetéseiben. Az ismérvek szinte minden ismérvkimenete szignifikáns volt, ezek közül is leginkább az, amelyik arra vonatkozott, hogy az illető heti néhány

alkalommal kommentel vagy tesz fel kérdést. Emellett az összes ismervkimenet pozitív regressziós együtthatóval rendelkezett.

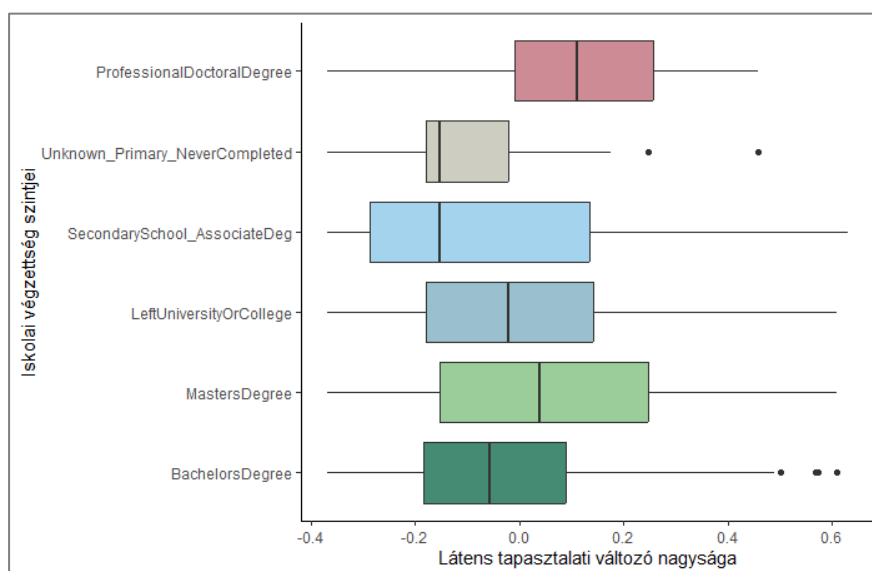
Azt gondolom, hogy munkáltatói szemszögből kevésbé van annak jelentősége, hogy egy fejlesztő aktív tagja – e a Stack Overflow közösségének, így értelemszerűen nem közvetlenül azért keres egy informatikus többet, mert többet kommentel vagy kérdez a Stack Overflow fórumán. Ennél a kérdéskörnél is felmerül a módszertani mellékletben tisztázott kauzalitás problémája, hogy vélhetően ebben a mintában magukat aktív Stack Overflow felhasználónak valló kitöltők nagy valószínűséggel rendelkeznek olyan egyéb tulajdonságokkal is, amelyeknek bérnövelő hatása van.

Ha például a fejlesztők jellemére koncentrálnunk, előfordulhat, hogy aki rendszeresen használja kérdésfeltevésre a Stack Overflowt, hajlamosabb a kollaborációra és nyitott arra, hogy a mások által javasolt, adott esetben optimálisabban működő kódokat vagy programcsomagokat kipróbálja, ezáltal pedig a munkában nyújtott teljesítménye is javul. Az is elképzelhető, hogy a fórumon gyakran kérdezők az egyes szituációkban gyorsabban kapnak segítséget, mintha kollégáikat kérdeznék. Munkám során nekem is volt alkalmam a fórum működését kipróbálni, ahol egy olyan kódrészletet csatoltam, amely hibára futott, de nem értettem miért. A kérdésem közzététele után 10 percen belül használható választ kaptam, amely megoldotta a problémám. Emellett valószínűsíthető, hogy a gyakran kommentelő felhasználók pedig olyan fejlesztők, akik már a jártasabbak a szakmában, ezáltal seniorabb pozíciókban dolgozhatnak, amely szintén magasabb kompenzációval jár.

8.2. Nem – metrikus többdimenziós skálázás

Dimenziócsökkentést nemcsak szelekcióval, hanem olyan eljárásokkal is végrehajthatunk, amelyek során több, hasonló viselkedésű változó hatásait sűrítjük egy változóba. A *YearsCoding* és *YearsCodingProf* változók esetén én ezt a módszert alkalmaztam. Ennek oka ezen két változó káros multikollinearitása volt, amely egyértelműen mutatta a két ismerv közötti szoros kapcsolatot, ugyanakkor az egyes változókra számolt parciális t – próbák p - értékeit pedig torzította. Kézenfekvő volt tehát ezt a két változót egybevonni, amelynek eredményeként egy látens „tapasztalati” változót hoztam létre. Mivel a két ismerv kategorikusan volt adott, így az ordinális mérési skála miatt a főkomponens - elemzés helyett a nem – metrikus többdimenziós skálázással hajtottam végre az egyesítést. Az eredményül kapott látens változó és az eredeti *YearsCoding* és *YearsCodingProf* változók kapcsolatát megnézve látható, hogy előbbinél körülbelül 8 - 10, míg utóbbinál 6 – 8 évvel bezárólag a látens változó értéke negatív,

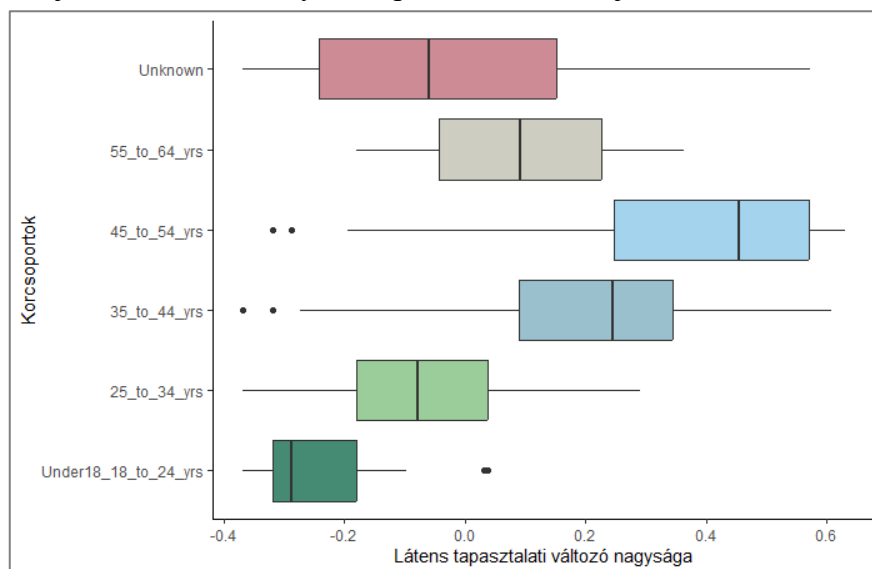
majd az évek növekedésével egy egyre nagyobb pozitív szám. Ezáltal elmondható, hogy a látens változó jelentéstartalmában pozitív korrelációban áll a kódolással eltöltött évek számával, tehát ha értéke nő, az várhatóan magasabb szintű programozási tapasztalatot jelent. A látens „tapasztalati” változó bevonása egyébként az információs kritériumok és a Wald – teszt szerint is javított a modellen, a regressziós együtthatója pedig minden szokásos szignifikanciaszint mellett szignifikáns volt.



10. ábra: Az iskolai végzettség és a látens kódolási tapasztalatokat reprezentáló változó kapcsolata.

Más változókkal összevetve az mondható el, hogy iskolai végzettség szerinti bontásban mediánok szerint a doktori és mesterfokozattal rendelkezők látens változója a legmagasabb, ugyanakkor a dobozok ezekben a részkategóriákban a legszélesebbek (lásd: 10. ábra), tehát elképzelhető, hogy többen vannak ezekben a kategóriákban, akik csak egyetemen kezdtek el kódolni. Összességében tehát akiknek magasabb iskolai végzettségük van, átlagosan nézve több kódolási tapasztalattal rendelkeznek. Ugyanakkor erre az eredményre az is hatással lehet, hogy akiknek magasabb iskolai végzettsége van, feltehetőleg idősebbek azoknál, akiknek alacsonyabb, így értelemszerűen több év kódolási tapasztalatuk van a fiatalabbaknál.

Vélhetőleg az érettségivel vagy középfokú szakképesítéssel rendelkezőknek ebből az okból kifolyólag van jellemzően alacsonyabb tapasztalati változójuk.



11. ábra: Az életkor és a látens kódolási tapasztalatokat reprezentáló változó kapcsolata.

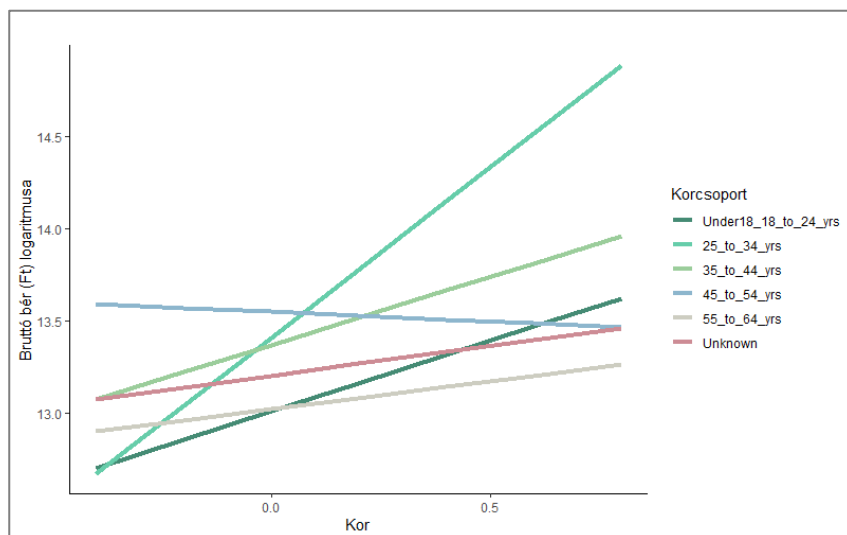
Érdekes tendenciát mutat továbbá a látens változó és életkor kapcsolatának vizsgálata. A dobozábrákat megnézve kivehető a kor nemlineáris hatása a tapasztalatra, amely körülbelül 55 éves korig folyamatosan növekszik, majd ezután csökken. Ezen a ponton ugyanakkor érdemes megjegyezni, hogy az 55 – 64 év közöttiek száma az adatbázisban rendkívül alacsony (kevesebb mint 5 fő) volt, így az erre a részcsoporthoz vonatkozó eredmények nem feltétlenül reprezentatívak a magyar informatikai szakemberekre nézve.

A 11. ábrán szembeűnő továbbá, hogy a legszélesen doboza annak a részkategóriának van, ahol a válaszadók nem adták meg a korukat. Valószínűsíthető, hogy ez a csoport számos korcsoporthoz magába foglal, így a tapasztalat szerinti nagymértékű szóródás ennek köszönhető.

8.3. Specifikáció, nemlineáris tagok bevonása

A nem – metrikus többdimenziós skálázás után számos lehetséges interakciót és kvadratis tagot kipróbáltam, melyekről – hasonlóan a szelekcióhoz – az információs kritériumok és a Wald – teszt p – értéke alapján döntöttem. Végezetül a kor és a „tapasztalat” látens változójának interakciója volt az egyetlen, amely mind az információs kritériumok és a Wald – teszt alapján is javított a modellen. A RESET teszt eredménye már az interakció bevonása előtt is H_0 volt, de a p – érték az interakció bevonása után méginkább megnőtt, jelezve, hogy a modell specifikációja tovább javult ezzel a taggal. A két változó interakciójának jelentése: A kor bérre gyakorolt marginális hatására hatással van a kódolási tapasztalat szintje és fordítva, a kódolási tapasztalat növelésével a bérek korcsoporthoz szerint eltérő meredekséggel nőnek vagy éppen

csökkennek. Ez konkrétan értelmezve azt jelenti, hogy a magyar informatikusok fizetései kor szerint nézve a tapasztalat függvényében eltérő mértékben nőnek vagy csökkennek. A marginális hatásokat a 12. ábrán szemléltettem, ahol gyakorlatilag 95% -os szignifikanciaszint mellett azok a kor szerinti kétváltozós regressziós egyenesek vannak ábrázolva, ahol csak a látens „tapasztalat” változóval becsüljük a jövedelmet.



12. ábra: Az életkor és tapasztalat marginális hatásainak szemléltetése.

Az interakció bevonása után a modellben található kor – tapasztalat interakciós tagok együttthatóinak jelentése: az eredetileg kijelölt, 24 év alatti referenciacsoporthoz képest átlagosan milyen mértékben csökkenti vagy növeli a fejlesztők jövedelmét egységnyi tapasztalatváltozás. A 12. ábrán látható, hogy a kor és tapasztalat nem teljesen lineáris kapcsolatban áll a bérrel, mivel a görbék meredeksége korcsoportonként jelentősen különbözik és néhol negatív irányú, illetve összességében teljesen más tengelymetszetekből induló egyeneseket látunk. Ezáltal elmondható, hogy egy egység kódolási tapasztalatváltozás hatására korcsoportonként eltér a jövedelem átlagos növekedése, nem mondhatjuk, hogy minden csoportban azonos hozzáadott értéke van a programozási tudásnak. Ez a jelenség nagyon látványos például a 25 – 34 éves korosztály esetén, ahol egységnyi tapasztalatváltozásnak átlagosan sokkal nagyobb keresetnövekedés az eredménye.

Az interakciós tagok bevonása után a modellbéli együttthatók előjele 35 év felett negatívvá vált. Ez azt jelenti, hogy 35 éves kor felett a meglévő tapasztalat hasznossága csökken és a 24 év alatti referenciacsoporthoz képest a 35 éves kor feletti informatikusok bérezési görbéje ellapul, ha csak a tapasztalat hatását vesszük figyelembe. A magyarázat erre az eredményre az lehet, hogy az évről évre történő technológiai változások következtében az idő előrehaladtával a fejlesztők egy – egy egyre inkább elavuló technológiában lesznek tapasztaltabbak (feltételezve,

hogy nem képzik tovább magukat más technológiákban). Az eredmény egybevág a 2.2.1. fejezetben tárgyaltakkal, hogy vélhetőleg az ilyen korú vagy ennél idősebb fejlesztők seniorabb pozíciókban dolgoznak és kevesebb operatív fejlesztési és több vezetési – koordinálási feladatot látnak el.

A látens tapasztaltot reprezentáló változó regressziós együtthatójából a 35 – 44 éves korosztály interakciós tagjának együtthatóját kivonva egyébként a visszakapott 35 – 44 éves csoportra vonatkozó meredekség továbbra is pozitív volt, amely a 12. ábrán is megjelenik. Ez alapján nem mondható, hogy ez a korosztály a fiatalabb referenciacsoporthoz képest egységnyi tapasztalatnövekedéssel már kevesebbet kereshet, viszont bérezésük esetén már kevésbé lehet fontos szempont a kódolási tapasztalat.

Az eredmény ugyanakkor számos további kutatási kérdést vet fel. Érdekes kérdéskör például, hogy vajon mekkora a piac nyomása az idősebb fejlesztőkön, amely az önképzésre ösztönzi őket. A munkáltató szempontjából sem elhanyagolandó a téma, hogy ezt a továbbképzést milyen eszközökkel tudja hatékonyan támogatni, hogy a cégen belüli senior kompetenciát is naprakészen tartsa vagy adott esetben jobban megéri – e egy vállalatnak inkább mindig a frissebb tudással rendelkező juniorabb munkaerőt felvenni, a tapasztaltabb kollégákat pedig vezetői – koordinátori feladatokkal ellátni. A naprakész és stabil IT kompetencia fejlesztése és megtartása a 2.1. fejezetben bemutatott erős munkaerőhiány miatt komoly feladat lehet a vállalatok számára a versenyképesség kapcsán.

8.4. Blinder – Oaxaca dekompozíció

A kéttagú dekompozíciós modellezést többfajta szempont szerint is elvégeztem, amelyek alapjai az előzetes kérdésfeltevéseim, illetve a modellépítés közben kapott részeredmények voltak. Fontos megjegyezni, hogy a dekompozíciót megelőzően az egyéb fejlesztői pozíciót magába foglaló *OtherDevType* és az 55 - 64 éves korosztály ismérvkimeneteit a saját változójuk referencia kategóriába kellett soroljam, mert a kis elemszámuk miatt a dekompozíciós modell nem tudott megfelelően standard hibát kalkulálni ezen két ismérvkimenet esetén.

8.4.1. A diploma hatásainak vizsgálata

A diploma hatásainak szétbontásához bevezettem egy *HasDegree* technikai változót, amelynek értéke 1 volt, amennyiben az illető rendelkezett Bsc, Msc vagy PhD fokozatú diplomával. Minden egyéb esetben (hiányzó érték esetén is) a változó a 0 értéket vette fel. Az ismérv R faktorváltozónak alakítása során a diplomával nem rendelkezők részcsoporthoz lett a referencia kategória. Az eredmények összesítve a következők lettek:

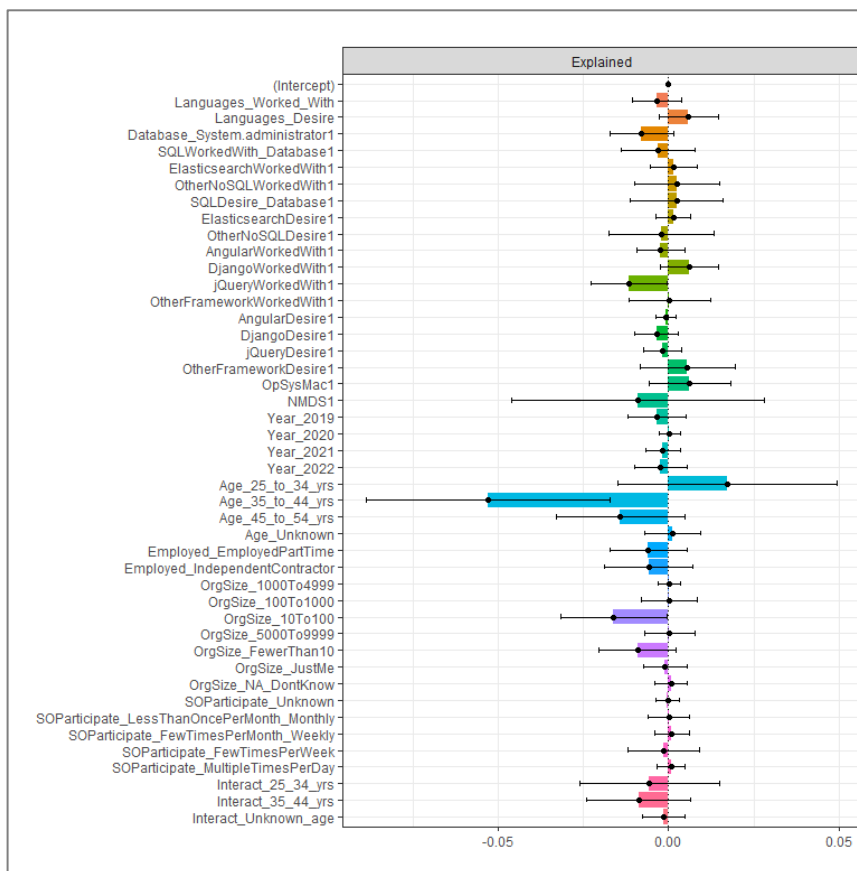
Részcsoporthoz átlagának bruttó bérkülönbsége	Szignifikáns összetételhatások	Szignifikáns összetételhatások mértéke a bérre	Szignifikáns részhatások	Szignifikáns részhatások mértéke a bérre
~ -157500 Ft	Dolgozott -e jQueryvel?	~ -1,14%	SQL -el szeretne a későbbiekben foglalkozni?	~ +22%
	35 - 44 éves -e?	~ -5,15%	Egyéb NoSQL nyelvekkel szeretne -e a későbbiekben foglalkozni?	~ -26%
	10 - 100 fős cégnél dolgozik -e?	~ -1,6%		

3. táblázat: A diplomával rendelkezők és nem rendelkezők dekompozíciós vizsgálatának eredményei.

A 3. táblázat a dekompozíció eredményeit mindig a referenciacsoportéhoz viszonyítva mutatja be. Jelen esetben például a bruttó bérkülönbség jelentése, hogy a diploma nélküli kitöltők részcsoporthoz átlagbéré kb. bruttó 157500 forinttal volt kevesebb a diplomával rendelkezők csoportjánál. A különbség – ahogy azt már korábban említettem – a kéttagú felbontás alapján összetétel – és részhatás különbségekre bomlik, melyekből előbbi a magyarázóváltozók által megmagyarázott, utóbbi a megmagyarázatlan/nem megfigyelt hatások következtében bekövetkező differencia.

8.4.1.1. Összetételhatások

Az összetételhatást a következőképp ábrázoltam:



13. ábra: A diploma hatásának vizsgálata a bérekre, összetételhatás.

A fekete vonalakkal jelölt konfidenciaintervallumok alapján számos változó nem szignifikáns összetételhatással bír 5% -os szignifikanciaszint mellett, mivel 95% -os konfidenciaintervallumaik a nullát is magukba foglalják, tehát nemcsak a hatásuk mértéke nem határozható meg pontosan, hanem még az sem, hogy a bért várhatóan pozitív vagy negatív irányba tudják módosítani. A szignifikánsak közül a következőket állapíthatjuk meg: A jQuery -vel dolgozók, a 35 – 44 éves korosztály, a és a 10 és 100 fős vállalatoknál dolgozók átlagosan kevesebbet keresnek és a nem diplomások körében többen fordulnak elő, ezzel csökkentve a diploma nélküliek csoportátlagát.

A jQuery elavultságát szintén már korábban tisztáztam. Ebből kifolyólag nem meglepő az eredmény, hogy az ezen technológiával dolgozó informatikusok várhatóan kevesebbet keresnek azoknál, akik más keretrendszert használnak munkájuk során. A jQuery változójának modellbeli együtthatójának negatív előjele szintén ezt támasztja alá. A 3. táblázatban összesített eredmények alapján az eleve kevesebbet kereső jQuery -vel dolgozók a diploma nélküliek

körében többen fordulnak elő, ezzel átlagosan 1,14% -kal rontva az egyetemi képzést nem végzett kitöltők csoportjának átlagjövedelmét.

A korra már a marginális hatások kapcsán korábban is kitértem, ahol megfigyelhető volt, hogy a 35 – 44 évesek bérezési görbéje ellapul és például a fiatalabb, 24 év alatti korosztály adott esetben megelőzheti ezeket az informatikusokat kereset szempontjából. A 3. táblázat szerint a diploma nélküliek körében jellemzőbben fordulnak elő a kevesebb jövedelemmel rendelkező 35 – 44 évesek, így a felsőoktatást otthagyló vagy nem végzett récsoprt átlagbére kb. 5,15% -kal romlik. Ez más aspektusból is megerősítette a korábban tett megállapításomat, hogy minden egyéb tényező változatlansága mellett, feltételezve, hogy az IT szektor dolgozói nem képzik magukat tovább, egy bizonyos pont után átlagosan kevesebbet kereshetnek, mint a fiatalabb, frissebb technológiákat ismerő korosztály.

A 10 – 100 főt számláló vállalatok a KKV szektorba tartoznak, ahol a cég költségkeretei sokszor szűkebbek, mint a nagyvállalatoké, ezért feltehetően az alacsonyabb átlagbér ebből a tényezőből is adódik. Azon informatikusok, akik ilyen vállalatoknál dolgoznak, jellemzően kisebb jövedelemmel rendelkeznek és a diplomával nem rendelkezők körében többen fordulnak elő, ezáltal kb. 1,6 %-kal csökkentve ezen részcsoport bruttó átlagjövedelmét.

8.4.1.2. Részhatások

A részhatáskülönbségek ábrázolása – hasonló formában, mint az összetételhatásokat bemutató 13. ábra – megtalálható a 14.2. fejezet ábramellékletei között. Ezen hatások között is hasonlóan az összetételhatások esetéhez azok tekinthetők 5% -os szignifikanciaszint mellett szignifikánsnak, amelyek 95% -os konfidenciaintervalluma nem tartalmazta a nullát. Emellett a részhatások pontos értékei megtalálhatók a 3. táblázatban is.

Az eredmények szerint SQL szabványon alapuló relációs sémás adatbáziskezelő programok használatát elsajátítani akarók valamilyen egyéb tulajdonságuk vagy nem megfigyelt hatás következtében többet keresnek a diplomások körében a diploma nélkülieknél, ezáltal a diplomával rendelkező részcsoport átlagbérét kb. 22% -kal növelik (a 3. táblázatban összefoglaltak alapján). Ugyanakkor az egyéb NoSQL nyelveket megismerni vágyók átlagfizetése szintén további nem megfigyelt hatások miatt alacsonyabb a diplomával nem rendelkezők körében a diplomásokénál, emiatt ennek a részcsoportnak a bruttó átlagbére kb. 26% -kal kevesebb a diplomával rendelkező kitöltők csoportjának átlagánál.

Az első állításra az egyetemi képzések struktúrájában találtam egy lehetséges választ. Általánosságban elmondható, hogy az informatikai képzést biztosító szakokon kötelező tárgyként szerepel az adatbázis rendszerek ismerete, amelynek tananyaga elsősorban a hagyományos relációs sémák logikája és az ilyen környezetben történő lekérdezésekhez használt SQL nyelv, míg a NoSQL módszertanok kevésbé szerepelnek az informatikai képzések tantárgyi programjaiban (BME-VIK, 2015; ELTE-IK, 2018a). Aki tehát egyetemi szinten végez el egy informatikai képzést, várhatóan több jártasságot és rálátást szerez a témában, illetve – akár a kötelező szakmai gyakorlat idején – olyan cégeknél helyezkedik el, ahol esetlegesen elvárt hasonló kompetencia. Éppen ezért érthető, hogyha valaki ilyen jellegű állással rendelkezik, a továbbiakban is szeretné tudását bővíteni a témában.

Az állítást az is alátámasztja, hogy Magyarországon valószínűleg még nem nagy a kereslet az ilyen technológiákat ismerő szakemberekre a hagyományos SQL alapú adatbáziskezelő kompetenciához képest. Ennek oka részben az ország technológiai szempontból perifériás helyzete, részben pedig a NoSQL, mint megoldás sajátossága, hogy feltételezésem szerint nem feltétlenül szükséges minden helyzetben alkalmazni, amikor a probléma egy egyszerűbb relációs sémás adatbázissal is megoldható.

A végső modellben kitüntetett Elasticsearch a módszertani mellékletben tárgyaltak alapján például JSON formátumhoz hasonló, nem strukturált adatok beágyazott lekérdezésére alkalmas keresőmotor. Az ELK Stack megoldás², amelynek alapját ez a technológia képezi, meglátásaim szerint alapvetően olyan vállalatok számára alkalmas, amelyek például IoT tevékenységekkel operálnak, ahol szenzoros adatfeldolgozás (is) folyik, ezért nagy mennyiségű (több ezer sor/perc) strukturálatlan adat keletkezik, amelyeket üzleti – üzletfolytonossági szempontból elsősorban a monitoring és anomáliakeresés céljából érdemes ábrázolni. Úgy gondolom, a magyar piacon működő, analitikai szempontból legalább közepesen érett vállalatok inkább hagyományos adattárház – architektúrát használnak, amely az adatokat relációs sémában tárolja és amelyhez standard konnektorokon keresztül számos elemzéshez, riportinghoz ... stb. használható eszköz csatlakoztatható. Egy további, előzőből következő indok lehet még, hogy mivel a NoSQL megoldások speciálisabb, nem strukturált adatok kezelésére alkalmasabbak, bevezetésük és üzemeltetésük pénzt és erőforrást tekintve is költségesebb lehet. Ugyanakkor mivel ezek a hatások a magyarázóváltozók által megmagyarázatlan komponensbe kerültek, így

² Az ELK Stack csomag részletesebb bemutatása az Elasticsearch -el egyetemben a 15.1. módszertani mellékletben található.

számos további, itt nem említett tényező lehet, amely befolyásolja az SQL – NoSQL módszertanok által tapasztalt bérkülönbszeteket részhatás szempontjából.

A második állításra adott lehetséges magyarázatom, hogy a NoSQL technológiákat tanulni akaró, de diplomával nem rendelkező csoport kevesebbet keres, részben kapcsolódik az előzőekben felvázoltakhoz. Aki nem végez egyetemi képzést, bár lehet, hogy az SQL nyelvet önmagában képes autodidakta módon elsajátítani, feltételezhető, hogy a relációs sémák architektúrájáról és a magyar piaci keresletről kevesebb ismeretet szerez, mert nem szükséges kötelező szakmai gyakorlatot végeznie, amely ösztönző hatással bírhat a piac feltérképezésére. Számos szerző, köztük El-Seoud, El-Sofany, Abdelfattah és Mohamed (2017) is megállapították, hogy a big data forradalmában valóban felmerült a szükség olyan adattárolási koncepciókra, ahol az adat a relációs sémáknál kötetlenebbül, objektumorientáltan lekérdezhető és skálázható módon tárolható. A NoSQL keresőmotorok erre próbáltak választ adni, ezért a big data kifejezés mellett ezek is népszerűvé, divatossá váltak.

A korábbi érvelésem alapján ezen kérdéskörben is hasonlóan úgy gondolom, hogy a magyar piacon működő vállalatok többsége technológiailag kevésbé érett vagy a főtevékenységéből adódóan nem is tarthat igényt egy NoSQL megoldásra. Ugyanakkor véleményem szerint azok, akik nem végeztek informatikai egyetemi képzést, ettől függetlenül a külföldi példák láttán, valamint a hiányosabb piaci ismeretek miatt úgy gondolhatják, hogy a NoSQL ismeretekkel egyedibb – és ezzel kifizetődőbb – tudásra tesznek szert a magyar piacon is, mint egy hagyományos relációs sémás logikát követő SQL nyelvvel. Ezek – ahogy már korábban említettem – viszont nagy valószínűséggel olyan speciális és magyar piacon még viszonylag ritka esetekben megfelelő megoldások, amikor nagy mennyiségű, nem strukturált adat keletkezik. Természetesen, mivel ezen hatás is a megmagyarázatlan hatások csoportjába került, itt szintén igaz a megállapítás, hogy számos más lehetséges magyarázat is van a tapasztalt jelenségre, amely sok más kutatási kérdést is felvet.

8.4.2. Alapszakos és mesterdiploma közti különbségek

A diploma fontosságán túl, arra is kíváncsi voltam, hogy vajon a piac mennyire honorálja egy mesterdiploma meglétét az alapszakhoz képest. A már korábban felvázolt munkaerőpiaci hiány és a kötelező szakmai gyakorlat miatt az alapszak végére számos hallgató talál stabil munkahelyet, emiatt pedig feltehetőleg kevésbé érezhetik szükségességét még egy mesterszak elvégzésének. Ezt a Felvi.hu idej, már korábban bemutatott felsőoktatási jelentkezési adatai is alátámasztják. Ebből az okból kifolyólag kíváncsi voltam, hogy amennyiben valaki az alapszak

befejeztével még továbbtanul, annak milyen mértékű pozitív hozadéka van később a keresetekre nézve.

Ehhez a dekompozícióhoz csak azokra a kitöltőkre szűrtem le az adatbázist, akiknek alap – vagy mesterszakos diplomája van. Az előzőekhez hasonlóan itt is bevezettem egy technikai változót, amely Bsc diploma esetén 0, Msc végzettség esetén pedig 1 értéket vesz fel. A referencia kategória az alapszakot végzettek csoportja volt. A szűrés után megmaradó 814 kitöltőből 432 rendelkezett alapszakos és 382 pedig mesterszakos oklevéllel. Az eredményeket a 4. táblázatban összesítettem, hasonlóan, mint a diplomával rendelkező és nem rendelkező csoport összehasonlítása esetén. A teljes, rész – és összetételhatásokat szemléltető ábrák a 15.2. mellékletben találhatók.

Részcsoporthoz tartozók átlagának bruttó bérkülönbsége	Szignifikáns összetételhatások	Szignifikáns összetételhatások mértéke a bérre	Szignifikáns részhatások	Szignifikáns részhatások mértéke a bérre
~ -198400 Ft	35 - 44 éves -e?	~ -5,35%	Szeretne egyéb frameworköket tanulni?	~ -19,41%
	Részmunkaidőben dolgozik -e?	~ -1,23%		
	25 - 34 éves korosztály interakciós tagja a látens "tapasztalat" változóval	~ -2,53%		

4. táblázat: Az alapszakos és mesterszakos diplomával rendelkező kitöltők dekompozíciós vizsgálatának eredményei.

Összességében a két csoport átlaga közti különbség a mesterszakosoknak kedvez. Pusztán az átlagok különbségeit összevetve pedig az látható, hogy a mesterszakos – alapszakos különbség nagyobb mértékű a megfigyelt adatok körében, mint a diplomás – nem diplomások csoportátlagának eltérése, előbbi átlagosan kb. bruttó 198400 Ft, míg utóbbi kb. bruttó 157000 Ft.

8.4.2.1. Összetételhatások

Az összetételhatást vizsgálva – hasonlóan az előzőekhez – számos változó összetételhatása a konfidenciaintervallumokból adódóan nem szignifikáns 5% -on, viszont a 35 – 44 éves korcsoport hatása itt is megjelenik. A 4. táblázat és a 8.4.1. fejezetben tisztázottak alapján az mondható el, hogy ezen korcsoport tagjai alapvetően alacsonyabb jövedelemmel rendelkeznek és az alapszakos diplomával rendelkező csoportban többen vannak, így csökkentve ezen csoport átlagjövedelmét kb. 5,35% -kal.

A teljes idejű munkavállalóknál eleve kevesebbet kereső részmunkaidőben foglalkoztatottak is magasabb összetételben fordulnak elő az alapszakokkal rendelkezők körében, amely tovább csökkenti ennek a részcsoporthoz tartozók átlagjövedelmét kb 1,23% -kal. Véleményem szerint ennek

az is lehet az oka, hogy azok is az alapszakos diplomát jelölhették meg legmagasabb iskolai végzettségnek, akiknek folyamatban lévő mesterszakos tanulmányaik vannak, de még nem szerezték meg a mesterdiplomát és emellett esetleg részmunkaidőben dolgoznak. Mivel az adatbázisban nem volt arra vonatkozó adat, hogy ki milyen képzési formában végezte el a felsőoktatási képzését, így nem lehetett elkülöníteni a nappali – levelező – esti munkarend hatását a kérdéskör vizsgálatánál. Ugyanakkor az informatikai mesterszakokat indító intézményeknek mindenképp szembe kell nézniük az IT szektorban fennálló munkaerőhiány elszívó hatásával, amelyre úgy gondolom, válasz lehet például a munkabarát képzési rend vagy akár duális informatikai képzések kialakítása. Ez a kérdéskör számos további oktatásszervezési problémát is felvet, amelyek jelenleg nem állnak dolgozatom fókuszában.

A 25 - 34 éves korcsoportban a látens változó összetételhatása negatívabb, mint a 18 év alatti/18 – 24 év közötti referencia korcsoportban. Ennek jelentése, hogy az alapszakkal rendelkezők közt kevesebb 25 - 34 éves van, akiknek a nagyobb tapasztalat miatt nagyobb bér járhat, ezért ez a jelenség pedig csökkenti vagy kevésbé növeli átlagosan az alapszakot végzett csoport átlagjövedelmét a mesterszakot végzettekéhez képest. A jelenségre értelemszerű magyarázat lehet, hogy az alapszakkal rendelkezők vélhetően eleve fiatalabb korosztály, mivel – feltételezve, hogy az adott kitöltő nem szünetelteti a tanulmányait – hét féléves alapszakos és négy féléves mesterszakos képzéssel várhatóan leghamarabb 26 – 27 éves korban reális a mesterdiploma megszerzése. Emellett viszont a csak alapszakot végzett informatikusok részcsoportja eleve nagyobb, ezért széles skálán mozoghat a tapasztalat, hiszen ugyanúgy előfordulhatnak itt pályakezdők és olyanok is, akik esetleg egyetem előtt már dolgoztak és utána kezdtek el valamilyen felsőfokú képzést.

8.4.2.2. Részhatások

A megmagyarázatlan csoportkülönbségeket megnézve egyedül egy változó, az egyéb frameworkok megismerésével kapcsolatos ismerv tűnt szignifikánsnak. Az eredmény szerint azok, akik az Angulartól, jQuerytől és Django-tól különböző, más framework használatát szeretnék elsajátítani, valamilyen egyéb tulajdonságuk vagy nem megfigyelt hatás következtében kevesebbet keresnek alapszakos diplomával a mesterdiplomásoknál. A tapasztalt összefüggés hátterében több ok is állhat.

Egyfelől a kérdéskör vizsgálata elég komplex, mivel a magyar informatikai felsőoktatásban sem alap – sem mesterszinten nem jellemző a haladóbb webfejlesztés oktatása, legfeljebb a kötelezően választható tárgyak szintjén (ELTE-IK, 2018b). A felsőoktatásban oktatott

webfejlesztés tárgyak zömének fókuszában az alapvető HTML, CSS, Java, JavaScript, C# és C++ ismeretek oktatása áll és kevésbé ezeké a konkrét keretrendszereké, így valószínűleg a kitöltők ezeket az egyetemi ismeretekre alapozva a munkahelyen szerezték meg. Előfordulhat például, hogy aki ezen három framework valamelyikét ismeri, mesterdiplomát igénylő magasabb beosztásban dolgozik vagy már több éves, komplexebb tapasztalatokkal rendelkezik, amelyek közvetlenül növelhetik a bért. Továbbá az egyéb frameworkök csoportjában a kiindulási adatbázisban minden olyan keretrendszer bekerült, amelynek alacsony (kisebb, mint 100) volt a számossága az adatbázisban, valamint a későbbi modellszelekciós lépéseknél a szignifikancia hiánya miatt szintén soroltam ide változókat (például: React). Ebből az okból kifolyólag a sok, adott esetben alacsony számosságú, nem túl elterjedt keretrendszer egybesűrített hatása is megjelenhet a dekompozícióban, mint negatív tényező.

8.4.3. A macOS-t használók bérezése a többi operációs rendszert használókhoz képest

Az előzetes kérdésfeltevéseimben nem szerepelt a macOS-t használók vizsgálata, mivel nem gondoltam, hogy a bérré nézve szignifikáns lenne, hogy az adott fejlesztő milyen operációs rendszert használ. Ugyanakkor már a legelső, minden változót tartalmazó modellben is kiugróan szignifikáns volt az *OpSysMac* változó, melynek jelentése, hogy az illető macOS operációs rendszerben dolgozik -e vagy sem. A végső modell regressziós együtthatója $\sim 1,32$ lett, tehát átlagosan kb. 32% -al keresnek többet a Macet használók a más operációs rendszert preferálókhoz képest. A dekompozíció eredményeként általánosan elmondható, hogy 1019 olyan kitöltő volt, aki valamilyen egyéb operációs rendszert használt a munkájában és 276, aki macOS -t. Összességében nézve a két csoport átlagfizetésének különbsége szerint a Macet használók átlagosan kb. bruttó 321000 Ft -al keresnek többet a referencia kategóriánál. Az

Részcsoportok átlagainak bruttó bérkülönbsége	Szignifikáns összetételhatások	Szignifikáns összetételhatások mértéke a bérré	Szignifikáns részhatások	Szignifikáns részhatások mértéke a bérré
~ -321000 Ft	Adatbázis - rendszeradminisztrátorként dolgozik -e?	$\sim -1,62\%$	Dolgozott -e egyéb NoSQL nyelvekkel?	$\sim -32\%$
	45 - 54 éves -e?	$\sim +1,87\%$	Szeretne -e későbbiekben SQL -el dolgozni?	-16,35%
	Részmunkaidőben dolgozik -e?	$\sim -1,6\%$	Dolgozott -e Angularral?	$\sim +9,7\%$
	Szerződéses keretek közt dolgozik -e?	$\sim -3,74\%$	Heti néhány alkalommal aktív a Stack Overflow fórumán	$\sim -4,46\%$
	Havonta vagy kevesebb mint havonta aktív a Stack Overflow fórumán	$\sim -1,3\%$		

5. táblázat: A macOS -t használó és nem használó kitöltők dekompozíciós vizsgálatának eredményei.

eredményeket az 5. táblázatban összesítettem, a teljes rész – és összetételhatásokat ábrázoló diagramok pedig a 14.2. mellékletben tekinthetők meg.

8.4.3.1. Összetételhatások

Az összetételhatásokat megnézve látható, hogy ebben a dekompozícióban több változó hatása lett szignifikáns, mint az előző kettőben. Az adatbázis – és rendszeradminisztrátorok, valamint a részmunkaidőben és szerződésesen foglalkoztatottak, illetve a Stack Overflow fórumokon havi vagy kevesebb mint havi rendszerességgel aktív felhasználók általánosan nézve kevesebbet keresnek azoknál, akikre ezek a tulajdonságok nem igazak, és mivel többen vannak a nem Macet használók csoportjában, csökkentik ezt a csoportátlagot. A 45 – 54 évesek részcsoportjára viszont éppen az ellenkezők jellemzőek.

Az adatbázis – és rendszeradminisztrátorokról és a részmunkaidős foglalkoztatottakról már korábban is esett szó, ezen csoportok hatása pedig itt is megjelenik, mint negatív bérmodosító tényező a nem Macet használók tekintetében. A szerződéses keretek közt foglalkoztatott munkavállalók nemcsak például a KATA konstrukcióban foglalkoztatottak lehetnek, hanem akár a diákszövetkezeteken keresztül alkalmazott gyakornokok is, akik érthető módon lényegesen kevesebbet keresnek például egy diplomás teljes állású munkavállalónál.

A 45 – 54 évesek korosztálya általánosságban magasabb jövedelemmel rendelkezik és nagyobb gyakorisággal fordul elő a Macet használó kitöltők körében, ezért ennek a csoportnak az átlagos bére kb. 1,87% -kal magasabb. Ugyanakkor ezen korcsoport tagjai valószínűleg a korábban tisztázottak alapján azért keresnek többet, mert feltételezhetően középvezetői – vezetői pozíciókban dolgoznak. A 8.3. fejezetben tisztázottakkal összevetve viszont az tapasztalható, hogy csak a kódolási tapasztalat hatásai szerint szemlélítve ennek a korosztálynak a bérezési görbéje a legfiatalabbakéhoz képest negatív meredekségű. Ez érdekes kontrasztot mutat az itteni információkkal, továbbá szintén alátámasztja azt a hipotézisem, hogy senior szint felett a magyar fejlesztők körében a kódolási tapasztalat kevesebb vagy negatív hatással van a bérré a fiatalabb korosztályhoz viszonyítva.

A Stack Overflow fórumokon kb. havi vagy kevesebb mint havi egyszer aktív felhasználók kisebb bérére nehéz magyarázatot adni. Vélhetően ezek az emberek rendelkezhetnek valamilyen nem megfigyelt tulajdonsággal, amelynek bércsökkentő hatása van, viszont mindannyiukra igaz, ilyen gyakorisággal használják a Stack Overflow fórumát, valamint nem Macen végzik munkájukat. A 8.1.10. fejezetben említettekkel összevetve előfordulhat, hogy

ezeknek a fejlesztőknek kevésbé kollaboratív attitűdjük van, vagy rosszabb problémamegoldó képességekkel rendelkeznek, ezért a teljesítményük is alacsonyabb.

8.4.3.2. *Részhatások*

A részhatáskülönbségek esetén első látásra kissé ambivalens eredményeket kaptam. Azok, akik NoSQL lekérdezőnyelvekkel dolgoznak, a nem Macet használók körében valamilyen egyéb hatás miatt kevesebbet keresnek, ugyanakkor ez igaz azokra is, akik SQL – el szeretnék foglalkozni. Mivel ezek mind részhatáskülönbségek, számos olyan - itt nem vizsgált – tényező lehet, amely jövedelmet csökkentő hatással bír ezen részcsoporthoz képest és ezek a hatások nem feltétlenül függenek össze. Előfordulhat például, hogy akik SQL – t szeretnék tanulni éppen alacsonyabb beosztásban vannak és nem Macet használnak (ugyanis arra, hogy például kik dolgoznak középvezetői – vezetői és kik beosztott pozíciókban, csak a korból és tapasztalatokból tudunk következtetni, de pontos adatok nincsenek). Ugyanez igaz lehet a NoSQL – t munkában használókra is. Emellett a korábban is tisztázott válaszadói torzítás hatása itt is megjelenhet, hogy előfordulhat, hogy olyanok is megjelöltek a NoSQL nyelveket, hogy munkában használják őket, akik csak felszínesen ismerik ezeket a technológiákat.

Hasonlóan értelmezhető a Stack Overflow fórumán heti néhány alkalommal aktív felhasználók változójának hatása is, amelynek konfidenciaintervalluma egyébként szinte érinti a nullát, így az eredmény eleve kevésbé tűnik szignifikánsnak. Nagy valószínűséggel az ebbe a szegmensbe tartozó részcsoporthoz van valamilyen mintán kívüli attribútuma, amely csökkenti a keresetet, de mindegyikükre igaz, hogy nem Macet használnak és heti néhány alkalommal aktívak a Stack Overflow weboldalon.

Egyedüli bérnövelő részhatás a Macet használók csoportjára nézve a dekompozíció szerint annak van, ha a fejlesztők Angulart használnak. Önmagában egyébként az Angular modellbéli együttműködésének is pozitív előjele lett, tehát függetlenül az operációs rendszertől, annak, ha valaki Angulart használ, várhatóan bérnövelő hatása van azokhoz képest, akik nem. Összességében az itt felfedezett összefüggést szintén a részhatáskülönbségek tulajdonságával és a kauzalitással magyaráznám, hogy valószínűleg nem azért keres többet egy informatikus, aki Angulart használ, mert azt macOS operációs rendszerkörnyezetben használja (és fordítva), hanem mert lehet más tulajdonsága, amely pozitívan hat a bérére. Általánosságban az Angulart a frontend fejlesztésben használják és alkalmas például kisebb webalkalmazások fejlesztésére is. Akik Macen dolgoznak szoftverfejlesztőként, azok véleményem szerint azért teszik ezt, mert valamilyen olyan szoftvert vagy alkalmazást fejlesztenek, amely macOS - el vagy iOS – el is

kompatibilis, így előfordulhat, hogy lehet a két ismerv közt valamilyen gyengébb összefüggés, ha az adott kitöltő ilyen jellegű kisebb alkalmazások fejlesztéséhez éppen használja az Angulart is. Ugyanakkor úgy gondolom, hogy mivel a népszerű programnyelvek, keretrendszerek vagy platformok kompatibilisek minden operációs rendszerrel, így inkább valamilyen nem megfigyelt hatás lehet az összefüggés hátterében, például az, ha az illető korábban is említett kedvező KATA konstrukcióban egyéni vállalkozóként számlázott cégeknek és éppenséggel a Macet preferálta a munkavégzéshez. Viszont ezen a ponton szintén fontos megemlíteni a kérdőív, mint torzító elem fontosságát, hogy bár a kérdésekben külön volt választva az otthoni és munkahelyi környezetben használt operációs rendszer, előfordulhat, hogy voltak a mintában olyanok, akik az otthonit jelölték munkahelyinek is.

9. AZ ELEMZÉSEK ALAPJÁN TETT VÉGSŐ MEGÁLLAPÍTÁSOK

9.1. Munkatapasztalat, betöltött pozíció

1. Az egyik leginkább bérnövelő hatású tényező a magyar szoftverfejlesztők körében a kódolási tapasztalat, mely szoros összefüggésben áll a korral és iskolai végzettséggel is. A korral kombinálva viszont a tapasztalat növekedésének életkor szerint eltérő hatása is lehet. Ez a jelenség abból is adódhat, hogy az adott informatikus egy elavult technológiában (például a jQuery, melynek ismerete átlagosan rontja a bért) lesz jártasabb, így bérének görbéje 35 – 44 éves kor felett olyan mértékben tud ellapulni, hogy minden egyéb tényező változatlansága mellett a fiatalabb korosztály esetén egységnyi tapasztalatnövekedés magasabb jövedelememelkedést is eredményezhet. Ez alapján tehát megállapítható, hogy 35 év felett a kódolási tapasztalatoknak kevesebb hozzáadott értéke van a jövedelemhez, mint pályakezdőként.
2. A magyar informatika szektorban nincs szignifikáns szóródás az informatika egyes részterületeinek pozíciói között, mivel a bérek egységesen magasak. Amennyiben az elemzés más ágazatok béreivel hasonlította volna össze az informatikusok bérét, a differenciák jelentősebbek lennének. Összességében egyedül a rendszeradminisztrátor – rendszerüzemeltető munkakört érdemes megkülönböztetni a magyar IT szektorban a többitől, ezen részcsoport átlagbére ugyanis szignifikánsan kevesebb a többiénél. Ennek vélhetően olyan okai lehetnek, hogy egy ilyen pozíció betöltése kevesebb speciális tudást igényel technológiai szempontból.

9.2. Iskolai végzettség, technológiai ismeretek

1. Összességében megállapítható, hogy bár megjelentek szakképesítést biztosító, nem felsőfokú kurzusok, Magyarországon továbbra is mérvadó, ha egy informatikus legalább alapszakos diplomával rendelkezik és nem hagyta ott az egyetemet. Az egyetemet be nem fejező informatikusok átlagos bére – feltéve, hogy minden további tényező változatlan – várhatóan kb. 6% -kal lehet kisebb egy alapszakot végzett informatikushoz képest. Emellett a piac honorálja a mesterdiplomát, a mesterszakot végzettek bruttó bére átlagosan nagyobb mértékben tér el az alapszakosokétól, mint a diplomásoké a diploma nélküliektől. Minden tényező változatlansága mellett várhatóan kb. 8,8% -al magasabb egy mesterfokozatú diplomával rendelkező IT szakember keresete egy alapfokozatú egyetemi képzést végzett informatikushoz képest. Az égető munkaerőhiány (és az emiatt megjelent intenzív kurzusok) viszont az informatikai képzések folyamatos fejlesztésére és rugalmas képzési rendűvé tételére kell, hogy ösztönözzék az egyetemeket, hogy a hallgatók naprakész tudást kapjanak és még ha korán is állnak munkába, ne legyen hátráltató tényező egy munkahely a mesterszak választásában. Ésszerű opció lehet például egyes informatikai szakok duális képzési rendben történő meghirdetése is, ahol a tanulók az egyetemi tanulmányokat a munkatapasztalatokkal már karrierjük egészen korai szakaszától kezdve tudják összehangolni.
2. Külön – külön vizsgálva szinte egy programnyelvnek sem volt saját bérnövelő hatása, az viszont már szignifikáns volt, hogy hány programnyelvet ismer vagy szeretne tanulni az illető. Hazánkban a magas informatikai munkaerőhiány miatt elképzelhető, hogy a munkáltatók számára az is elegendő, ha a leendő munkavállaló akár más nyelvek terén szerzett már kódolási tapasztalatokat és rendelkezik az alapvető algoritmikus gondolkodással, tehát könnyen átképezhető más nyelven való programozásra is. A platformok ismerete szintén nem volt szignifikáns, viszont ott még az sem, hogy a kitöltő hányat ismer vagy hányat szeretne megismerni. Nagy valószínűséggel, amennyiben a leendő munkavállaló nem találkozott egy adott platformmal még, annak elsajátítása alapvető informatikai kompetenciákkal nem jelent nagy erőforrást a munka közben sem.

9.3. Demográfiai tényezők

1. A nemek (és szexualitás) esetén megállapítható, hogy nincs szignifikáns bérkülönbség nők és férfiak között, ugyanakkor ez nem feltétlenül a megkülönböztetés hiányából

fakad, hanem abból, hogy a nők és nem heteroszexuális emberek hatása nem mérhető a mintában a kis elemszám miatt. Ez a kis létszám a nemek esetén továbbá olyan szempontból hordoz magában információtartalmat, hogy a női nem valószínűleg még mindig alulreprezentált a szoftverfejlesztői piacon Magyarországon a pálya vélhetően maszkulin megítélése miatt.

2. Az életkornak szignifikáns hatása van a bérre, ugyanakkor hatása összemosódik a korábban említett tapasztalattal, az egyes bérsávok görbéje kor és kódolási tapasztalat szerint különböző. Értelemszerűen az idősebb fejlesztők tapasztaltabbak a fejlesztésben, ugyanakkor az elemzésből megállapítható, hogy Magyarországon fontos az önképzés és a tudás naprakészen tartása, különben az idő előrehaladtával a bér tapasztalat szerinti görbéje 35 – 44 éves kor felett ellapulhat.

10. TOVÁBBI FELMERÜLŐ KUTATÁSI KÉRDÉSEK

10.1. A senior informatikai szakértelem fejlesztése és megtartása egy munkaerőhiány sújtotta piacon

A kódolási – fejlesztési tapasztalat és életkor kapcsán felmerülhet a kérdés, hogy a magyar IT szektor seniorabb dolgozóinak jellemzően milyen kompetenciákat érdemes fejleszteni, hogy továbbra is versenyképes tudással rendelkezzenek. Egy olyan ágazatban, ahol évről évre hatalmas változások történnek, a munkavállaló szempontjából is fontos a kompetitív attitűd és az önképzésre való hajlandóság. Emellett érdekes lenne további szempontok szerint vizsgálni akár azt is, hogy a vállalatokon mekkora nyomás lehet a munkavállalói tudásvagyon megtartását és folyamatos fejlesztését illetőleg, különösen egy olyan piacon, ahol eleve nagymértékű munkaerőhiány áll fenn.

Vizsgálandó kérdés lehet például, hogy mekkora alternatív költsége van annak, ha a drágább seniorabb munkaerő folyamatos továbbképzésébe és megtartásába investálnak ahelyett, hogy frissebb technológiai tudással rendelkező junior kollégákat alkalmazzanak a leszállítandó típusú feladatokra, az idősebb kollégákat pedig több munkaszervezési feladattal lássák el. Az itt felmerült kérdéseket fókuszáltabban, tapasztalat szerint rétegzett mintavétel mellett akár mélyinterjúk módszerekkel is lehetne vizsgálni, hogy fény derüljön arra, hogy az informatikusként dolgozó munkavállalók (tovább)képzési igényei mennyire találkoznak a vállalatok által nyújtott lehetőségekkel és mekkora szükség van még a munkahelyi képzéseken túl a saját önerőből végzett önképzésre.

10.2. A magyar informatikai munkaerő szerepköre és helyzete a KKV és nagyvállalati szektorban

Az elemzésem során fény derült arra, hogy átlagosan nézve a magyar informatikusok várhatóan kevesebbet keresnek, ha start – upoknál vagy kis – és középvállalkozásoknál helyezkednek el. A kérdéskört olyan szempontok mentén lehetne tovább vizsgálni, hogy mennyiben különbözik egy fejlesztő szervezetileg definiált szerepköre egy start – upban vagy KKV -ban egy multinacionális vállalati munkakörtől, illetve, hogy az egyes munkavállalók hol értékelik szakmailag kihívóbbnak a kapott feladatokat. Ebben az elemzésben továbbá feltárássra kerülhetnének a juttatások és bérezés különbségei mellett az egyéni attitűd jellemzői is, hogy melyik szervezeti típusban nagyobb az általános munkavállalói elégedettség és hogy ezt milyen hatások tudják közvetlenül vagy közvetetten befolyásolni. A 8.1.9. fejezetben említett KATA – törvény módosításának piacot átformáló hatása ezen kérdéskörben szintén tovább vizsgálható szempontokkal gazdagítja ezt a témát.

10.3. Az informatikai képzést biztosító magyar felsőoktatási intézmények jövőképe

Bár kutatásom eredményei szerint a magyar informatikai piacon továbbra is fontos a diploma, a különböző felnőttképzések megjelenése, valamint a mintában is nagy számban (átlagosan kb. évi 18-20%) előforduló egyetemet otthaggyók felteszik a kérdést, hogy az egyetemek népszerűsége a későbbiekben is fennmarad -e az informatikai képzéseket illetőleg. Érdekes kutatási kérdés lehet például a duális informatikai képzések lehetőségeinek modellezése vagy akár egy olyan fókuszált mintavételes vagy mélyinterjú vizsgálat, melyben kifejezetten csak egyetemet és csak felnőttképzést végzett szakembereket hasonlítunk össze egységes szempontok mentén.

Az informatikai mesterdiploma kérdésköre szintén olyan téma, melyet érdemes tovább kutatni. Egyfelől fontos lenne a hallgatói szemszög megismerése és azon gyökérokok feltárása, amelyek miatt valaki inkább nem választja vagy inkább választja tanulmányai után a mesterképzést. Emellett felsőoktatási aspektusból nézve további kérdéseket vet fel, hogy mennyire lehet a piac igényeihez igazítani a képzési struktúrát. Például, hogy milyen módszerekkel lehetne a hagyományos hat félév és szakmai gyakorlatos informatikai alapképzések perspektívájából kilépve azokat úgy átalakítani, hogy egyszerre legyenek gyakorlatorientáltak de megtartó erővel is rendelkezzenek, hogy a hallgatók érdemesnek tartsák a továbbtanulást az első diploma megszerzése után.

Szintén mintavételes közvéleménykutatás vagy mélyinterjúk módszerek segítségével továbbá érdemes lenne feltárni és kettéválasztani, hogy mi az a piac által elvárt kompetencia, amit az egyetemet befejező hallgatók csak a munkahelyen tudnak elsajátítani és hogy a felsőoktatási képzések milyen alapot tudnak ezen tudás elsajátításához adni. Ebben nagy szerepe van továbbá az oktatói tudás naprakészen tartásának is, amely szintén tovább vizsgálendő kérdéskör, hogy egy egyetem miképpen tudja az oktatóit a piaci trendek nyomán követésére ösztönözni és a tapasztaltakat a tananyagba építeni.

11. KÖSZÖNETNYILVÁNÍTÁS

Szeretnék köszönetet mondani mindenkinek, aki támogatott abban, hogy ez a dolgozat elkészülhessen.

Köszönöm témavezetőm, Kovács László munkáját, aki a folyamatos segítségnyújtás mellett számos hasznos tanáccsal látott el. Hozzáállásával és mély szakértelemmel példaképpé vált számomra.

Köszönet illeti továbbá Balázsne Mócsai Andreát, aki nemcsak megalapozta a statisztikai ismereteimet, de példamutatásával és oktatói munkájával megszerettette velem ezt a tudományterületet.

12. IRODALOMJEGYZÉK

2022. évi XIII. törvény a kisadózó vállalkozók tételes adójáról. (2022.07.18). Letöltés dátuma: 2023.04.14., forrás: Jogszabálykereső: <https://net.jogtar.hu/jogszabaly?docid=a2200013.tv>

5/2020. (I. 31.) Kormányrendelet a Nemzeti Alapnterv kiadásáról, bevezetéséről és alkalmazásáról szóló 110/2012. (VI. 4.) Kormányrendelet módosításáról. (2020.01.31.). Letöltés dátuma: 2023.04.14. forrás: Jogszabálykereső: <https://net.jogtar.hu/jogszabaly?docid=a1200110.kor>

Arbidāne, S., & Silickis, M. (2018). *How Latvian Developers Learn New Technologies. A Conclusive Review and Analysis of Survey Data*. Students International Scientific and Practical Conference. <http://dx.doi.org/10.17770/het2018.22.3644>

Árgilán, V. S., & Kelemen, A. (2016. 10. 06 - 09). *Az elvárt digitális kompetenciák a XXI. században és az informatika oktatás gyakorlata a közoktatásban Magyarországon*. [Konferenciaelőadás]. XVII. ENELKO – XXVI. SzámOkt Nemzetközi Energetikai-elektrotechnikai és Számítástechnikai Konferencia, Kolozsvár. <http://publicatio.bibl.u-szeged.hu/id/eprint/14770>

Békés, G., & Kézdi, G. (2021). *Data Analysis for Business, Economics, and Policy*. Cambridge University Press. <https://doi.org/10.1017/9781108591102>

Billionniere, E., Rahman, F., Marshall, B., & Forman, T. (2020). Debugging the Gender Gap: (Re-)entry Initiatives in Emerging Technologies for Women. *ACM Transactions on Computing Education*, 21(2), 1-5. <https://doi.org/10.1145/3368308.3415422>

Blinder, A. S. (1973). Wage Discrimination: Reduced Form and Structural Estimates. *The Journal of Human Resources*, 8(4), 436. <https://doi.org/10.2307/144855>

Brown, C. D., & Davis, H. T. (2006). Receiver operating characteristics curves and related decision measures: A tutorial. *Chemometrics and Intelligent Laboratory Systems*, 80(1), 24–38. <https://doi.org/10.1016/j.chemolab.2005.05.004>

Budapesti Műszaki és Gazdaságtudományi Egyetem, Villamosmérnöki Kar. (2015). *Adatbázisok*. Tantárgyi adatlap. Budapest: Budapesti Műszaki és Gazdaságtudományi Egyetem. <https://portal.vik.bme.hu/kepzes/targyak/VITMA311/>

- Charbuty, B., & Abdulazeez, A. (2021). Classification Based on Decision Tree Algorithm for Machine Learning. *Journal of Applied Science and Technology Trends*, 2(1), 20–28. <https://doi.org/10.38094/jastt20165>
- Cotton, J. (1988). On the Decomposition of Wage Differentials. *The Review of Economics and Statistics*, 70(2), 236. <https://doi.org/10.2307/1928307>
- Centor, R. M. (1991). Signal Detectability: The Use of ROC Curves and Their Analyses. *Medical Decision Making*, 11(2), 102–106. <https://doi.org/10.1177/0272989X9101100205>
- Daniya, T., Geetha, M., & Suresh Kumar, K. (2020). Classification and Regression Trees with Gini Index. *Advances in Mathematics: Scientific Journal*, 9(10), 8237–8247. <https://doi.org/10.37418/amsj.9.10.53>
- Dorfman, R. (1979). A Formula for the Gini Coefficient. *The Review of Economics and Statistics*, 61(1), 146. <https://doi.org/10.2307/1924845>
- Elasticsearch B.V. (n.d.). *What is ELK Stack?* <https://www.elastic.co/what-is/elk-stack>
- El-Seoud, S. A., El-Sofany, H. F., Abdelfattah, M. A. F., & Mohamed, R. (2017). Big Data and Cloud Computing: Trends and Challenges. *International Journal of Interactive Mobile Technologies (IJIM)*, 11(2), 34. <https://doi.org/10.3991/ijim.v11i2.6561>
- Eötvös Loránd Tudományegyetem, Informatikai Kar. (2018a). *Adatbázisok I. Tantárgyi adatlap.* Budapest: Eötvös Loránd Tudományegyetem. <https://www.inf.elte.hu/dstore/document/1051/Adatb%C3%A1zisok%20I..pdf>
- Eötvös Loránd Tudományegyetem, Informatikai Kar. (2018b). *Full-stack webprogramozás. Tantárgyi adatlap.* Budapest: Eötvös Loránd Tudományegyetem. <https://www.inf.elte.hu/dstore/document/1091/Full-stack%20webprogramoz%C3%A1s.pdf>
- Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8), 861–874. <https://doi.org/10.1016/j.patrec.2005.10.010>
- Felvi.hu. (2023). *2023 szeptemberében induló képzések.* https://www.felvi.hu/felveteli/ponthatarok_statistikak/friss_statistikak!/FrissStatistikak/index.php/friss_statistikak/karonkent?page=1&orderBy=sta_kar_kod&sortingBy=ASC
- GitHub Engineering. (2018. 09. 06). *Removing jQuery from GitHub.com frontend.* <https://github.blog/2018-09-06-removing-jquery-from-github-frontend/>

Google LLC. (n.d.). *Introduction to the Angular Docs*. Letöltés dátuma: 2023.04.14.
<https://angular.io/docs>

Gower, J. C. (1971). A General Coefficient of Similarity and Some of Its Properties. *Biometrics*, 27(4), 857. <https://doi.org/10.2307/2528823>

Guttman, L. (1968). A general nonmetric technique for finding the smallest coordinate space for a configuration of points. *Psychometrika*, 33(4), 469–506.
<https://doi.org/10.1007/BF02290164>

Hanck, C., Arnold, M., Gerber, A., & Schmelzer, M. (2019). *Introduction to Econometrics with R*. University of Duisburg-Essen.

Hays Hungary. (2022). *Hays Hungary Salary Guide 2022*.
<https://image.email.hays.com/lib/fe4515707564057c751477/m/6/f7f76436-3389-444b-b0e3-b83800f868c4.pdf>

Hays Hungary. (2023). *Hays Hungary Salary Guide 2023*. <https://www.hays.hu/salary-guide/trendek#salary-guide>

Henrion, R. (1994). N-way principal component analysis theory, algorithms and applications. *Chemometrics and intelligent laboratory systems*, 25(1), 1-23.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2021). *An Introduction to Statistical Learning with Applications in R*. Springer.

Jann, B. (2008). The Blinder–Oaxaca Decomposition for Linear Regression Models. *The Stata Journal: Promoting Communications on Statistics and Stata*, 8(4), 453–479.
<https://doi.org/10.1177/1536867X0800800401>

Janssen, M., & Joha, A. (2006). Motives for establishing shared service centers in public administrations. *International Journal of Information Management*, 26(2), 102–115.
<https://doi.org/10.1016/j.ijinfomgt.2005.11.006>

Kleiber, C., & Zeileis, A. (2008). *Applied econometrics with R*. Springer.

Kruskal, J. B. (1964). Nonmetric multidimensional scaling: A numerical method. *Psychometrika*, 29(2), 115–129. <https://doi.org/10.1007/BF02289694>

Kovács, E. (2019). *Többváltozós adatelemzés*. Typotex Kiadó.
http://etananyag.ttk.elte.hu/FiLeS/downloads/14_KOVACS_E_Tobbvalt_adatelemzes.pdf

Központi Statisztikai Hivatal. (2023a). 20.1.1.9. A foglalkoztatottak száma nemzetgazdasági ágak, ágazatok szerint, nemenként – TEÁOR'08 [ezer fő]. https://www.ksh.hu/stadat_files/mun/hu/mun0009.html

Központi Statisztikai Hivatal. (2023b). 20.1.1.72. Üres álláshelyek száma és aránya nemzetgazdasági áganként. https://www.ksh.hu/stadat_files/mun/hu/mun0072.html

Központi Statisztikai Hivatal. (2023c). 9.1.1.18. Magyarországon működő szolgáltató központok száma, teljes- és nem teljes munkaidőben alkalmazásban állók állományi létszáma, havi bruttó átlagkeresetük alakulása nemzetgazdasági ág szerint. https://www.ksh.hu/stadat_files/gsz/hu/gsz0064.html

Központi Statisztikai Hivatal. (2023d). 20.1.1.21. A munkanélküliek száma előző munkahelyük nemzetgazdasági ága, ágazata szerint, nemenként – TEÁOR'08 [ezer fő]. https://www.ksh.hu/stadat_files/mun/hu/mun0028.html

Lavery, M. R., Acharya, P., Sivo, S. A., & Xu, L. (2019). Number of predictors and multicollinearity: What are their effects on error and bias in regression? *Communications in Statistics - Simulation and Computation*, 48(1), 27–38. <https://doi.org/10.1080/03610918.2017.1371750>

Lux, G. (2017). A külföldi működő tőke által vezérelt iparfejlődési modell és határai Közép-Európában. *Tér és Társadalom*, 31(1), 30–52. <https://doi.org/10.17649/TET.31.1.2801>

Mansfield, E. R., & Helms, B. P. (1982). Detecting Multicollinearity. *The American Statistician*, 36(3), 158–160. <https://doi.org/10.1080/00031305.1982.10482818>

Mantovani, R. G., Horvath, T., Cerri, R., Vanschoren, J., & de Carvalho, A. C. P. L. F. (2016. 10. 09 - 12). *Hyper-Parameter Tuning of a Decision Tree Induction Algorithm*. [Konferenciaelőadás]. 2016 5th Brazilian Conference on Intelligent Systems (BRACIS), Recife. <https://doi.org/10.1109/BRACIS.2016.018>

Mayer-Schönberger, V., & Cukier, K. (2014). *Big Data: Foradalmi módszer, amely megváltoztatja munkánkat, gondolkodásunkat és egész életünket*. HVG Kiadó.

Mead, A. (1992). Review of the Development of Multidimensional Scaling Methods. *The Statistician*, 41(1), 27. <https://doi.org/10.2307/2348634>

Meta Platforms Inc. (n.d.). *React*. Letöltés dátuma: 2023.04.14. <https://react.dev/>

- MacSources. (2022. 01. 20). *Why Is MacOS Better For Graphic Designing?*
<https://macsources.com/why-is-macos-better-for-graphic-designing/>
- Microsoft Corporation. (2022). *ASP.NET Overview*. <https://learn.microsoft.com/en-us/aspnet/overview>
- Microsoft Corporation. (2023). *What is .NET? Introduction and Overview*.
<https://learn.microsoft.com/en-us/dotnet/core/introduction>
- MongoDB Inc. (n.d.). *Welcome to the MongoDB Documentation*. Letöltés dátuma: 2023.04.14.
<https://www.mongodb.com/docs/>
- Nemzeti Szakképzési és Felnőttképzési Hivatal. (2023). *2.sz. melléklet: képzési jegyzék GINOP-6.2.8-VEKOP-20-2020-00001 kiemelt projekt keretében 2023. februárban indítani tervezett képzések jegyzéke*.
https://ujratervezes.nive.hu/api/static/doc/kepzesi_jegyzek_3.0%20halad%C3%B3.pdf
- Neumark, D. (1988). Employers' Discriminatory Behavior and the Estimation of Wage Discrimination. *The Journal of Human Resources*, 23(3), 279. <https://doi.org/10.2307/145830>
- Oaxaca, R. (1973). Male-Female Wage Differentials in Urban Labor Markets. *International Economic Review*, 14(3), 693. <https://doi.org/10.2307/2525981>
- O'Dea, R. E., Lagisz, M., Jennions, M. D., & Nakagawa, S. (2018). Gender differences in individual variation in academic grades fail to fit expected patterns for STEM. *Nature Communications*, 9(1), 3777. <https://doi.org/10.1038/s41467-018-06292-0>
- OECD. (2022). *Gender wage gap*. Organisation for Economic Co-operation and Development.
https://www.oecd-ilibrary.org/employment/gender-wage-gap/indicator/english_7cee77aa-en
- OpenJS Foundation. (n.d.). *jQuery API Documentation*. Letöltés dátuma: 2023.04.14.
<https://api.jquery.com/>
- OpenJS Foundation. (n.d.). *Express – Fast, unopinionated, minimalist web framework for Node.js*. Letöltés dátuma: 2023.04.14. <https://expressjs.com/>
- Peres-Neto, P., Legendre, S., Dray, D., & Borcard, D. (2006). *Variation partitioning of species data matrices: estimation and comparison of fractions*. *Ecology*, 87, 2614–2625.
- Plackett, R. L. (1949). A Historical Note on the Method of Least Squares. *Biometrika*, 36(3/4), 458. <https://doi.org/10.2307/2332682>

Powers, D. M. W. (2007). *Evaluation: From Precision, Recall and F-Factor to ROC, Informedness, Markedness & Correlation*. School of Informatics and Engineering Flinders University of South Australia.

Pranggono, B., & Arabo, A. (2021). COVID-19 pandemic cybersecurity issues. *Internet Technology Letters*, 4(2), e247. <https://doi.org/10.1002/itl2.247>

Redis Ltd. (n.d.). *Introduction to Redis*. Letöltés dátuma: 2023.04.14. <https://redis.io/docs/about/>

R Foundation, (n.d.). *What is R?* Letöltés dátuma: 2023.04.14. <https://www.r-project.org/about.html>

Silveira, K. K., Musse, S., Manssour, I., Vieira, R., & Prikladnicki, R. (2019). *Reinforcing Diversity Company Policies: Insights from StackOverflow Developers Survey*. School of Technology, Pontifícia Universidade Católica do Rio Grande do Sul.

Shepard, R. N. (1962). The analysis of proximities: Multidimensional scaling with an unknown distance function. I. *Psychometrika*, 27(2), 125–140. <https://doi.org/10.1007/BF02289630>

Stack Overflow. (n.d.). *Stack Overflow Annual Developer Survey*. Letöltés dátuma: 2022.10.22. <https://insights.stackoverflow.com/survey>

Stack Overflow. (n.d.). *Empowering the world to develop technology through collective knowledge*. Letöltés dátuma: 2023.04.14. <https://stackoverflow.co/>

Stack Overflow. (2022). *Stack Overflow Developer Survey 2022*. Letöltés dátuma: 2022.10.22. <https://survey.stackoverflow.co/2022/>

Stock, J. H., & Watson, M. W. (2020). *Introduction to econometrics*. Pearson.

Stroustrup, B. (2010). What should we teach new software developers? Why? *Communications of the ACM*, 53(1), 40–42. <https://doi.org/10.1145/1629175.1629192>

Tangirala, S. (2020). Evaluating the Impact of GINI Index and Information Gain on Classification using Decision Tree Classifier Algorithm. *International Journal of Advanced Computer Science and Applications*, 11(2). <https://doi.org/10.14569/IJACSA.2020.0110277>

TIOBE. (n.d.). *TIOBE Community Index Definition*, Letöltés dátuma: 2022.09.26. https://www.tiobe.com/tiobe-index/programminglanguages_definition/

Tharwat, A. (2021). Classification assessment methods. *Applied Computing and Informatics*, 17(1), 168–192. <https://doi.org/10.1016/j.aci.2018.08.003>

Torgerson, W. S. (1952). Multidimensional scaling: I. Theory and method. *Psychometrika*, 17(4), 401–419. <https://doi.org/10.1007/BF02288916>

Van Buuren, S. (2018). *Flexible Imputation of Missing Data*. CRC Press. <https://stefvanbuuren.name/fimd/>

VMware Inc. (n.d). 1. *Introduction to Spring Framework*. <https://docs.spring.io/spring-framework/docs/3.2.x/spring-framework-reference/html/overview.html>

Wilkins, J. (2021.08.30). *What is a Full Stack Developer? Back End + Front End = Full Stack Engineer*. FreeCodeCamp.Org. <https://www.freecodecamp.org/news/what-is-a-full-stack-developer-back-end-front-end-full-stack-engineer/>

Wooldridge, J. M. (2013). *Introductory econometrics: A modern approach*. South-Western Cengage Learning.

World Economic Forum. (2021). *Global Gender Gap Report 2021*. https://www3.weforum.org/docs/WEF_GGGR_2021.pdf

World Economic Forum. (2022). *Global Gender Gap Report 2022*. https://www3.weforum.org/docs/WEF_GGGR_2022.pdf

You, E. (n.d.). *What is Vue?* Letöltés dátuma: 2023.04.14. <https://vuejs.org/guide/introduction.html>

13. ÁBRAJEGYZÉK

1. ábra: Az információ, kommunikáció szektorban foglalkoztatottak száma (ezer fő). Forrás: KSH (2023a). Saját készítés.	5
2. ábra: Egyéb, elsősorban nem informatikai tudást igénylő állások bruttó átlagbérei (juttatások és bónusz nélkül) 2023-ban tapasztalat szerint (eFt). Forrás: Hays Hungary Salary Guide (2023). Saját készítés.	7
3. ábra: Mesterszakra jelentkezők arányának összehasonlítása a gazdasági és informatikai területek közt 2 - 2 népszerűbb fővárosi és vidéki egyetemen a 2023 szeptemberben induló képzésekre. Forrás: Felvi.hu (2023). Saját készítés.	10
4. ábra: Munkanélküliek száma nemenkénti bontásban, ha az adott személy előző munkahelye szálláshely – szolgáltatás, vendéglátás szektorba esett, 2009 és 2022 között (ezer fő). Forrás: KSH (2023d). Saját készítés.	13
5. ábra: A bruttó bérek (Ft) eloszlása logaritmálás után.	22
6. ábra: Döntési fa modell a jövedelem megosztásával kapcsolatban.	24
7. ábra: A döntési fa modellt kiértékelő ROC görbe.	25
8. ábra: A webfejlesztés területén dolgozók megoszlása iskolai végzettség szerint a Stack Overflow adatbázisban.	29
9. ábra: A front - end fejlesztéshez használt nyelvek (HTML, CSS, PHP, JavaScript és TypeScript) megoszlása a webfejlesztéssel foglalkozók között a Stack Overflow adatbázisban.	30
10. ábra: Az iskolai végzettség és a látens kódolási tapasztalatokat reprezentáló változó kapcsolata.	40
11. ábra: Az életkor és a látens kódolási tapasztalatokat reprezentáló változó kapcsolata.	41
12. ábra: Az életkor és tapasztalat marginális hatásainak szemléltetése.	42
13. ábra: A diploma hatásának vizsgálata a bérekre, összetételhatás.	45

14. TÁBLÁZATOK JEGYZÉKE

1. táblázat: A kiinduló adatbázis leíró statisztikájának összesítése.....	28
2. táblázat: A modellszelekció főbb lépései.	32
3. táblázat: A diplomával rendelkezők és nem rendelkezők dekompozíciós vizsgálatának eredményei.	44
4. táblázat: Az alapszakos és mesterszakos diplomával rendelkező kitöltők dekompozíciós vizsgálatának eredményei.	49
5. táblázat: A macOS -t használó és nem használó kitöltők dekompozíciós vizsgálatának eredményei.	51

15. MELLÉKLETEK

15.1. Módszertani mellékletek

Ökonometriai módszertanok, alapfeltevések

Az ökonometria alapvetően a statisztikai matematikán alapuló, abból továbbfejlődött tudományág, amely (köz)gazdasági, makroökonómiai, politikai és egyéb társadalmi kérdésekre keres választ empirikus megközelítéssel. A hagyományos értelemben vett statisztikától az különbözteti meg, hogy míg az számos esetben szimulált, laboratóriumi körülmények között vizsgálódik, addig az ökonometriai elemzések alapja mindig a tapasztalati megfigyelés. Ebből fakadóan az ökonometriai méréseket szokták „nem – experimentális”, avagy megfigyeléseken alapuló módszertanoknak is hívni, ezzel kiemelve, hogy az adatgyűjtés szempontjából az elemzés passzív, például nem egy valamilyen előre adott eloszlás mentén generált mesterséges mintán dolgozunk. A megfigyelés – alapú kutatás fő oka a vizsgált társadalmi és gazdasági problémák komplexitása, amelyeket lehetetlen vagy rendkívül költséges lenne szimulálni rájuk (Stock és Watson, 2020).

Az empirikus modellezések egyik fő hátulütője a kauzalitás problémája, amelyet érdemes minden megállapításkor szem előtt tartanunk. A vizsgált mintában található egyedek nagy valószínűséggel nemcsak az általunk megfigyelt tulajdonságaikban térhetnek el, hanem lehetnek rájuk ható, nem megfigyelt hatások is. Egy egyszerűbb példa a problémára - amelyet elemzésemben én is érinteni fogok - a bérezés és iskolai végzettség kérdése. Az iskolai végzettségnek bizonyosan van önmagában is hatása a bérekre, ugyanakkor hibás lenne csak ezzel az egy faktorral magyarázni azt, hogy valaki mennyit keres, mivel az, hogy milyen végzettsége van, rengeteg egyéb tényezőből adódik (például: szociokulturális háttér, anyagi helyzet, egyéni preferenciák). Ebből fakadóan a végzettség – kereset közti korreláció nem jelent egyértelmű kauzalitást.

A kauzalitás problémáját ökonometriai modellezéssel tudjuk kezelni. Ennek alapja, hogy egy mintára illesztett regressziós függvényben a magyarázó – és eredményváltozók közti kapcsolatot nem determinisztikusnak, hanem csak sztochasztikusnak tekintjük, tehát feltételezzük, hogy a becslésben mindig lesz hibatag. Másfelől kikötjük, hogy a modellben szereplő összes változónak és a hibatagnak is van valamilyen valószínűségi eloszlása, tehát valószínűségi változókként tekintünk (Wooldridge, 2013).

A legkisebb négyzetek módszere

Az elemzésemben az ökonometriai modellezést a legkisebb négyzetek (Ordinary Least Squares, a továbbiakban OLS) módszerén alapulva kezdtem el. Amikor lineáris regressziós modellt illesztünk a mintánkra, egy olyan lineáris függvényt keresünk, amellyel adott hibatag mellett a legpontosabban megbecsülhető az általunk választott eredményváltozó a mintában adott magyarázóváltozók függvényében (Wooldridge, 2013).

$$\hat{Y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_k x_k + \varepsilon \quad (8.)$$

A (8.) egyenlet egy lineáris regressziós modellt mutat be, ahol a becsült eredményváltozót k darab magyarázóváltozóval becsültük ε hibatag mellett. A modellezés során egy optimalizálási problémát oldunk meg, hogy megtaláljuk azon β paramétereket, amelyekkel a függvényünk által becsült értékek a legközelebb helyezkednek el az eredeti értékekhez. A becsült és eredeti pontok távolságát négyzetesen tudjuk leírni, ezeket reziduumoknak hívjuk. A cél, hogy ezeket a reziduumokat minimalizáljuk, hiszen ekkor közelítjük meg becslésünkkel a leginkább az eredeti pontokat, azaz ekkor érjük el a legjobb illeszkedést a mintára. A legkisebb négyzetek módszere tehát a (9.) képletben bemutatva a négyzetes távolságok minimalizálására törekszik minden megfigyelés és becsült érték között (Wooldridge, 2013).

$$\arg \min \sum_{i=0}^n (y_i - \hat{y}_i)^2 \quad (9.)$$

A túlillesztés problémája

Természetesen egy lineáris függvény a sztochasztikusságból adódóan a legtöbb esetben nem fog tökéletesen illeszkedni a minta egyes pontjaira. A függvénnyel már nem megmagyarázott szóródást (variáciát) a modell hibatagja foglalja magába, a modell magyarázóerejét pedig, hogy a variancia hány százalékát sikerült az adott függvénnyel lefednünk, a determinációs együttható (R^2) mutatja meg. 100%-os R^2 esetén tökéletes illeszkedést értünk el (Kleiber és Zeileis, 2008).

Ugyanakkor modellezésnél sosem cél a tökéletes illeszkedés, mivel a függvény, amelyet megalkotunk, mindig csak a mintára vonatkozik, amely a teljes sokaságnak csak egy része, amit soha nem fogunk teljesen ismerni (hiszen akkor nem is lenne szükségünk ökonometriai modellekre). A lineáris regressziókba bármikor bevonhatunk új változókat, vagy akár el is szakadhatunk a linearitástól, harmad – negyedfokú vagy egyéb tagokat bevonva, ezzel a mintára való illeszkedést javítva, ugyanakkor az így kapott bonyolult modell nagy valószínűséggel csak kis mértékben fogja a valóságot tükrözni, mivel túlzott mértékben rákoncentráltunk a

mintánkra, ezáltal az általános sokaságra tett becslés sem lesz pontos. A gépi tanuló modellek – melyek közül számos ökonometriai modelleken alapul – ezt a problémát próbálják kiküszöbölni az adathalmaz tanító – teszt halmazokra bontásával. Az ilyen módszertanok során a modell csak a tanító halmazokra „tanul rá”, majd a jóságának minősítésekor a tesztalmazon futtatjuk, hogy megnézzük, hogy a tanítóhalmazon megbecsült paraméterekkel mennyire képes az új adatok alapján helyes eredményeket adni. A mostani elemzésemnek ugyanakkor nem fókusza a gépi tanulás. Az ökonometriai modellek a sokaság mintán kívüli viselkedésének problémáját hipotézisvizsgálattal igyekeznek kiküszöbölni. Ennek előnye, hogy a modellezéshez kevesebb megfigyelés is elegendő, hiszen nem szükséges az adathalmaz tanító – teszt felbontása, amely nagy számosságú adatot igényel, hogy a tanítást hatásosan tudjuk elvégezni (James, Hastie, Tibshirani és Witten, 2021).

Az ökonometriai modellezés standard modellfeltevései

A statisztikai és ökonometriai becsléseknek három szükséges tulajdonsága kell, hogy létezzen:

- **Torzítatlanság:** A több mintavételből számított regressziós együtthatók várható értéke legyen egyenlő a teljes sokaságra illesztett OLS regresszió együtthatóinak értékével.
- **Konzisztencia:** A mintára illesztett OLS regresszió együtthatóinak standard hibája nullába konvergáljon, amennyiben a mintaelemszám növekszik.
- **Hatásosság:** Az OLS regresszióhoz képest ne létezzen olyan alternatív becslési módszer, amely kisebb standard hibát eredményez annál.

A három tulajdonság együttes fennállása esetén mondhatjuk, hogy az OLS regresszió együtthatóbecslése a legjobb lineáris torzítatlan becslés. Az angol szakirodalom ezt BLUE tulajdonságnak nevezi (Best Linear Unbiased Estimator) (Wooldridge, 2013).

A probléma a fenti három tulajdonság fennállásának tesztelése, ugyanis ehhez több mintavételre lenne szükség, az ökonometriai modellezésnél pedig számtalan esetben csak egyetlen minta áll rendelkezésünkre az egész sokaságból. A 6 standard modellfeltevést egyszerre alkalmazó Gauss – Markov tételt Plackett (1949) összegezte, amelyek együttes teljesüléséből következik, hogy az OLS becslés BLUE tulajdonságú. Ezek a feltevések már a modellezés során többfajta eljárással kezelhetők, fennállásuk pedig különböző próbákkal tesztelhető is.

1. A minta legyen nagy (≥ 30) vagy az eredményváltozó legyen normális eloszlású: Az utóbbi elérhető például logaritmálással, erős jobbra elnyúlás esetén.

2. Lineáris regressziós függvényforma: A regressziós egyenes egyenlete nem tartalmazza a magyarázóváltozók nemlineáris transzformáltját (például logaritmált vagy kvadratikus módon).
3. Erős exogenitás: A regressziós hibatag várható értéke nulla és nem korrelál egy magyarázóváltozóval sem. Amennyiben ez nem teljesül és a modell hibatagja endogén (korrelál valamelyik magyarázóváltozóval), akkor fennáll annak a veszélye, hogy a magyarázóérték egy része a hibatagba került.
4. Homoszkedaszticitás: A hibatag állandó szórással jelentkezik, tehát kilengéseinek amplitúdója nem függ egy megfigyeléstől sem. Amennyiben a modell heteroszkedasztikus, fennáll annak a veszélye, hogy a modell nem lesz hatásos és konzisztens.
5. Autokorrelálatlanság: A hibatag és annak semmilyen késleltetett értéke közt nem áll fenn korreláció.
6. Nem áll fenn egzakt multikollinearitás: Egy olyan magyarázóváltozó sincs a modellben, amelyet előállíthatunk a többi magyarázó tag lineáris kombinációjával.

A BLUE tulajdonság elérése kapcsán az 1. 2. és 3. feltétel szükséges a torzítatlan, a 4. 5. 6 feltételek pedig a hatásos és konzisztens becsléshez (Wooldridge, 2013).

A pooled OLS becslés

Attól függően, hogy milyen szerkezetű adathalmaz áll rendelkezésünkre, az adatbázisok három nagyobb csoportját tudjuk elkülöníteni: keresztmetszeti, idősoros és panel struktúrájú adatok. A keresztmetszeti adattáblákban a megfigyelési egységek az egyes sorok, és a hozzájuk tartozó attribútumok az oszlopok. A *Stack Overflow Annual Developer Survey* egyes éveinek egy – egy adattáblája például keresztmetszeti tábla, ahol egy egyed egy kitöltő, a hozzá tartozó tulajdonságok (pl.: nem, kor ... stb.) az egyes oszlopok. Az idő, mint tulajdonság megjelenthet egy oszlopban (például a Stack Overflow kérdőív adataiban a válasz rögzítésének éve), ugyanakkor a tábla ettől még nem válik idősorossá. Az idősoros táblában a megfigyelési egység maga az idő. Az ilyen táblázatokban idő függvényében láthatjuk az adatok megváltozását (pl.: egy ország GDP-jének alakulása évről évre), amely ezért legtöbbször időpont szerint rendezett is. A paneladatbázis a kettő ötvözte: több keresztmetszeti tábla különböző időpontokban. A panelökonometriai elemzés során el tudjuk különíteni az egyedekre vonatkozó fix és véletlen hatások megváltozását az idő függvényében (Békés és Kézdi, 2021).

Mivel a Stack Overflow kérdőívének eredményei minden évben egy – egy keresztmetszeti tábla formájában állnak elő, így akár paneladatbázisként is kezelhetnénk azt. Ugyanakkor a panelökonometria elemzés egyik alapja, hogy az egyedek azonosíthatók és megkülönböztethetők legyenek, hogy számszerűsíteni tudjuk az egyedenkénti fix hatásokat. Mivel az adatokat anonim módon gyűjtötték, nem garantálható az, hogy aki például 2018-ban kitöltő volt, az a többi évben is kitöltötte a kérdőívet, valamint az egyes felhasználók semmilyen módon nem azonosíthatók, tehát aki az egyik évben a 39-es sorszámmal került be az adatbázisba, nem biztos, hogy a rákövetkező évben is a 39-es ID-vel rendelkezett. Ilyen értelemben az adatbázisnak egy gyengesége, hogy nem használható paneladatbázisként a fix egyedatások mérésére, ugyanakkor a bérezést meghatározó tényezők a legkisebb négyzetek módszerével, nemlineáris változók bevonásával és egyéb dekompozíciós eljárásokkal is jól becsülhetők és jellemezhetők. Ehhez az adatbázison egy olyan módosítást szükséges végrehajtani, hogy az egyes évek keresztmetszeti tábláit egy külön Év változót bevezetve egymás alá másoljuk a korábban tisztázott OLS becslésen alapuló ökonometria modellt építünk a kialakult összesített adattáblára. Ezáltal az idő hatását megtartjuk, ugyanakkor feltételezzük, hogy nincs kimutatható egyedenkénti fix hatás. Az ilyen jellegű módszertant hívjuk pooled OLS becslésnek (Békés és Kézdi, 2021).

Alkalmazott modellszelekciós eljárások

A modellszelekció során a két információs kritérium és a Wald – teszt eredményét együttesen vettem figyelembe, tehát csak akkor hagytam el vagy vettem be új változót, amennyiben mindhárom eljárás együttesen ezt javasolta.

Információs kritériumok

A modellszelekció során kétfajta információs kritériumot is alkalmaztam: Schwarz (BIC) és Akaike (AIC). Az információs kritériumok olyan mutatószámok, amelyek egyszerre büntetik, ha a modellben túl sok magyarázóváltozó található és ha túl nagy hibát vét a becslésben. A különbség köztük, hogy melyikre helyezik a hangsúlyt. Mivel ezek a mutatók hiba jellegű tulajdonságokat számszerűsítene, így a cél a minimalizálás. A (10.) és (11.) képletekben szemléltetve logikailag a modell teljes eltérésnégyzetösszegéből (továbbiakban SSE) és a paraméterek számának (k) valamely függvényéből állnak, amely maga a büntetőparaméter. Plusz egy változó bevonásával SSE szükségszerűen csökken vagy stagnál, ezek a büntetőparaméterek pedig nőnek. Amennyiben SSE csökkenése meghaladja a büntetőparaméterek növekedését, a kérdéses változónak van akkora hozzáadott értéke a

modellhez, hogy érdemes legyen bevenni. A két információs kritérium közül a Schwarz szigorúbban bünteti egy további változó bevetését.

$$AIC = \frac{SSE}{n} e^{\frac{2k}{n}} \quad (10.)$$

$$BIC = \frac{SSE}{n} n^{\frac{k}{n}} \quad (11.)$$

Modellszelekció esetén a szűkített és eredeti modell közül azt érdemes választanunk amelyiknek kisebb az információs kritériuma (James és munkatársai 2021).

Wald – teszt

A Wald – teszt szintén egy modellszelekció során használható módszer, amely hipotézisvizsgálaton alapszik. A próba során a szűkített és eredeti modell közt döntünk, amelyek közül az eredeti modell m db változóval többet tartalmaz a szűkítetténél és az eredeti modell tartalmazza a szűkített összes változóját. A teszt nullhipotézise azt állítja, hogy az eredeti modellben található plusz m db változó regressziós együtthatójának mindegyike nulla, tehát még együttesen sem bírnak olyan mértékű szignifikanciával, hogy érdemes legyen őket benntartani, elhagyhatóak anélkül, hogy a modell ténylegesen romlana. Az alternatív hipotézis állítása ezzel szemben, hogy a bővebb modell plusz m db változójának együtthatója szignifikáns, a szóban forgó változókat nem érdemes elhagyni, mert romlana a modell (Wooldridge, 2013).

A modellszelekcióhoz használt kritériumokat és a Wald – tesztet nemcsak a kiinduló változók szelektálásakor használtam fel, hanem az elemzés későbbi fázisaiban akkor is, amikor arról döntöttem, hogy egy általam létrehozott újabb változót bevegyek -e a modellbe vagy sem. Ilyenkor hasonlóan azt figyeltem, hogy az új változó hatására romlanak -e az információs kritériumok, illetve, hogy a Wald – teszt p – értéke alapján az alternatív hipotézist szükséges -e választani.

Alkalmazott modellspecifikációs tesztek és eljárások

RESET teszt

A Ramsey – féle RESET teszt a modellspecifikáció általános tesztje, amelynél (sok más hasonló teszttel ellentétben) nem kell előre sejtenünk, milyen nemlineáris hatások miatt nem jól specifikált a lineáris regressziós modell, hanem általában vizsgálja a kérdést, hogy a modell jól specifikált -e vagy sem. Továbbá nem kell figyelembe vennünk az interakciókat és kvadratikus hatásokat sem. A teszt működési logikája, hogy az általunk becsült eredményváltozó négyzetes

és harmadik hatványra emelt értékét hozzáadja magyarázóváltozóként a modellünkhöz. A teszt nullhipotézise: az újonnan beillesztett nemlineáris hatások regressziós együtthatója a modellben nulla, tehát modell specifikációja megfelelő, nem igényel további nemlineáris tagokat. Az alternatív hipotézis, hogy a modell specifikációja nem megfelelő, mert az újonnan bevett nemlineáris tagok együtthatója nem nulla, tehát van szignifikanciaszintjük, ezért van valamilyen nem megfigyelt, lineáristól eltérő hatás. Amennyiben a modell nem jól specifikált, a nemlineáris hatások torzítják a lineáris függvény becslését, ezért nem fog megvalósulni a korábban tisztázott torzítatlan becslés, a modell elveszti a BLUE – tulajdonságát (Wooldridge, 2013).

Interakciók

A specifikáción különböző nemlineáris tagok bevonásával, például interakciókkal lehet javítani. Az interakció a lineáris modell additivitását oldja fel. Amikor interakcióba vonunk két változót egy regressziós modellben, akkor feltételezzük, hogy az egyik marginális hatását befolyásolja a másik szintje és fordítva (az interakció szimmetrikus). Ezt olyan változók esetében érdemes megtenni, amelyek hatása eltérhet egy másik magyarázóváltozó által felvett értékek függvényében (például kor és tapasztalat). A marginális hatás jelentése, hogy amennyiben a magyarázóváltozót (nem feltétlenül egységgel) megnöveljük, mekkora az eredményváltozó változása az egységnyi magyarázóváltozó – változtatásra vetítve. Matematikai szempontból, a marginális hatás a modell adott magyarázó szerinti parciális deriváltja. Lineáris esetben a marginális hatás éppen megegyezik a magyarázóváltozó modellben szereplő koefficiensével, ugyanis ebben az esetben azt éppen egységnyit kell növelnünk, hogy egységnyi eredményváltozó – megváltozást érzünk el átlagosan (James és munkatársai, 2021).

Alkalmazott eljárások a heteroszkedaszticitás mérésére

A korábban említett hat standard modellfeltevések közé tartozik a homoszkedaszticitás feltétele, amely kimondja, hogy a regressziós modell hibatagjának szórása legyen állandó, tehát a kilengések ne függjenek a becsült értékek nagyságától. Amennyiben ez nem teljesül, egyfelől a modell hibatagjainak becslése inkonzisztenssé és torzítottá válik, továbbá a regressziós együtthatók parciális szignifikanciatesztjeinek konfidencia-intervallumai is érvénytelenné válnak, mert a standard hibák nem normális eloszlásúak lesznek (ez a korábban említett hat standard modellfeltevés egyike, amely így nem teljesül) (Wooldridge, 2013).

White – teszt

A White – teszt logikája azon a feltevésen alapul, hogy amennyiben a modell homoszkedasztikus, a hibatagok nem függenek a magyarázóváltozóktól, sem pedig azok kvadratikus értékeitől. Ebből kifolyólag egy segédregressziót hoz létre, melyben a regresszió hibatagjainak négyzetét magyarázza az eredeti modellben szereplő magyarázóváltozókkal és azok négyzetre emelt értékeivel. A teszt ezután jobboldali Khi – négyzet próbával teszteli, hogy a segédregresszió determinációs együtthatója szignifikáns -e a mintán kívül is. H_0 esetén a segédmodell nem szignifikáns, tehát az eredeti modell magyarázóváltozói nem magyarázzák a hibatag négyzetét, így homoszkedaszticitás áll fenn. Az alternatív hipotézis állítása ezzel szemben, hogy a segédregresszió szignifikáns, tehát van összefüggés a magyarázóváltozók és a hibatag közt, így az eredeti modell heteroszkedasztikus. A White – tesztnek létezik további, interakciókkal kibővített verziója is, amelyben a hibatag négyzetét magyarázó segédregresszió nemcsak az eredeti magyarázóváltozókat, hanem azok interakcióit is tartalmazza, minden kombinációban. A hipotézisvizsgálat ezután hasonló elven működik, mint az interakció nélküli esetben (Kleiber és Zeleis, 2008).

Breusch – Pagan próba

A Breusch – Pagan próba a White – teszt egy változata, ami egy olyan segédregresszióra épít, amely csak az eredeti magyarázóváltozókkal magyarázza a hibatag négyzetét (tehát a magyarázóváltozóknak nincs kvadratikus értelemben transzformált, sem pedig interakciós verziója sem benne). Ezek a változók kiválaszthatók a sejtés alapján, hogy melyik okozza a heteroszkedaszticitást. A hipotézisvizsgálat ezután hasonló a White – tesztéhez: ha a modell homoszkedasztikus (H_0 esetén), akkor a segédregresszió összes béta paramétere nulla, tehát nem szignifikáns, hiszen ekkor a magyarázott hibatag maga a konstans. Az alternatív hipotézis esetén a segédregresszióknak van olyan becsült együtthatója, amely szignifikáns, tehát a hibatag függ a magyarázóváltozóktól, vagyis a modell heteroszkedasztikus (Wooldridge, 2013).

A döntési fa

A döntési fát az elemzésben abból a célból építettem, hogy megvizsgáljam, van -e valamilyen általánosítható mintázat abban, hogy kik azok, akik várhatóan megadják vagy nem adják meg a fizetésüket. A döntési fa az egyik leggyakoribb, adatbányászatban is használt klasszifikációs algoritmus, amely az egyedek sokaságon belüli 0 - 1 részcsoportokhoz való tartozását becsli (Abdulazeez és Jijo, 2021).

A döntési fa által meghatározott részcsoportokat leveleknek hívjuk. Egy döntési fa annál pontosabban becsül, minél homogénebb az eredményváltozó szerint egy levél, hiszen ekkor tudjuk az adott részcsoportot a legjobban körülírni és lehatárolni. A fát hiperparaméterekkel tudjuk hangolni, amellyel azt szabályozzuk, hogy hogyan végezze el a becslést az adott adathalmazon. A három leginkább használt hiperparaméter a mélység, levelek száma és komplexitás. A mélységgel azt állíthatjuk be, hogy maximum hány szintre bomoljon tovább a fa. Ha túl kis mélységet adunk meg, a fa nem tudja megfelelően homogén csoportokra bontani az adatbázis elemeit, viszont, ha túl nagyot, akkor pedig túlzottan elaprózódik a becslés nagyon kis, nem szignifikáns elemszámú levelekre. A levelek számának megadásával a részcsoportok számát lehet maximalizálni (Mantovani és munkatársai, 2016). A komplexitást többfajta módszer alapján beállíthatjuk, a leggyakoribb ilyen módszer (amellyel a legtöbb döntési fa alapbeállításként számol) a Gini index, amely 0 – 1 skálán méri azt, hogy adott statisztikai eloszlás mennyire homogén az eredményváltozó szerint (minél kisebb az érték, annál homogénebb az adott részhalmaz a teljes halmazon belül) (Dorfman, 1979). Az általam használt *rpart* csomag döntési fája is a Gini indexszel számolt. Amennyiben például a komplexitás értékét 0,01-re állítjuk, az azt jelenti, hogy a megadott hibafüggvény (jelen esetben a Gini index) legalább 0,01 -el kell, hogy javuljon egy új ág létrehozásánál, különben az új ág nem jön létre.

$$G = 1 - \sum_{i=1}^k g_i^2 \quad (12.)$$

A (12.) képletben bemutatott Gini index két paramétere k (a kategóriák száma) és g_i (a kategóriák relatív gyakorisága adott levélen).

A döntési fa algoritmusnak számos további, felügyelt gépi tanulásban is használt fajtája van, amelyek rengeteg egyéb hiperparaméterrel rendelkeznek, ugyanakkor ezek jelenleg nem állnak a kutatásom fókuszában.

A döntési fa kiértékelésének módja

A döntési fa – ahogy azt korábban említettem – egy klasszifikációs algoritmus, amely egy adott csoporthoz való tartozást becsül, amelyet egy Igen – Nem (1 – 0) jelentéstartalmú eredményváltozóval fogalmazhatunk meg. A modell kiértékeléséhez a klasszifikációs (konfúziós) mátrixot alkalmazhatjuk, amelyben összevethetjük, hogy a modell mennyire jól becsülte a valódi egyeseket és nullákat (Powers, 2007).

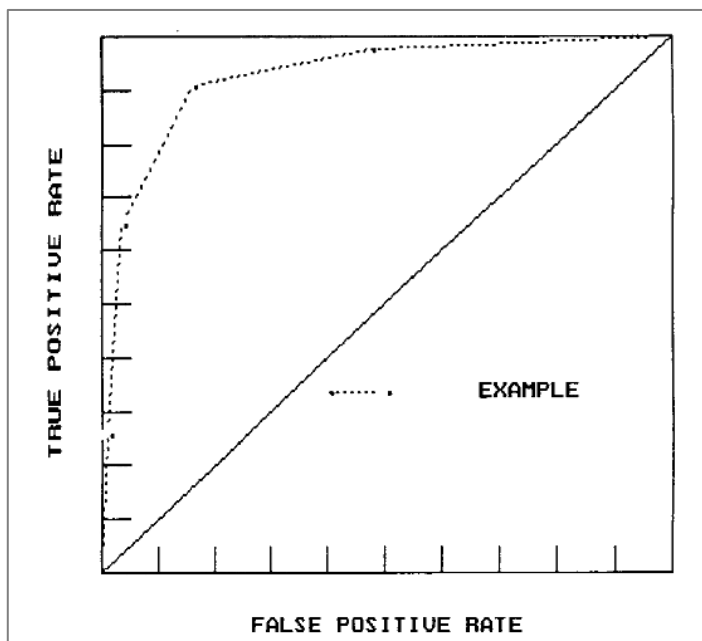
		Valós értékek	
		Nem (0)	Igen (1)
Becsült értékek	Nem (0)	Helyesen eltalált hamisak	"Miss": Helytelenül hamisnak becsült igazak (másodfajú hiba)
	Igen (1)	"False alarm": helytelenül igaznak becsült hamisak (elsőfajú hiba)	Helyesen eltalált igazak

5. ábra: A klasszifikációs mátrix bemutatása. Forrás: Powers (2007). Saját készítés.

Eszerint többfajta értékeléshez használt mutatót is képezhetünk, melyekből Tharwat (2018) a következőket emeli ki:

- *Pontosság (Accuracy)*: A helyesen becsült egyesek és nullák aránya az egész adathalmazban. Megmutatja, hogy a modell általánosságban nézve mennyire találta el az igazakat és hamisakat helyesen.
- *Szenzitivitás (Sensitivity)*: A valódi igazak aránya, azt mutatja meg, hogy az összes egyesből hány százalékot talált el a modell.
- *Specifitás (Recall)*: A valódi hamisak aránya, amely az előzőhöz hasonlóan a helyesen eltalált nullák arányát mutatja meg a mintában található összes nullához viszonyítva.

A fentebb említett mutatók mind különböző perspektívából értékelik a döntési fa modellünket és kontextustól függően más lehet éppen fókuszban, amelyen javítani szeretnénk (például egy csődelőrejelzésben fontosabb, hogy a helytelenül hamisnak sorolt igazak számát minimalizáljuk, tehát maximális szenzitivitásra kell törekednünk). Amennyiben viszont összehasonlítani is szeretnénk különböző modelleket, a ROC görbe a számunkra alkalmas viszonyítási alap.



14. ábra: A ROC görbe bemutatása. Forrás: Centor (1991).

A ROC (Receiver Operating Characteristic) görbe bináris (0 - 1) eredményváltozókat becslő modellek kiértékelésére szolgáló grafikus ábra. A görbe két tengelye a szenzitivitás (valódi igazak aránya) és a specificitás komplementere (hamisan becsült igazak aránya). A tengelyek közti 45°-os egyenes a véletlen találgatást jelenti, az ábra bal felső sarka pedig a tökéletesen becslő modellt, ahol minden „igaz” elemet helyesen azonosítunk anélkül, hogy tévesen egy „igaz” elemet „hamisnak” detektálnánk. A görbe minél inkább távolodik el a 45°-os egyenestől és közelíti meg a bal felső sarkot, annál megfelelőbben klasszifikál. A bal felső sarok megközelítését a görbe alatti területtel (area under the curve, a továbbiakban AUC) írhatjuk le, amelyet százalékosan értelmezünk. A tökéletesen becslő modell AUC-ja 100%, a 45°-os egyenesé 50%, két tetszőleges modell közül pedig klasszifikáció szempontjából az a megbízhatóbb, amelynek magasabb AUC értéke van (Fawcett, 2005).

A magyarázóváltozók átalakítása

A munkakörök esetén a válaszadóknak számos válaszlehetőség állt a rendelkezésére, valamint egyszerre több pozíciót is megjelölhettek. A leggyakoribb munkakörök mind a webfejlesztéshez kapcsolódtak, amelyet a leíró statisztikai részben is kifejték. Azokat a válaszadókat, akik egyszerre jelölték a full – és backend fejlesztői állásokat, a full – stack fejlesztők közé soroltam. A full-stack, front-end és back-end webfejlesztés között gyakorlatilag az tesz különbséget, hogy ki melyik szegmensével foglalkozik a weboldalnak. A back-end fejlesztők felelnek azért, hogy a weboldal felhasználó számára nem látható része megfelelően, a kívánt logika szerint működjön, ők írják meg például az API (Application Programming Interface) végpontokat, amiken keresztül a front – és back end felület találkozik, amikor például a weboldal egy mögöttes adatbázisból jelenít meg a felhasználónak adatokat. A front – end fejlesztők ezzel ellentétben a weboldal designjáért és a könnyedén használható felhasználói felületért felelősek. A full – stack fejlesztők azok, akik egyszerre foglalkoznak a front – és back - end fejlesztéssel is (Wilkins, 2021).

Ezen kívül létrehoztam egy olyan ismérvet is, amely a data scientist, data engineer és data analyst állásokat betöltő embereket foglalta magába. Ebben az esetben, ahol ez az érték 1, ott a kitöltő megjelölt legalább egyet vagy többet ezekből a pozíciókból, 0 esetén pedig egyiket sem jelölte, vagy egyáltalán nem válaszolta meg a kérdést. Ezeken kívül külön változókat kaptak a DevOps specialisták és termékmenedzserek, mobil – és játékfejlesztők, adatbázis – és rendszeradminisztrátorok és akadémiai kutatók, óraadó oktatók. A 0 – 1 kódolásnak itt is hasonló jelentése volt, mint ahogy azt korábban már tisztáztam.

Az egyéb technológiai magyarázóváltozók átalakítása

A *Stack Overflow Annual Developer Survey* kérdőívében minden évben vannak kérdések az egyén által ismert és munkában használt különböző technológiákról, amelyek a következő kategóriákba sorolhatók (mindhez 1-1 kérdés tartozott a kérdőívben): ismert programnyelvek, ismert adatbázistechnológiák, ismert platformok és ismert keretrendszerek. Mindegyik kategóriában a már ismert technológiákon túl volt egy – egy olyan kérdés is, amely arra irányult, hogy melyek azok az adott kategórián belül, amelyeket a kitöltő a későbbiekben szeretné elsajátítani. Ezeknél a kérdéseknél a kitöltő bármennyi válaszlehetőséget megjelölhetett.

A nyelvek kiválogatásakor elsősorban a TIOBE – indexet vettem figyelembe. A mutató egy hollandiai központú vállalat, a TIOBE mérőszáma. A cég többek között különböző minőségi tanúsítványokat bocsát ki a szoftverfejlesztéssel foglalkozó vállalatok egyes termékeire. A

TIOBE – index elsősorban azt veszi figyelembe, hogy az adott programnyelvek milyen rendszerességgel fordulnak elő különböző keresőmotorokban (pl: Google, MSN, Yahoo!, Wikipedia és YouTube). Az újrakalkulált adatokat havi rendszerességgel frissítik, amelyek honlapjukon ingyenesen elérhetők, bár a hosszabb távú, statisztikai mérésekhez felhasználható adatokat csak megvásárolni lehet. A TIOBE – index csak a Turing - teljes nyelveket veszi figyelembe, ezért például a HTML -t vagy XML -t nem kezeli programnyelvként és nem is vizsgálja őket (TIOBE, n.d.).

A TIOBE – index által 2022 szeptemberében világszerte 20 leggyakoribb nyelvnek ítélt programnyelv közül azoknak hoztam létre külön változót, amelyeknek a mintában legalább száz elemű volt a gyakorisága. Az elsőnek kiválogatott nyelvek a C#, SQL és Python voltak, gyakoriságuk a teljes adatbázisra nézve rendre 388, 680 és 499 volt. Ezen kívül külön kategóriát képeztem a webfejlesztéshez használt nyelveknek, amelybe egybevonam őket, mivel ezeket a kitöltők jellemzően együtt jelölték meg. Ebbe az ismervbe a HTML, CSS, PHP, JavaScript és TypeScript nyelvek lettek egybevonva, így itt az 1 ismérvkimenet jelentése az, hogy a kitöltő megjelölt legalább egyet vagy többet ezen nyelvek valamelyike közül. E ismérvkategória gyakorisága a teljes adatbázisban 978 volt. További két ismérvet képeztem még egymáshoz közel álló nyelvek egybevonásával: a C és C++ nyelveket, illetve a Bash, Shell, PowerShell nyelveket vontam össze 1 – 1 változóba. A C -t vagy C++ -t ismerők gyakorisága az összesített táblázatban 327, míg a Bash, Shell vagy PowerShell nyelvek valamelyikét 490 – en jelölték meg úgy, hogy dolgoztak már vele. Végezetül a Java és Assembly nyelvet ismerőket különítettem el egy - egy külön változóba. Az Assembly nyelv különválasztásának oka az a személyes feltételezésem volt, hogy ez az a nyelv, amely a legközelebb áll a gépi kódoláshoz, és úgy gondoltam, hogy amennyiben valaki ismeri, az biztosan szignifikánsan befolyásolhatja a bérét. Minden nyelv – változónak (és általánosságban nézve minden technológiai változónak) hasonló módszerekkel megképeztem a „*Desire*” elnevezésű párját arra vonatkozólag, hogy az illető szeretné -e az adott technológiát a közeljövőben elsajátítani. A 0 és 1 jelentése ezen változók esetén is hasonló volt az előzőekhez. Azon nyelvek, amelyek nem kaptak külön változót, az Other kategóriába kerültek. Itt a 0 jelentése a következő: a kitöltő nem jelölt meg a fentebb említett nyelvektől különböző nyelveket, vagy pedig üresen hagyta az ismert nyelvek mezőjét, tehát nem jelölt meg semmit sem. Az 1 jelentése pedig, hogy használ legalább egy, az előzőleg felsorolt nyelvektől különböző programnyelvet. Az Other kategóriát ezután hasonlóan létrehoztam a vágyott nyelvekre is.

Az ismert adatbázistechnológiák esetén szintén az adatokban lévő gyakoriságokat és a népszerűséget vettem figyelembe. Az SQL, mint adatbázishoz használt technológia itt is megjelent, bár biztosan van benne átfedés az SQL-t, mint ismert nyelvet megadó emberekkel. A kettő közt a különbség az volt, hogy a nyelv változó esetén csak az SQL, mint lekérdezőnyelv ismeretének megadására volt lehetőség, míg az adatbázistechnológiával kapcsolatos kérdések esetén olyan válaszok közül lehetett választani, amelyek az SQL-n, mint szabványon alapuló relációs adatbáziskezelő rendszerek voltak (pl.: PostgreSQL, SSMS, MySQL ... stb). Utóbbiakat az adatbázistechnológiákra vonatkozó változók esetén összevontam egy SQL_Database változóba, amelynek így a „*WorkedWith*” változó esetén 1598, míg a „*Desire*” oszlopban 1338 volt a gyakorisága. Az adatbázisban ezen felül fennmaradó ismérvkimenetek mind NoSQL technológiák voltak, amelyek már nem a relációs sémákon alapulnak. Ezek közül szintén azokat emeltem ki külön változóba, amelyek feltételezéseim szerint népszerűek és az adatbázisban is megfelelően nagy számosságuk volt: MongoDB, Redis és Elasticsearch.

A MongoDB saját MQL (Mongo Query Language) nyelvet használ, amely a JSON struktúrájához hasonló, beágyazott lekérdezések írásához tud segítséget nyújtani. Ez egy dokumentum - orientált nyelv, amely az adatokat rugalmasabb és skálázhatóbb formában tárolja, manipulációkor nincs szükség előzetesen definiált sémára (mongodb.com, n.d.). A Redis egyik fő tulajdonsága, hogy „in – memory” tárolja az adatokat, ezáltal a beolvasás – kiírás sokkal gyorsabbá válik. Emellett a „distributed caching” technikának köszönhetően, amennyiben a kliens által lekért adat megtalálható a gyorsítótárban, azonnal ki is szolgálja a kérést, egyébként lekérdezi az adatokat az adattárházból és eltárolja őket a gyorsítótárban későbbi lekérdezésre (redis.io, n.d.). Az Elasticsearch főleg üzleti és logfile adatok elemzésében használatos NoSQL nyelv, amely az Elastic által fejlesztett ELK stack (Elasticsearch, Logstash és KIBANA) része. Ezen háromból a KIBANA egy adatvizualizációs felület, míg a Logstash például a különböző rendszer logadatok esetén is használható adatmenedzsment rendszer, amely az adatok összegyűjtését, rendszerezését, transzformálását és áramlását optimalizálja (elastic.co, 2023). A további, mintában kis gyakoriságú adatbázistechnológiák az egyéb NoSQL technológiákba lettek sorolva, ugyanis az SQL szabványon alapuló technológiákat már lefedte az erre vonatkozó változó. Így ennek a változónak a jelentése 1 esetén az, hogy az illető megjelölt a fenti nem relációs adatbázistechnológiáktól eltérő NoSQL nyelvet azon kérdésnél, hogy használja – e („*WorkedWith*”) vagy hogy a későbbiekben használni szeretné – e („*Desire*”).

Az ismert keretrendszerek esetén – hasonlóan az előzőekhez – szintén az adatbázisban megtalálható gyakoriságra és a népszerűségre támaszkodtam. Egyfelől egy változóba vontam a .NET és ASP .NET keretrendszereket, valamint külön oszlopokat hoztam létre a Django, Spring, Angular, React.js és jQuery frameworkok számára. A .NET, ASP.NET keretrendszerekkel Microsoft .NET platformokkal kompatibilis desktop vagy webes felületű alkalmazásokat fejleszthetünk főleg a backend oldalról (learn.microsoft.com, 2022, 2023). A Django Python nyelvbe leforduló egyszerűbb webalkalmazások backend komponenseinek megírására alkalmas, bár a frontend felületet is tudjuk vele szerkeszteni (djangoproject.com, 2023). A Spring Java alapú alkalmazásfejlesztésre használható, szintén inkább a backend oldal tekintetében (docs.spring.io, n.d.). Az Angular a Google, míg a React.js a Facebook által kezelt frontend fejlesztésre használható open source keretrendszerek (angular.io, 2023; react.dev, 2023). A jQuery szintén frontend oldalon a fejlesztés mellett a frontend oldali kompatibilitási problémák kiküszöbölésére alkalmas (api.jquery.com, n.d.). A Vue.js, Node.js és Express.js ismérvkimeneteket egy közös kategóriába vontam, ugyanis ezek bár egyenként kevésbé voltak gyakoriak, mindhárom JavaScriptbe leforduló webalkalmazások fejlesztéséhez használatos keretrendszer. A három együttes használata például a full – stack fejlesztők számára hasznos, akik a Vue.js segítségével tudják az interfaceket megalkotni, majd a Node.js-t és Express.js-t a backend folyamatok és logika megképezéséhez használhatják (expressjs.com, n.d.; nodejs.com, n.d.; vuejs.org, n.d.).

Az utolsó technológiához tartozó változó a munkához használt operációs rendszer volt, amelyet három külön változóra bontottam: Linux, Windows és macOS felhasználók.

Végezetül a Stack Overflow használati szokásokhoz kapcsolódó ismérvek maradtak, amelyekből három került be a kiinduló adattáblába: *SOParticipate*, miszerint az illető milyen gyakorisággal vesz részt a Stack Overflow kérdezz – felelek fórumán, *SOVisit*, amely azt méri, milyen rendszerességgel látogatja a weboldalt és *SOHasAccount*, amelynek jelentése, hogy a kitöltőnek van -e felhasználója az oldalon.

15.2. Mellékelt ábrák

Modellszelekció

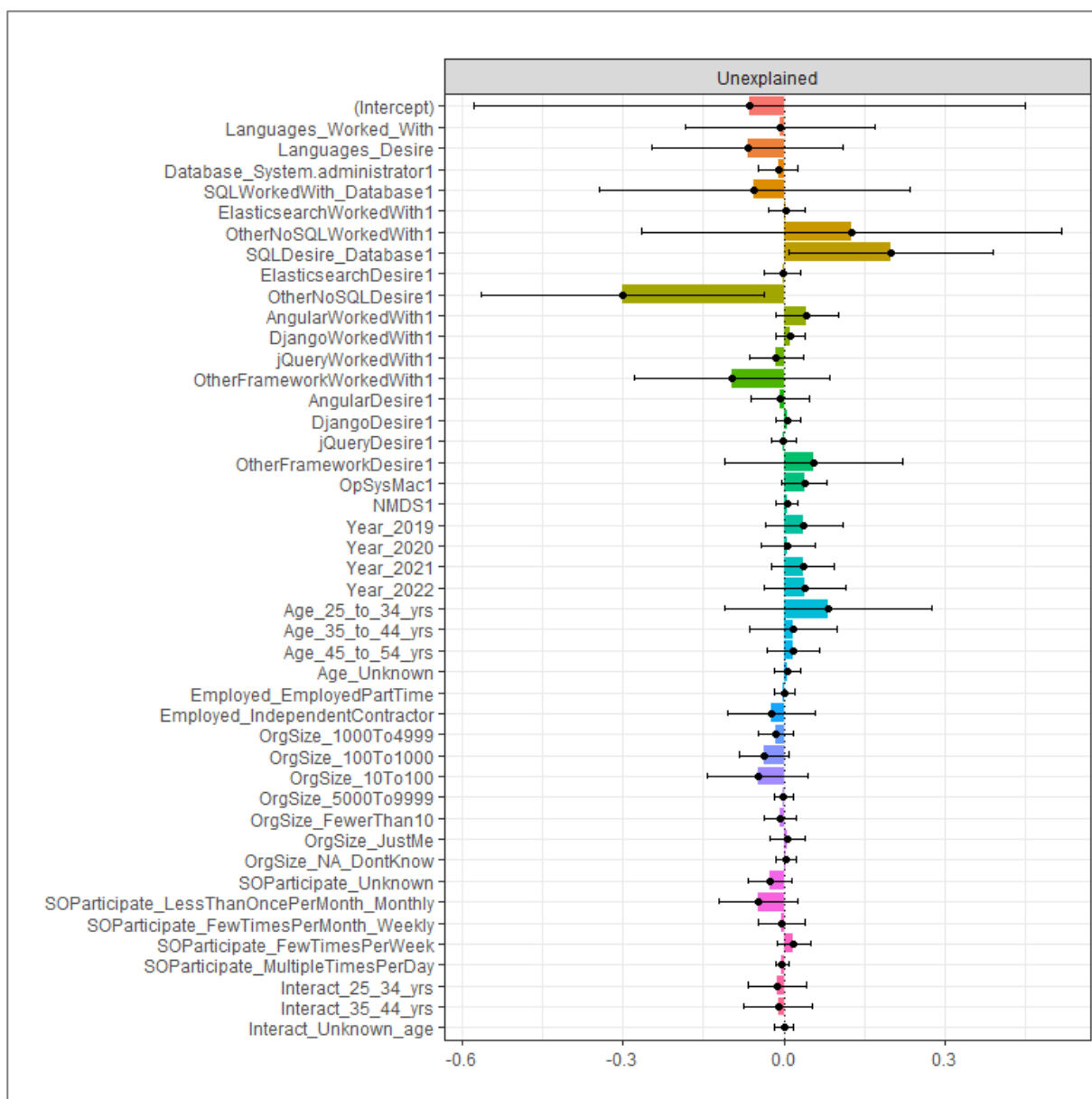
Állapot	Modell változójának neve az R kódban	Leírás	Kihagyott változók (előző állapothoz képest)	Változók száma (tengelymetszet nélkül)	AIC	BIC	Wald - teszt p-érték
1	model_full_log	Kiinduló modell, amely az összes változóval (kivéve a deviza) becsülte a logaritmált béreket	Currency	110	2391	2970	-
2	model_szelektalt_3	Első szűkített modell, amelyben csak a p - érték alapján nem szignifikáns változókat szelektáltam	Sexuality, Ethnicity, Gender, SOHasAccount, SOVisit	93	2387	2878	4,77%
3	vegso_szelektalt	Végso szűkített modell	Az összes programnyelvi változó (WorkedWith és Desire, összesen 18db) + az összes platform változó és (WorkedWith és Desire, összesen 20db) és később az ismert és ismerni vágyott platformok száma is (2db) + az összes betöltött pozíció változója (7db), kivéve: Database_System_Administrator + OpSysWindows, OpSysLinux, MongoDBWorkedWith, MongoDBDesire, RedisWorkedWith, RedisDesire, SpringWorkedWith, SpringDesire, .NET/ASP.NET_WorkedWith, .NET/ASP.NET_Desire, NodeJS/VueJS/ExpressJSWorkedWith, NodeJS/VueJS/ExpressJSDesire, ReactWorkedWith, ReactDesire	70	2364	2736	55,62%

1. melléklet: A modellszelekció főbb lépései részletesen.

Blinder – Oaxaca dekompozíció

A diploma hatásainak vizsgálata

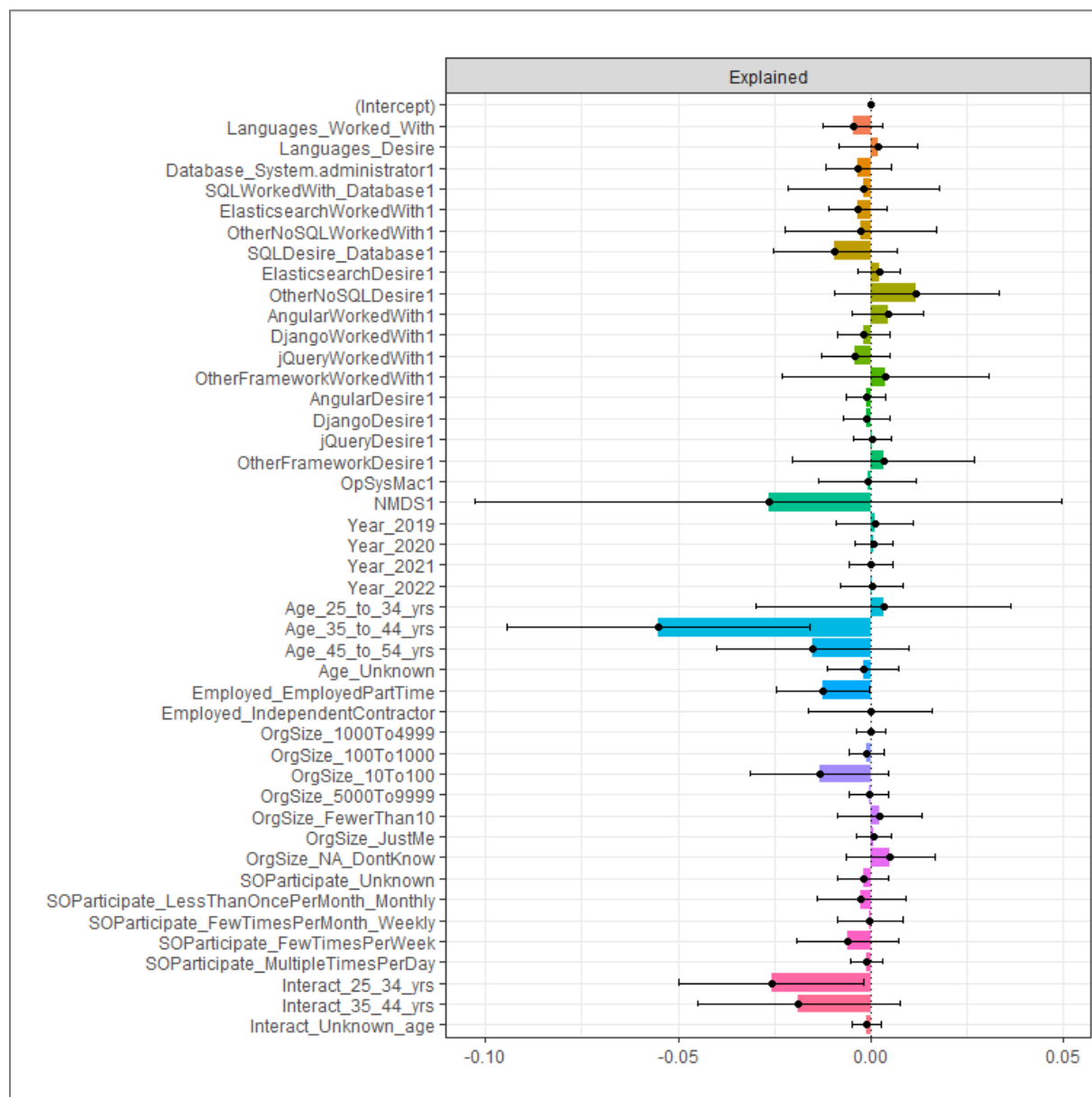
Részhatások



2. melléklet: A diploma hatásainak vizsgálata, részhatás.

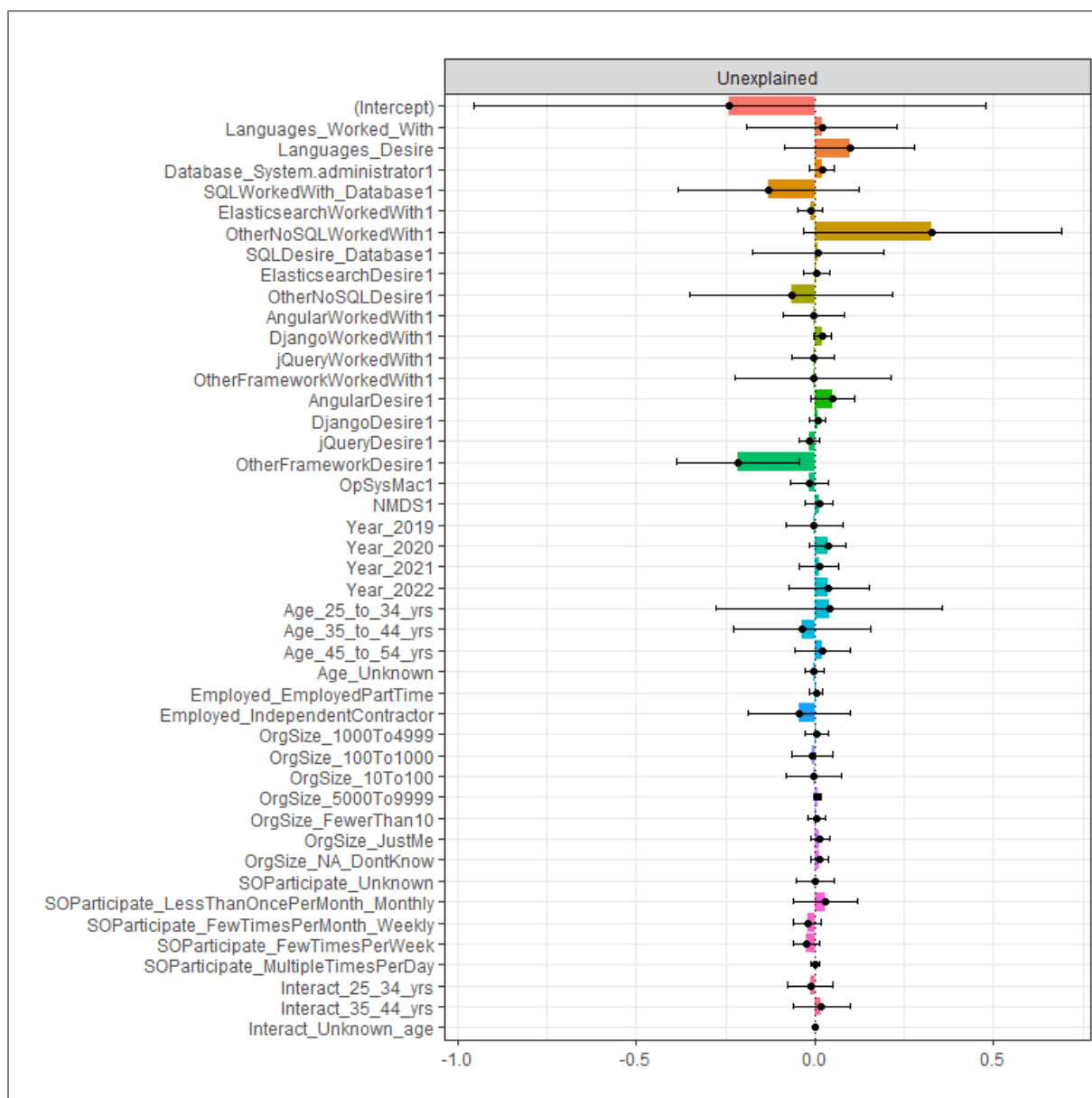
Alapszakos és mesterdiploma közti különbségek

Összetételhatások



3. melléklet: Alapszakos és mesterdiploma közti különbségek, összetételhatás.

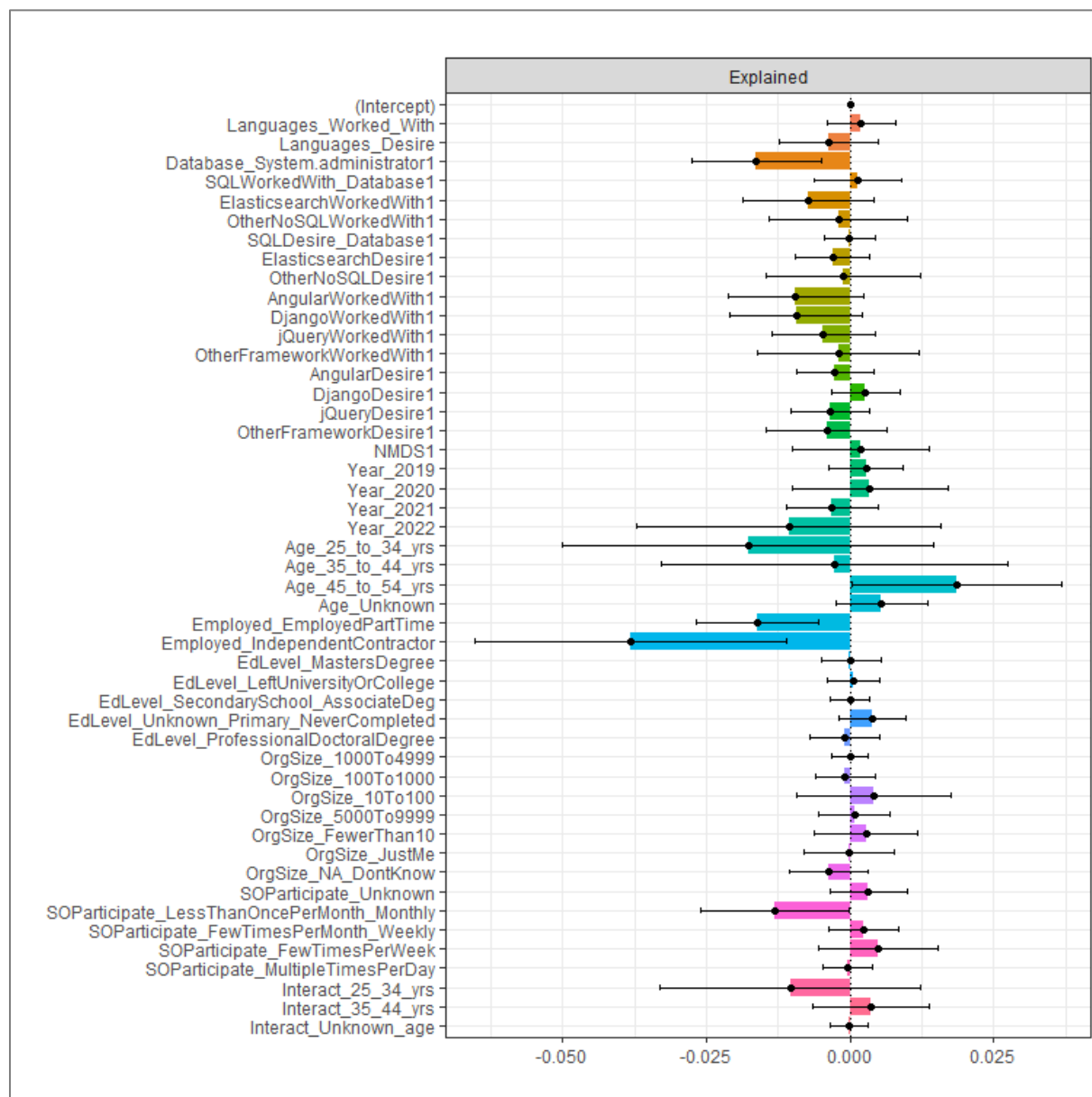
Részhatások



4. melléklet: Alapszakos és mesterdiploma közti különbségek, részhatás.

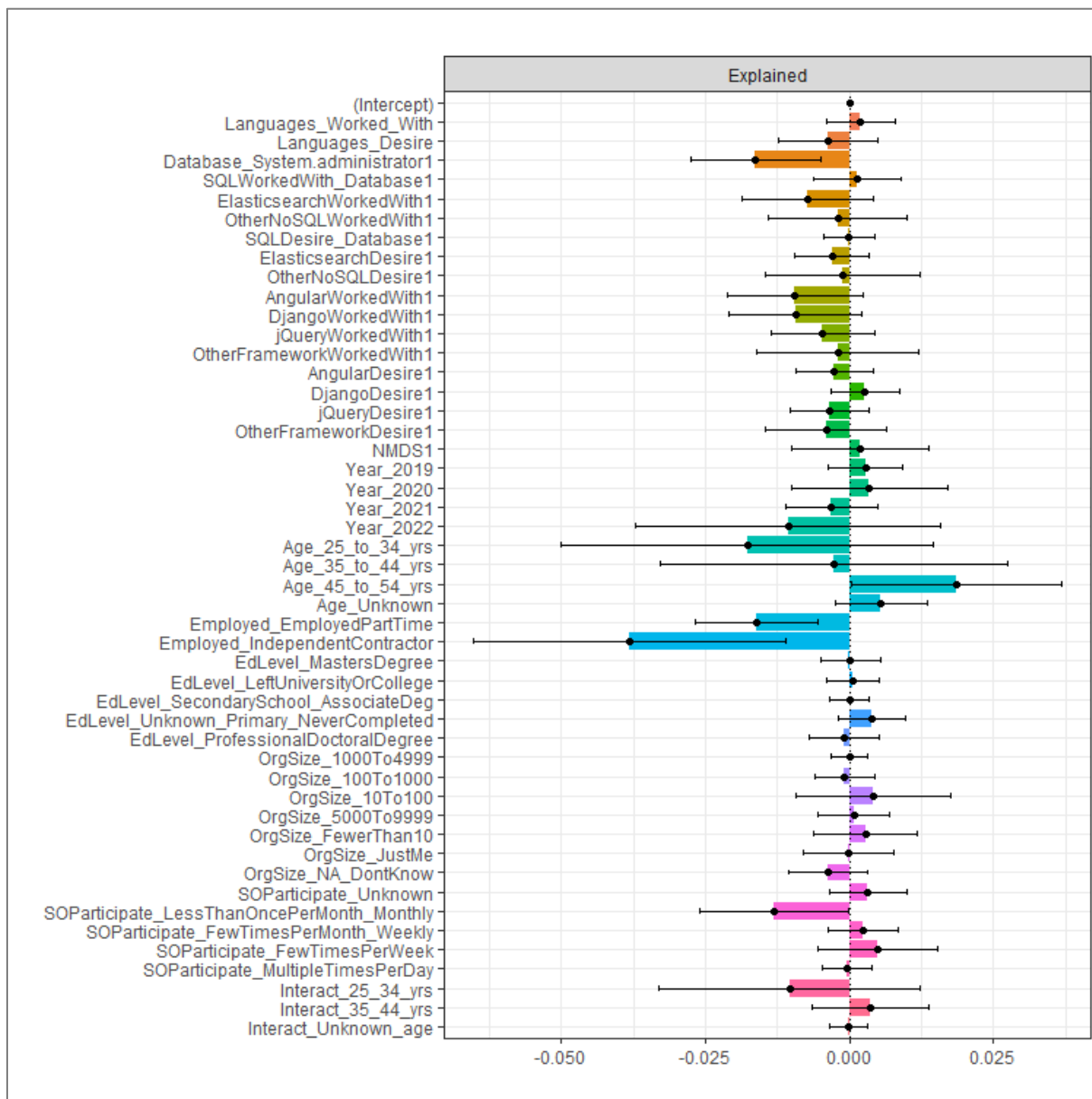
A macOS -t használók bérezése a többi operációs rendszert használóhoz képest

Összetételhatások



5. melléklet: A macOS -t használók névezése a többi operációs rendszert használóhoz képest, összetételhatás.

Részhatások



6. melléklet: A macOS-t használók névezése a többi operációs rendszert használóhoz képest, részhatás.