**Szent István Egyetem**

**Gazdaság- és Társadalomtudományi Kar**

TATA Kiválósági Központ és Informatikai Intézet

Feleletválasztós tesztek feltöltésére, tárolására alkalmas online keretrendszer fejlesztése az oktatás elősegítése érdekében

Konzulens: Dr. Pitlik László, Egyetemi docens,

SZIE, GTK TKI

Intézeti igazgató: Dr. Kovács Árpád Endre,

Egyetemi docens, SZIE, GTK TKI

Készítette: Varga Viktor, SZIE GTK Vezetés és szervezés MSC, levelező tagozat, II. évf.

Gödöllő, 2011

Tartalom

[Bevezetés – Motiváció, a témaválasztás indoklása 4](#_Toc307519677)

[Cél 5](#_Toc307519678)

[Célcsoport 5](#_Toc307519679)

[Hasznosság 6](#_Toc307519680)

[A dolgozat szerkesztéséről 6](#_Toc307519681)

[1. Irodalmi áttekintés 8](#_Toc307519682)

[1.1. A feleletválasztós tesztek ismertetése 8](#_Toc307519683)

[1.1.1. A feleletválasztós tesztek alkalmazásának előnyei 8](#_Toc307519684)

[1.1.2. A feleletválasztós tesztek alkalmazásának korlátai 9](#_Toc307519685)

[1.1.3. A feleletválasztós tesztek fajtái 10](#_Toc307519686)

[1.1.4. Hogyan kell megfelelő tesztkérdéseket, tesztsorokat előállítani? 11](#_Toc307519687)

[1.2. Az online tesztrendszer megírásához alkalmazandó programnyelvek 13](#_Toc307519688)

[1.2.1. HTML 13](#_Toc307519689)

[1.2.2. PHP 14](#_Toc307519690)

[1.2.3. SQL 17](#_Toc307519691)

[1.3. E-learning alkalmazása az oktatásban, különös tekintettel a nyelvoktató portálokra 19](#_Toc307519692)

[1.3.1. Mit értünk e-learning alatt? 19](#_Toc307519693)

[1.3.2. Az e-learning lehetőségei 20](#_Toc307519694)

[1.3.3. Az e-learning korlátai 22](#_Toc307519695)

[1.3.4. Gyakorlat, vagy teszt? 23](#_Toc307519696)

[1.4. Az adatbányászat és lelke, a gépi tanulás 25](#_Toc307519697)

[1.4.1. Gépi tanulás ismertetése, a gépi tanulási módszerek használatának jelentősége 25](#_Toc307519698)

[1.4.2. Adatbányászat ismertetése, céljai 26](#_Toc307519699)

[1.4.3. Az adatbányászat feladatai 27](#_Toc307519700)

[1.4.4. A tudásfeltárás folyamata 28](#_Toc307519701)

[1.4.5. Az adatbányászat etikai vonatkozásai 30](#_Toc307519702)

[1.4.6. Adattárházak 30](#_Toc307519703)

[1.5. Hasonlóságelemzés: A COCO módszer 31](#_Toc307519704)

[1.5.1. COCO online - standard verzió 31](#_Toc307519705)

[1.5.2. COCO-OPTI2 online - standard verzió (MCM) 32](#_Toc307519706)

[2. Anyag és módszertan 34](#_Toc307519707)

[2.1. Az Excel táblázatkezelő szoftverrel előállított feleletválasztós tesztről 34](#_Toc307519708)

[2.1.1. A feleletválasztós tesztsor-modell előállításának lépései 35](#_Toc307519709)

[2.1.2 A tesztsor tartalmának kialakítása 38](#_Toc307519710)

[2.2. A tesztek adatainak feldolgozása, és modellalkotás 39](#_Toc307519711)

[3. Eredmények 41](#_Toc307519712)

[3.1. A kitöltött Exceles tesztek eredményei 41](#_Toc307519713)

[3.2. A sorrendek mérésének eszköze: Az MCM futtatás eredményeinek értelmezése 44](#_Toc307519714)

[4. Javaslatok 46](#_Toc307519715)

[4.1. A tesztrendszer mérési problémájának kiküszöbölése 46](#_Toc307519716)

[4.2. Hogyan viszonyul a teszt keretrendszer az ötletül szolgáló, magyarázó alrendszerrel ellátott nyelvi tesztrendszerhez? 46](#_Toc307519717)

[4.3. Milyen módokon válhatna a rendszer hasznossá a tanárok számára? 48](#_Toc307519718)

[4.4. Milyen eszközökkel igyekszik a rendszer megakadályozni, hogy rossz adatokból téves következtetésekre jusson a rendszer? 48](#_Toc307519719)

[5. Összefoglalás 50](#_Toc307519720)

[6. Irodalomjegyzék 52](#_Toc307519721)

[Definíciós jegyzék 53](#_Toc307519722)

[Rövidítések jegyzéke 54](#_Toc307519723)

[Ábrajegyzék 55](#_Toc307519724)

[Mellékletek 56](#_Toc307519725)

[Függelékek 57](#_Toc307519726)

# Bevezetés – Motiváció, a témaválasztás indoklása

Bár a cím azt sejteti, hogy jelen dolgozat az oktatást általánosságban igyekszik egy tesztrendszer kialakításával elősegíteni, valójában nem ez irányú vágyak ihlették azt. A téma és címválasztás kiindulópontja egy, már létező online német nyelvi tesztrendszer (lásd: http://miau.gau.hu/myx-free/index.php3?x=de1), melynek sajátossága, hogy a válasz helyességének megállapításán túlmenően a helyes, illetve helytelen válaszok indokaival is megismerteti a felhasználóját.

E sorok írója, ki az angol nyelv szépségei után – egyelőre – lelkes amatőrként érdeklődik (itt kell megjegyezni, hogy az „egyelőre” nem a lelkes, hanem az amatőr jelző miatt ékelődött be), nem csupán a már meglévő megoldást szerette volna alkalmazni az angol nyelvre, hanem a rendszer jobbá tétele érdekében is kívánt erőfeszítéseket tenni. Az így kialakuló keretrendszert azután bárki felhasználhatja tesztek feltöltésére és tárolására (például KRESZ teszteket is lehetne feltölteni), a forráskód adott lenne, teljesen függetlenül attól, hogy készítője egyetlen igazi célja az angol nyelvoktatás elősegítése. Ilyen körülmények között maradt ki a címből az angol nyelvoktatás, ami a dolgozat kulcsfontosságú eleme abból a szempontból, hogy ez adta a dolgozat megírásának a motivációját, holott a készülő rendszer univerzalitásából fakadóan kiesett annak fókuszából.

Az említett újítási tervek közé tartozik a tesztek nehézségének meghatározása, objektíven, a helyes/helytelen kitöltések arányait felhasználva, felhasználók kezelése az oldalon, és több más egyéb, amely a javaslatokról szóló fejezetben lesznek tárgyalva. Mindezen célok elérése érdekében ismertetésre kerülnek azok a technológiai tudásbeli, és egyéb igények, amelyek ha hosszú távon nem is lehetetlenné, de a dolgozat adta keretek (leginkább idő, de egyéb erőforrások is) között nem megvalósíthatóvá teszik azokat.  
A szakirodalmi fejezetben az alkalmazott rendszerek előnyei és hátrányai, sajátosságai, illetve az angol nyelv oktatásának gazdasági szükségessége kerül elemzésre a megfelelő statisztikák ismertetésén keresztül. Aki ilyen rendszert épít, tisztában kell, hogy legyen azokkal az előnyökkel, amivel az jár, és azzal is, gazdaságilag mennyire releváns az a témakör, amely köré az épül, hiszen az előnyök hangsúlyozása a sarokköve bármilyen tudatos marketing akciónak, a hátrányok ismertetésére, azaz hogy mit ne várjon a kedves felhasználó a rendszer használatától, pedig az erkölcsi kötelesség készteti az alkotót. A tesztrendszer nem fog megtanítani senkit angolul, pont úgy, ahogy autót vezetni sem lehet tanulni KRESZ-tesztek kitöltésével. Segédeszközként lehet tekinteni a rendszerre, amely hozzájárulhat a sikerhez.

# Cél

A hosszú távú cél egy olyan feleletválasztós teszt keretrendszer kialakítása, amely elősegíti az oktatást, elsősorban a nyelvtanítást. E cél elérése több lépésből áll, ennek első lépése azoknak a szabályoknak, mechanizmusoknak a leírása, amelyek a rendszer minőségbiztosítását jelentik. A tesztrendszernek úgy kell működnie, hogy nagy biztonsággal kizárható legyen annak a lehetősége, hogy a tesztalany tudásának megmérése sikertelen. Vannak olyan tipikus, a szakirodalomban részletesen ismertetett, a tesztsorok elkészítése során elkövethető hibák (tesztsor-elemek egymásra utalnak, hibás nyelvtan, mint árulkodó nyom, tesztkérdések nem megfelelő homogenitása) amelyek a tesztek kitöltését aránytalanul könnyűvé, vagy nehézzé, esetleg azt okozzák, hogy a tesztek nem azt a tudást mérik, amelyet a tesztkérdés írója mérni szándékozott.  
Tekintettel arra, hogy egy nagyméretű tesztadatbázis készítése a hosszú távú cél, reális elvárás, hogy a hibák kiszűrése nem történhet kézzel, hanem automatizálni kell. Fontos a hibák kiszűrésének automatizálása, amelyre a jelen dolgozat kutatási része a hangsúlyt helyezi. Az automatizálás kapcsán adatbányászati eszközökre kell hagyatkoznia a rendszernek.  
A dolgozatnak ugyanakkor nem célja az abban leírt, ideálisnak tartott oktató teszt keretrendszer teljes körű megvalósítása a dolgozat védésének időpontjáig, hiszen az olyan mérvű feladat, ami több ember munkáját dicsérné.

# Célcsoport

A teszt keretrendszer célcsoportja minden olyan ember (elsősorban tanárok), akik valamilyen témakörben tesztek segítségével oktatási tevékenységet szeretnének megvalósítani. Leginkább tanároktól, vagy egy-egy téma egyéb szakértőitől lehetne elvárni, hogy egy teszt adatbázist megfelelő tartalommal tudjanak megtölteni, mert többféle előnyre tehetnek szert a rendszer használata által. Egyrészt kapnak egy eszközt, amivel tanítványaik teljesítményét könnyen, gyorsan, és pontosan meg tudják mérni. A kapott adatokat felhasználhatják arra, hogy eldönthessék, melyek azok a kritikus részei a tananyagnak, amelyeket a tanulók megfelelően, illetve nem megfelelően sajátítottak el, így mire kell összpontosítani az energiájukat a tanórákon. Másrészt láthatják, hogy egy-egy tesztkérdés kapcsán melyik válaszlehetőségre érkezett aránytalanul sok, vagy kevés „tipp”, és ezek közül melyik rossz válasz (amennyiben egy rossz válaszra sok jelölés érkezik, akkor az hatékony zavaró tényezőnek tekinthető, a diákok által gyakran elkövetett hiba, fordítva viszont, ha nagyon sok jó válasz érkezik be, akkor a kérdés túl triviális, és a zavarók esetleges felülvizsgálata indokolt). Egy jól elkészített rendszerben a tanulók önállóan és helyesen mérhetik meg a teljesítményüket és dolgozhatják fel a lexikális tudáselemeket (pl. nyelvtant), így a több teret engedve az órai foglalkozásokon a gyakorlati jellegű tudáselemeknek (mint például a beszédkészség idegen nyelven).  
Így egyrészt a tanárokat megtanítjuk tanítani, másrészt a diákokat megtanítjuk a tananyag ismeretére.

# Hasznosság

A dolgozat hasznossága egy olyan tesztrendszer létrejötte, amely lehetővé teszi a hibás tesztkérdések és tesztsorok adatbányászati, statisztikai alapú kiszűrését, a kitöltött tesztek adatainak eltárolásán keresztül az egyes tesztkérdések, vagy tesztsorok nehézségi szintjének objektív mérését, azoknak a zavaró tényezőknek a kiszűrését, amelyek nem fejtik ki a hatásukat. A kutatás fókuszában azok a mutatók vannak, amelyek a jövőbeni rendszer naplófájlját alkotják, amelyekkel egy a teszt kitöltőjének a tesztsorral kapcsolatos viselkedése mérhető, amely az alkalmazott eszközökkel elemezhető.

# A dolgozat szerkesztéséről

A műben kétféle cél mentén lehet elkülöníteni a leírtakat: a szakirodalmi és javaslati fejezetekben a nagyobb cél, a címben foglalt feleletválasztós tesztek feltöltésére, tárolására, alkalmas online keretrendszerhez szükséges eszközök, tudáselemek kerülnek ismertetésre, valamint a javaslati fejezetben azok az ötletek, amelyek az eddig létező hasonló tesztrendszerek megújítását célozzák. A szakirodalmi alfejezetek előtt egy két mondatban megmagyarázásra kerül, hogy az adott alfejezet miért is került bele a dolgozatba.

Az ezektől eltérő főfejezetekben (anyag és módszertan, eredmények) az előbbi nagy cél egy részcéljának, a feleletválasztós tesztek minőségbiztosításának kapcsán lefolytatott kutatás kapcsán alkalmazott eszközök, illetve kapott eredmények kerülnek bemutatásra.

# Irodalmi áttekintés

## A feleletválasztós tesztek ismertetése

A feleletválasztós teszt az értékelés olyan formája, melyben a válaszadóknak a lehető legjobb választ kell kiválasztaniuk a lehetséges válaszok listájából. A feleletválasztós tesztek két alapvető részből állnak: a ***problémából***, amelynek formája lehet egy kérdés, vagy befejezetlen állítás, valamint az***alternatívák*** listájából, amely tartalmazza a ***választ****,* ami az egyetlen korrekt, vagy a legjobb alternatíva, valamint bizonyos számú ***zavaró*** *tényezőt*.

A zavaró tényezők célja, hogy plauzibilis megoldásnak tűnjön azon kitöltők számára, akik nem érték el a célt, melynek elérését a tesztkérdés méri. Viszont, a zavaró tényezőknek helytelen megoldásnak kell tűnnie azon kitöltők számára, akik elérték a célt. Számukra csak a válasznak szabad plauzibilisnek tűnnie.

### A feleletválasztós tesztek alkalmazásának előnyei

A feleletválasztós tesztek alkalmazásának egyik nagyon fontos előnye a sokoldalú felhasználhatóság. A tanulás eredményének több szintjére is alkalmazható, az elsajátított tudás felidézésétől összetettebb szintekig, mint például a tanuló képességeire: jelenségek elemzése, elvek alkalmazása új szituációkban, koncepciók és elvek megértésére, tények és vélemények megkülönböztetése, ok-okozati összefüggések értelmezésére, táblázatok és grafikonok értelmezése, az információ relevanciájának megítélése, következtetések levonása adatokból. A tesztek nehézsége könnyen befolyásolható az alternatívák homogenitásának változtatásával, azaz: minél inkább hasonlóak az alternatívák, annál árnyaltabbak különbségek, amiket a tanulóknak helyesen kell felmérniük a helyes alternatíva kiválasztása érdekében.

További előnye másmilyen vizsgálati módszerekkel szemben, mint amilyenek például az esszékérdések, az, hogy a diákok által gyorsabban megválaszolható, a tanárok által gyorsabban kiértékelhető, ezáltal a tanárok adott időkeretek között nagyobb mennyiségű tananyag elsajátításának mérését tudják lebonyolítani.

Itt kell megemlíteni a nagyobb megbízhatóságot is másféle tesztrendszerekkel szemben. A feleletválasztós tesztek kevésbé hajlamosítanak a találgatásra, ezáltal sokkal megbízhatóbb értékelést tesznek lehetővé. A pontozás sokkal világosabb, mint olyan értékelési módszerek esetében, ahol a tanulóknak röviden válaszolniuk kell a feltett kérdésre, mert nem fordulhatnak elő rossz helyesírással írt, vagy részleges válaszok. Az értékelés objektív, így nem kell számolni az értékelő inkonzisztenciájával, mint az esszék esetében, a blöffölés valamint az íráskészség sem játszik szerepet az értékelésben.  
  
És végül szót kell emelni az előnyök között a hatékonyságról is, ami abból fakad, hogy a tesztek gyorsan kiértékelhetők, akár gép által is, szemben az esszékérdésekkel, amelyek csak „manuálisan” értékelhetőek. A gyorsaságból fakadóan a tanuló még azelőtt értesülhet a teszt eredményeiről, hogy számottevő előrehaladás történne a tananyagban.

### A feleletválasztós tesztek alkalmazásának korlátai

Természetesen a feleletválasztós tesztekről nem beszélhetünk egyfajta üdvözítő megoldásként, melynek háttérbe kell szorítania minden egyéb módszert, hanem beszélhetünk hátrányokról, korlátokról, olyan esetekről, ahol a módszer alkalmazása kevésbé hatékony más módszerekkel szemben.

Mivel a tanulók a választ az alternatívák listájából választják ahelyett, hogy maguk szolgáltatnák, alakítanák ki a választ, a feleletválasztós tesztek nem adaptálhatók bizonyos tanulmányi eredmények mérésére, úgymint a tanuló képessége: a világos magyarázatadásra, a gondolatfolyamok megjelenítésére, az információ szolgáltatására, a személyes gondolatok összeszedésére, bizonyos feladatok elvégzésére, eredeti ötletek kreálására, példák bemutatására. Ezek a készségek jobban mérhetők az esszé jellegű kérdésekkel.

Bár az igaz-hamis állítások közötti választást nagyobb mértékben befolyásolhatja a találgatás, bizonyos mértékig a feleletválasztós tesztek esetében is fennáll ez a probléma.

Megfelelő feleletválasztós teszteket írni általában nehezebb és időigényesebb, mint más típusú teszteket. A megfelelő zavaró tényezők kitalálásához bizonyos fokú készségre van szükség. Ez a készség, azonban a gyakorlás, tanulás, tapasztalat útján elsajátítható.

### A feleletválasztós tesztek fajtái

**Egyetlen helyes válasz**

Az egyetlen helyes válasszal rendelkező teszt elemek esetében egy alternatívát kivéve az összes inkorrekt, az egy fennmaradó pedig a helyes válasz. A tanuló feladata ez utóbbit megtalálni.

**Legjobb Válasz**

Ezeknél a teszt elemeknél az alternatívák a korrektség mértékében változóak. Egyesek teljesen inkorrektek vagy korrektek lehetnek, de csak egy van, ami korrektebb a többinél. Ez az alternatíva a válasz, többi pedig zavaró tényező. A tanuló dolga az előbbit felismerni.

**Negatív teszt elemek**

A negatív típusnál a tanuló feladata lehet az inkorrekt, vagy a legrosszabb válasz megtalálása. Bármely más típusú teszt elem konvertálható erre a formára.

A legtöbb oktatási cél esetében a tanuló eredménye hatékonyabban mérhető, hogy ha a helyes választ kell megtalálnia, mint ha a helytelent. Csak azért, mert a diák felismer egy helytelen választ, még nem biztos, hogy ismeri a helyes választ. Ezért a negatív típusba tartozó tesztkérdések használata általában nem javasolt. Bizonyos esetekben a negatív elemek olyan mérési célok esetében, mint amilyen az egészség, vagy biztonság kérdése, ahol fontos annak ismerete, hogy mit szabad, vagy nem szabad csinálni. Ugyanakkor a negatív típusba tartozó teszt elemeknél a negációt megfelelően jelölni kell *dőlt*, vagy **félkövér**, vagy NAGYBETŰVEL, valamint a negatívan megfogalmazott kérdéseknél az alternatívákat pozitívan kell megfogalmazni, hogy a dupla tagadás elkerülhető legyen.

**Több válasz**

A több válasz típusú teszt elemeknél kettő vagy több alternatíva van megadva, mint helyes válasz, a tanuló feladata mindet megtalálni. Ez a fajta teszt elem többféle módon is értékelhető. Lehet „mindent vagy semmit” alapon, ha minden helyes válasz és nulla zavaró tényező kerül kiválasztásra, az ér egy pontot. Ugyanakkor lehet úgy is pontozni, hogy minden kiválasztott helyes válaszért és minden ki nem választott zavaró tényezőért egy-egy pont jár. Az első esetben az a tanuló, aki csak egy helyes választ nem jelöl meg, ugyanannyi pontot érdemel, mint az, aki egyet sem jelöl meg helyesen. Ezért a több válasz típusú kérdések eme fajtájának használata nem ajánlott. Mivel a több válasz típusú teszt elem gyakran egymással összefüggő, igaz-hamis típusú kérdések láncolata, ezért a fenti értékelési probléma elkerülésének jó alternatívája az **igaz-hamis** típusú kérdés, ahol meg kell jelölni, hogy melyik válasz igaz, vagy hamis.

**Kombinált válasz**

A kombinált válaszok esetén egy vagy több alternatíva lehet a helyes válasz. A tanuló feladata, hogy beazonosítsa a helyes választ, vagy válaszokat úgy, hogy egyet ki kell választania a megadott betűhalmazok közül (A és B, A és C, stb.), amelyek valamennyien egy-egy alternatíva-kombinációt képviselnek. (FORRÁS: Burton, S.J., Merril, P.F., Sudweeks, R.R., Wood, B.(1991): How to prepare Better Multiple Choice Test Items: Guidelines for University Faculty)

### Hogyan kell megfelelő tesztkérdéseket, tesztsorokat előállítani?

Az alábbi listát T.M. Haladyna S.M. Dowling[[1]](#footnote-1) szolgáltatta:

**Általános tesztkérdés írás (eljárás tekintetében)**

1. Használd a legjobb válasz, vagy a korrekt válasz formájú tesztet!

2. Kerüld el a komplex feleletválasztós teszteket (a és b)!

3. A kérdés és az opciók függőlegesen legyen elhelyezve, ne vízszintesen!

4. Hagyj időt a szerkesztésre, vagy egyéb felülvizsgálatra!

5. Használd jól a nyelvtant, központozást, helyesírást

6. Minimalizáld a tesztkérdés elolvasására fordítandó időt!

7. Kerüld el a trükkös kérdéseket, amelyek megtévesztik a tanulókat!

**Általános tesztkérdés írás (tartalom tekintetében)**

8. Minden tesztkérdést alapozz valamilyen oktatási célra!

9. Egyetlen problémára fókuszálj!

10. Az alkalmazott szókincs legyen konzisztens a tesztalanyok szókincsével!

11. Ne fűzd össze a teszt elemeket, maradjanak függetlenek!

12. Használd a szerző példáit a saját teszt elemeid alapjaként!

13. Kerüld el a túl specifikus tudást tesztelemek írásakor!

14. Kerüld el a tankönyv ízű tesztelemeket!

15. Kerüld el a vélemények alapján írt teszt elemeket!

16. Használj feleletválasztós teszteket a magasabb szintű gondolkodást igénylő feladatok mérésére!

17. Szignifikáns, vagy fontos tudást mérjél, kerüld el a triviális tudást!

**Tesztkérdés írás**

18. A tesztkérdés legyen kérdés, vagy mondat kiegészítés formájában!

19. Kitöltős tesztkérdés esetén a hely ne az elején, vagy a közepén legyen kihagyva!

20. A tesztkérdésben az utasítás legyen világos és érthető!

21. Kerüld el a bőbeszédűséget a tesztkérdés megfogalmazásakor!

22. Kerüld el a negatívan megfogalmazott kérdéseket!

23. A központi gondolat és a szöveg döntő részét helyezd el a tesztkérdésben!

**Általános opciófejlesztés**

24. Használj minél több opciót! (Kutatások négyet-ötöt javasolnak)

25. Az opciókat logikai, vagy numerikus sorrendbe rakd!

26. Az opciók legyenek függetlenek, ne fedjék egymést!

27. Az opciók tartalma legyen homogén!

28. Az opciók hossza legyen konzisztens!

29. Kerüld el, vagy ritkán használd az „egyik sem ezek közül” kifejezést!

30. Kerüld el, vagy ritkán használd a „mind ezek közül” kifejezést!

31. Kerüld el a „nem tudom” válaszlehetőséget!

32. Pozitívan fogalmazd meg az opciókat!

33. Kerüld el azokat a zavaró tényezőket, amelyek segíthetik a tesztelőket: túl általános, vagy túl specifikus nyomok, abszurd opciók, stb.

34. Kerüld el a „soha” és „mindig” válaszokat!

35. Kerüld el a nyomravezető nyelvtani konstrukciókat!

**Helyes opciók fejlesztése**

36. A választ úgy pozícionáld, hogy ne mindig ugyanazon pozícióban legyen a helyes válasz (a ,b, c, d pozíció)!

37. Csak egyetlen helyes válasz legyen!

**Zavaró tényezők**

38. Használj plauzibilis zavarókat, de kerüld el az illogikus zavarókat!

39. Használj gyakran elkövetett hibákat zavarókként!

40. A megfogalmazás ne legyen túl technikai!

41. Használj ismert, de helytelen frázisokat!

42. Használj igaz állításokat, melyek nem helyes válaszok a kérdésre!

43. Ne használj humort zavaró tényezőknél!

## Az online tesztrendszer megírásához alkalmazandó programnyelvek

Ebben a fejezetben az online feleletválasztós tesztrendszer megalkotása során alkalmazandó informatikai eszközök rövid bemutatására kerül sor. Ezek azok a programnyelvek, amelyekre szükség lesz egy online működő, teszt adatbázisokat kezelni tudó portálhoz.

### HTML

#### Mi a HTML?

Az információk globális méretű disztribúciójához szükség van egy egyetemesen értett nyelvre, egyfajta publikációs anyanyelvre, melyet valamennyi számítógép potenciális megérthet. A World Wide Web, azaz a világháló publikációs nyelve a HTML (HyperText Markup Language = hiperszöveges jelölőnyelv)

A szerzők számára a HTML nyújtja az eszközt arra, hogy:

Online dokumentumokat publikáljanak címsorokkal, táblázatokkal, listákkal, fotókkal, stb.

Online információkat nyerjenek ki a hiperszöveges hivatkozásokon (linkek) keresztül, egy gombnyomásra.

Kialakíthassák a távoli szolgáltatásokkal bonyolított tranzakciók formáit, az információk keresése, helyek lefoglalására, termékek megrendelésére, stb.

Táblázatokat, videoklipeket, hangokat és további alkalmazásokat tegyenek közvetlenül a dokumentumaikba. (FORRÁS: <http://www.w3.org/TR/html401/intro/intro.html#h-2.2>)

### PHP

#### Mi az a PHP?

A PHP egy széles körben használt, nyílt forrású, univerzális szkriptnyelv, amely különösen alkalmas web fejlesztésre, és beágyazható a HTML-be.

Ami megkülönbözteti a PHP-t a kliensoldali programnyelvektől, mint a Javascript, az nem más, mint hogy a kód a szerveren fut, HTML-t generál, amely ezután elküldésre kerül a kliensnek. A kliens megkapja a szkript futtatásának eredményeit, de nem tudja, hogy mi volt a programkód mögötte. Lehetőség van a web szerver t úgy konfigurálni, hogy az összes HTML állományt PHP-vel dolgozza fel, ily módon a felhasználók végképp nem tudják megmondani, hogy

A legjobb dolog a PHP használatával kapcsolatban az, hogy nagyon könnyű a kezdők számára, de sok funkciót kínál a profi programozók számára is. (FORRÁS: <http://hu.php.net/manual/en/intro-whatis.php>)

#### Mit tud a PHP?

*„Bármit. A PHP főleg szerver-oldali szkriptek írására készült, azaz bármire képes, amit más CGI programok el tudnak végezni, ilyen funkciók az űrlap adatok feldolgozása, dinamikus tartalom generálása, vagy sütik küldése és fogadása. De a PHP ennél sokkal többet tud.*

*Három fő területen használnak PHP programokat.*

*Szerver oldali programozás. Ez a hagyományos, és fő használati formája a PHP-nek. Három komponens szükséges ahhoz, hogy ezt a formát használhasd. Az első a PHP értelmező (CGI vagy szerver modul formájában), egy webszerver és egy webböngésző. Egy webszerverrel mindenképpen rendelkezni kell, megfelelően telepített és beállított PHP-vel. A PHP program kimenetét a webböngészővel lehet megtekinteni, a szerveren keresztül elérve a szkriptet. Mindezek képesek elutni a te otthoni gépeden is, ha csupán csak ismerkedni kívánsz a nyelvvel.*

*Parancssori programozás. PHP programok szerver és böngésző nélkül is futtathatóak. Ha ilyen környezetben szeretnéd használni a PHP-t, csak a PHP értelmezőre van szükséged. Ebben a formában gyakran valamilyen ütemező program segítségével (cron \*nix és Linux alatt, Task Scheduler Windows alatt) futtatott programokat írnak, vagy egyszerű szövegfeldolgozó szkripteket készítenek.*

*Ablakozós alkalmazások írása. A PHP valószínűleg nem a legjobb nyelv grafikus felületű asztali alkalmazások írásához, de ha nagyon jól ismered a PHP-t, és szeretnél néhány fejlett PHP szolgáltatást használni a kliens-oldali programjaidban, a PHP-GTK-t is használhatod ilyen programok írásához. Ezt használva lehetőséged van operációs rendszerfüggetlen programok írására is. A PHP-GTK a PHP egy kiterjesztése, nem érhető el a hivatalos PHP csomagban.”*

*„A PHP használható a legfontosabb operációs rendszereken, beleértve a Linuxot, sok Unix változatot (beleértve a HP-UX, Solaris és OpenBSD rendszereket), a Microsoft Windows-t, a Mac OS X rendszert, a RISC OS-t, és másokat. A PHP a legtöbb webszervert is támogatja, beleértve az Apache, Microsoft Internet Information Server, Personal Web Server, Netscape és iPlanet szervereket, az Oreilly Website Pro, Caudium, Xitami, OmniHTTPd, és más szervereket. A legtöbb szerverhez a PHP modul szintű támogatást nyújt, de más a CGI szabványt támogató szerverekkel is együtt tud működni CGI feldolgozóként.*

*Összességében a PHP használatakor szabadon választhatsz operációs rendszert és webszervert. Ráadásul a függvény-alapú és objektum orientált programozás, vagy ezek keveréke közötti választás is rajtad áll. Bár nem minden szokásos OOP szolgáltatás került megvalósításra a PHP 4-es változatában, sok eljáráskönytár és nagyobb alkalmazás is az OOP-t használja, például a PEAR könyvtár. A PHP 5-ös változata helyreteszi a PHP 4 OOP terén mutatott gyengeségeit, teljes objektum modell áll rendelkezésünkre.*

*A PHP képességei nem csak HTML kimenet előállítására korlátozódnak. Képeket, PDF állományokat vagy akár Flash mozikat (libswf vagy Ming kiterjesztéssel) is létrehozhatsz futásidőben. Természetesen egyszerűen generálhatsz bármilyen szöveges kimenetet, mint az XHTML vagy bármilyen más XML. A PHP elő tudja állítani ezeket az állományokat, és el tudja menteni a szerven a közvetlen kiküldésük helyett, valamilyen szerver-oldali gyorsítótárat valósítva meg ezzel.*

*Az egyik legjobb és legfontosabb tulajdonsága a nyelvnek az adatbázisok széles körű támogatása. Adatbázisokat kezelő weblap készítése PHP segítségével hihetetlenül egyszerű. A következő adatbázisok támogatja jelenleg:*

*Adabas D*

*dBase*

*Empress*

*FilePro (csak olvasásra)*

*Hyperwave*

*IBM DB2*

*Informix*

*Ingres*

*InterBase*

*FrontBase*

*mSQL*

*Direct MS-SQL*

*MySQL*

*ODBC*

*Oracle (OCI7 és OCI8)*

*Ovrimos*

*PostgreSQL*

*SQLite*

*Solid*

*Sybase*

*Velocis*

*Unix dbm*

*A PHP rendelkezik egy adatbázis absztrakciós kiterjesztéssel is (PDO), amellyel egyöntetűen és áttetsző módon lehet kezelni bármilyen adatbázist, amit ez a kiterjesztés támogat. Ezen kívül a PHP támogatja az ODBC-t, ezért bármilyen más, ezt a szabványt támogató adatbázishoz is lehet kapcsolódni.*

*A PHP támogatja a kommunikációt más szolgáltatásokkal is különböző protokollok segítségével, úgy mint LDAP, IMAP, SNMP, NNTP, POP3, HTTP, COM (Windows rendszeren), és számos más. Sőt, nyithatsz hálózati foglalatokat is (socket) és kommunikálhatsz más protokollokkal is. A PHP támogatja a WDDX komplex adatcseréket, ami bármely más web programozási nyelvvel való kommunikációt megkönnyítheti. A PHP szintén rendelkezik a Java objektumok létrehozásának, és átlátszóan PHP objektumokként való kezelésének képességével. A CORBA kiterjesztés távoli objektumok elérésére használható.*

*A PHP rendkívül jó szövegfeldolgozó képességekkel rendelkezik, a POSIX és Perl reguláris kifejezésektől az XML állományok kezeléséig. Az XML dokumentumok feldolgozásához és eléréséhez PHP4-ben a SAX és DOM szabványok is használhatóak, Az XSLT kiterjesztés XML dokumentumok átalakítására használható. A PHP 5 az XML kapcsolatos feladatokat egységesen, a libxml2 függvénykönyvtárra támaszkodva látja el. Ezen biztos alapokon a PHP 5 bevezeti a SimpleXML és az XMLReader támogatást is.*

*Végül, de nem utolsósorban a PHP számos más érdekes kiterjesztéssel szolgálhat, mint például az mnoGoSearch kereső függvények, az IRC átjáró függvények, tömörítő eszközök (gzip, bz2, zip), naptár átalakítás, fordítás...” (FORRÁS:* <http://www.php-blog.hu/php-magyar-kezikonyv/intro-whatcando.html>*)*

### SQL

Az SQL (Structured Query Language = strukturált lekérdező nyelv) azon instrukciók halmaza, melyeket a relációs adatbázisokkal történő interakciókhoz használnak. Tulajdonképpen az SQL az egyetlen nyelv, melyet a legtöbb adatbázis képes megérteni. Valahányszor interakcióba lépünk egy ilyen adatbázissal, a szoftver lefordítja a parancsainkat (legyenek azok egérkattintások, vagy űrlapbejegyzések) SQL-parancsokká, amelyeket az adatbázisok értelmezni tudnak. Az SQL-nek három komponense van: Az adatmanipulációs nyelv (DML), az adatdefiníciós nyelv (DDL), és a vezérlő nyelv (DCL).

Amikor adatbázis-alapú programokat használunk, valószínűleg SQL-t használunk, még ha nem is feltétlenül tudunk róla. Például egy adatbázis alapú dinamikus weboldal a felhasználók inputjait megkapja űrlapok és kattintások formájában, amelyet arra használ, hogy összeállítson egy lekérdezést, ami kinyeri az adatbázisból azokat az információkat, amelyek a következő weboldal generálásához szükségesek.

Vegyünk például egy egyszerű online katalógust, amelynek keresési funkciója van. A keresőoldal állhat csak egy egyszerű űrlapból, amely csak egy szövegdobozt, ahova a felhasználó beírja a keresőszót, valamint egy keresés gombot tartalmaz. A gombra kattintással a webszerver kinyeri az összes rekordot, amelyben a keresőszó megtalálható, és generál egy olyan specifikus weboldalt, amely megfelel a felhasználó kérésének.

Ha a felhasználó olyan termékekre keres, amelyek az „Irish” szót tartalmazzák, a web szerver az alábbi SQL-parancsot használhatja a kapcsolódó termékek lekérdezéséhez.

SELECT \*

FROM products

WHERE name LIKE '%irish%'

Lefordítva, ez a parancs visszaadja az összes olyan rekordot az adatbázis „termékek” táblájából, melyek az „irish” karakterláncot a termék nevében bárhol tartalmazza.

Adatmanipulációs nyelv

Az adatmanipulációs nyelv (DML) tartalmazza a leggyakrabban alkalmazott SQL parancsok részhalmazát – azokat, amelyek az adatbázisok tartalmát valamilyen formában manipulálják. A négy leggyakoribb DML-parancs közé tartozik a SELECT, amely kinyeri az információt az adatbázisból, az INSERT, amely új információt ad az adatbázishoz, az UPDATE, amely módosítja az adatbázisban korábban eltárolt információt, és végül a DELETE parancs, amely törli az információt az adatbázisból.

Adatdefiníciós nyelv

Az adatdefiníciós nyelv (DDL) kisebb gyakorisággal alkalmazott parancsokat tartalmaz. A DDL parancsok az adatbázisok struktúráját módosítják, nem az adatbázis tartalmát. A gyakorta használt DDL parancsok között említhető példaként a CREATE TABLE, amely új adattáblát generál az adatbázisban, az ALTER TABLE, amely módosítja a tábla struktúráját, és a DROP TABLE, amely törli az adattáblát.

Vezérlő nyelv

A vezérlő nyelv (DCL) menedzseli a felhasználók hozzáférését az adatbázishoz. Két parancsból áll, az egyik a GRANT parancs, amely az adatbázishoz hozzáférést ad a felhasználónak, és a REVOKE parancs, amely a meglévő kiadott jogosítványokat visszavonja. Ez a két parancs adja a relációs adatbázis biztonsági modelljének a magját.

Az SQL parancs struktúrája

Szerencsére, mivel nem vagyunk számítógépek, az SQL parancsok szintaxisa az angol nyelvhez nagyon hasonlóan van kialakítva. Általában azzal a paranccsal kezdődik, amely megadja, hogy mit szeretnénk tenni, majd folytatódik a kikötéssel, amely megadja a parancsnak a célját (az adatbázis azon tábláját, amely a parancs által érintett) és végül további kikötésekkel végződik, amelyek további instrukciókat adnak.

Többnyire, ha hangosan felolvasunk egy SQL parancsot, elég jól meg tudjuk állapítani, hogy mi lehet a célja az adott parancsnak. Vegyük például az alábbi SQL parancsot:

DELETE

FROM students

WHERE graduation\_year = 2011

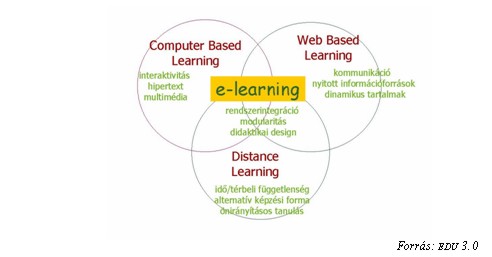
Hogy mit csinál ez a parancs? Hozzáfér az adatbázis „tanulók” táblájához, és a 2011-ben végzett tanulók összes rekordját törli. (FORRÁS: Chapple, M: What is SQL? Introduction to the Structured Query Language (2011)

## E-learning alkalmazása az oktatásban, különös tekintettel a nyelvoktató portálokra

Ebben a fejezetben megvizsgáljuk, hogy mit értünk computer milyen képességeket lehet mérni online keretek között, és mi a különbség a gyakorlatok és tesztek között, és hogy milyen előnyökkel, hátrányokkal jár az e-learning alkalmazása. Bármilyen online működő tanítóportálnál figyelembe veendőek az alábbiak.

### Mit értünk e-learning alatt?

*„Az e-learning olyan, számítógépes hálózaton elérhető nyitott - tér- és időkorlátoktól független - képzési forma, amely a tanítási-tanulási folyamatot hatékony, optimális ismeretátadási, tanulási módszerek birtokában megszervezve mind a tananyagot és a tanulói forrásokat, mind a tutor-tanuló kommunikációt, mind pedig az interaktív számítógépes oktatószoftvert egységes keretrendszerbe foglalva hozzáférhetővé teszi a tanuló számára.”[[2]](#footnote-2)*



1. ábra: Az e-learning kapcsolata a számítógép alapú, web alapú, illetve távoktatással Forrás: Bodó Balázs

### Az e-learning lehetőségei

**„*Hozzáférés***

*Az internet alapú oktatás nagy előnye szemben a hagyományos oktatással, hogy elvileg képes a megfelelő információt, a megfelelő időben a megfelelő személyhez eljuttatni. Azért csak elvileg, mert az internet elérése bár rohamosan növekszik, világviszonylatban a megfelelő infrastruktúra hiányában mégsem általános. Ahol azonban van internet elérhetőség, bármikor bekapcsolódhatnak a résztvevők az oktatásba és elérhetik a számukra szükséges információt, akár a munkahelyen is.*

***Just-in-time oktatás***

*Az internet segítségével a szolgáltató az oktatási tartalmakat mindig naprakészen tudja tartani. Ez főleg a gyorsan változó technikai szektorokban jelent nagy előnyt, ahol az információ szinte naponta elévül. Az elektronikus oktatás másik előnye, hogy a szükséges információ rögtön elérhető.*

***Interaktivitás***

*Ebből a szempontból különbséget kell tenni a szinkron és az aszinkron rendszerek között. Míg a szinkron felületek határtalan lehetőséget nyújtanak az interaktivitáshoz (real time oktatás), addig az aszinkron felületeket csak korlátozott interaktivitás a jellemző. A korlátot az oktató hiánya jelenti, aki így csak csúszva tud a diákok, hallgatók kérdésére reagálni. Az előre megírt változók segítségével azonban a program maga is tud válaszolni a diákok kérdéseire.*

***Személyre szabhatóság***

*Előre rögzített kritériumok alapján a program ki tudja választani azokat az oktatási egységeket, amik a legjobban illeszkednek a diák igényeihez, így a diáknak nem kell végigmennie azokon az egységeken, amelyek már ismertek számára, viszont hosszabb időt tud eltölteni azoknál az egységeknél, amelyeket nehezebben tud elsajátítani.*

***Eredmények figyelemmel kísérése***

*Az e-learning rendszerekbe, programokba be lehet építeni egy úgynevezett tracking rendszert, mely lehetőséget biztosít a tanár számára, hogy a diák teljesítményét valamint aktivitását folyamatosan figyelemmel kísérhesse.*

***Kapcsolattartás***

*Itt nem csak a tanár-tanuló kapcsolattartásról van szó, hanem lehetőség nyílik arra, hogy a diák egy központi adatbankból, amelyet a tanár is használ, lehívja eredményeit, figyelemmel kísérje saját teljesítményét.*

***Személytelenség***

*Sokakat frusztrál, ha csoport előtt kell szerepelnie. A csoport együttes interaktivitását csökkenti, ha a csoporton belül néhány tag az esetleges negatív élmények miatt nem mer felszólalni. Ha azonban az ember csupán egy név a képernyőn, sokkal bátrabban szólal fel, mégis továbbra is ismeretlen marad a többiek előtt. Másik előny, hogy bátrabban állnak neki egy-egy feladatnak, akár többször is nekikezdenek, mint egy hagyományos csoportos kurzuson.”*

### Az e-learning korlátai

*„****Személytelenség***

*Az egyik embernél előny, a másiknál korlát. Az elektronikus oktatás jellegéből adódóan hiányzik a csoportnyomás, amely az emberek egy részénél szükséges ahhoz, hogy a feladatokat végrehajtsa. Ahol nincs folyamatos kontroll, egyszerűbb a határidőket kitolni, esetleg mással foglalkozni.*

***Csoport motiváció hiánya***

*Sok feladatot könnyebb csoportban végrehajtani, mert a csoport motiváló hatással bír tagjaira. A tagok képesek kölcsönösen ösztönözni egymást, mint például a sportteljesítményeknél.*

***Gyakorlati bemutatás hiánya***

*Bizonyos tananyagoknál fontos a személy kipróbálás lehetősége. Gondoljunk például a virágkötészetre, vagy a kerámiakészítésre, ahol nem elég az elméletet megtanulni, hiszen a kézmozdulatokat nem elég a képernyőn keresztül látni. Az internet segítségével csak az elméleti tudást lehet átadni, de a gyakorlati tananyag bizonyos tanulmányoknál semmiképpen sem hagyható el.*

***Hozzáférhetőség és technikai nehézségek***

*Bizonyos régiókban a megfelelő infrastruktúra és az internet hiánya korlátot szab az e-learning terjedésének és alkalmazásának. Az újonnan elért régiókban viszont probléma lehet a csatlakozni vágyók technikai tudásának hiánya, így ez újabb korlátokat szabhat a megfelelő alkalmazásnak.*

***Oktatók felkészületlensége***

*Az internetes oktatás sokszor nem már, mint a megszokott tanóra online köntösben, mert az oktató nem használja ki az internet adta lehetőségeket. A lehetőségeket vagy nem ismerik fel, vagy nem képesek átültetni, ezért sokszor a diáknak elmegy a kedve a tanulástól.” (FORRÁS:* <http://e-learning.vacau.com/elearning.html>*)*

### Gyakorlat, vagy teszt?

A számítógép-alapú gyakorlatok és tesztek gyakran hasonló formákat öltenek. A lényegi különbség gyakorlatok és tesztek között a cél, amire alkalmazzuk őket. A gyakorlatok általában azonnal visszacsatolást nyújtanak a tanulónak, valamint lehetőséget arra, hogy kijavítsanak bármilyen hibát, amit vétettek, míg a tesztek kevés visszacsatolást nyújtanak a tanulónak, csupán az elért pontszámot a teszt végén, vagy akár semmilyen választ nem kapnak a teszt eredménye kapcsán, például akkor, amikor a teszt eredményeit elemzési céllal a tanár vagy a vizsgabizottság eltárolja. A gyakorlatok általában úgy vannak kialakítva, hogy lehetőséget nyújtsanak a hallgató számára a gyakorlásra bizonyos témákban, valamint, hogy motiválják, bátorítsák a tanulókat, míg a teszteket úgy alakítják ki, hogy azok mérjék a tanulók előrehaladását valamilyen téren, vagy az önvizsgálat céljára, vagy a tanár, vagy a vizsgabizottság számára. A tesztek legfőbb formái a következők:

**Felmérő teszt:** Ezeket úgy alakítják ki, hogy a tanulókat tanulócsoportokba lehessen sorolni, így közel azonos képességekkel kerülnek egy adott csoportba. Az felmérő tesztek az adaptív tesztek alakját is felvehetik (lásd lejjebb).

**Diagnosztikus tesztek**: Kialakításukkor arra összpontosítanak, hogy a tanuló vagy tanár képes legyen erősségek, vagy gyengeségek beazonosítására annak érdekében, hogy a megfelelő intézkedéseket megtehessék.

**Eredmény / tudás teszt**: Ezek többnyire formálisabbak, úgy alkotják meg, hogy egy-egy tematika elsajátítását mutassák meg, ahelyett, hogy a tanulót motiválják a tanulót, vagy megerősítsék bizonyos nyelvi képességeiben.

**Jártasság teszt:** Ezek a tesztek annak a megmérésére hivatottak, hogy a tanuló eredményeit egy olyan feladattal kapcsolatban mérjék, amelyet később el kell végezniük, például egy olyan tantárgyat hallgatni az egyetemen, amelyet az anyanyelvüktől eltérő nyelven oktatnak. Az autóvezetési teszt egy tipikus példája a jártassággal kapcsolatos teszteknek: azt méri, hogy a tanuló mennyire uralja az autót a közúti forgalomban.

**Alkalmassági vizsga:** Ezeknek a teszteknek az a célja, hogy megjósolják, hogy a tanuló hogyan fog teljesíteni egy bizonyos tantárgy, vagy egy tantárgy egy része kapcsán. (FORRÁS: <http://www.ict4lt.org/en/en_mod4-1.htm>)

## Az adatbányászat és lelke, a gépi tanulás

A készülő tesztrendszer egyik célja az, hogy felismerjen a tesztek írásakor elkövethető hibákat. Az ember képes arra, hogy ezeket a hibákat úgymond „ránézésre” felfedezze, a tesztkérdések tartalmának megismerése után. A számítógépek nem tudják értelmezni az ember által alkotott szöveg tartalmát, ha így lenne, akkor a Google Translate szolgáltatás miatt rengeteg, a nyelviparban dolgozó ember veszíthetné el a megélhetésének forrását egyik napról a másikra. Ezért a számítógépnek a tesztek kitöltése kapcsán megismerhető statisztikai adatokra lesz szüksége ahhoz, hogy helyes következtetéseket tudjon levonni, és magukban tesztkérdésekben, vagy a tesztkérdések kapcsolatában felismerhető hibákra rátaláljon. Az adatokkal való munka automatizációját lehetővé számítástechnikai eljárásokat nevezzük gépi tanulásnak, a gépi tanulás eredményeit feldolgozó módszertant pedig adatbányászatnak.

### Gépi tanulás ismertetése, a gépi tanulási módszerek használatának jelentősége

A tanulás, éppúgy, mint az intelligencia, folyamatok olyan széles tartományát fedi le, amelyet nehéz precízen definiálni.

A gépek tekintetében nagyon tágan értelmezve azt mondhatjuk, hogy egy gép akkor tanul, ahányszor csak megváltoztatja a struktúráját, programját, vagy adatait (a bemeneti értékeire alapozva, vagy külső információra adott válaszként, olyan módon, amitől annak a jövőbeli teljesítménye megváltozik. Néhány e változások közül, mint például egy rekord hozzáadása egy adatbázishoz, könnyedén beletartozhat más diszciplínák hatáskörébe, és nem feltétlenül érthető meg könnyebben attól, ha tanulásnak nevezzük őket. De például amikor egy beszédfelismerő gép teljesítménye javul, miután hallotta néhány személy beszédmintáit, azt állíthatjuk, hogy tanulásról van szó.  
  
A gépi tanulás rendszerint olyan rendszerekben történ változásokra vonatkozik, amelyek olyan feladatokat végeznek, amelyek a mesterséges intelligenciához kapcsolhatók. Ezen feladatok közé tartozik a felismerés, diagnózis, tervezés, robotirányítás, előrejelzés, stb.

Felmerülhet a kérdés, miért kell a gépeknek tanulnia? Miért nem terveznek olyan gépeket, amelyek rögtön el tudják látni a feladatokat?

* Néhány feladatot nem lehet egyszerűen definiálni, csak példákon keresztül. Lehet, hogy adott esetben meg tudunk szabni input/output párokat, de a kapcsolatokat az inputok és a kívánt outputok között nem. Azt szeretnénk, hogy a gépek képesek legyenek szabályozni a belső struktúrájukat annak érdekében, hogy nagy inputmennyiség esetében is képesek legyenek korrekt outputokat produkálni, így megfelelően kikényszerítve azt, hogy megközelítsék az értelemszerű kapcsolatokat a példákban.
* Lehetséges, hogy nagy adathalmazokban fontos kapcsolatok és korrelációk rejlenek. A gép tanulási módszerek gyakran alkalmazhatók e kapcsolatok kinyerésére.
* A tervezők gyakran gyártanak gépeket, amelyek nem az elvárásoknak megfelelően dolgoznak azokban a környezetekben, ahol használják őket. Tulajdonképpen a munkakörnyezet néhány jellemzője nem teljesen ismert a tervezés pillanatában. A gépi tanulás arra is használható, hogy gépeket a munkavégzés helyén fejlesszék tovább.
* Bizonyos feladatokról több tudás áll rendelkezésre, mint amit az emberek kódolni akarnának. A gépek, amelyek maguktól ismerik meg ezt a tudást néha többet tanulnak belőle, mint amennyit az emberek hajlandóak volnának leírni.
* A környezet változhat idővel. A gépek, amelyek képesek adaptálódni a változó környezethez csökkentenék az állandó újratervezés szükségét.
* A feladatokról rendelkezésre álló tudás állandóan növekszik az emberek felfedezései által. A szókincs változik. Új események keletkezése állandó folyamat a világban. Folytatólagosan újratervezni az MI rendszereket azért, hogy illeszkedjenek az új tudáshoz nem praktikus, de a gépi tanulás sokat tud ezekből nyomon követni.

(Forrás: NILSSON, 1996)

### Adatbányászat ismertetése, céljai

Az adatbányászat nem más, mint korábban ismeretlen és potenciálisan hasznos információ kinyerése az adatokból, valamint a zajok, az irreleváns adatok kiszűrése. (Forrás: Varga Viktor, 2009)

Az adatbányászat gyakran negatív fényben tűnik fel, mivel gyakran összemossák az data snooping-gal (amit magyarra szó szerint adat szimatolásnak fordíthatunk). Ez azt jelenti, hogy valaki előre megfontolt módon keres látszólag, de nem szükségszerűen reprezentatív mintázatokat adatokban. (Forrás: Varga Viktor, 2009)

Néhány adatbányászati alkalmazás célja az előrejelzés: előrejelezni mi fog történni egy jövőbeli szituációban, olyan adatok alapján, amelyek a múltbeli szituációkat írnak le, azaz következtetni arra, hogy milyen osztályba lesz sorolható a szituáció. Más alkalmazások esetében a modellek tanulásának célja a egy olyan struktúra leírása, amely később bármikor felhasználható osztályozásra. Míg a gépi tanulás annak tudománya, hogyan kell olyan algoritmusokat tervezni, illetve fejleszteni, amelyek képesek tanulni valamilyen adatokból, addig az adatbányászat a már kész algoritmusok megfelelő alkalmazásának tudománya.

(Forrás: Varga Viktor, 2009)

### Az adatbányászat feladatai

„Feltehetjük, hogy az adatbázis valamilyen objektumok (ügyfelek, betegségek, vásárlók, telekommunikációs események, . . . ) különböző tulajdonságait írja le. A tulajdonság helyett gyakran használjuk majd az attribútum szót. Az adatbányászat feladata a rejtett összefüggések, kapcsolatok felderítése. Az összefüggések típusa szerint a következő adatbányászati alapproblémákról beszélhetünk:

**Gyakori minták kinyerése**: Adott objektumok egy sorozata. Célunk megtalálni a gyakran előforduló (rész-) objektumokat. Az objektumok lehetnek elemhalmazok vagy sorozatok, esetleg epizódok (részben rendezések), gráfok stb.

**Attribútumok közötti kapcsolatok:** Gyakran hasznos, ha az objektumokra úgy tekintünk, mint az attribútumok megvalósulásaira és keressük az összefüggéseket az attribútumok között. Többféle összefüggés létezik. Ilyenek például az asszociációs-, korrelációs szabályok, a funkcionális függőségek és hasonlóságok. Az osztályozás is attribútumok közötti összefüggések felfedezésére szolgál. Az osztályozásnál egy kitüntetett attribútum értékét kell megjósolnunk a többi attribútum értéke alapján. Ezt egy modell felépítésével teszi. Leggyakrabban a modell egy döntési fa, de lehet if-then szabályok sorozata, valamilyen matematikai formula, vagy akár egy neurális hálózat stb. is.

**Klaszterezés**: Objektumokat előre nem definiált csoportokba (klaszterekbe) kell sorolnunk úgy, hogy az egy csoportba tartozó objektumok hasonlóak legyenek, míg a különböző csoportba kerültek különbözzenek egymástól. Két pont hasonlóságát egy előre megadott (távolságszerű) függvény segítségével szokás értelmezni.

**Sorozatelemzés**: A sorozatelemzésbe többféle adatbányászati feladat tartozik. Kereshetünk egymáshoz hasonlító (akár rész-) sorozatokat. Ezen kívül elemezhetjük a sorozat alakulását, és különböző regressziós módszerekkel próbálhatjuk megjósolni a jövőbeli valószínűleg előforduló eseményeket.

**Eltéréselemzés**: Azokat az elemeket, amelyek nem felelnek meg az adatbázis általános jellemzőinek, tulajdonságaik nagymértékben eltérnek az általánostól, különc pontoknak nevezzük. A legtöbb adatbányászati algoritmus az ilyen különc pontoknak nem tulajdonít nagy jelentőséget, legtöbbször zajnak vagy kivételnek kezeli őket. Azonban az élet egyre több területén merül fel az igény, hogy éppen az ilyen különc pontokat találjuk meg. Eltéréselemzés főbb alkalmazási területe a másolás-, koppintáskeresés, továbbá a csalások, visszaélések, vírusok, hackertámadások kiszűrése.

**Webes adatbányászat**: Az Interneten óriási adattömeg található, így az Interneten alapuló információ-kinyerő algoritmusok is az adatbányászat területéhez tartoznak. „ (BODON, 2010)

### A tudásfeltárás folyamata

„ *I. Az alkalmazási terület feltárása és megértése, fontosabb előzetes ismeretek begyűjtése, és a felhasználási célok meghatározása.*

*II. Céladatbázis létrehozása: kiválasztani a használni kívánt adatbázist, (vagy annak csak egy részét), amiből a tudást ki akarjuk nyerni.*

*III. Adattisztítás: itt olyan alapvető operációkat értünk, mint a téves bejegyzések eltávolítása, hiányos mezők pótlása, zajok szűrése stb. Zajon az adatba épült véletlen hibát értünk. Vannak zajok, amelyeket egyszerű felfedezni és javítani. Például sztring érték ott, ahol számot várunk, vagy felsorolás típusú attribútumnál érvénytelen érték található. Sajnos sok esetben a hiba észrevétlen marad (például 0.53 helyett 0.35 érték gépelése).*

*IV. Adatintegráció: a feldolgozás számára fontos, esetleg elosztott adatbázisok egyesítése.*

*A harmadik és negyedik lépést együtt gyakran nevezik az adatok előfeldolgozásának.*

*A különböző forrásból vett adatok integrációja során sok problémába ütközhetünk. A különböző osztályok különböző módon tárolják adataikat, különböző konvenciókat követnek, különböző mértékegységeket, elsődleges kulcsokat és elnevezést használhatnak, és különféle hibák lehetnek jelen. Az egész céget átfogó adatintegrációt adattárházban tárolják, mely egy speciális, az elemzést támogató adatbázis.*

*V. Adattér csökkentés: az adatbázisból a cél szempontjából fontos attribútumok kiemelése.*

*VI. Adatbányászati algoritmus típusának kiválasztása: eldönteni, hogy a megoldandó feladat klaszterezés, vagy szabály-, illetve mintakeresés, esetleg osztályozás.*

*VII. A megfelelő adatbányászati algoritmus meghatározása. Előnyeinek, hátrányainak, paramétereinek vizsgálata, futási idő- és memóriaigény elemzése.*

*VIII. Az algoritmus alkalmazása.*

*IX. A kinyert információ értelmezése, esetleg visszatérés az előző lépésekhez további finomítások céljából.*

*X. A megszerzett tudás megerősítése: összevetés elvárásokkal, előzetes ismeretekkel.*

*Eredmények dokumentálása és átadása a felhasználónak. Egy adatbányászati elemzés eredménye akkor „nem felel meg az elvárásainknak", ha nem sikerül semmilyen új, hasznos és természetesen valós összefüggést feltárni. Ennek nyilván több oka is lehet, a következőökben két példát mutatunk.*

*1. Előfordulhat, hogy rosszul választottuk meg az elemzéshez (adatbányászathoz) használt algoritmust vagy ennek paramétereit, és egy másik eljárással (vagy más paraméterekkel) találni fogunk valamilyen érdekes összefüggést. Szemléletesen szólva: más oldalról ránézve az adathegyre, lehet, hogy látunk rajta valami érdekeset.*

*2. Természetesen az is lehetséges, hogy az adatok egyáltalán nem rejtenek semmiféle új, a gyakorlatban hasznosítható összefüggést. Ekkor sajnos teljesen elölről kellkezdeni a folyamatot, új adatok gyűjtésével. „ (BODON, 2010)*

### Az adatbányászat etikai vonatkozásai

Az adatok - legfőképpen emberekre vonatkozó adatok – adatbányászati célokra való felhasználásának komoly etikai következményei lehetnek, ezért az adatbányászatot végzőknek felelősségteljesen, az ő adatbányászat alkalmazásukra vonatkozó etikai megfontolásoknak megfelelően kell cselekedniük. WITTEN, FRANK, 2000)

Ha emberekre vonatkozó adatokra alkalmazzuk, az adatbányászat könnyen diszkriminatívvá válhat. A diszkrimináció bizonyos fajtái - faji, nemi, vallási, stb. – nemcsak etikátlanok, hanem illegálisak is. Orvosi diagnózis felállításának érdekében etikus nemi vagy faji információkat gyűjteni, kölcsön visszafizetési szokásokról információt gyűjteni nem etikus. Mivel bizonyos földrajzi területekhez könnyen társíthatók az ott élő etnikumok, ezért a földrajzi területegység azonosítására alkalmas adatok felhasználása adatbányászati projektekben azok etnikai alapú interpretációjának kockázatát vonhatják maguk után. Az adatbányászathoz az adatot biztosító embereknek fel kell világosítani az adatgyűjtés céljáról, meg kell határozni, ki férhet hozzá az adatokhoz, és milyen jellegű következtetésekre szabad jutni. (WITTEN, FRANK, 2000)

### Adattárházak

„ Adatbányászkodni nagy mennyiségű adaton lehet és érdemes. Ezek lehetnek egy adattárházban, ami megkönnyíti a munkát, de nem feltétlenül szükséges. Sőt, érdemes az adatokat előbb elemezni, és az alapján építeni az adattárházakat, kialakaítani az elemzési szempontokat (például multidimenzionális adatkockákat). Kár az adattárházat teletömni, ha már előzetes elemzésekkel meghatározhatók az elvárt eredményt igazán befolyásoló változók. Ugyanígy egy jelentés esetében gyakori feladat az értékek kategorizálása értéksávok meghatározásával. Miért határozzuk meg önkényesen ezeket a sávokat, amikor azok az eloszlásokból sokal pontosabban és hasznosabban előállíthatók lennének? Nem vezet helyes marketingdöntésekhez, ha eddig az 50 mFT feletti vevőinket tartottuk kiemelt ügyfeleknek, holott a lényeges ugrás 35 mFT-nál van.” (SOMFAI, 2006)

## Hasonlóságelemzés: A COCO módszer

A diplomadolgozat részét képező kutatás során a COCO néven ismert hasonlóságelemzés módszer két különböző, fajtája került alkalmazásra. A hasonlóságelemzés célja, csakúgy mint az adatbányászat egyéb eszközeinek, hogy nagy tömegű adatból információkat nyerjünk ki, és a zajokat, irreleváns tényezőket kiszűrjük. A kétféle alkalmazott COCO - módszer a COCO-STD (standard) és a COCO-MCM (Monte-Carlo módszer) volt.

### COCO online - standard verzió

*„A COCO online additív standard verzió alkalmazásának előfeltételei:*

*Adott egy OAM, mely kialakításakor az irányvektorok meghatározása magától értetődő volt, ill. nem volt szükség/lehetőség felárak és attribútum-arányok kezelésére.*

*Az Y értékek egész számok, ill. (eltolás nélkül vagy eltolás után) nem negatívak.*

*Az attribútumok bármelyikének hiányában az Y nem kell, hogy nulla legyen (=additív hatásmechanizmus).*

*Azonos hatásmechanizmusú oszlopok a futásgyorsítás érdekében összevonásra kerültek (vö. a futtatás során az ismétlődő oszlopok hatásukat vesztik).*

*A lépcsők első sora az Y (genetikai) potenciáljaként értelmezhető.*

*A ceteris paribus összefüggések monotonak, ill. Liebig-függvényt megengedők.*

*Alkalmazási területek:*

*Benchmarking (ár/ vagy bér/teljesítmény-elemzés, üzem-összehasonlítás, regionális összevetések)*

*Előrejelzések (pl. tőzsdei elemzések monoton (Xi<->Y) fordított arányosság mellett)*

*Termelési függvények.”* (FORRÁS: Pitlik László: Magyar Internetes Agrárinformatikai Újság (2011))

### COCO-OPTI2 online - standard verzió (MCM)

„*A COCO-OPTI2 online additív standard verzió alkalmazásának előfeltételei:*

*Adott egy OAM, mely kialakításakor az irányvektorok meghatározása magától értetődő volt.*

*A lépcsők száma nem több mint az objektumok 10%-a, minimum azonban 5, (ugyanis a modellből bizonyos korlátozó feltételek elhagyásra kerülnek annak érdekében, hogy a monoton lépcső-alakzatok egy/több ponton eltörhessenek). Túl sok lépcső (s így potenciálisan sok töréspont) nem feltétlenül előnyös polinomokhoz vezet. Túl kevés lépcső esetén az optimum-hatás túl durva lesz.*

*Az Y értékek egész számok, (eltolás nélkül vagy eltolás után) nem negatívak.*

*Az attribútumok bármelyikének hiányában az Y nem kell, hogy nulla legyen (=additív hatásmechanizmus).*

*Az azonos hatásmechanizmusú oszlopok a futásgyorsítás érdekében összevonásra kerültek (vö. a futtatás során az ismétlődő oszlopok hatásukat vesztik).*

*A lépcsők első sora az Y (genetikai) potenciáljaként értelmezhető.*

*A ceteris paribus összefüggések vélelmezhetően nem monotonak (primer értelmezés szerint egy extrém értékük van, melyet maximumnak tekintünk).*

*MCM, avagy lépcsős függvény lépcső-restrikciók nélkül: Abban az esetben, ha rel. kevés lépcsővel alakítunk ki egy OAM-t, s éppen arra vagyunk kíváncsiak, milyen ceteris paribus alakzatok képzelhetők el egy tanulási minta hátterében, nem szükséges előírni, hogy a jobb helyezés milyen viszonyban álljon egy gyengébbel. 3 lépcső esetén minden lépcsőlefutás értelmezhető. Ettől felfelé a lépcsők hajlamossá válnak a hullámzásra (vö. polinom-hatás). Ha a lépcsők száma megegyezik az objektumok számával, akkor egy klasszikus MCM-t (azaz Monte-Carlo Módszert / elsődlegesen irányítatlan keresésvezérlést) kapunk, mely quasi véletlenszámokkal próbálja letapogatni a kombinatorikai teret (további részletek és az LP-kiváltás demo-ja részlegesen célirányos keresésvezérléssel).*

*Az optimalizáló modellek esetén mindenkor érdemes figyelembe venni, hogy a látszólag zavaros ceteris paribus hullámzások mögött döntési fa jellegű klasszifikációkkal növelhető a megoldás robosztussága (vö. COCO STEP).*

*Alkalmazási területek:*

*Optimum-hatásokat feltételezni engedő modellek építése (pl. termelési függvények).*

*Döntési fák szimulációja (vagyis HA/AKKOR elven magyarázható polinom-részletek keresése).*

*Rendszer-viselkedés feltárása.*

*Előrejelzések készítése.*

*Nem ajánlott: benchmarking, ill. ár/teljesítmény-vizsgálat esetén!”*(FORRÁS: Pitlik László: Magyar Internetes Agrárinformatikai Újság (2011))

# Anyag és módszertan

Annak érdekében, hogy megfelelőképpen ki tudjunk alakítani egy olyan rendszert, ami önállóan képes kiszűrni a teszt elemek írása kapcsán előforduló esetleges hibákat, többféle módszerrel is éltem. Először egy olyan tesztkérdés-sorozatot állítottam elő, amelybe az elkövethető hibák listájából, ami a szakirodalmi fejezetben megtalálható, néhányat szándékosan kiválasztottam, majd pedig odaadtam kevés számú embernek. A második módszer, amit alkalmaztam az úgynevezett COCO módszer ahol megadtam az ideális értékét a tesztsor valamennyi kitöltendő kérdésének sorrendje kapcsán. Ezeket a módszereket fogom most bemutatni.

## Az Excel táblázatkezelő szoftverrel előállított feleletválasztós tesztről

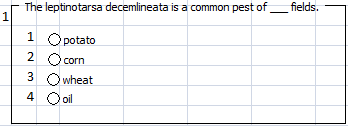
Amikor szembesültem azzal a szakirodalmi elemmel, amely a tesztkérdések írása kapcsán elkövethető hibákkal foglalkozik, rögtön felmerült a kérdés, mit is lehet tenni annak érdekében, hogy azok elkerülhetők legyenek. Bár az ezzel foglalkozó irodalmi fejezetben rengeteg féle hibával találkozhatunk, néhány alapvető felismerésre sikerült jutni ezekkel kapcsolatban:

* A problémák két nagy csoportba sorolhatók: az egyik részük abból fakad, hogy egy-egy teszt elem önmagában lehet rossz, és ezért megtévesztő vagy éppen túl könnyű a hallgatók számára. A második nagy csoportja az elkövethető hibáknak az, hogy az egyik tesztkérdésben, vagy a teszt elemnek az alternatívái között olyan nyomot „sikerül” hagyni, ami egy ettől különböző tesztkérdésnek a megválaszolását segítheti.
* Az előbbi ponthoz kapcsolódóan azt feltételeztem, hogy mindezeket a hibákat sikerülhet kiszűrni azáltal, ha a tesztalanyoknak a reakcióit kellő pontossággal sikerül megmérni. A problémák pontos okát természetesen statisztikailag nem tudjuk kimutatni, csak arra vagyunk képesek, hogy kimutassuk, hogy valamely kérdéssel baj van-e, avagy sem. Meg tudjuk mérni a kérdések kapcsán a helyes válaszadások arányát, egyenként az lehetséges opciókra adott válaszok arányát az összes válaszadáshoz képest, mérhetjük a válaszadások időpontját, és ezen keresztül a válaszadások sorrendjét.
* Az ezt megelőző pontokban tárgyalt két nagy probléma-csoportot, valamint a mérendő mutatókat a következőképpen lehet összhangba hozni: a tesztek egyéni problémái kapcsán, azaz ahol az a cél, hogy a helyes válasz, és a zavaró tényezők hogyan fejtik ki hatásukat az egyetlen helyes válasz formájú teszt elemekben, a helyes válaszok arányát, valamint a lehetséges opciókra adott válaszok arányát kell mérni. Amikor viszont azt kívánjuk felderíteni, hogy egy-egy kérdés, vagy a hozzákapcsolódó alternatívák esetleg segítették egy másik kérdés megválaszolását, a kitöltési sorrendre hagyatkozunk, azt feltételezve, hogy ha a kitöltési sorrend eltér a megadott kérdés-sorrendtől, akkor meg kell vizsgálni, hogy melyik kérdések következtek egymás után, és miért.

Azért, hogy olyan módszereket lehessen alkotni, amelyekkel hatékonyan lehet mérni a problémákat egy olyan tesztsor készült, ahol a fent felsorolt hibák egy része szándékosan elkövetésre került. Azért nem követtem el az összes hibát, mert egyrészt nem volt rá szükség ahhoz, hogy ki lehessen mutatni a kétféle hibatípust, másrészt pedig azért, mert egy túl hosszú tesztsor elvehette volna a kedvét a tesztalanyok kedvét a teszteléstől, amit el kívántam kerülni. A tesztkérdések idegen nyelven (angolul) lettek megfogalmazva, és olyan lexikális tudást mérnek, amire a legtöbb embernek nincs igazán szüksége, annak érdekében, hogy a tesztkérdések ne lehessenek már önmagukban is egyszerűek, és a tesztalanyok minél nagyobb mértékben támaszkodjanak az árulkodó nyomokra.

### A feleletválasztós tesztsor-modell előállításának lépései

A tesztsor kérdéseire az első munkalapon nyílt válaszlehetőség a tesztalanyok számára, ahol a tesztkérdések a csoportpanelekbe helyezett választógomb űrlap vezérlőelemek formájában jelentek meg, ahogy az alábbi kép is mutatja (2. ábra). Az űrlap vezérlőelemeket lehet úgy formázni, hogy az egy csoportpanelbe tartozó választógomboknak az értékéhez egy cellát csatolunk, amibe a megfelelő érték fog kerülni. Így például az első (a képen a legfelső) válaszlehetőséget kiválasztva a kiválasztott cellába egy fog kerülni. A hivatkozott cella a munkafüzet második munkalapján volt megtalálható, ahol a kalkulációk lefolytatásra kerültek.



2. ábra: Tesztkérdés az Excel-táblázatban. Forrás: saját eredmények

A kalkulációs munkalap megalkotása szembeállított néhány érdekes kihívással. A kitöltési sorrendeket úgy lehet meghatározni, hogy ha bármely tesztkérdés opcióira adott legutolsó válasz idejét mérjük meg. Ehhez szükség volt a MOST() függvényre, amiről az kell tudni, hogy mindig újraszámolja az időértéket, ha valamit történik az adott munkalapon. Ezért HA függvénnyel biztosítani kellett, hogy csak a megfelelő feltétel fellépése esetén írja ki, hogy mennyi az idő. Ez a feltétel pedig az, hogy a másik a munkalapról rögzített cella értéke, azaz a válaszérték megváltozzon.

Ahhoz, hogy a program meg tudja állapítani, hogy a változás megtörtént, több cellára volt szükség, amelyekben körkörös hivatkozások vannak.  
Először is szükség volt egy olyan cellára a C oszlopban, amely megállapítja, hogy el kell-e indítani a számítást, és amiben a következő képlet van:  
  
HA(B3<>F3;2;0)

Ebben a képletben a B oszlop értéke a csatolt cella (válasz) értéke, az F oszlopban pedig a B oszlop mindenkori értéke van (arról, hogy miért kellett még egyszer eltárolni ugyanazt az értéket, lejjebb lehet olvasni) Ha a B és F oszlopok értéke nem egyenlő, akkor a 2 értéket, egyébként 0-t ad vissza.

A következő, D jelű oszlopban az „iteráció számláló” kapott helyet. Ez a számítás elindítónak, valamint önmaga+1-nek veszi a minimumát, ami azt jelenti, hogy értéke a maximálisan megengedett iteráció számig fog növekedni, ami az Excel alapbeállításában 100, egészen addig, amíg a C értéke nem lesz egynél nagyobb:  
  
MIN(C3;D3+1)

A harmadik oszlop, amire szükség van ahhoz, hogy a MOST() függvény értéke akkor és csak akkor frissüljön, amikor a válaszban változás áll be, az előzőekben említett F oszlop, aminek az értéke a válasz jelenlegi értékét tárolja. Ebben az oszlopban a következő képlet található:

HA(D3=2;B3;F3)

Ezek után már csak magára a MOST() függvényt tartalmazó cellára volt szükség,amelyben a következő képlet található:

HA(D3=2;MOST();J3)

Ebből következik, hogy a cellában csak akkor fog kiszámítódni az idő értéke, amikor az iterációs oszlop (D) értéke 2-vel lesz egyenlő.

Ahhoz hogy, megértsük ezeknek a körkörös hivatkozásoknak a működését, álljon itt a hatásmechanizmus:

A csatolt B oszlop értéke megváltozik 🡪 ebből következően már nem lesz egyenlő F oszloppal, ahol mindenkor a B értékének kell állnia 🡪 C oszlop értéke, ahol a kalkuláció akkor indul el, ha B nem egyenlő F-el, 2 lesz 🡪 Mivel a C oszlop értéke 2, az iteráció korlátig (100-ig) számoló D oszlop minimuma a 2 lesz 🡪 Ha D oszlop egyenlő kettővel, akkor az előző értéket tároló F oszlop értéke megváltozik B oszlop értékére 🡪Ennélfogva a B és F oszlop értéke meg fog egyezni, azaz C oszlop újra 0 lesz, emiatt D oszlop minimuma is 0 lesz, nem 2.

Ezeken kívül volt még egy olyan oszlop is, amely a B oszlop előző értékét tárolta, az alábbi képlet segítségével:

HA(D3=1;F3;E3)

D oszlop értékének minimuma csak abban az idő pillanatban lehet egyenlő egyel, amikor a C oszlop értéke már kettő, de a léptetés (iteráció) még nem jutott el 2-ig. Ebben a pillanatban az F oszlop értéke még nem vált át a B új értékére.  
  
Azáltal, hogy ismerjük a korábbi értéket, meg tudjuk mondani, hogy egy-egy tesztkérdés esetében egynél többször módosították-e a választ: mivel a kitöltetlen tesztben a csatolt cella (B oszlop) értéke 0, a tárolt korábbi érték is 0 lesz egészen addig, amíg legalább egyszer nem módosul a válasz.

Az Excelben ily módon kialakított „naplózási” lehetőséggel a következő dolgokat lehetett kimutatni: az első és utolsó válaszadás között eltelt időt, minden kérdésre a legutoljára adott választ, azt, hogy legalább egyszer módosítottak-e a válaszon, természetesen magukat a válaszokat, amik összehasonlításra kerültek a helyes válasszal, így mérhető a „találati arány”.

2.1.2 A tesztsor tartalmának kialakítása

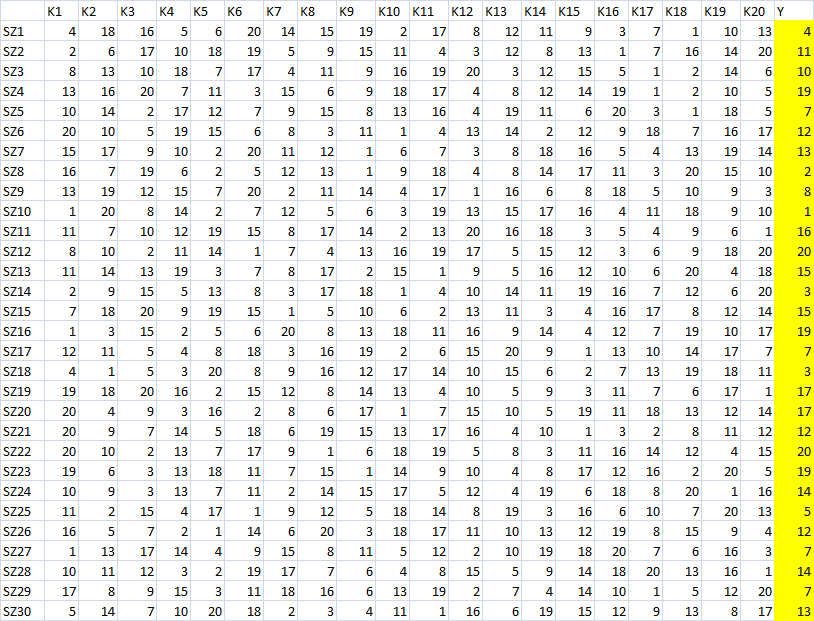
Miután a teszt formája, működése bemutatásra került, említést kell tenni arról, hogy a tartalom, azaz maguk a tesztkérdések hogyan és miért kerültek kialakításra.

A tesztkérdések által megcélzott célcsoport a Vezetés és Szervezés MSC képzés végzős hallgatói voltak. Azért, hogy a teszt ne legyen túl egyszerű ahhoz, hogy a tesztalanyok könnyedén megoldhassák, angol nyelven lett elkészítve, valamint olyan lexikális jellegű tudáselemek lettek kiválasztva, amelyekről azt feltételeztem, hogy a tesztalanyok nem tudják azokra a válaszokat, ha nem kapnak valamilyen formában segítséget hozzá. A segítség, vagy éppen a „kelepce” több féle formában is fel lett kínálva. Voltak azonban olyan kérdések is, amelyek nem tartalmaztak semmi olyan szándékos hibát, ami bármiféle tudásnak a megmérését akadályozta volna. Ezeknek a szerepe az volt, hogy ki lehessen deríteni, hogy egy olyan tesztkérdés-sorozatban vannak-e jelentős hibák, vagy csak azt tudjuk kimutatni, amit mi magunk követünk el. Ezek úgy lettek elhelyezve a tesztsorban, hogy két, szándékosan hibás tesztkérdés között sok hibátlannak vélt tesztkérdés legyen. Ha ugyanis előbbiek túl közel kerültek volna egymáshoz, akkor a tesztalanyok még a válaszadás előtt rájönnek a „turpisságra”, ami úgy kisebb eséllyel látszik meg a teszteken.

## A tesztek adatainak feldolgozása, és modellalkotás

Miután kialakítunk egy olyan módszert, amivel különböző adatokat tudunk begyűjteni, rögtön adódik a következő kérdés: hogyan fogjuk azokat feldolgozni? A választ jelen esetben a COCO hasonlóságelemzési módszer additív modulja jelentette. Az egyik ilyen modell az MCM (Monte-Carlo módszer) lényege, hogy véletlenszámokkal közelítünk egy problémát, a kapott adatokat feljegyezzük, és ebből tudunk következtetni a valóságra.  
Minderre azért volt szükség, mert mint ahogy az eredményekben ismertetésre kerül, az Excel táblázatkezelőben elkészített teszt nem tudta hatékonyan mérni a sorrendekből fakadó asszociációs lehetőségeket. Ezért véletlenszámokból álló táblázatot alkalmaztunk, ahol az alkalmazott adatbányászati módszer, a hasonlóságelemzés objektum-attribútum mátrixa a következőképpen épült fel: az objektumok (sorok) a tesztalanyok voltak, az attribútumok (oszlopok) pedig az kérdések voltak. Így egy olyan véletlenszámokkal modellezett táblázatot kaptunk, amelyben 30 kitöltőnek megvolt az adata arra vonatkozólag, hogy melyik kérdést hányadikként válaszolt meg összesen húsz kérdésből. A táblázat legutolsó oszlopában a tesztet kitöltők által elért jó válaszok száma volt, ez volt az elemzés cél-attribútuma. Így létrejött egy összesen 30\*21-es méretű táblázat, amely a hasonlóságelemzés alapjául szolgált (3. ábra).

Az online hasonlóságelemzési modulok bemeneti értékei mindenkor a rangsorszám-táblázatok (itt a rangsort a sorrendek képezték), valamint az utolsó értékoszlop a cél-attribútum. Az online futtatás kimenete az úgynevezett lépcsőtáblázat, ami megmutatja, hogy az egyes pozíciókat hogyan értékelte a rendszer. Ha ezeket a lépcső értékeket visszahelyettesítjük az objektum-attribútum mátrixba, akkor megkapjuk, sor irányba megkapjuk, hogy a modell szerint az adott objektum esetében mennyinek kéne lennie az eredménynek, oszlopirányban pedig azt, hogy az adott attribútum milyen mértékben járul hozzá az eredményhez. Lefordítva mindezt a jelen elemzésre: meg fogjuk tudni mondani, hogy az adott kitöltő (objektum) a kitöltési sorrendjével mennyit érhetett volna el a modell szerint (erre az értékre később, az eredményeknél a „becslés” kifejezés fog utalni), illetve meg fogjuk tudni mondani azt, hogy az egyes kérdéseknél mennyire játszott szerepet, hogy hányadikként válaszolták meg. Amennyiben az utóbbi értéke nagy valamely kérdés esetében, az azt jelenti, hogy annak a kérdésnek a megválaszolása kapcsán a sorrend lényeges különbséget jelentett, azaz itt kell keresni az asszociációt.



3. ábra: A COCO hasonlóságelemzés eredeti, véletlenszámokat és célértéket tartalmazó táblázata

A kérdések (attribútumok), valamint az eredményoszlop (amely a 3. ábrán sárgával van jelölve) korrelációira szükség volt. Ez az érték adta meg a COCO hasonlóságelemzés inputjának, a sorszám függvénnyel előállított rangsorszámoknak az irányát. Ha a korreláció negatív volt valamely attribútum kapcsán, akkor annak az iránya 1 volt, fordított esetben 0. A sorszám függvények egy számnak (adott objektum adott attribútumát) a számhalmazon belül (az attribútum oszlop) a csökkenő sorrend szerint helyét adják meg abban az esetben, ha sorrend-paraméternél 0-t adunk meg. Bármely 0-nál nagyobb érték esetében a sorrend növekvő.

Az ily módon előállított, rangsorból és az eredményoszlopból álló táblázat készen állt az online futtatásra.

# Eredmények

Amiképpen többféle módszerről beszéltünk, úgy többféle eredményről is beszélhetünk.

Egyrészt beszélhetünk a kiadott tesztsor eredményeiről, másrészt pedig a véletlen számokkal futtatott MCM – hasonlóságelemzésről. A két féle eszköz teljesen másfajta célokat szolgált. Az első módszer lényege az volt, hogy adatokat gyűjtsünk (bármennyit, mivel itt semmilyen jól körülhatárolható csoportról nem beszélhetünk, így reprezentativitásról sem lehet beszélni. Természetesen minél több adatot sikerül begyűjteni, annál jobb.) A második módszer alkalmazásának szükségességét az adta, hogy az első módszer nem tudta hatékonyan megmérni a tesztkitöltők viselkedésének valamennyi aspektusát. Ezért volt szükség a Monte-Carlo módszerre, amit a COCO hasonlóságelemzési modullal sikerült szimulálni. Mindazonáltal a jövőbeni rendszer, ami a bemenet tekintetében az eddiginél megbízhatóbban fogja mérni a kitöltési sorrendeket, ugyanúgy a COCO módszerre fog támaszkodni az adatok kiértékelésekor. A különbség csupán annyi lesz, hogy nem véletlenszámokkal, hanem valódi tényadatokkal fog működni.

## A kitöltött Exceles tesztek eredményei

Az Excel táblázatkezelővel készített tesztet öten töltötték ki. A kitöltők egyedi jellemzői (mint amilyen a kérdőívekben a nem, a szervezet, amihez a tartoznak, stb.), nem játszott szerepet. A cél egy olyan rendszer kialakítása, ami hatékonyan képes mérni a tesztekből levonható tanulságokat, és lehetőleg minél kevesebb adatból ismerjen fel hibákat a tesztben, és azokról a teszt keretrendszer üzemeltetői számára a megfelelő riasztásokat megtegye. A kitöltött tesztek kapcsán a legfontosabb felismerés az volt, hogy a kitöltési sorrendek nem jól korreláltak az eredményekkel. Csupán a 19. illetve a 20. kérdés megválaszolása kapcsán sikerült magas korreláció értéket kimutatni. Tulajdonképpen azt lehet mondani, hogy az esetek túlnyomó többségében a kérdéseket abban a sorrendben válaszolták meg, ahogy azok fel lettek téve. Ennek az lehet az oka, hogy a teszt alanyok valamennyi tesztkérdést megismerhették, még mielőtt bármelyikre válaszoltak volna. Így lehetőségük nyílt arra, hogy a tesztet csak azután töltsék ki, miután a tesztkérdésekre adandó válaszokat „fejben” kigondolták. Mivel a tesztalanyok gondolkodási folyamatát nem lehet lemérni, ezért ez a rövid, kevés szereplős teszt máris egy fontos, jövőbeni fejlesztési irányt szabott meg: **Meg kell akadályozni, hogy a tesztsorban, amelyben az asszociáció lehetőségét fel akarjuk kutatni, a tesztalanyok előre megismerhessék az összes kérdést, ugyanakkor biztosítani kell, hogy módosíthassák bármely korábban megválaszolt kérdésre a válaszadásukat!** Olyan rendszert kell építeni, amely egyszerre csak egy tesztkérdést mutat meg, és addig nem engedi tovább a felhasználót a következő kérdésre, amíg az éppen aktuális kérdést meg nem válaszolta. Ily módon a jövőbeni adatoknál egészen pontosan ki fog rajzolódni, hogy mely kérdések kapcsán vette észre a felhasználó, hogy egy korábbit elrontott.

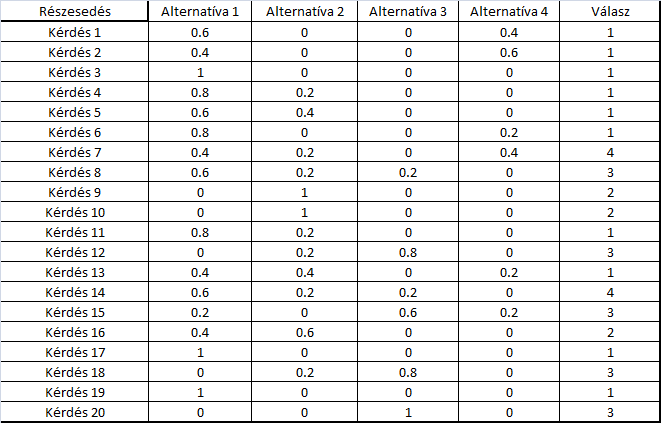
A másik kérdés volt, hogy fel tudjuk-e mérni, hogy egy-egy tesztkérdés kapcsán melyek azok a helyes válasz opciók, amelyeket gyakran, vagy ritkán választanak a tesztalanyok, így a kérdés túl triviális, vagy nehéz volt, valamint melyek azok a zavaró tényezők, amelyeket gyakran, vagy ritkán választanak, azaz hatékonyan, vagy nem hatékonyan fejtik ki a zavaró funkciójukat.

Ennek kapcsán a válaszadásokat megmértem mind a négy opcióra, majd elosztottam az összes válaszadás számával. Mivel összesen négyféle válaszlehetőség van, és optimális esetben az alternatívák kellően homogének, azaz a válaszok csak kevéssé térnek el egymástól és csak a kellően felkészült tanuló tudja megmondani a helyes választ, azt feltételeztem, hogy hosszútávon valamennyi alternatívának 25 %-ban kell részesednie az összes válaszadásból, ha a tesztkérdés megfelelően működik. Ezért többféle kimutatást is készítettem.

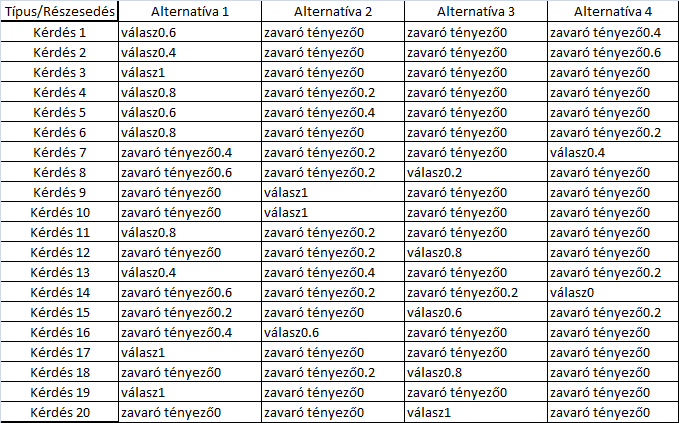
Először is szükségem volt arra a táblázatra, ami megmutatja, hogy az egyes kérdések egyes alternatívái mekkora arányban részesednek az összes kitöltésből a kitöltési statisztikák alapján (4. ábra)

Ezután készítettem egy olyan táblázatot, amiben látszik, hogy az egyes alternatíváknak mennyi az értéke, és hogy azok válaszok, avagy zavaró tényezők-e. (5. ábra)

Ebből a táblázatból le lehet olvasni, hogy az adott alternatíva válasz vagy zavaró tényező-e, illetve mennyiben részesedik a kitöltésekből. Ebből meg lehet állapítani például, hogy az válasz, aminek a részesedése 0.8 (80%), mint a 11-es kérdés esetében, az egy nagyon könnyű (túl könnyű) kérdés volt, mert a teszt alanyok nyolcvan százaléka azt választotta. Ugyanakkor a második kérdésénél a 4. alternatíva zavaró tényező, és mégis a tesztalanyok 60 %-a ezt választotta. Ebből következik, hogy az egy hatékonyan működő zavaró tényező, mert sokakat megtéveszt. Ugyanakkor észre kell venni, hogy van olyan kérdés (3, 9, 10, 17. 19), ahol valamennyi kitöltő helyesen válaszolt. És van olyan is, ahol senki nem válaszolt jól (14-es kérdés). Ilyen, és ehhez hasonló megállapításokat lehet levonni az összes alternatíva kapcsán, és a megfelelő lépéseket megtenni (Túl triviális kérdések kizárása, a jól működő zavaró tényezők felhasználása más tesztkérdésekben, illetve megvizsgálni, hogy van-e olyan hibája, ami a fenti listában található, és kijavítani.)



4. ábra: Alternatívák részesedése az összes kitöltésből, valamint helyes válasz. Forrás: Saját eredmények



5. ábra: Alternatívák részesedése, típusonként. Forrás: Saját eredmények

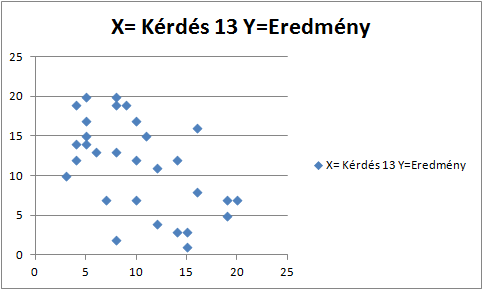
További eredményként kaptuk még a tesztekből a teszteléssel eltöltött időt, amelyből azt lehetetett levonni, hogy a teszteléssel átlagosan eltöltött idő: 00:08:55 volt, azaz 8 perc 55 másodperc. A tesztelésre fordított idő, és a helyes válaszok számának korrelációja 0.581170069 volt, ami azt jelenti, hogy az eddig begyűjtött adatok alapján a tesztelésre fordított idő nem befolyásolja jelentősen az eredményeket, ez a korreláció alacsonynak mondható.  
Ilyen eredményeket tetszőlegesen sok kitöltésen meg tudunk mérni. Ezek az eredmények a kitöltések számának függvényében dinamikusan tudnak változni egy online keretrendszerben, ahol éppúgy megmérhetőek.

## A sorrendek mérésének eszköze: A COCO futtatás eredményeinek értelmezése

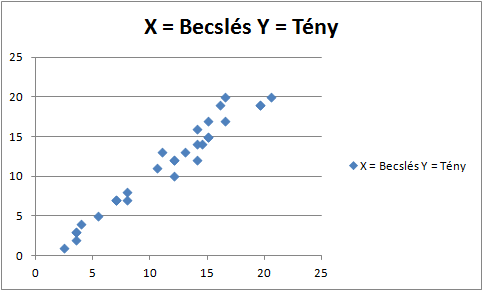
A COCO online módszernek több futtatása volt, ami többféle eredménnyel szolgált. Mivel a módszer lényege az, hogy véletlenszámokat generálva vonjuk le következtetéseket valós problémákra, ezért az eredményeknek azok a szövegsablonok tekinthetők, amit egy-egy ilyen elemzésből ki lehet olvasni. Ezek a szövegsablonok ugyanis előállíthatók a valós adatokból is, és hasonló modell-pontosságot lehet elérni a valós adatok körében is.

Az első eredménynek és legfontosabb eredmény, amit figyelembe kell venni, az a tény és a becslés értékek közötti korreláció. Az attribútumonkénti korrelációnál a negatív, illetve pozitív előjel a fontos. A korreláció a két értékhalmaz közötti lineáris összefüggést jelenti, értéke 1 és -1 közé eshet. Ha a korreláció negatív, akkor fordított arányosság van, és a függvény csökkenő, ha pozitív, akkor növekvő. Ha a korreláció magas, akkor a függvény lineáris, monoton függvény, egyébként pedig nem lineáris a függvény. A 13-as kérdés és az eredményoszlop között volt a legnagyobb korreláció, ami negatív irányú volt (6. ábra). Amint az ábrán látszik, minél később válaszolta meg valaki a 13-as kérdést, annál rosszabb eredményt ért el. De ez nem lineáris függvény, ami egyenes következménye annak, hogy a korreláció alacsony, mindössze -0,53.

Ugyanakkor a tény és becslés korrelációja magas volt, ami jelzi a modell nagy megbízhatóságát. A korreláció ebben az esetben 0,977449 volt, ami az alábbi ábrán (7. ábra) jól látszik. Mivel a korreláció ebben az esetben pozitív volt, a függvény növekvő.



6. ábra: a 13-as kérdés és az eredmény korrelációja



7. ábra: A tény és a becslés korrelációja a COCO direkt futtatásánál

Az attribútumonkénti korreláció ugyan módszertani kérdés, ahogy az a módszertani fejezetben le lett írva, csak a rangsortáblázat irányultságának meghatározásánál játszik szerepet. Csupán a korreláció fogalmának feloldása szempontjából került ide a diagram a tény-becslés diagramhoz, ami eredmény, és a modell jóságát hivatott szemléltetni. Azt, hogy egyes attribútumok milyen mértékben járulnak hozzá a célfüggvényhez, az online hasonlóságelemzés eredményeiből kapjuk meg. Ehhez a már említett, az objektum-attribútum mátrixba visszahelyettesített lépcsőket kell megvizsgálni. Az attribútumok hozzájárulását úgy kapjuk meg, ha az egyes lépcsőértékeket tartalmazó attribútum-oszlopoknak az értékeinek az összegét elosztjuk a teljes objektum-attribútum mátrixnak az összegével. Az így kapott százalékok az elsőtől az utolsó kérdésig a következőképpen alakultak a véletlen-generált adatsorban:

* Kérdés 1: 20 %
* Kérdés 2: 0 %
* Kérdés 3: 4 %
* Kérdés 4: 0 %
* Kérdés 5: 4 %
* Kérdés 6: 8 %
* Kérdés 7: 6 %
* Kérdés 8: 7 %
* Kérdés 9: 1 %
* Kérdés 10:6 %
* Kérdés 11: 18 %
* Kérdés 12: 8 %
* Kérdés 13:6 %
* Kérdés 14: 0 %
* Kérdés 15: 3 %
* Kérdés 16: 6 %
* Kérdés 17: 1 %
* Kérdés 18: 0 %
* Kérdés 19: 3 %
* Kérdés 20: 0 %

Amit egy ilyen adatsorból le lehet olvasni: például, hogy az első kérdésre adott válasz nagymértékben befolyásolja az eredményt. Mivel a korrelációból ismerjük az irányt, azt is tudjuk, hogy azok értek el jobb eredményt, akik az első kérdést minél később válaszolták meg. Az első kérdés és a válasz korrelációja pozitív volt. A második, illetve negyedik kérdésnél nem számított, hogy milyen sorrendben válaszolták meg azokat. Ezek az attribútumok zajnak minősültek, az eredményre gyakorolt hatásuk irreleváns. Így kizárható esetükben az asszociáció lehetősége. A 11-es kérdés szintén nagymértékben befolyásolta az eredményt, de iránya ellentétes, a korreláció negatív volt. Ez azt jelenti, hogy a feladat szempontjából fontos volt, hogy ki milyen korán válaszolta meg ezt a kérdést. Tekintve, hogy COCO rövidítés objektum-összehasonlítást takar, meg kell emlékeznünk arról, hogy milyen következtetésekre lehet jutni az elemzés eredményeiből a teszt kitöltőkre. A feljebb látható, tény-becslés kapcsolatot bemutató ábra mutatja meg, hogy milyen pontosan becsülte meg a hasonlóságelemzési algoritmus azt, hogy adott válaszadási sorrenddel hány pontot illene elérnie adott kitöltőnek. Amennyiben a becslés értéke kevesebb, mint a tény, azt mondhatjuk, hogy az adott sorrenddel a tesztalany valamit megérthetett volna, felismerhetett volna valamilyen összefüggést két tesztkérdés között (asszociációt) de valamilyen oknál fogva ezt nem tette. Ha a becslés értéke nagyobb, mint a tény, akkor kreatív tesztelőről beszélhetünk.

A modellek helyességének mérésére van eszköz a korreláción kívül is: az inverz futtatás. Az inverz futtatás esetén a rangsormátrix az eredeti futtatáshoz képest fordított irányokkal működik. Ha a direkt és az inverz futtatás esetében a tény és becslés különbségeinek előjelei megegyeznek, akkor az adott objektum helyzete a többihez képest nem határozható meg kellő pontossággal. A direkt, illetve indirekt futtatások eredményei a mintában 16-szor tértek el. Ezekben az esetekben a direkt és inverz futtatásoknál a tény és a becslés közötti különbségek előjelei nem egyeztek meg, ami jónak tekinthető.

## Milyen következtetésekre enged juttatni a COCO-MCM?

A COCO-MCM abban tér el a korábban ismertetett, szintén additív hasonlóságelemzési módszertől, hogy az irány itt nem játszik semmiféle szerepet. Azaz nem jelentjük ki semmilyen értékoszlop kapcsán, hogy „minél kisebb annál jobb”, vagy fordítva. Azt feltételezzük, hogy a legjobb érték nem csak a legmagasabb, vagy legkisebb érték lehet, hanem lehetséges, hogy egy középső érték is lehet a legjobb.

Ennek következtében az MCM - futtatás eredményei másmilyenek lettek, mint amilyenek az irányított keresésvezérlés esetében voltak, miután itt már nem játszott szerepet a monotonitás.

Az MCM futtatás esetében az alábbi eredmények jöttek ki az attribútumok kapcsán:

* Kérdés 1: 6 %
* Kérdés 2: 1 %
* Kérdés 3: 1 %
* Kérdés 4: 20 %
* Kérdés 5: 0 %
* Kérdés 6: 0 %
* Kérdés 7: 3 %
* Kérdés 8: 6 %
* Kérdés 9: 0 %
* Kérdés 10: 6 %
* Kérdés 11: 4 %
* Kérdés 12: 2 %
* Kérdés 13:14 %
* Kérdés 14: 0 %
* Kérdés 15: 11 %
* Kérdés 16: 8 %
* Kérdés 17: 1 %
* Kérdés 18: 8 %
* Kérdés 19: 0 %
* Kérdés 20: 10 %

A számok itt nem minden esetben alakultak hasonlóképpen, mint a COCO-STD esetében. Azt lehet mondani, hogy ha nem adjuk meg, hogy minél később, vagy korábban célszerű megválaszolni egy-egy kérdést, akkor a negyedik kérdés megválaszolásának a sorszáma alakítja a leginkább az elért eredményt.

Ahol az MCM futtatás eredménye megegyezik, arról magabiztosan ki lehet jelenteni, hogy ott valaminek a nyomára bukkantunk. Ahol pedig nem, ott további vizsgálatok indokoltak.

A COCO-MCM esetében a tény-becslés korreláció értéke 0,905005 volt, ami alacsonyabb, mint a COCO-STD esetében, de nem számottevően.

# Javaslatok

## A tesztrendszer mérési problémájának kiküszöbölése

A kiadott tesztek eredményeinek tükrében egyértelműen ki lehet jelenteni, hogy amennyiben szeretnénk kiszűrni a lehetséges hibákat a feleletválasztós tesztsorokban, akkor nem szabad megengednünk, hogy a teljes tesztet láthassa a hallgató már a teszt kitöltésének megkezdésekor. Ha ugyanis ezt megengedjük, akkor adott a lehetőség, hogy a tesztalany a tesztet csak azután töltse ki, miután valamennyi tesztkérdést megismert és a „véleményét” az egyes teszt elemekkel kapcsolatban tetszőleges sorrendben kialakítsa, ami akár a kérdések feltevésének sorrendjével is megegyezhet. Ez azért jelent problémát, mert a naplófájlba csak azt a tevékenységet tudjuk dokumentálni, amit a tesztalany az online keretrendszerben hajt végre, azaz: tudjuk dokumentálni, hogy mikor és hova kattint, de nem tudjuk, hogy mire gondol.  
A probléma kezelésére egy egyszerű szabályt sikerült kialakítani. Az elkészítendő online rendszerben a kérdések egyesével, sorban kerülnének felkínálásra. A tesztalany csak akkor tekintheti meg a következő kérdést, amint az éppen aktuális kérdésre válaszolt, így rögzítésre kerül az első gondolata kérdéssel kapcsolatban. Miután a kérdést megválaszolta, a következő kérdést megkapja. A korábbi kérdésekre bármikor visszamehetne, és azokra adott válaszát módosíthatja. Ily módon, ha a tesztsor bármely későbbi eleméből bármilyen, akár pozitív, vagy negatív következtetést vonnak le a tesztalanyok, le tudjuk majd szűrni abból, hogy hányadik kérdés megismerésénél döntenek gyakran úgy, hogy valamely korábbi kérdésre a válaszukat módosítani szükséges.

## Hogyan viszonyul a teszt keretrendszer az ötletül szolgáló, magyarázó alrendszerrel ellátott nyelvi tesztrendszerhez?

Amint az a 1.3.4.-es e-learningről szóló szakirodalmi fejezetben leírásra került, az online tesztrendszereknek két elkülöníthető válfaja van: lehetnek gyakorlatok, ahol a tanulók azonnali visszacsatolást kapnak, és lehet teszt, ahol semmilyet, vagy csak a helyes válasz/összes kérdés arányát kapják meg. A bevezetőben említett, és URL-el hivatkozott német nyelvi tesztrendszer a vonatkozó irodalmi fejezetben megismert elvek alapján a gyakorlatok közé sorolható: a válaszadó azonnal megismeri a válasz helyességének/helytelenségének okát.

Az új rendszerben viszont a tesztelésre kerülne a hangsúly, a megismert hibákat kiszűrve biztosítani lehetne, hogy bármilyen tananyag megismerését hatékonyan meg lehessen mérni a tesztalanyok körében. Mindazonáltal az e-learing rendszer akkor érhetné el igazán a célját, az automatizált tanítást, ha a tananyag megismertetését, amit a tesztek mérnek, éppúgy biztosítaná. Ezért a magyarázó rendszer nem veszne el, csak átalakulna. A magyarázó rendszer alapját jelenthetné a feldolgozandó tananyag (ami, a nyelvtanítás példájánál maradva lehetne: nyelvtani szabálygyűjtemény, olvasókönyv a szövegértési feladatokhoz, vagy akár hanganyag is hallott szöveg feldolgozásának mérése kapcsán.)

Miután az e-learning rendszer adatbázisába egy-egy tananyag felkerült, az üzemeltetők következő, könnyűnek nem mondható feladata lenne, hogy a tananyag különböző tartalmi egységei (bekezdés, alfejezet, fejezet) alapján a tesztkérdéseket megfogalmazza. A tesztkérdések így rögtön hozzárendelődhetnek a tananyag különböző részeihez. Az üzemeltetők akkor lennének készen a munkával, amikor a könyv teljes egészében, lehetőleg minél több kérdéssel lefedésre kerül.

Ily módon, lehetőség nyílik arra, hogy a tesztalanyok a tesztsor kitöltése után ne a helyes, illetve helytelen válaszokkal, és azok okaival szembesüljenek, hanem a tananyag részegységeivel, amelyeket hiányosan tanultak meg. A tesztalany akkor éri el célját, azaz tanulja meg a tananyagot, ha a teszt összes kérdésére helyesen tudja a választ, amit nem azáltal tanul meg, hogy rögtön a kitöltés után szembesül a helyes válasszal, hanem úgy, hogy a rendszer egészen addig fel fogja kínálni az átnézendő tananyagot, amíg az összes, az adott mérési célt lefedő kérdésre helyes választ nem ad.

Akik regisztrálnak a rendszerben (aminek használata több okból is regisztrációhoz kötött lenne), azok egy nagyon jó képet kaphatnak arról, hogy hol állnak a tananyagban. Meg lehetne jeleníteni numerikusan és grafikusan, hogy melyik témakör (mérési cél) tekintetében a kérdések hány százalékára adott már helyes választ, így azonnal világossá válna, hogy mely tudáselemeket szükséges a továbbiakban elsajátítania, illetve melyek a gyenge pontjai, amelyeken dolgoznia kell.

## Milyen módokon válhatna a rendszer hasznossá a tanárok számára?

A tesztsorok kitöltésével kapcsolatban nem csak maguk a tesztalanyok egyenként kaphatnának információkat, hanem az összes kitöltött teszt alapján az összes nyelvtanuló képességeiről képet lehetne kapni, ami így hatékonyan le tudná fektetni a tanórák követendő irányvonalát. Egy pillanat alatt ki lehetne rajzolni, hogy a tananyag mely pontjai jelentenek problémát a tanulók legnagyobb részének.  
Mindazonáltal, egy tanárt valószínűleg sokkal inkább érdekel, hogy a saját tanulói, tanulócsoportjai mennyire sikeresen sajátítják el a tananyagot. Ezért létre lehetne hozni az úgynevezett „social networking” portálok mintájára, mint amilyen például a Facebook, egy belső közösségi hálót az e-learning portálon, ahol a tanárok és a hallgatók összekapcsolódhatnának. A tanár ezután a személyes foglalkozásokon kialakított csoportoknak megfelelően kialakíthatna online csoportokat is, és az online tananyagban kiadhatná a házi feladatot, amit a tanulóknak a következő foglalkozásra meg kell tanulni, és le is kell tesztelni. A tanár így az összes csoportjával kapcsolatban egy képet kaphatna arról, hogy mit kell tanítania az órán, vagy a tananyag olyan elemeit, amit a tesztrendszeren keresztül el tudnak sajátítani, rábízná arra, és a személyes találkozók alkalmával több idő maradna a gyakorlati, személyek közötti interakciót igénylő feladatok elvégzésére (azaz, idegen nyelvi órákon a beszélgetésre).

## Milyen eszközökkel igyekszik a rendszer megakadályozni, hogy rossz adatokból téves következtetésekre jusson a rendszer?

A teszt készítése során felmerült néhány olyan probléma, amely akadályozhatja a hibás tesztek detektálását. Az egyik ilyen probléma azoknak a tesztalanyoknak a kezelése, akik nem tudatosan, hanem csak véletlenszerűen kattintgatva, tippelnek a válaszokra. Az ilyen, irracionális viselkedést tanúsító tanulókat nem lenne ugyan helyes eltiltani a tanulás lehetőségétől, Viszont, okszerűnek tűnik, hogy az asszociációs lehetőségek felderítésénél ne játsszanak szerepet ezeknek a tesztalanyoknak az eredményei.

Ennek elkerülésére valamennyi új regisztrálót kipróbált, szándékosan rossz tesztek elé kellene állítani, hogy kiderülhessen az, milyen módszerekkel, attitűddel közelíti meg az elé állított teszteket. A kitöltött tesztek értékelésénél szerepet játszana, hogy felismerte-e a nyomokat a tesztekben, valamint, hogy az átlagosnál szignifikánsan rövidebb idő alatt töltötte-e ki a tesztet, ami szintén a kapkodó, tippelő tesztelő ismérve.

Az ily módon felderített tesztelőknek az eredményei ugyan tárolva lenne, de nem lenne kiértékelve. Az efféle attitűd felmérés észrevétlenül, bizonyos időközönként megismételhető lenne, és ha egy későbbi felmérés pozitív eredményeket mutatna, a tesztalany adatainak felhasználása engedélyezhető lenne. Itt kell megemlíteni, hogy a tesztalanyok attitűdjére, „racionalitására” vonatkozó megállapítások sosem kerülnének a széles nyilvánosság elé, és a rendszer felhasználóktól a felhasználónéven, jelszón kívül semmit sem kérne, így az adatbányászati fejezetben megfogalmazott, az adatbányászathoz köthető etikai megfontolások figyelembe vételre kerülnének.

# Összefoglalás

E dolgozat egy olyan online, feleletválasztós tesztrendszernek az alapkövét fekteti le, mely képes felderíteni a feleletválasztós tesztsorok megírásakor elkövethető hibákat, amelyek adódhatnak a tesztelem (tesztelem) saját hibáiból, azaz lehet önmagában is megtévesztő, vagy nyomravezető. A dolgozat megírásához a motivációt egy régebb óta létező, német nyelvi tesztrendszer adta, amelynek sajátossága, hogy nem csak a helyes/helytelen válaszadás tényét, hanem az indoklást is közölte a felhasználóval  
**Szakirodalmi** téren a dolgozat kitér az említett feleletválasztós tesztek oktatásban való felhasználhatóságára, annak előnyeivel és hátrányaival, nyomatékosítva: a tesztrendszer nem a tudás elsajátításának kulcsa, hanem (remélhetőleg) hatékony segédeszköz csupán. Bemutatásra kerül a feleletválasztós tesztek valamennyi formája, valamint azok az elkövethető hibák, amelyek hátrányosan befolyásolják a tesztek mérési céljainak elérését Kitér az e-learning rendszerek sajátosságaira, mely eszközök közé sorolható a tesztrendszer is, azok előnyeivel és hátrányaival együtt. A tesztrendszer elkészítése során, miután szembesültem azzal, hogy a feleletválasztós tesztek során az említett hibák elkövethetők, célként jelent meg az elvárás, hogy a rendszer lehetőleg önmagától legyen képes ezeknek a felderítésére. Ennek eszközei a gépi tanulás, valamint az adatbányászat, amelyeknek együttes célja, hogy adatokban rejtett kapcsolatokat fedezzenek fel. A szakirodalmi rész zárásaként bemutatásra kerülnek azok a fejlesztő eszközök, amelyek szükségesek ahhoz, hogy a keretrendszer online környezetben megvalósulhasson.  
Az **anyag és módszertan** fejezetben a célok eléréséhez alkalmazott eszközök, illetve elvek kerülnek bemutatásra. A fejezet két módszert is bemutat, amelyek alkalmazásra kerültek. Az egyik egy olyan, Excel táblázatkezelő segítségével készített feleletválasztós tesztsor, amely képes dokumentálni, hogy mely tesztkérdésre melyik válasz-alternatívát választották, a válaszadás helyes/helytelen volt-e, mennyi időt töltött a tesztalany a tesztkérdések megválaszolásával, illetve, hogy milyen sorrendben töltötte ki azokat. A másik a Monte Carlo módszeren alapuló, additív hasonlóságelemzés, amelynek célja volt relatíve nagy tömegű adat szimulálása annak érdekében, hogy a megfelelő fejlesztési irányok meghatározhatók legyenek a tesztrendszer hibafelismerő eljárásai számára. Az additív hasonlóságelemzés lesz az eszköze annak, hogy a kitöltési sorrendek és az eredmények kapcsolatából a gép következtetéseket vonjon le a tesztkérdések között fennálló esetleges asszociációs lehetőségekre. Az **eredmények** fejezetben ismertetésre kerülnek a két alkalmazott módszer eredményei, és az azokból levonható következtetések.Az eredményekből közvetlenül az éles működtetés során majdan levonható következtetés lettek kialakítva, bemutatva, hogy az alkalmazott módszerekkel milyen eredményekre lehet jutni. A **javaslatokról** szóló fejezetben az eredményekből közvetlenül levonható, az online keretrendszer fejlesztésére irányuló lépések mellett néhány alfejezet lett szentelve olyan elképzeléseknek, amelyek az eredményektől ugyan függetlenek, de már a kutatások lefolytatása előtt elérendő célként voltak kezelve. Az alkalmazott módszerekből következnek azok a javaslatok, amelyek arra vonatkoznak, hogyan kell kiküszöbölni a lefolytatott mérések pontosságát veszélyeztető problémákat, illetve hogyan kell elkerülni az elemzés szempontjából rossz adatok bekerülését az elemző rendszerbe. Az alkalmazott módszerektől független javaslatok a tesztrendszernek a tananyaghoz való viszonyulásáról, a dolgozat megírását motiváló, magyarázó alrendszerrel ellátott német nyelvi tesztrendszer megújítását célzó törekvésről, valamint a rendszer adatainak felhasználására vonatkozik.

# Irodalom- és hivatkozásjegyzék

Könyvek:

1. FORGÓ Sándor: Az eLearning fogalma. In: HUTTER Otttó – MAGYAR Gábor - MLINARICS József: E-LEARNING 2005 (eLearning kézikönyv), Műszaki Könyvkiadó, 2005. 14.
2. NILSSON, N. J. (1996): Introduction to Machine Learning, Stanford University
3. SOMFAI Z. (2006): Adatbányászat: Az Információbróker elemző eszköze. *Információból üzleti érték*, Magyar információbrókerek Egyesülete, Budapest
4. WITTEN, I. H., FRANK, E (2000): Data mining: Practical Machine Learning Tools and Techniques with Java Implementations, Morgan Kaufmann Publishers

Hivatkozások:

1. Atkinson T. & Davies G. (2011) Computer Aided Assessment (CAA) and language learning. Module 4.1 in Davies G. (ed.) Information and Communications Technology for Language Teachers (ICT4LT), Slough, Thames Valley University [Online]. Available at: <http://www.ict4lt.org/en/en_mod4-1.htm> [Letöltve: 2011.10.25.]
2. Bodó Balázs E-learning módszertan-, tananyag- és technológiafejlesztés a felsőoktatásban (2006): <http://www.feek.pte.hu/feek/feek/index.php?ulink=665>
3. Bodon Ferenc: Adatbányászati algoritmusok (2010): http://www.cs.bme.hu/~bodon/magyar/adatbanyaszat/tanulmany/adatbanyaszat.pdf

[Letöltve: 2011.10.25.]

1. Burton, S.J., Merril, P.F., Sudweeks, R.R., Wood, B.(1991): How to prepare Better Multiple Choice Test Items: Guidelines for University Faculty [Letöltve: 2011.10.25.]
2. Chapple, M: What is SQL? Introduction to the Structured Query Language (2011): <http://databases.about.com/od/sql/a/What-Is-Sql.htm>
3. Haladyna, T.M., and Downing, S.M. (1989): A taxonomy of multiplechoice

item-writing rules, Applied Measurement in Education. pp. 37-50. <http://testing.byu.edu/info/handbooks/Multiple-Choice%20Item%20Writing%20Guidelines%20-%20Haladyna%20and%20Downing.pdf> [Letöltve: 2011.10.25.]

1. MIAU-WIKI: Liebig-féle minimum-elv (2011) <http://miau.gau.hu/mediawiki/index.php/Liebig-féle_minimum-elv> [Letöltve: 2011.10.25.]
2. MIAU-WIKI: Monte-Carló módszer (2011): <https://miau.gau.hu/mediawiki/index.php/Monte_Carlo_Módszer> [Letöltve: 2011.10.25.]
3. Mit tud a PHP? <http://www.php-blog.hu/php-magyar-kezikonyv/intro-whatcando.html> [Letöltve: 2011.10.25.]
4. Pitlik László: Magyar Internetes Agrárinformatikai Újság (2011): COCO OPTI 2 <http://miau.gau.hu/myx-free/index_e4.php3?x=e04> [Letöltve: 2011.10.25.]
5. Pitlik László: Magyar Internetes Agrárinformatikai Újság (2011): COCO STD <http://miau.gau.hu/myx-free/index_e.php3?x=e01> [Letöltve: 2011.10.25.]
6. Pitlik László: Magyar Internetes Agrárinformatikai Újság (2011): Német nyelvi teszt <http://miau.gau.hu/myx-free/index.php3?x=de1> [Letöltve: 2011.10.25.]
7. Pitlik László: Magyar Internetes Agrárinformatikai Újság (2004): Lépcső <http://miau.gau.hu/miau/74/mtn2004_full.doc> [Letöltve: 2011.10.25.]
8. Varga Viktor 2009: Mezőgazdasági Szaktanácsadás online támogatással: Online Benchmarkin Egyedi üzemi adatok alapján <http://miau.gau.hu/myx-free/files/studies/tanacsadas_vv_full_hu.pdf> [Letöltve: 2011.10.25.]
9. Wikipedia 2011: Algoritmusok <http://hu.wikipedia.org/wiki/Algoritmus> [Letöltve: 2011.10.25.]
10. Wikipedia 2011: Korreláció <http://hu.wikipedia.org/wiki/Korreláció> [Letöltve: 2011.10.25.]
11. What is Html? <http://www.w3.org/TR/html401/intro/intro.html#h-2.2> Letöltve: [2011.10.25]
12. What is PHP? <http://hu.php.net/manual/en/intro-whatis.php> [Letöltve: 2011.10.25.]
13. E-learning <http://e-learning.vacau.com/elearning.html> [Letöltve: 2011.10.25.]

# Definíciós jegyzék

**Algoritmus:** Algoritmuson vagy inkább eljáráson olyan megengedett lépésekből álló módszert, utasítás(sorozato)t, részletes útmutatást, receptet értünk, amely valamely felmerült probléma megoldására alkalmas. Például eljárást, algoritmust, receptet lehet adni egy „kombo” asztal (vagy egyéb bútor) összeszerelésére, valamilyen élelmiszer, mondjuk sajt (vagy bármilyen tejipari termék) elkészítésének módjára, a Deák térről a Lánchídhoz vezető út megtalálására, vagy éppen két egész szám legnagyobb közös osztójának kiszámolására. A számítógépes programok általában tartalmaznak algoritmusokat, ezekkel utasítják a gépet az adott feladat végrehajtására.

(Forrás: http://hu.wikipedia.org/wiki/Algoritmus)

**Korreláció:** A matematikában (a statisztikában) a korreláció jelzi két tetszőleges érték közötti lineáris kapcsolat nagyságát és irányát (avagy ezek egymáshoz való viszonyát). Az általános statisztikai használat során a korreláció jelzi azt, hogy két tetszőleges érték nem független egymástól. Az ilyen széles körű használat során számos együttható, érték jellemzi a korrelációt, alkalmazkodva az adatok fajtájához.

(Forrás: http://hu.wikipedia.org/wiki/Korreláció)

**Lépcső:** A COCO keretében a rangsorszámokat egy speciális helyettesítési érték váltja fel. EZ a helyettesítési érték azt fejezi ki, melyik magyarázó tényező milyen mértékben járult hozzá ahhoz, hogy a tényleges állományváltozások bekövetkezhettek. A behelyettesítési értékek lépcsős függvényt alkotnak attribútumonként, melyben a jobb rangsorszámhoz nagyobb (vagy azonos) helyettesítési érték tartozik, mint az ezt követő lépcsőfokhoz. A lépcsőket jelen esetben az Excel solver funkciójára támaszkodva kaphatjuk meg. (FORRÁS: miau.gau.hu/miau/74/mtn2004\_full.doc)

**Liebig-függvény:** "1837": Justus von Liebig német mezőgazdasági vegyész 1837-ben fedezte fel, hogy a növények által a talajból felvett tápanyagok voltaképpen ásványi sók. Innen már csak három év választotta el, hogy közzétegye híres törvényét. E szerint hiába áll rendelkezésre egy adott tápanyag (de akár más erőforrás), ha egy másik nélkülözhetetlen tápanyag/erőforrás nem áll elegendő mennyiségben jelen. Tehát a minimumban lévő erőforrások határozzák meg a maximális teljesítményt. (Forrás: http://miau.gau.hu/mediawiki/index.php/Liebig-féle\_minimum-elv)

**Monte-Carlo módszer:** 1965:"A „Monte Carlo-módszerek” néven ismert számítási eljárások a numerikus matematika legkülönbözőbb területein alkalmazhatók. Közös alapelvük: sztochasztikus modellek szimulációja számítástechnikai eszközök segítségével, majd a kísérletek eredményeként kapott numerikus jellemzők feljegyzése és kiértékelése. Ezért minden ilyen módszerre a sztochasztikus szimuláció vagy sztochasztikus modellezés, illetőleg a Monte Carlo-módszer elnevezést alkalmazzuk. A Monte Carlo-módszerrel kapott eredmény hibája nem becsülhető meg előre elég jól, ezért a hibabecslés rendszerint a modellben szereplő valószínűségi változók szórásának kiszámítása alapján történik. Az eredmény az esetek többségében nem reprodukálható pontosan, viszont a felhasznált elektronikus számológépek szórványos hibáival szemben elég stabilis." (Forrás: <https://miau.gau.hu/mediawiki/index.php/Monte_Carlo_Módszer>)

# Rövidítések jegyzéke

API: Application Programming Interface

CGI: Common Gateway Interface

COCO: Component-based object-comparison for objectivity: komponens alapú objektum összehasonlítás az objektivitás érdekében

COM: Component Object Model

CORBA: Common Object Request Broker Architecture

DCL: Data Control Language; Vezérlő nyelv

DDL: Data Definition Language; Adatdefiníciós nyelv  
DML: Data Manipulation Language; Adatmanipulációs nyelv  
DOM: Document Object Model  
HTML: Hypertext Markup Language

IMAP: Internet Message Access Protocol

LDAP: Lightweght Data Access Protocol

MCM: Monte-Carlo módszer

NNTP: Network News Transfer Protocol

OOP: Object-Oriented Programming, objektum-orientált programozás

PHP: Hypertext Preprocessor

POP3: Post Office Protocol 3

POSIX: Portable Operating System Interface for Unix

SAX: Simple API for XML

SNMP: Simple Network Management Protocol  
SQL: Structured Query Language; Strukturált lekérdező nyelv

XML: Extensible Markup Language; Kiterjeszhető leíró nyelv

WDDX: Web Distributed Data eXchange

# Ábrajegyzék

1. ábra: Az e-learning kapcsolata a számítógép alapú, web alapú, illetve távoktatással Forrás: Bodó Balázs

2. ábra: Tesztkérdés az Excel-táblázatban. Forrás: saját eredmények

3. ábra: A COCO hasonlóságelemzés eredeti, véletlenszámokat és célértéket tartalmazó táblázata

4. ábra: Alternatívák részesedése az összes kitöltésből, valamint helyes válasz. Forrás: Saját eredmények

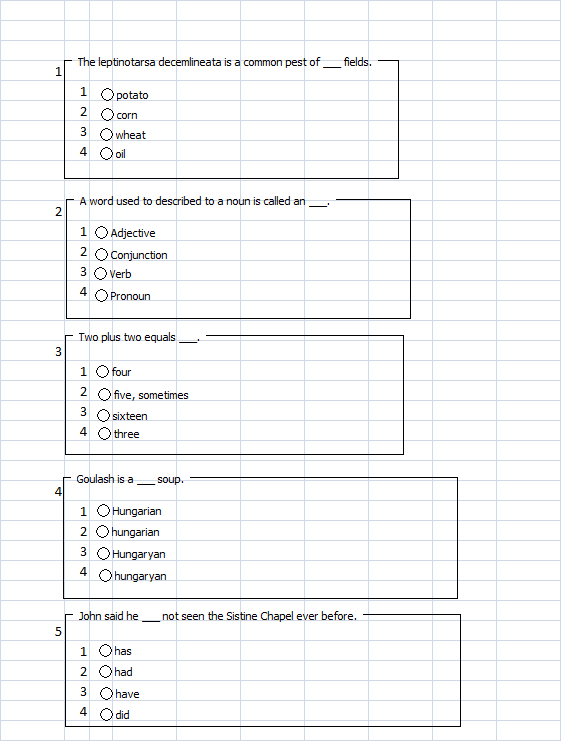
5. ábra: Alternatívák részesedése, típusonként. Forrás: Saját eredmények

6. ábra: a 13-as kérdés és az eredmény korrelációja

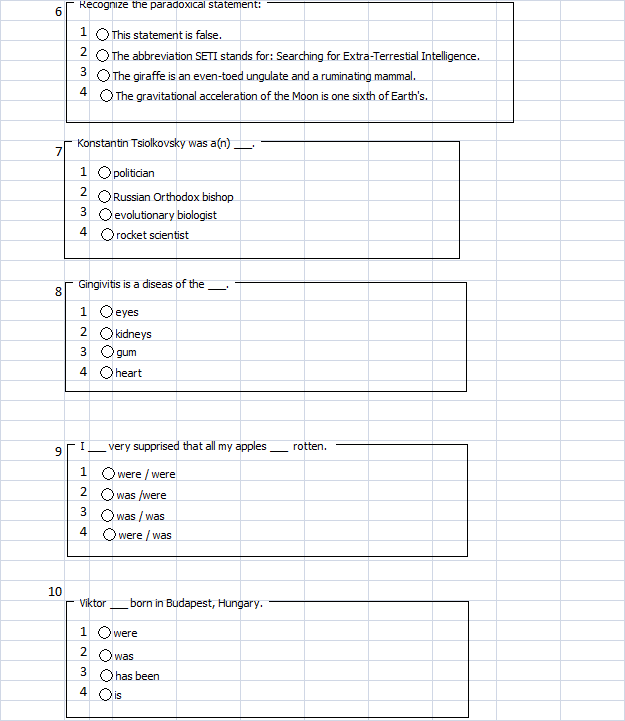
7. ábra: A tény és a becslés korrelációja a COCO direkt futtatásánál

# Mellékletek

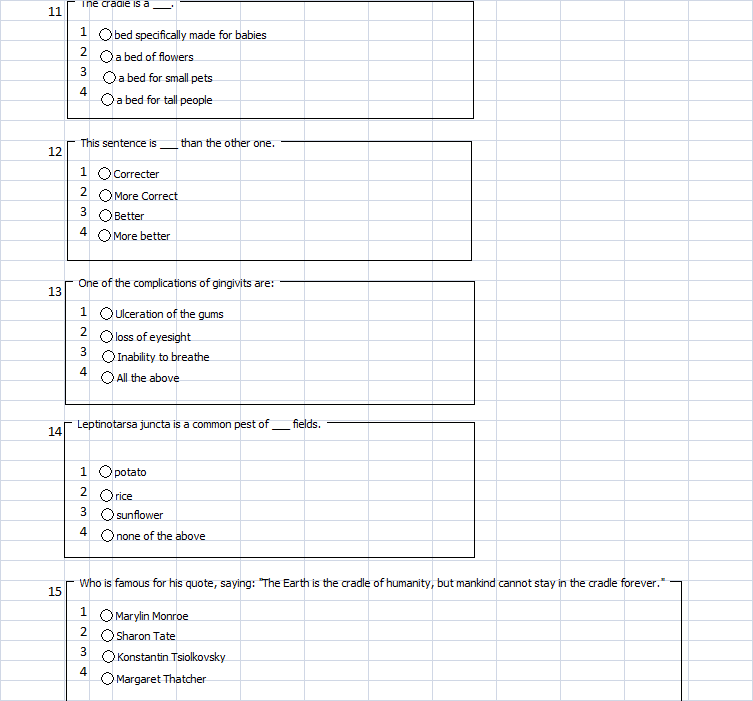
1. **Melléklet: A kiadott teszt kérdései 1/4**



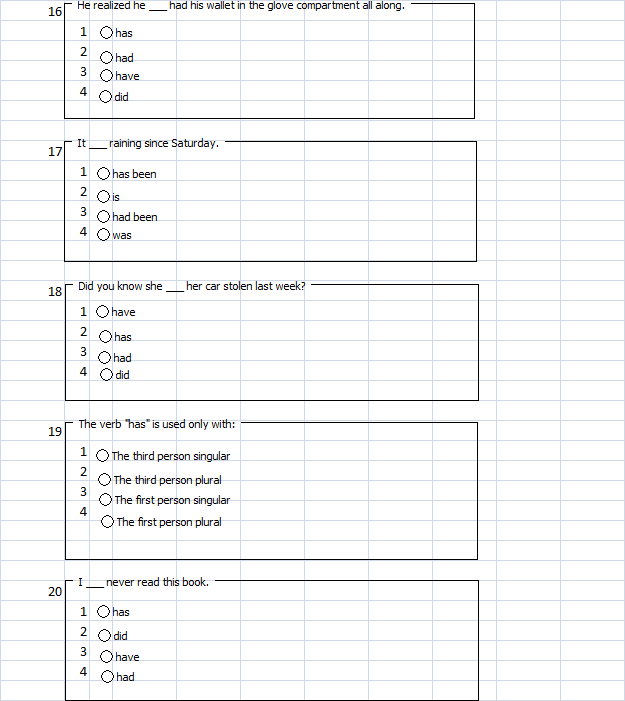
1. **Melléklet: A kiadott teszt kérdései 2/4**



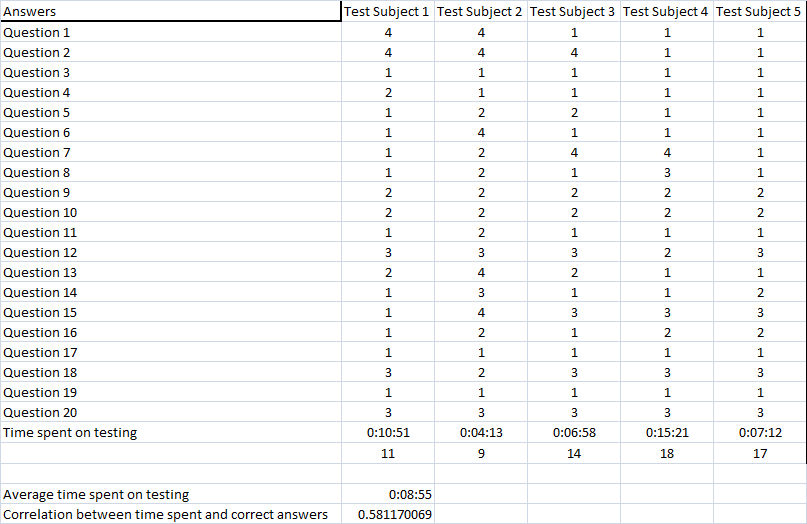
1. **Melléklet: A kiadott teszt kérdései 3/4**



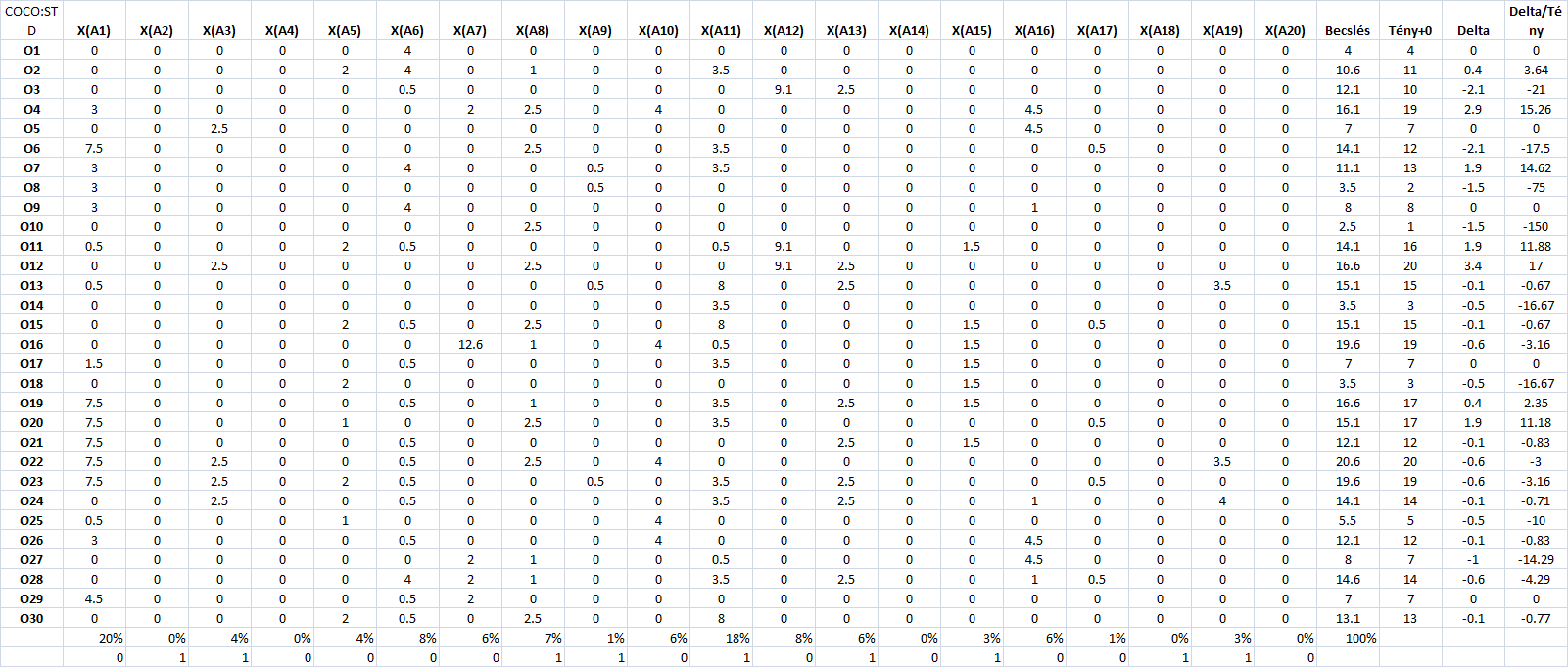
1. **Melléklet: A kiadott teszt kérdései 4/4**



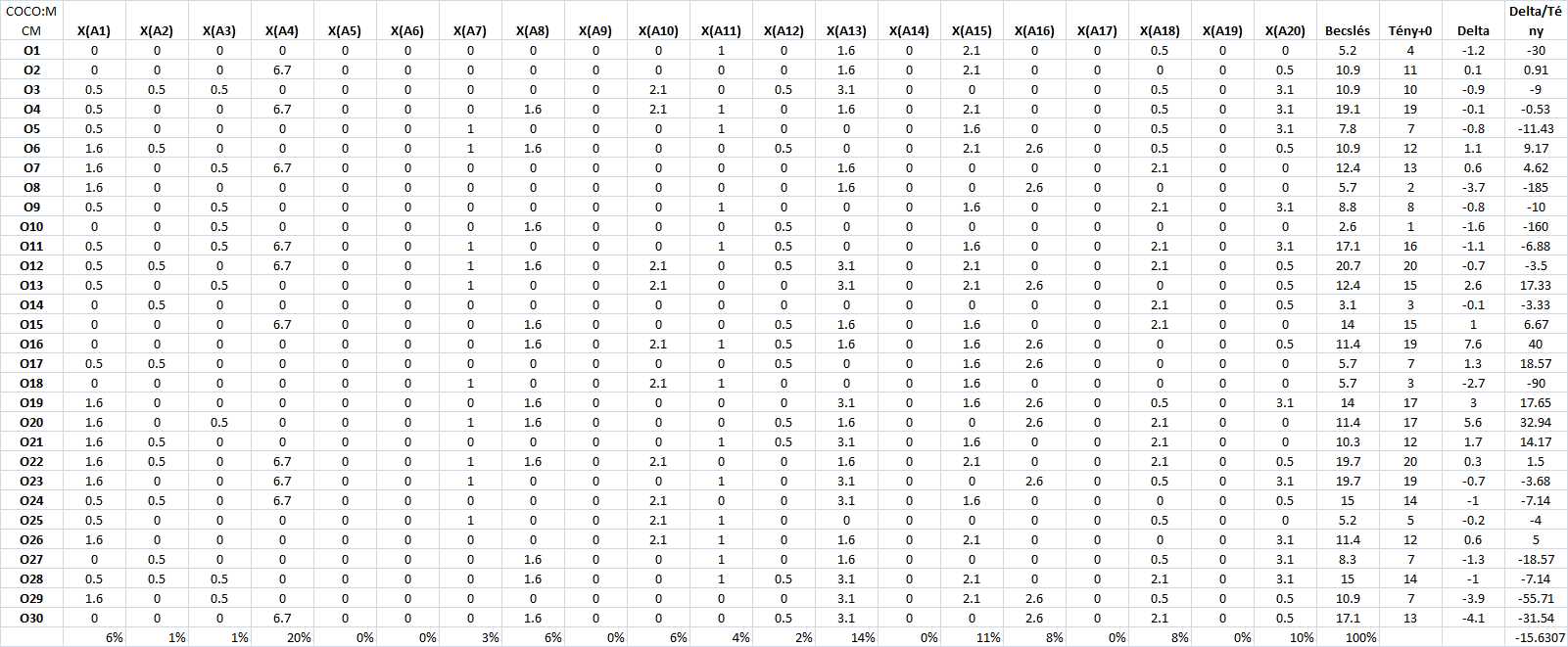
1. **Melléklet: A tesztek eredményeit tartalmazó táblázat**



1. **Melléklet: A COCO direkt futtatásának eredménytáblázata**



1. **Melléklet: A COCO-MCM futtatásának eredményei**

****

# Függelékek

**Konzultációkon való részvétel igazolása**

A hallgató neve: Varga Viktor

A belső konzulens neve és beosztása: Dr. Pitlik László, egyetemi docens, SZIE GTK TKI

A témát kiadó önálló oktatási szervezeti egység neve: TATA Kiválósági Központ és Informatikai Intézet

Nevezett hallgató a 2010/2011 tanévben a diplomadolgozatkészítésével kapcsolatos konzultációkon rendszeresen részt vett. Az elkészített dolgozatot

*Feleletválasztós tesztek feltöltésére, tárolására alkalmas online keretrendszer fejlesztése az oktatás elősegítése érdekében*

címmel bemutatta. A dolgozatnak a Záróvizsgához kapcsolódó bírálati eljárásra való beadásával egyetértek.

Gödöllő, 2011. év ........................... hó .................. nap

.......................................

konzulens aláírása

**Nyilatkozat**

Alulírott **Varga Viktor** a Szent István Egyetem Gazdaság- és Társadalomtudományi Kar Vezetés és Szervezés mesterszak levelező tagozatának végzős hallgatója nyilatkozom, hogy a

**Feleletválasztós tesztek feltöltésére, tárolására alkalmas online keretrendszer fejlesztése az oktatás elősegítése érdekében**

címmel védésre benyújtott diplomadolgozatsaját munkám eredménye, amelynek elkészítése során a felhasznált irodalmat a szerzői jogi szabályoknak megfelelően kezeltem.

Gödöllő, 2011. év …………… hó ……… nap

(a hallgató aláírása)

**Diplomadolgozat rövid bemutatása**

**Készítette:** Varga Viktor

**A diplomadolgozat címe:** Feleletválasztós tesztek feltöltésére, tárolására alkalmas online keretrendszer fejlesztése az oktatás elősegítése érdekében

**Belső konzulens neve, beosztása:** Dr. Pitlik László, egyetemi docens

**A témát kiadó önálló szervezeti egység neve:** TATA Kiválósági Központ és Informatikai Intézet

**Kulcskifejezések:** feleletválasztós tesztek, online, oktatás, e-learning, adatbányászat, hasonlóságelemzés

**A dolgozat rövid leírása:**Jelen dolgozatot az interneten már ma is nagy számban előforduló, nyelvoktató weboldalak ihlették, amelyeken feleletválasztós teszteket tölthetnek ki a tanulni vágyók. A dolgozat irodalmi áttekintésében megvizsgálja a feleletválasztós tesztek fontos ismérveit, egy, a jövőben elkészítendő online keretrendszerhez megalkotásához szükséges eszközök sajátosságait, az elkészítendő rendszer gazdasági jelentőségét a nyelvismerettel kapcsolatos statisztikákon keresztül, az e-learning rendszerek sajátosságait, amelyek közé a tesztrendszer is tartozik. A kutatás középpontjában a feleletválasztós tesztsorok írásakor elkövethető hibák statisztikai-adatbányászati alapú kiszűrésére tesz kísérletet a dolgozat szerzője, amelyet első és legfontosabb lépésként tekint a vázolt tesztrendszer elkészítése felé.

1. *Haladyna, T.M., and Downing, S.M. (1989) A taxonomy of multiplechoice*

   *item-writing rules, Applied Measurement in Education. pp. 37-50.* [↑](#footnote-ref-1)
2. FORGÓ Sándor: Az eLearning fogalma. In: HUTTER Otttó – MAGYAR Gábor - MLINARICS József: E-LEARNING 2005 (eLearning kézikönyv), Műszaki Könyvkiadó, 2005. 14. [↑](#footnote-ref-2)