Eötvös Loránd Tudományegyetem

Természettudományi Kar

Időfüggő preferenciaadatok modellezése

Berger Máté

BSc szakdolgozat

Témavezető:

Prőhle Tamás

Valószínűségelméleti és Statisztikai Tanszék
Matematikai Intézet

Budapest, 2017

Tartalomjegyzék

1. Köszönetnyilvánítás

Szeretnék köszönetet mondani témavezetőmnek, Prőhle Tamásnak, hogy elvállalta a konzulensi teendőket, akihez bármikor fordulhattam. Ötletei és lelkesedése nagyban hozzájárult szakdolgozatom elkészültéhez.

Továbbá köszönetet szeretnék mondani családomnak és barátaimnak, akik tanulmányaim alatt végig biztattak és támogattak.

2. Bevezetés

Az idősorok modellezése napjaink egyik legfontosabb statisztikai területe. Szabályosságot, ok-okozati összefüggéseket tárunk fel, statisztikai módszerek segítségével, mely segít megmagyarázni az idősor viselkedésének miértjét. Az esetek részében a jövőre vonatkozó előrejelzés is célunk.

Dolgozatomban teniszmérkőzések győzteseinek kilétét vizsgálom a múltbéli, adataik segítségével. A választás elsősorban személyes érdeklődés alapján esett erre a konkrét feladatra, de a rendelkezésre álló adatok mennyisége is számított. Magam is lelkes teniszjátékos vagyok, így a játék oldaláról is van némi rálátásom, mely segített a releváns és nem releváns adatok szétválasztásában. Népszerű sport lévén jól dokumentált adatok álltak rendelkezésre.

A konkrét idősor specialitása, hogy folytonos, diszkrét és minősítő változókat is tartalmaz, így nem elemezhető a klasszikus módszerekkel. Az adatok letisztázása és megfelelő formára alakítása után, az erre a célra megfelelő általánosított lineáris modellt, a GLM-et használtam. A modell kiértékelte a releváns változókat, majd a dolgozat végén a kiértékelt modellek alapján összegzésre kerültek a kapott eredmények.

3. Alkalmazott módszer áttekintése

Ebben a fejezetben a dolgozat szempontjából lényeges elméleti ismeretek kerülnek áttekintésre. A fejezetben a következő források lettek felhasználva : [1], [2], [3], [4], [5], [6], [7]; pontos megnevezése a dolgozat végén, a felhasznált irodalom fejezetben találhatók.

3.1. Regresszió

A regressziónak – a legegyszerűbb esetben – a rendelkezésre álló $(y_k, x_k), \quad k = 1, ..., n$ méréspárok értékelésekor az a feladata, hogy megtalálja azt az $a, b \in \mathbb{R}$ számpárt, amelyre az

$$y_k = a + bx_k + e_k, \quad k = 1, \ldots, n$$

egyenletek teljesülnek olymódon, hogy a mérések hibáit jelölő $e_k, \quad k = 1, \ldots, n$ értékek $\sum_{k=1}^{n} e_k^2$ négyzetösszege minimális legyen. A regressziós modellben az y_k értékeket célváltozóknak, az x_k értékeket pedig magyarázó változóknak nevezzük. Kiterjesztve pedig az egyenletek jobb oldalán p darab magyarázó változó van, az (y_k, x_k) pedig bővebb, $(y_k, x_{k,1}, \ldots, x_{k,p})$ formában jelenik meg. A mérésekre az

$$y_k = a + b_1 x_{k,1} + ... + b_p x_{k,p} + e_k, \quad k = 1, \ldots, n$$

egyenletek érvényesek. Általában a lineáris modellek magukban foglalnak determinisztikus és véletlen komponenst is.

$$Y_{i,j} = \alpha_i + \beta u_{i,j} + \gamma_j v_{i,j} \quad (i = 1, \ldots, n, \quad j = 1, \ldots, p),$$

ahol α_i n db diszkrét változót, $\beta u_{i,j}$ n db folytonos változót, $\gamma_j v_{i,j}$ n db független változót jelent. Általános lineáris modellnek azt a modellt nevezzük, amikor a megfigyelt változókra az

$$Y = X\beta + \epsilon$$

egyenlet mellett azt feltételezzük, hogy az ϵ hibák nulla várható értékűek, normális eloszlásúak, de nem feltétlenül azonos szórásúak, és nem feltétlenül függetlenek.

Több oka lehet annak, hogy egy regressziós feladat megoldása során a legkisebb négyzetek módszerét és a normalitás feltételezését is elhagyjuk. Példa:

- A legkisebb négyzetes módszer a közelítés hibáit négyzetesen veszi, és így a kilógó értékek túlságosan 'elhúzzák' a modellt, a négyzetes hiba – a szimmetria okán – a pozitív és a negatív hibákat azonos súllyal tekinti, miközben az adatokra az egyik irányú eltérés jellemzőbb, mint a másik.

3.2. Általánosított lineáris modell (GLM)

Az általánosított lineáris regresszió modellben az $X\beta$ prediktornak egy olyan, megfelelő $h : \mathbb{R} \to \Theta$ függvény szerinti leképezését vesszük, ahol a Θ megegyezik azzal a halmazzal, ami a célváltozó eloszlás paramétereiből áll, ami a legegyszerűbb esetben a célváltozó adott körülmények közti várható értéke. Vagyis alapesetben a

$$E(Y|X) \approx h(X\beta)$$

közelítés megoldását keressük. Ezt a h függvényt nevezzük az általánosított lineáris regresszió átviteli függvényének. A szokásos jelölés szerint g függvény, amelyre tehát

$$g(E(Y|X)) \approx X\beta.$$

Ezt a g függvényt nevezzük a regresszió kapcsoló (link) függvényének. Ha a célváltozó feltételezhető eloszlásai exponenciális eloszláscsaládot alkotnak, akkor a Newton-Raphson iterációt a megfelelő likelihood maximalizálására alkalmazva, a Fisher scoringnak vagy iterated weighted least squarenek (IWLS, iteratív súlyozott legkisebb négyzetek módszere) nevezett eljárás használatos. A modell illesztési folyamatában, minimális modelltől – mely a lehető legkevesebb változót tartalmazza – lépésenként jutunk el a teljes modellig.

A maximalizálandó függvény négy speciális eloszlásra a következő:

- Normális eloszlás:

$$\sum (z - \widehat{\mu})^2 / \sigma^2$$

- Poisson eloszlás:

$$2[\sum z ln(z/\widehat{\mu}) - \sum (z - \widehat{\mu})]$$

- Binomiális eloszlás:

$$2[\sum z ln(z/\widehat{\mu}) + \sum (n - z) ln[(n - z)/(n - \widehat{\mu})]]$$

- Gamma eloszlás:

$$2p[-\sum ln(z/\widehat{\mu}) + \sum (z - \widehat{\mu})/\widehat{\mu}]$$

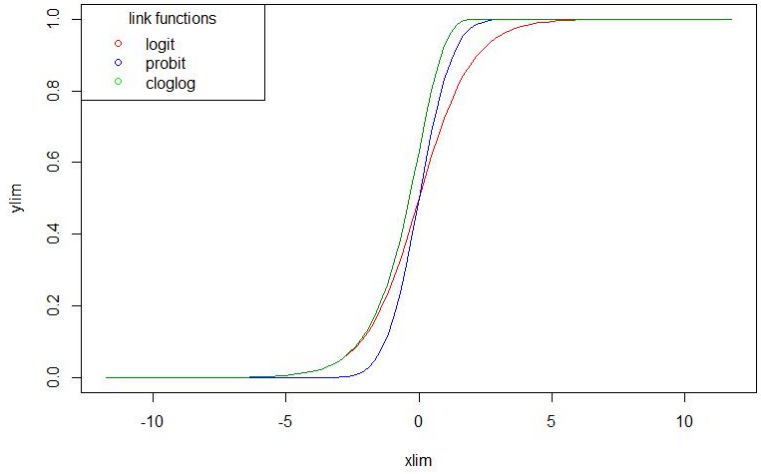
3.3. Kapcsoló függvények

Következzen a leginkább használatos kapcsoló függvényekről egy rövid áttekintés.

- log-log kapcsoló függvény

  R. A. Fisher angol statisztikus 1922-ben végzett vizsgálatához köthető, mikor azt a kísérletet írja le, ahogy higítással próbálja meghatátozni talaj illetve vízmintákban, bizonyos mikróbák jelenlétét vagy hiányát. Ehhez használta az általa alkotott függvényt és az alkalmazott likelihood számítást. Észrevette, hogy az organizmusok minden esetben Poisson-eloszlást követnek, így az x-edik higítást követően előállt a most használatos complementary $\log - \log(c \log - \log)$ függvény:

$$log(-log(1 - \pi_x)) = \alpha + \beta x.$$



- probit kapcsoló függvény

  Chester Ittner Bliss nevéhez fűződik, aki peszticidek hatékonyságát figyelte 1933-ban, a kártevők eltávolítása szempontjából. A dózis növelésével arányosan változott a hatékonyság, így a logaritmus segítségével megszületett a probit függvény. Probit regresszió esetén az eredeti célváltozó egy valószínűség, magyarázó változókkal az adott valószínűséghez a standard normális eloszlás szerint tartozó kvantilist, tehát egy $p \in [0, 1]$ valószínűség esetén a $\Theta^{-1}(p)$ értéket közelítjük.

$$Pr(Y = 1|X) = \Phi^{-}1(X^T\beta)$$

- logit kapcsoló függvény

  Joseph Berkson, orvos nevéhez fűződik, aki a probit alapján dolgozott, de a logisztikus valószínűséget használta a normál helyett. A logit regresszió esetén a célváltozó szintén egy valószínűség, és a magyarázó változókkal a valószínűség helyett a valószínűséghez tartozó esélyhányados (odds) logaritmusát igyekszünk a magyarázó változók lineáris függvényével közelíteni. Vagyis, ha a célváltozó szerinti valószínűség p, akkor a magyarázókkal közelített érték:

$$log(p/(1-p)).$$

3.4. Exponenciális eloszláscsalád

Egy–egy exponenciális családot a neki megfelelő $\vartheta(t)$, $T(t)$, $A(t)$, $B(y)$ illetve az $a(\varphi)$, $b(t)$, $c(y, \varphi)$ függvények határoznak meg, és

$$f_t(y) = exp(\vartheta(t)T(y) - B(t) + C(y)) = exp(\frac{\vartheta(t)T(y) - b(t)}{a(\varphi)} + c(y, \varphi))$$

formában írhatók fel. Ez azt jelenti, hogy egy-egy ilyen eloszlás logaritmált sűrűségfüggvénye a t paraméter ϑ és $B(t)$ függvényétől és az y megfigyelés $T(y)$ és $C(y)$ függvényétől egyaránt lineárisan függ.

GLM esetén, ha a célváltozó eloszlása exponenciális családbeli, valamint ha a modellben alkalmazott kapcsoló függvény ϑ, g-t alkalmazva

$$g(\mu) = \vartheta,$$

akkor mivel $b(\vartheta)$ ϑ szerinti deriváltja,

$$b'(\vartheta) = \mu,$$

a célváltozó várható értékére a következő is érvényes:

$$b' \equiv g^{-1} \equiv h.$$

Tehát, ha a magyarázó változók $l(X)$ lineáris függvényével ϑ-t közelítjük akkor az is érvényes, hogy

$$\mu \approx h(l(X))$$

3.5. Binomiális eloszlás

Belátható, hogy a binomiális eloszlás (n, p), $p \in [0, 1]$, rögzített n-re az exponenciális eloszláscsalád tagja.

Ha rögzített n-re $Y \sim Binom(n, p)$, akkor

$$P_p(Y = y) = \binom{n}{y} p^y (1-p)^{n-y} \quad y = 0, 1, 2, \ldots, n$$

ami felírható

$$P_p(Y = y) = \binom{n}{y} exp(log(\frac{p}{1-p})y + nlog(1-p))$$

formában is.

Tehát rögzített n-re a binomiális eloszlások egy exponenciális eloszlás családot alkotnak. A binomiális eloszláscsalád paraméterei a binomiális szokványos, p valószínűséggel paraméterezett esetben:

- $\vartheta(p) = log(p/1-p))$, kanonikus paraméter
- $T(y) = y$, elégséges statisztika
- $b(\vartheta) = nlog(1 + e^v artheta)$ partíció függvény
- $c(y) = log(\binom{n}{y})$

Az eloszlás tehát a természetes paramétere szerint felírva a következő:

$$P_\vartheta(Y = y) = exp(\vartheta y - nlog(1 + e^\vartheta) + log(\binom{n}{y}).$$

Az eloszlás paramétere tehát

$$\vartheta = log\frac{p}{1-p} \ .$$

4. Adatsor bemutatása, motiváció

4.1. Motiváció

Dolgozatomban az ATP World Tour, vagyis az Association of Tennis Professionals által szervezett teniszversenyek mérkőzéseit vizsgálom. A választás azért került erre az adatsorra, mert a sportok rengeteg érdekes adatsorral bírnak, és közülük ez illett személyes érdeklődési körömbe. A választás mellett szól az is, hogy ebben a konkrét témában nehéz fellelni hasonló elemzést, így várhatóan nem olyan eredményre jutunk, mely már közismert. A tenisz, mint globálisan népszerű sport az utóbbi 20 évben tett szert mai népszerűségére, főként a televíziós és internetes közvetítések elterjedésének, illetve a számos sportfogadási lehetőségeknek köszönhetően. A dolgozat célja kideríteni, hogy például a futballhoz hasonlóan fellelhetők-e olyan azonosságok, melyek matematikai, statisztikai szempontból segítenek eldönteni egy-egy mérkőzés kimenetelét.

4.2. ATP World Tour ismertetése

Az Association of Tennis Professionals által szervezett versenysorozatot ATP World Tournak hívják. 1973-tól vezet világranglistát, amelyen versenyzők megszerzett pontjaik alapján kapnak helyezést a listán. Megalakulása óta sok változáson ment keresztül. A sportfogadás terjedése - a futball mellett - talán a teniszt érintette a legelőnyösebben. A nagyrészt szponzorok által finanszírozott versenyek összdíjazása egyre nő, a legtöbb torna pedig akár élőben is követhető. Bár az ATP hivatalos adattára nehezen kezelhető, és további elemzésre alkalmatlan, az adatok sok internetes oldalon megtalálhatók, könnyen használható formában és szabadon felhasználhatók, alakíthatók.

4.3. Szabályismertetés

A dolgozatnak nem célja a szabályok teljes bemutatása, csak felületes, az adatok értelmezéséhez szükséges tudást ad.

A jelenlegi fontosabb szabályok a következők:

- Egy teniszmérkőzést ketten, illetve négyen (párosban) lehet játszani. A dolgozat csak a férfi, egyéni mérkőzésekkel foglalkozik.
- Egy mérkőzés két nyert szettig (Grand Slam versenyen, esetleg Davis kupa eseményen háromig) tart. Egy szett megnyeréséhez hat játék megnyerésére van szükség, úgy, hogy kettővel több legyen, mint az ellenfélnek. Egyenlőség esetén hat-hatos állásnál rövidítés következik, mely legalább hét nyert pontig tart és szintén kettővel többet kell gyűjteni, mint az ellenfél. Egy játék pedig 4 pontig tart legalább, illetve ameddig az egyik játékosnak nincs kettővel több pontja a másiknál.
- A játékosok játékonként felváltva szerválnak, rövidités esetén két pontonként.
- Az ATP szabályozása szerint egy játékos egyszerre egy versenyen vehet részt.
- A versenyeken való indulás nevezéshez kötött, ranglista helyezés alapján dől el, hogy a nevező személy egyből a főtáblán indulhat-e, vagy selejtező mérkőzést kell játszania ennek érdekében.
- A versenyek egyenes kieséses rendszerben folynak. Ez azt jelenti, hogy a győztes továbbjut a következő körbe, a vesztes pedig kiesik.

A számítási mód többször változott az elmúlt negyven évben. Jelenleg a következő van érvényben:

- a világranglista hetente frissül, pontszám alapján
- a pontszám az elmúlt 52 hétben Grand Slam és Masters tornákon elért eredményekből, illetve az ezeken kívüli hat legjobb eredményéből áll össze

- Az ATP alapvetően 6 versenykategóriát különböztet meg az ott megszerezhető pontok alapján: Grand Slam(2000), Masters(1000), 500 Series(500), 250 Series(250), Challenger(80-125), Futures(18-35), zárójelben a győztesnek járó pontszám. A felsoroltakon kívül rendeznek egy ATP Masters nevű versenyt is, mely rendszerint az év utolsó tornája, melyen a világranglista első nyolc helyezettje mérkőzik előbb csoportmérkőzéseken, majd a legjobb négy versenyző egyenes kiesésben. A győztesnek 1100-1150 pont jár.

- A mérkőzéseket emellett négy borításon - kemény pályán, salakon, füvön, szőnyegen(beltéren) - játsszák.

5. Adatsor elemzése

A következőkben bemutatom, miként alakítottam az idősort. Az átalakítás során fontos volt, hogy az időbeliség megmaradjon, illetve, hogy a végső adattábla modellezhető legyen. Felhasznált irodalom: [8]

5.1. Az adatsor ismertetése, kezdeti lépések

Az adatok forrása : https://github.com/JeffSackmann/tennis_atp (2017)

Minden, adatokkal kapcsolatos átalakítást, műveletet és modellezést az R programban végeztem el.

Az adatok ATP, ATP Challenger, Futures felosztásban évenként álltak rendelkezésre, 1968-tól. Ezeken kívül egy lista minden hét világranglistájáról, illetve egy játékosazonosító párokat tartalmazó tábla. Mivel ez hatalmas adatbázis, úgy döntöttem, hogy csak az elmúlt három év adataival dolgozom (2014-2016). Minden sor egy mérkőzés adatait tartalmazza. A felhasznált adatbázisok sor és oszlopszámmal a következők:

	adattabla	sorok szama	oszlopok szama	fejlec
1	atp_2014	2901	49	van
2	atp_2015	2958	49	van
3	atp_2016	2941	49	van
4	atp_challenger_2014	6424	49	van
5	atp_challenger_2015	6811	49	van
6	atp_challenger_2016	9218	49	van
7	futures_2014	20854	49	van
8	futures_2015	20361	49	van
9	futures_2016	19407	69	nincs

5.2. Adattábla szerkesztése

Látható, hogy az állományok egyenlő oszlopszámúak, kivéve a 2016-os ITF adattáblát, amely valamiért 69 oszloppal rendelkezik, de ellentétben a többi állománnyal, fejléccel nem. A 69-ből 49-et csináltam, a fejlécet pedig pótoltam a többi adattábla alapján, így már lehetett velük dolgozni. Az állományokat összefésültem, egy nagy táblába, mely a *matches* nevet kapta. A 49 adott változónk a következők:

torna azonosítója(tourney_id),torna neve(tourney_name),borítás(surface),
főtábla mérete(draw_size),torna ATP szerinti besorolása(tourney level),
torna dátuma/torna hetének első napja/(tourney_date),azonosító(match_num),
győztes azonosító(winner_id),győztes kiemelése(winner_seed),winner_entry,
győztes neve(winner_name),győztes ütőkeze(winner_hand),
győztes nemzetisége(winner_ioc),győztes kora(winner_age),
győztes világranglista helyzése(winner_rank),
győztes világranglista pontszáma(winner_rank_points),
vesztes azonosítója(loser_id),vesztes kiemelése(loser_seed),loser_entry,
vesztes neve(loser_name),vesztes ütőkeze(loser_hand),loser_ht,
vesztes nemezetisége(loser_ioc),vesztes kora(loser_age),
vesztes világranglista helyezése(loser_rank),
vesztes világranglista pontszáma(loser_rank_points),
eredmény/szetteket tekintve/(score),
hány nyert szett szükséges a győzelemhez(best_of),
tornán belüli forduló(round),időtartam(minutes),
győztes ászainak száma(w_ace),
győztes kettős hibáinak száma(w_df),
győztes szervapontjainak száma(w_svpt),
győztes sikeres első szerváinak száma(w_1stIn),
győztes megnyert első szerváinak száma(w_1stWon),
győztes megnyert második szerváinak száma(w_2ndWon),
győztes által megnyert szervagémek száma(w_SvGms),

győztes által kivédett bréklabdák(w_bpSaved),

győztes által kivédendő bréklabdák száma(w_bpFaced),

vesztes ászainak száma(l_ace),vesztes kettős hibáinak száma(l_df),

vesztes sikeres első szerváinak száma(l_1stIn),

vesztes megnyert első szerváinak száma(l_1stlon),

vesztes megnyert második szerváinak száma(l_2ndlon),

vesztes által megnyert szervagémek száma(l_SvGms),

vesztes által kivédett bréklabdák(l_bpSaved),

vesztes által kivédendő bréklabdák száma(l_bpFaced)

Ennyi változóra nincs szükség. A következő szempontok alapján választottam ki a megmaradt változókat:

- gyenge minőségűeket, sok hibás sorral rendelkezőket kivettem
- logikailag irrelevánsakat kivettem

A következő változók maradtak:

- torna dátuma/torna hetének első napja/(tourney_date)
- torna azonosítója(tourney_id)
- torna neve(tourney_name)
- torna ATP szerinti besorolása(tourney_level)
- borítás(surface)
- győztes azonosítója(w_id)
- győztes neve(w_name)
- győztes ütőkeze(w_hand)
- győztes világranglista helyezése()w_rank)
- győztes világranglista pontszáma(w_rank_points)
- győztes kora(w_age)
- győztes nemzetisége(w_ioc)
- vesztes azonosítója(l_id)
- vesztes neve(l_name)
- vesztes ütőkeze(l_hand)

- vesztes világranglista helyezése(l_rank)
- vesztes világranglista pontszáma(l_rank_points)
- vesztes kora(l_age)
- vesztes nemzetisége(l_ioc)
- hány nyert szett szükséges a győzelemhez(best_of)

A időbeliség biztosítására az adatok elsődlegesen a tornák hetének első napjai szerint vannak rendezve, másodlagosan pedig a tornán belüli fordulónként.

	tourney_date	w_id	l_id	surface ...	w_rank	w_rank_points
33063	20140127	105650	111194	Hard ...	383	106
33064	20140127	104587	104980	Hard ...	473	76
33065	20140127	123896	106299	Hard ...	1790	1
33066	20140127	105178	122381	Hard ...	455	81
33067	20140127	106075	125852	Hard ...	346	125
33068	20140127	104804	106260	Hard ...	241	197
33069	20140127	105943	106204	Hard ...	446	84
33070	20140127	104993	104838	Hard ...	349	124

Egy fontos változót nem tartalmaz az adattáblánk, mely összefüggésben lehet egy mérkőzés kimenetelével, ez pedig a játékos formája. Az interneten fellelhető formák általában az elmúlt öt mérkőzésre térnek ki, így én is így teszek. A formák az adott borításon számított formát jelentik jelen esetben.

Először hozzáadunk négy, nullákat tartalmazó oszlopot az adattáblánkhoz. Ez a négy oszlop a győztes elmúlt öt mérkőzésén való győzelmeinek száma(win-win), vereségeinek száma(win-lost) és a vesztes játékosra ez a két adat (lost-win), (lost-lost) sorrendben.

Majd a mellékelt függvénnyel kitölti a program ezeket az oszlopokat. Az első sorokban nem vesszük figyelembe, hogy régebbről is van adatunk, így ott ezek az értékek nullák maradnak, mivel az adattáblánkban előbb nincs mérkőzés. A függvény először hibára futott. A probléma forrása pedig az, hogy három sorban ugyanaz a játékos szerepelt győztesként, és vesztesként is. Ezeket a sorokat töröltem.

Így előállt a *matches_4p* adattábla melynek mérete 91872 sor, és 18 oszlop.

	tourney_date	w_id ...	win-win	win-lost	lost-win	lost-lost
33063	20140127	105650 ...	2.00	2.00	0.00	0.00
33064	20140127	104587 ...	1.00	2.00	2.00	2.00
33065	20140127	123896 ...	0.00	2.00	2.00	2.00
33066	20140127	105178 ...	1.00	2.00	0.00	0.00
33067	20140127	106075 ...	1.00	2.00	0.00	0.00
33068	20140127	104804 ...	5.00	0.00	1.00	2.00
33069	20140127	105943 ...	3.00	1.00	1.00	0.00
33070	20140127	104993 ...	4.00	1.00	1.00	0.00
33071	20140127	105523 ...	4.00	1.00	1.00	2.00

Furcsának tűnhet, hogy a ranglista helyezés, illetve a pontszám is bekerült a változók közé, hiszen látszólag ugyanazt fejezi ki. Ez nem így van. Az ATP furcsa pontozásának köszönhetően egyáltalán nem arányos a helyezés és a pontszám nagysága. Egy szemléltető példa a ranglistákat tartalmazó tábla segítségével:

	datum	r.h.	azon	pont
1	20170102	42	104269	1035
2	20170102	43	105311	1030
3	20170102	44	104597	1013
4	20170102	45	106378	1001
5	20170102	46	105723	970
	⋮	⋮	⋮	⋮
7	20170102	219	105292	242
8	20170102	220	106105	239
9	20170102	221	104594	238
10	20170102	222	117360	238
11	20170102	223	122078	238

Látható, hogy míg a 42. és a 46. helyezett között 5 helyezés és 65 pont a különbség, a 219. és 223. helyezett szintén 5 helyezés mellett mindössze 4 pont a különbség.

Mivel a célváltozók - nyertes és vesztes kiléte - egy sorban helyezkednek el, olyan formára kell alakítani az adattáblánkat, hogy a célváltozók egyértelmű formában jelenjenek meg. Ez az *esely* tábla, mely a következőképpen jött létre: minden mérkőzés két sorban szerepel a győztes és vesztes szempontjából. Egy plusz változót is kiszámoltam, melyből kiderül, hogy a győztes vagy a vesztes adatai szerepelnek-e az adott sorban, mely változó számszerűsítve - *res*01 oszlop - a célváltozó lesz.

	result	surface	s_rank ...	res01
1	nyertes	Hard	39.00 ...	1.00
2	vesztes	Hard	136.00 ...	0.00
3	nyertes	Hard	61.00 ...	1.00
4	vesztes	Hard	35.00 ...	0.00
5	nyertes	Hard	46.00 ...	1.00
6	vesztes	Hard	29.00 ...	0.00

6. Modellépítés

6.1. Szignifikancia vizsgálat

Egy regressziós modell, az adatok struktúrája alapján megfelelő a célváltozó modellezésére. A célváltozó eloszlása binomiális, mely eloszlás az exponenciális eloszláscsalád tagja, így a választásom a fentebb bemutatott általánosított lineáris modellre esett.

Az első modell megalkotása előtt egy új változó került az adattáblába.

- *rank_kul*: a két játékos ranglista helyezései közti különbség (-1)-szerese
- *form_kul*: az adott játékos győztes illetve vesztes mérkőzéseinek különbsége a formájuk alapján

Az első modellbe minden, az esély táblázatban szereplő releváns változó bekerült, kivéve a *form_kul*, mivel, ha a játékos formájának mindhárom értéke szerepel, akkor a *form_kul* változó hibára fut a korreláció miatt. Az adattábla tanuló és teszt adatrészre lett bontva. A tanuló rész az egész adattábla 75%-a.

Kiértékelés után a következőt kaptam:

```
Call:
glm(formula = modell0, family = binomial, data = train)

Deviance Residuals:
Min       1Q   Median       3Q      Max
-2.7740  -1.0595   0.2343   1.0640   2.6207
Coefficients:
               Estimate   Std. Error z value Pr(>|z|)
(Intercept)    1.144e-01  4.463e-01   0.256  0.79765
surfaceCarpet -1.872e-01  1.841e-01  -1.017  0.30935
surfaceClay   -2.061e-01  1.795e-01  -1.148  0.25084
surfaceGrass  -2.212e-01  1.824e-01  -1.213  0.22508
surfaceHard   -2.016e-01  1.794e-01  -1.124  0.26122
s_handL        4.629e-02  3.349e-01   0.138  0.89007
```

```
s_handR         7.268e-02  3.342e-01   0.217  0.82786
s_handU        -8.383e-02  3.340e-01  -0.251  0.80181

⋮

s_iocMKD       -8.272e-01  2.793e-01  -2.961  0.00307 **
s_iocSYR       -1.380e+00  3.111e-01  -4.437 9.12e-06 ***

⋮

s_iocVEN       -6.832e-01  2.597e-01  -2.630  0.00853 **
s_iocVIE       -2.356e-01  4.047e-01  -0.582  0.56039
s_iocZIM       -6.893e-01  2.779e-01  -2.481  0.01311 *
s_rank          3.706e-04  2.184e-05  16.968  < 2e-16 ***
s_rank_points   2.137e-04  1.124e-05  19.025  < 2e-16 ***
s_age          -2.877e-03  1.793e-03  -1.604  0.10874
s_win           8.935e-02  6.827e-03  13.087  < 2e-16 ***
s_lost         -2.170e-02  6.975e-03  -3.110  0.00187 **
rank_kul        1.913e-03  1.824e-05 104.856  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 163344  on 117827  degrees of freedom
Residual deviance: 142656  on 117702  degrees of freedom
(19980 observations deleted due to missingness)
AIC: 142908

Number of Fisher Scoring iterations: 11
```

Bár külön az adatminőség részletes vizsgálatára a dolgozat nem tér ki, a modell a hibás adatokat nem veszi figyelembe.(... *observations deleted due to missingness*) Ezek a hibák adathiányokból adódnak. A kiértékelés értelmezése a következő:

- Estimate: adott változó becslése
- Std. Error: a becslés szórása
- z value: a becslés és a szórás hányadosa
- Pr(>|z|): $p = 2 * \Phi(-|z|)$

A csillaggal jelölt változóink szignifikánsak. A három csillaggal jelöltek a legerősebben. A modell pontosságát a teszt adatrészen nem sikerült kiszámolnia a programnak, mivel ott szerepelnek számára nem "megtanult" országok is. Az országok szignifikanciáját nézve látható, hogy néhány szignifikáns, de ezek kis országok, kevés versenyzővel - például: Szíria, Macedónia - így a nemzetiséget elhagyva futtattam újra majd értékeltem ki a modell pontosságát.

```
Confusion Matrix and Statistics
Reference
Prediction     0     1
0              17405 12180
1              2287  7529

Accuracy : 0.6328
95% CI : (0.628, 0.6376)
No Information Rate : 0.5002
P-Value [Acc > NIR] : < 2.2e-16

Kappa : 0.2658
Mcnemar's Test P-Value : < 2.2e-16
Sensitivity : 0.8839
Specificity : 0.3820
```

```
Pos Pred Value : 0.5883
Neg Pred Value : 0.7670
Prevalence : 0.4998
Detection Rate : 0.4417
Detection Prevalence : 0.7509
Balanced Accuracy : 0.6329

'Positive' Class : 0
```

A modell körülbelül 63%-os pontosságot ért el. A következőkben ezt szeretném növelni. Mivel az esély táblában minden mérkőzés kétszer szerepel, így minden mérkőzést két oldalról elemez, ami torzíthatja az eredményt, ezért a következő modellben minden mérkőzésből véletlenszerűen csak az egyik játékos szerepel az adattáblában. Ez az adattábla az *esely*2 nevet kapta. A könnyebb kiválasztás érdekében egy *azonosito* oszloppal bővült az *esely* adattábla. Az *esely*2 tábla elkészítése minden ezután következő modell elején megtörténik, így elősegítve, hogy minél több különböző esetet lefedjünk.

```
Call:
glm(formula = modell2, family = binomial, data = train)
Deviance Residuals:
Min        1Q    Median        3Q       Max
-2.75917  -1.06003  -0.01918   1.07354   2.63578
Coefficients:
               Estimate   Std. Error z value Pr(>|z|)
(Intercept)   -4.563e-01  7.932e-02  -5.753 8.77e-09 ***
s_rank         3.135e-04  2.845e-05  11.016  < 2e-16 ***
s_rank_points  2.157e-04  1.551e-05  13.913  < 2e-16 ***
s_age          9.099e-04  2.445e-03   0.372   0.7098
s_win          9.630e-02  9.303e-03  10.352  < 2e-16 ***
s_lost        -2.405e-02  9.497e-03  -2.532   0.0113 *
rank_kul       1.889e-03  2.540e-05  74.356  < 2e-16 ***
---
```

```
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 81567  on 58837  degrees of freedom
Residual deviance: 71485  on 58831  degrees of freedom
(10066 observations deleted due to missingness)
AIC: 71499

Number of Fisher Scoring iterations: 4

Confusion Matrix and Statistics

Reference
Prediction    0    1
0             8690 6150
1             1164 3709

Accuracy : 0.629
95% CI : (0.6222, 0.6357)
No Information Rate : 0.5001
P-Value [Acc > NIR] : < 2.2e-16

Kappa : 0.258
Mcnemar’s Test P-Value : < 2.2e-16

Sensitivity : 0.8819
Specificity : 0.3762
Pos Pred Value : 0.5856
Neg Pred Value : 0.7611
```

```
Prevalence : 0.4999
Detection Rate : 0.4408
Detection Prevalence : 0.7528
Balanced Accuracy : 0.6290

'Positive' Class : 0
```

Látható, hogy a kor(s_age) már nem szignifikáns , így a következő lépésben ezt a változót vettem ki, illetve a forma két változója(s_win és s_lost) helyett a $form\_kul$-t próbáltam.

```
Call:
glm(formula = modell3, family = binomial, data = train)
Deviance Residuals:
Min       1Q   Median       3Q      Max
-2.8366  -1.0599  -0.2569   1.0739   2.6420
Coefficients:
               Estimate   Std. Error  z value Pr(>|z|)
(Intercept)   -2.669e-01  1.954e-02  -13.66   <2e-16 ***
s_rank         3.052e-04  2.657e-05   11.49   <2e-16 ***
s_rank_points  2.295e-04  1.567e-05   14.65   <2e-16 ***
rank_kul       1.882e-03  2.522e-05   74.61   <2e-16 ***
form_kul       6.123e-02  4.263e-03   14.36   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
(Dispersion parameter for binomial family taken to be 1)
Null deviance: 81897  on 59075  degrees of freedom
Residual deviance: 71838  on 59071  degrees of freedom
(9828 observations deleted due to missingness)
AIC: 71848

Number of Fisher Scoring iterations: 4
```

```
Confusion Matrix and Statistics

Reference
Prediction    0    1
0             8693 6168
1             1140 3665

Accuracy : 0.6284
95% CI : (0.6216, 0.6352)
No Information Rate : 0.5
P-Value [Acc > NIR] : < 2.2e-16

Kappa : 0.2568
Mcnemar's Test P-Value : < 2.2e-16

Sensitivity : 0.8841
Specificity : 0.3727
Pos Pred Value : 0.5850
Neg Pred Value : 0.7627
Prevalence : 0.5000
Detection Rate : 0.4420
Detection Prevalence : 0.7557
Balanced Accuracy : 0.6284

'Positive' Class : 0
```

Így már minden változónk erősen szignifikáns. Nincs jelentős eltérés a $form\_kul$ változó használatával, mégis maradtam ennél, mert erősen szignifikáns, az előző esetben pedig látható volt, hogy a vesztes mérkőzések száma nem volt erős.

6.2. Tanuló és teszt adattáblák aránya

Következő lépésben a tanuló, és a teszt adattáblák arányát változtatom. Eddig 75%-25% volt, következő modellekben 55%-45%, 63%-47%, 83%-17% és 90%-10% lesz.

55%-45%:

```
Confusion Matrix and Statistics
Reference
Prediction     0     1
0              17376 12190
1              2361  7500
Accuracy : 0.6309
95% CI : (0.6262, 0.6357)
No Information Rate : 0.5006
P-Value [Acc > NIR] : < 2.2e-16
Kappa : 0.2614
Mcnemar's Test P-Value : < 2.2e-16
Sensitivity : 0.8804
Specificity : 0.3809
Pos Pred Value : 0.5877
Neg Pred Value : 0.7606
Prevalence : 0.5006
Detection Rate : 0.4407
Detection Prevalence : 0.7499
Balanced Accuracy : 0.6306
'Positive' Class : 0
```

63%-47%:

```
Confusion Matrix and Statistics
Reference
Prediction     0     1
0              12871 9088
1              1750  5498
Accuracy : 0.6289
95% CI : (0.6234, 0.6345)
No Information Rate : 0.5006
P-Value [Acc > NIR] : < 2.2e-16
Kappa : 0.2574
Mcnemar's Test P-Value : < 2.2e-16
Sensitivity : 0.8803
Specificity : 0.3769
Pos Pred Value : 0.5861
Neg Pred Value : 0.7586
Prevalence : 0.5006
Detection Rate : 0.4407
Detection Prevalence : 0.7518
Balanced Accuracy : 0.6286
'Positive' Class : 0
```

83%-17%:

```
Confusion Matrix and Statistics

Reference
Prediction    0    1
0             5929 4122
1             766 2541

Accuracy : 0.6341
95% CI : (0.6258, 0.6423)
No Information Rate : 0.5012
P-Value [Acc > NIR] : < 2.2e-16

Kappa : 0.2673
Mcnemar's Test P-Value : < 2.2e-16

Sensitivity : 0.8856
Specificity : 0.3814
Pos Pred Value : 0.5899
Neg Pred Value : 0.7684
Prevalence : 0.5012
Detection Rate : 0.4439
Detection Prevalence : 0.7524
Balanced Accuracy : 0.6335

'Positive' Class : 0
```

90%-10%:

```
Confusion Matrix and Statistics

Reference
Prediction    0    1
0             3485 2461
1             461  1522

Accuracy : 0.6315
95% CI : (0.6208, 0.6421)
No Information Rate : 0.5023
P-Value [Acc > NIR] : < 2.2e-16

Kappa : 0.2647
Mcnemar's Test P-Value : < 2.2e-16

Sensitivity : 0.8832
Specificity : 0.3821
Pos Pred Value : 0.5861
Neg Pred Value : 0.7675
Prevalence : 0.4977
Detection Rate : 0.4395
Detection Prevalence : 0.7499
Balanced Accuracy : 0.6326

'Positive' Class : 0
```

Habár jelentős eltérés nincs, azt mondhatjuk, hogy 83%-os tanuló adattábla teljesített a legjobban.

6.3. Módosított forma változó

A forma verség változójának (*s_lost*) gyengeségével kapcsolatban azt a szabályszerűséget fedeztem fel, hogy a tornák elődöntőiben és döntőiben az elmúlt mérkőzések alapján alacsony, hiszen aki egy elődöntőig eljut az nyilvánvalóan előtte nem szenvedett vereséget azon a tornán. Ennek kiküszöbölésére a formát 5 mérkőzés helyett 15 mérkőzésre is kiszámoltam, és ebben az előbbi modellben ellenőriztem, hogy valóban szignifikánsabbak lesznek-e a forma változók.

```
Call:
glm(formula = modell3, family = binomial, data = train)
Deviance Residuals:
Min       1Q   Median       3Q      Max
-2.8550  -1.0606   0.2756   1.0677   2.5766

Coefficients:
Estimate Std. Error z value Pr(>|z|)
(Intercept)   -3.114e-01  1.935e-02  -16.09   <2e-16 ***
s_rank         3.651e-04  2.580e-05   14.15   <2e-16 ***
s_rank_points  2.181e-04  1.505e-05   14.49   <2e-16 ***
rank_kul       1.869e-03  2.398e-05   77.96   <2e-16 ***
form_kul       3.987e-02  2.172e-03   18.35   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 90639  on 65383  degrees of freedom
Residual deviance: 79398  on 65379  degrees of freedom
(10870 observations deleted due to missingness)
AIC: 79408
```

```
Number of Fisher Scoring iterations: 4

Confusion Matrix and Statistics

Reference
Prediction    0    1
0             5831 4161
1              796 2583

Accuracy : 0.6293
95% CI : (0.621, 0.6375)
No Information Rate : 0.5044
P-Value [Acc > NIR] : < 2.2e-16

Kappa : 0.2617
Mcnemar's Test P-Value : < 2.2e-16

Sensitivity : 0.8799
Specificity : 0.3830
Pos Pred Value : 0.5836
Neg Pred Value : 0.7644
Prevalence : 0.4956
Detection Rate : 0.4361
Detection Prevalence : 0.7473
Balanced Accuracy : 0.6314

'Positive' Class : 0
```

A modell eredményessége nem változott a módosított forma változótól sem.

7. Összefoglalás

Dolgozatom során az ATP World Tour versenysorozat mérkőzéseit vizsgáltam. Az adatok három évet öleltek fel, összesen több, mint 90000 mérkőzést. Az adatok formázása, és néhány új adat levezetése után, az adatok összetettsége miatt, az általánosított lineáris regressziót használtam modellként, mely tökéletesen megfelelt az adatstruktúrának.

A változók jól elkülönültek egymástól szignifikanciaszint szempontjából, érdekesség, hogy a kor változó is jó eredményt mutatott. Végül egy olyan modellt sikerült elérnem, melyben minden változó erősen szignifikáns. Ezután kiválasztottam a legmegfelelőbb bontás arányát a tanuló és teszt adattáblák szempontjából. Végül érdekességképpen, a forma változó viselkedésének okát megsejtve, egy bővebb változatát is kipróbáltam az elkészült modellnek. A modell több változó esetében mért magas szignifikancia ellenére nem teljesített jól. Ebből azt a következtetést vonhatjuk le, hogy a tenisz egy jó szabályrendszerrel rendelkező, izgalmas sport, a mérkőzések eredményeit nehéz előrejelezni.

A végső modellben megmaradt, erősen szignifikáns változók:

- ranglista helyezés
- ranglista pontok
- ranglista helyezés különbsége
- forma

8. Felhasznált irodalom

Dolgozatom során a következő forrásokat használtam:

- Peter McCullagh; John Nelder: Generalized Linear Models. Chapman and Hall (1989) [1]
- A.J. Dobson; A.G.Barnett: Introduction to Generalized Linear Models. Chapman and Hall (2008) [2]
- Prőhle Tamás jegyzete a regresszióról (2012) [3]
- W. N. Venables; B. D. Ripley: Modern Applied Statistics with S. Springer. (2002) [4]
- T. J. Hastie, D. Pregibon: Generalized linear models. Chapter 6 of Statistical Models in S eds J. M. Chambers and T. J. Hastie, Wadsworth and Brooks/Cole. (1992) [5]
- R. R. Davidson: On extending the Bradley-Terry model to accommodate ties in paired comparison experiments. J. Amer. Statist. Assoc 65, 317–328, (1970) [6]
- Mohsen Pourahmadi: Covariance Estimation: The GLM and Regularization Perspectives Statistical Science, Vol. 26, No. 3 pp. 369-387 (2011) [7]
- H. Turner, D. Firth: Generalized nonlinear models in R: An overview of the gnm package. R package version 1.0-6 (2012) [8]

9. R Kódok

```
x <- seq(-12,12,l=101)[-c(1,101)]
y1 <- logit(x,inverse = TRUE)
y2 <- probit(x,inverse = TRUE)
y3 <- cloglog(x,inverse = TRUE)
xlim <- c(min(x),max(x))
ylim <- c(min(y1,y2,y3),max(y1,y2,y3))
plot(xlim,ylim,t='n')
points(x,y1,t='l',col="red")
points(x,y2,t='l',col="blue")
points(x,y3,t='l',col="green4")
legend("topleft", c("logit","probit","cloglog"), pch = 1,
title = "link functions",col=c("red","blue","green"))

---------------

dim(read.csv("atp\_matches\_2014.csv",
header=TRUE, sep=",",stringsAsFactors=FALSE))# 2901 x 49
dim(read.csv("atp\_matches\_2015.csv",
header=TRUE, sep=",",stringsAsFactors=FALSE))# 2958 x 49
dim(read.csv("atp\_matches\_2016.csv",
header=TRUE, sep=",",stringsAsFactors=FALSE))# 2941 x 49
dim(read.csv("atp\_matches\_futures\_2014.csv",
header=TRUE, sep=",",stringsAsFactors=FALSE))# 20854 x 49
dim(read.csv("atp\_matches\_futures\_2015.csv",
header=TRUE, sep=",",stringsAsFactors=FALSE))# 20361 x 49
dim(read.csv("atp\_matches\_futures\_2016.csv",
header=TRUE, skip=1,sep=",",stringsAsFactors=FALSE))# 19407 x 69
dim(read.csv("atp\_matches\_qual\_chall\_2014.csv",
header=TRUE, sep=",",stringsAsFactors=FALSE))#  6424 x 49
```

```
dim(read.csv("atp\_matches\_qual\_chall\_2015.csv",
header=TRUE, sep=",",stringsAsFactors=FALSE))#  6811 x 49
dim(read.csv("atp\_matches\_qual\_chall\_2016.csv",
header=TRUE, sep=",",stringsAsFactors=FALSE))#  9218 x 49

---------------

matches_4p <- cbind(matches[,c("tourney_date","w_id","l_id","surface",
"w_hand","w_rank","w_rank_points","w_age","w_ioc",
"l_hand","l_rank","l_rank_points","l_age","l_ioc")],
"win-win"=0,"win-lost"=0,"lost-win"=0,"lost-lost"=0)

---------------

hibas <- which(matches[,"w_id"]==matches[,"l_id"]))
# 60502 78706 90727 harom hibas sor
matches <- matches[-hibas,]

---------------

for(k in nrow(matches_4p):2)
{ w_id <- matches_4p[k,"w_id"]
hol <- matches_4p[k,"surface"]
win <- (matches_4p[(k-1):1,"w_id"]==w_id &
matches_4p[(k-1):1,"surface"]==hol)
lost <- (matches_4p[(k-1):1,"l_id"]==w_id &
matches_4p[(k-1):1,"surface"]==hol)
csum <- cumsum(win+lost)
pos <- which(csum==min(5,csum[k-1]))[1]
win <- sum(win[1:pos])
```

```
lost <- sum(lost[1:pos])
matches_4p[k,"win-win"]<-win
matches_4p[k,"win-lost"]<-lost
l_id <- matches_4p[k,"l_id"]
win <- (matches_4p[(k-1):1,"w_id"]==l_id &
matches_4p[(k-1):1,"surface"]==hol)
lost <- (matches_4p[(k-1):1,"l_id"]==l_id &
matches_4p[(k-1):1,"surface"]==hol)
csum <- cumsum(win+lost)
pos <- which(csum==min(5,csum[k-1]))[1]
win <- sum(win[1:pos])
lost <- sum(lost[1:pos])
matches_4p[k,"lost-win"]<-win
matches_4p[k,"lost-lost"]<-lost
}

---------------

esely <- data.frame(result=1:(2*n))
esely <- cbind(esely,surface=0,
s_hand=0,s_rank=0,s_rank_points=0,s_age=0,s_ioc=0,s_win=0,s_lost=0,
e_hand=0,e_rank=0,e_rank_points=0,e_age=0,e_ioc=0,e_win=0,e_lost=0)
for(be in 1:n)
{
ki<-2*be-1 # a win versenyzo adatai
esely[ki,"result"] <- "nyertes"
esely[ki,"surface"] <- matches_4p[be,"surface"]
esely[ki,"s_hand"] <- matches_4p[be,"w_hand"]
esely[ki,"s_rank"] <- matches_4p[be,"w_rank"]
esely[ki,"s_rank_points"] <- matches_4p[be,"w_rank_points"]
```

```
esely[ki,"s_age"] <- matches_4p[be,"w_age"]
esely[ki,"s_ioc"] <- matches_4p[be,"w_ioc"]
esely[ki,"s_win"] <- matches_4p[be,"win-win"]
esely[ki,"s_lost"] <- matches_4p[be,"win-lost"]
esely[ki,"e_hand"] <- matches_4p[be,"l_hand"]
esely[ki,"e_rank"] <- matches_4p[be,"l_rank"]
esely[ki,"e_rank_points"] <- matches_4p[be,"l_rank_points"]
esely[ki,"e_age"] <- matches_4p[be,"l_age"]
esely[ki,"e_ioc"] <- matches_4p[be,"l_ioc"]
esely[ki,"e_win"] <- matches_4p[be,"lost-win"]
esely[ki,"e_lost"] <- matches_4p[be,"lost-lost"]
ki <- 2*be   # a lost versenyzo adatai
esely[ki,"result"] <- "vesztes"
esely[ki,"surface"] <- matches_4p[be,"surface"]
esely[ki,"s_hand"] <- matches_4p[be,"l_hand"]
esely[ki,"s_rank"] <- matches_4p[be,"l_rank"]
esely[ki,"s_rank_points"] <- matches_4p[be,"l_rank_points"]
esely[ki,"s_age"] <- matches_4p[be,"l_age"]
esely[ki,"s_ioc"] <- matches_4p[be,"l_ioc"]
esely[ki,"s_win"] <- matches_4p[be,"lost-win"]
esely[ki,"s_lost"] <- matches_4p[be,"lost-lost"]
esely[ki,"e_hand"] <- matches_4p[be,"w_hand"]
esely[ki,"e_rank"] <- matches_4p[be,"w_rank"]
esely[ki,"e_rank_points"] <- matches_4p[be,"w_rank_points"]
esely[ki,"e_age"] <- matches_4p[be,"w_age"]
esely[ki,"e_ioc"] <- matches_4p[be,"w_ioc"]
esely[ki,"e_win"] <- matches_4p[be,"win-win"]
esely[ki,"e_lost"] <- matches_4p[be,"win-lost"]
}
esely$res01 <- 2-as.numeric(as.factor(esely[,"result"]))
```

```
---------------

sample <- sample.split(esely$res01, SplitRatio = .75)
train <- subset(esely, sample == TRUE)
test  <- subset(esely, sample == FALSE)
modell0 <- paste("res01~surface",
"s_hand+s_ioc+s_rank+s_rank_points+s_age+s_win+s_lost+rank_kul",
sep="+")
M0 <- glm(modell0,data=train,family=binomial)
summary(M0)
M0.pred <- predict.glm(M0, newdata=test)
Error in model.frame.default(Terms, newdata, na.action = na.action,
xlev = object$xlevels) :
factor s_ioc has new levels AZE, CAF, CAM, CIV,...

---------------

sample <- sample.split(esely$res01, SplitRatio = .75)
train <- subset(esely, sample == TRUE)
test  <- subset(esely, sample == FALSE)
modell0 <- paste("res01~surface",
"s_hand+s_rank+s_rank_points+s_age+s_win+s_lost+rank_kul",
sep="+")
M0 <- glm(modell0,data=train,family=binomial)

M0.pred <- predict.glm(M0, newdata=test)
M0.predt <- function(t) ifelse(M0.pred > t , 1,0)
confusionMatrix(M0.predt(0.5), test$res01)
```

```
---------------

azonosito <- seq(1,(nrow(esely))/2)
azonosito <- rep(azonosito, each =2)
esely <- cbind(esely,azonosito)
esely2 <- esely %>% group_by(azonosito) %>% sample_n(1)
sample <- sample.split(esely2$res01, SplitRatio = .75)
train <- subset(esely2, sample == TRUE)
test  <- subset(esely2, sample == FALSE)
modell1 <- paste("res01~surface",
"s_hand+s_rank+s_rank_points+s_age+s_win+s_lost+rank_kul",
sep="+")
M1 <- glm(modell1,data=train,family=binomial)
summary(M1)
M1.pred <- predict.glm(M1, newdata=test)
M1.predt <- function(t) ifelse(M1.pred > t , 1,0)
confusionMatrix(M1.predt(0.5), test$res01)

---------------

esely2 <- esely %>% group_by(azonosito) %>% sample_n(1)
sample <- sample.split(esely2$res01, SplitRatio = .75)
train <- subset(esely2, sample == TRUE)
test  <- subset(esely2, sample == FALSE)
modell2 <- paste("res01~s_rank+s_rank_points+s_age+s_win+s_lost+rank_kul",
sep="+")
M2 <- glm(modell2,data=train,family=binomial)
summary(M2)
M2.pred <- predict.glm(M2, newdata=test)
M2.predt <- function(t) ifelse(M2.pred > t , 1,0)
```

```
confusionMatrix(M2.predt(0.5), test$res01)

---------------

esely2 <- esely %>% group_by(azonosito) %>% sample_n(1)
sample <- sample.split(esely2$res01, SplitRatio = .75)
train <- subset(esely2, sample == TRUE)
test  <- subset(esely2, sample == FALSE)
modell3 <- paste("res01~s_rank+s_rank_points+rank_kul+form_kul",
sep="+")
M3 <- glm(modell3,data=train,family=binomial)
summary(M3)

M3.pred <- predict.glm(M3, newdata=test)
M3.predt <- function(t) ifelse(M3.pred > t , 1,0)
confusionMatrix(M3.predt(0.5), test$res01)

---------------

esely2 <- esely %>% group_by(azonosito) %>% sample_n(1)
sample <- sample.split(esely2$res01, SplitRatio = .55)
train <- subset(esely2, sample == TRUE)
test  <- subset(esely2, sample == FALSE)
modell4 <- paste("res01~s_rank+s_rank_points+rank_kul+form_kul",
sep="+")
M4 <- glm(modell4,data=train,family=binomial)
summary(M4)

M4pred <- predict.glm(M3, newdata=test)
M4.predt <- function(t) ifelse(M4.pred > t , 1,0)
```

```
confusionMatrix(M4.predt(0.5), test$res01)

---------------

esely2 <- esely %>% group_by(azonosito) %>% sample_n(1)
sample <- sample.split(esely2$res01, SplitRatio = .63)
train <- subset(esely2, sample == TRUE)
test  <- subset(esely2, sample == FALSE)
modell5<-paste("res01~s_rank+s_rank_points+rank_kul+form_kul",
sep="+")
M5<-glm(modell5,data=train,family=binomial)
summary(M5)

M5.pred <- predict.glm(M5, newdata=test)
M5.predt <- function(t) ifelse(M5.pred > t , 1,0)
confusionMatrix(M5.predt(0.5), test$res01)

---------------

esely2 <- esely %>% group_by(azonosito) %>% sample_n(1)
sample <- sample.split(esely2$res01, SplitRatio = .83)
train <- subset(esely2, sample == TRUE)
test  <- subset(esely2, sample == FALSE)
modell6 <- paste("res01~s_rank+s_rank_points+rank_kul+form_kul",
sep="+")
M6 <- glm(modell6,data=train,family=binomial)
summary(M6)

M6.pred <- predict.glm(M6, newdata=test)
M6.predt <- function(t) ifelse(M6.pred > t , 1,0)
```

```
confusionMatrix(M6.predt(0.5), test$res01)

---------------

esely2 <- esely %>% group_by(azonosito) %>% sample_n(1)
sample <- sample.split(esely2$res01, SplitRatio = .90)
train <- subset(esely2, sample == TRUE)
test  <- subset(esely2, sample == FALSE)
modell7<-paste("res01~s_rank+s_rank_points+rank_kul+form_kul",
sep="+")
M7 <- glm(modell7,data=train,family=binomial)
summary(M7)

M7.pred <- predict.glm(M7, newdata=test)
M7.predt <- function(t) ifelse(M7.pred > t , 1,0)
confusionMatrix(M7.predt(0.5), test$res01)

---------------

esely2 <- esely15 %>% group_by(azonosito) %>% sample_n(1)
sample <- sample.split(esely2$res01, SplitRatio = .75)
train <- subset(esely2, sample == TRUE)
test  <- subset(esely2, sample == FALSE)
modell8 <- paste("res01~s_rank+s_rank_points+rank_kul+form_kul",
sep="+")
M8 <- glm(modell8,data=train,family=binomial)
summary(M8)

M8.pred <- predict.glm(M8, newdata=test)
M8.predt <- function(t) ifelse(M8.pred > t , 1,0)
```

```
confusionMatrix(M8.predt(0.5), test$res01)
```