

Primjena Raschovog modela u evaluaciji testa računalnog razmišljanja

Crmarić, Petra

Master's thesis / Diplomski rad

2024

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Split, Faculty of Science / Sveučilište u Splitu, Prirodoslovno-matematički fakultet**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:166:652119>

Rights / Prava: [In copyright](#)/[Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2024-11-27**

Repository / Repozitorij:

[Repository of Faculty of Science](#)



UNIVERSITY OF SPLIT



DIGITALNI AKADEMSKI ARHIVI I REPOZITORIJI

SVEUČILIŠTE U SPLITU
PRIRODOSLOVNO MATEMATIČKI FAKULTET

DIPLOMSKI RAD

**PRIMJENA RASCHOVOG MODELA U
EVALUACIJI TESTA RAČUNALNOG
RAZMIŠLJANJA**

Petra Crmarić

Split, rujan 2024.

*Veliko hvala mojoj obitelji, momku i prijateljima, bez vaše neizmjerne podrške
moji uspjesi ne bi bili mogući.*

*Posebno zahvaljujem mentorici dr. sc. Moniki Mladenović na stručnom vodstvu i
nesebičnoj pomoći tijekom cijelog procesa, kao i izv. prof. dr. sc. Ivici Boljatu koji
me usmjerio prema ovom području istraživanja.*

Temeljna dokumentacijska kartica

Diplomski rad

Sveučilište u Splitu
Prirodoslovno-matematički fakultet
Odjel za informatiku
Ruđera Boškovića 33, 21000 Split, Hrvatska

PRIMJENA RASCHOVOG MODELA U EVALUACIJI TESTA RAČUNALNOG RAZMIŠLJANJA

Petra Crmarić

SAŽETAK

U suvremenom obrazovanju, važnost računalnog razmišljanja raste, no potrebno je razviti pouzdane alate za njegovo mjerenje. Ovaj rad istražuje razvoj i validaciju testa za procjenu razine računalnog razmišljanja neovisno o programiranju. Cilj je bio izraditi instrument koji precizno mjeri kompetencije računalnog razmišljanja koristeći teoriju odgovora na zadatke (IRT) i Raschov model. Test je primijenjen većinom na srednjoškolcima i studentima, a analizom su utvrđena pitanja koja zahtijevaju reviziju radi poboljšanja unutarnje dosljednosti i preciznosti mjerenja. Raschov model omogućio je detaljan uvid u pouzdanost stavki i njihovu sposobnost diskriminacije različitih razina kompetencija. Istraživanje postavlja temelje za daljnju prilagodbu testa kako bi se postigla optimalna preciznost i valjanost.

Ključne riječi: pouzdanost mjerenja, testne stavke, teorija odgovora na zadatke, evaluacija

Rad je pohranjen u knjižnici Prirodoslovno-matematičkog fakulteta, Sveučilišta u Splitu

Rad sadrži: 61 stranice, 31 grafičkih prikaza, 2 tablice i 33 literaturnih navoda. Izvornik je na hrvatskom jeziku.

Mentor: **Dr. sc. Monika Mladenović**, *docent Prirodoslovno-matematičkog fakulteta, Sveučilišta u Splitu*

Ocjenjivači: **Dr. sc. Monika Mladenović**, *docent Prirodoslovno-matematičkog fakulteta, Sveučilišta u Splitu*

Dr. sc. Divna Krpan, *docent Prirodoslovno-matematičkog fakulteta, Sveučilišta u Splitu*

Dr. sc. Goran Zaharija, *docent Prirodoslovno-matematičkog fakulteta, Sveučilišta u Splitu*

Rad prihvaćen: **rujan 2024**

Basic documentation card

Thesis

University of Split
Faculty of Science
Department of Informatics
Ruđera Boškovića 33, 21000 Split, Croatia

IMPLEMENTATION OF THE RASCH MODEL FOR THE COMPUTATIONAL THINKING TEST EVALUATION

Petra Crmarić

ABSTRACT

In contemporary education, the significance of computational thinking is growing, yet there is a need for reliable tools to measure it. This study focuses on the development and validation of a test designed to assess the level of computational thinking independently of programming skills. The aim was to create an instrument that accurately measures computational thinking competencies using Item Response Theory (IRT) and the Rasch model. The test was administered primarily to high school students and university students and the analysis identified items requiring revision to enhance internal consistency and measurement precision. The Rasch model provided a detailed insight into item reliability and their ability to discriminate between different levels of competency. The study lays the groundwork for further test refinement to achieve optimal accuracy and validity.

Key words: measurement reliability, test items, item response theory, evaluation

Thesis deposited in library of Faculty of science, University of Split

Thesis consists of: 61 pages, 31 figures, 2 tables and 33 references

Original language: Croatian

Mentor: **Monika Mladenović, Ph.D.** *Assistant Professor of Faculty of Science, University of Split*

Reviewers: **Monika Mladenović, Ph.D.** *Assistant Professor of Faculty of Science, University of Split*

Divna Krpan, Ph.D. *Assistant Professor of Faculty of Science, University of Split*

Goran Zaharija, Ph.D. *Assistant Professor of Faculty of Science, University of Split*

Thesis accepted: **September 2024**

IZJAVA

kojom izjavljujem s punom materijalnom i moralnom odgovornošću da sam diplomski rad s naslovom PRIMJENA RASCHOVOG MODELA U EVALUACIJI TESTA RAČUNALNOG RAZMIŠLJANJA izradila samostalno pod voditeljstvom doc. dr. sc. Monike Mladenović. U radu sam primijenila metodologiju znanstvenoistraživačkog rada i koristila literaturu koja je navedena na kraju diplomskog rada. Tuđe spoznaje, stavove, zaključke, teorije i zakonitosti koje sam izravno ili parafrazirajući navela u diplomskom radu na uobičajen, standardan način citirala sam i povezala s fusnotama s korištenim bibliografskim jedinicama. Rad je pisan u duhu hrvatskog jezika.

Studentica

Petra Crmarić

Sadržaj

Uvod	1
1. Teorijska osnova.....	2
1.1. Računalno razmišljanje.....	2
1.1.1. Definiranje problema	3
1.1.2. Logičko zaključivanje.....	4
1.1.3. Dekonstrukcija	5
1.1.4. Apstrakcija.....	6
1.2. Teorija testova	9
1.2.1. Klasična teorija testiranja	10
1.2.2. Moderna teorija testiranja.....	12
1.2.3. Rasch model	20
1.2.4. Person Item Map.....	22
1.2.5. Partial Credit Model	23
2. Metodologija istraživanja	25
2.1. Vrsta istraživanja	25
2.2. Predmet i cilj istraživanja	25
2.3. Instrumenti.....	26
2.4. Sudionici.....	29
2.5. Postupak.....	30
3. Rezultati.....	31
3.1. Deskriptivna statistika	31
3.2. Korelacija rezultata testa i odabranih varijabli	34
3.3. Rasch Model	35
3.4. Item Characteristic Curve.....	37
3.5. Person Item Map.....	41

3.6.	Partial Credit Model	42
3.7.	Item Fit Statistic.....	44
3.8.	Item Information Curve	45
3.9.	Ograničenja istraživanja	46
4.	Zaključak	47
	Literatura	48
	Popis slika i tablica	51
	Prilozi.....	53
	Skraćenice.....	61

Uvod

Suvremeni život ne može se zamisliti bez digitalne tehnologije koja sve više utječe na način na koji radimo, učimo i povezujemo se s drugima. Kako bi društvo pratilo ove promjene, obrazovni sustavi suočeni su s potrebom za razvojem novih vještina koje omogućuju uspješno snalaženje u sve kompleksnijem tehnološkom okruženju. U ovom kontekstu, područje znanosti, tehnologije, inženjerstva i matematike (engl. Science, Technology, Engineering, and Mathematics, skraćeno STEM) postalo je ključno za oblikovanje budućih inovacija i napretka.

Posebno se ističe potreba za stručnjacima u programiranju, a ta se potreba iz godine u godinu intenzivno povećava. Ipak, unatoč rastućem naglasku na STEM obrazovanju, rezultati međunarodnih testova, poput PISA-e (engl. Programme for International Student Assessment, skraćeno PISA) često pokazuju da učenici u mnogim zemljama, uključujući i Hrvatsku, postižu ispodprosječne rezultate u ključnim područjima poput matematičke pismenosti i kritičkog razmišljanja. Ovi podaci ukazuju na nužnost unaprjeđenja obrazovnih pristupa kako bi učenici razvili vještine potrebne za uspjeh u digitalnom društvu.

Jedna od ključnih kompetencija u tom procesu je računalno razmišljanje, definirano kao sposobnost formuliranja problema i osmišljavanja rješenja na način koji omogućuje njihovu provedbu korištenjem računalnih sustava. Premda se računalno razmišljanje često povezuje s programiranjem, njegova primjena nadilazi područje računalnih znanosti i primjenjiva je u različitim disciplinama kao alat za analitičko rješavanje problema. Wing je ovu vještinu istaknula kao temeljnu analitičku sposobnost koju bi trebalo poučavati uz osnovne vještine poput čitanja, pisanja i aritmetike [1].

Uvođenje računalnog razmišljanja u obrazovni sustav zahtijeva i razvoj pouzdanih instrumenata za njegovu procjenu. Postojeći alati često ne ispunjavaju potrebne kriterije objektivnosti i preciznosti, što stvara potrebu za razvojem novih, valjanih instrumenata. Stoga je cilj ovog rada razviti i validirati test za procjenu računalnog razmišljanja primjenom Raschovog modela. Ovaj model omogućuje detaljnu analizu mjernih podataka, osiguravajući objektivnost i konzistentnost procjene. Razvijanjem ovog instrumenta, namjera je pružiti edukatorima i istraživačima učinkovit alat za pouzdanu procjenu jedne od ključnih vještina potrebnih za uspjeh u suvremenom, tehnološki orijentiranom društvu.

1. Teorijska osnova

1.1. Računalno razmišljanje

Računalno razmišljanje (engl. Computational thinking, skraćeno CT) podrazumijeva rješavanje problema, dizajniranje sustava te razumijevanje ljudskog ponašanja koje se temelji na konceptima koji su temelj računalne znanosti. Računalno razmišljanje predstavlja ključnu vještinu koju svatko treba usvojiti, bez obzira na struku. To nije samo alat za informatičare, već je način razmišljanja koji omogućuje rješavanje problema i dizajniranje sustava na temelju osnovnih koncepata računalne znanosti [1].

Razvoj računalne tehnologije i znanosti duboko je transformiralo naš svakodnevni život. Načini na koji obavljamo svakodnevne zadatke i interagiramo sa svijetom oko nas su potpuno izmijenjeni u odnosu na prošlo desetljeće. Počevši od jednostavnih računanja do kompleksnih analiza i otkrivanja novih znanja, računala su postala neizostavni dio naših života [2].

Računalno razmišljanje, koje se odnosi na rješavanje problema, je centralni aspekt računalne znanosti i programiranja koji zahtijeva apstraktno razmišljanje i omogućuje kreativno rješenje za dani problem. Štoviše, računalno razmišljanje je skup procesa rješavanja problema proizašlih iz računalne znanosti, ali primjenjivih u bilo kojoj domeni [3].

To nije samo puko programiranje; radi se o stvaranju apstrakcija i modularnih pristupa za rješavanje složenih problema. Ovaj način razmišljanja transformira način na koji sagledavamo i pristupamo problemima, omogućujući nam da ih razbijemo na manje dijelove te ih rješavamo sustavno i učinkovito. Također, računalno razmišljanje nije ograničeno na laboratorijske uvjete ili radne okoline. Ono postaje sveprisutno u svakodnevnom životu kada ljudi počnu koristiti koncepte iz računalne znanosti u svojim svakodnevnim aktivnostima.

Upravo zbog toga, računalno razmišljanje ima sveprisutan utjecaj u gotovo svim područjima. Primjerice, kad pokušavamo pronaći izgubljene stvari koristeći strategiju povratka na prethodne korake, to je primjer tzv. "backtrackinga" - pristupa kojeg koriste i računalni algoritmi.

Nadalje, ovaj pristup radikalno mijenja statistiku koristeći strojno učenje i automatizaciju Bayesian metoda, omogućujući prepoznavanje obrazaca i anomalija u velikim skupovima podataka, uključujući astronomske karte i medicinske snimke. U biologiji, računalno

razmišljanje ubrzava sekvenciranje ljudskog genoma i omogućuje modeliranje dinamičkih prirodnih procesa, poput ciklusa stanica i presavijanja proteina. U ekonomiji, doprinosi razvoju računalne mikroekonomije, koja optimizira razne aspekte poput oglašavanja i online aukcija. Računalno razmišljanje bit će ključno za analizu velikih količina podataka koje prikupljamo i generiramo. Primjenom računalnog razmišljanja moći ćemo otkriti i razumjeti saznanja koja su skrivena ili raspršena unutar tih podataka [2].

Seymour Papert prvi je detaljno istraživao ideju računalnog razmišljanja. Kao pionir u programiranju s programskim jezikom Logo i kao zagovornik učenja potpomognutog računalima, Papert je vjerovao da programiranje igra ključnu ulogu u razvoju proceduralnog razmišljanja kod djece. Smatrao je da će učenici, kroz stvaranje malih, simuliranih svjetova pomoću tehnologije i programiranja, postupno usvojiti vještine koje će moći primijeniti ne samo u programiranju, već i u drugim područjima života i izvan učionice.

Često se kaže da su stručnjaci u računalnim znanostima današnji majstori u rješavanju problema. Stoga moramo razumjeti različite vještine i tehnike rješavanja problema koju računalni znanstvenici koriste. Naravno, rješavanje problema nije započelo s računalima, niti su ona neophodna za rješavanje mnogih vrsta problema. Također, vještine koje stručnjaci za računalne znanosti koriste pri rješavanju problema često su korisne i u drugim područjima života izvan računalnog svijeta.

Nemoguće je obuhvatiti sve načine na koje računalni znanstvenici rješavaju probleme, ali postoje četiri strategije koje oni redovito koriste i koje su u srži ovog modernog stila rješavanja problema:

- Definiranje problema
- Logičko zaključivanje
- Dekonstrukcija
- Apsktrakcija

Svaka od ovih tehnika, osim što je izuzetno korisna za rješavanje problema u računalnim znanostima, ima i široku primjenu izvan tog područja, te ćemo ih detaljnije razmotriti [3].

1.1.1. Definiranje problema

Slično kao što znanstveni istraživački proces započinje s postavkom hipoteze i razvoj softvera započinje definiranjem problema. Definicija problema služi kao cilj softverskog razvojnog inženjera.

Prepoznamo najmanje tri glavne faze u procesu razvoja softvera:

- Analiza
- Dizajn
- Implementacija

Dizajn i implementacija uključuju stvaranje rješenja problema dok se analiza bavi definiranjem problema. Sama analiza je najvažnija faza jer su uspješan dizajn i implementacija mogući su samo uz odgovarajuću analizu [3].

Definiranje problema uključuje više osoba od samih računalnih znanstvenika. Svatko tko naručuje izradu softvera mora sudjelovati u analizi kako bi jasno prenio definirani problem. Dobro definirani softverski problem uglavnom uključuje popis zahtjeva. Kao što samo ime sugerira, zahtjevi opisuju ključne karakteristike softvera. Oni se zatim dalje razrađuju kako bi se osiguralo da softver ispunjava očekivane funkcionalnosti. Zahtjevi se nadalje dijele na dvije vrste:

- funkcionalni zahtjevi koji specificiraju određene zadatke koje softver mora izvršiti
- nefunkcionalni zahtjevi koji definiraju druge karakteristike i ograničenja vezana uz softver.

Dobar popis funkcionalnih zahtjeva mora biti jasan, dosljedan i potpun.

1.1.2. Logičko zaključivanje

Većina računalnog softvera oslanja se na logičke upute jer programi moraju donositi određene odluke tijekom izvršavanja. Na primjer, softver koji upravlja web-stranicom odlučuje kada naplatiti kupcu, program u digitalnom fotoaparatu određuje koliko dugo zatvarač treba ostati otvoren pri snimanju, dok glazbeni uređaj bira sljedeću pjesmu za reprodukciju. Sve ove odluke donose se putem računalnih uputa koje predstavljaju logičku strukturu programa. U većini programskih jezika za te se svrhe koristi naredba *if*.

If naredbe prisiljavaju nas da razmišljamo na logičan način, da promatramo stvari kao složenu zbirku uzročno-posljedičnih veza. Uzročno-posljedična veza sastoji se od logičkog uvjeta (uzrok) koji prisiljava program da izvrši neki zadatak (posljedica). Tablica sadrži primjere uzročno-posljedične veze koje se često koriste u računalnim programima. Velik dio rada u računalnom programiranju sastoji se od identificiranja takvih uzročno-posljedičnih veza i prevođenja istih u upute.

Uzrok	Posljedica
Korisničko ime i lozinka su uneseni.	Provjeriti jesu li korisničko ime i lozinka valjani.
Korisnik klikne gumb za zatvaranje prozora.	Ukloniti zatvoreni prozor s prikaza.
Napon baterije prijenosnog računala ispod 10%.	Prikazati skočni prozor s upozorenjem o niskoj razini baterije.
Nova e-mail poruka stiže s interneta.	Obavijestiti korisnika zvučnim signalom.

Tablica 1.1. Primjeri uzročno-posljedičnih veza

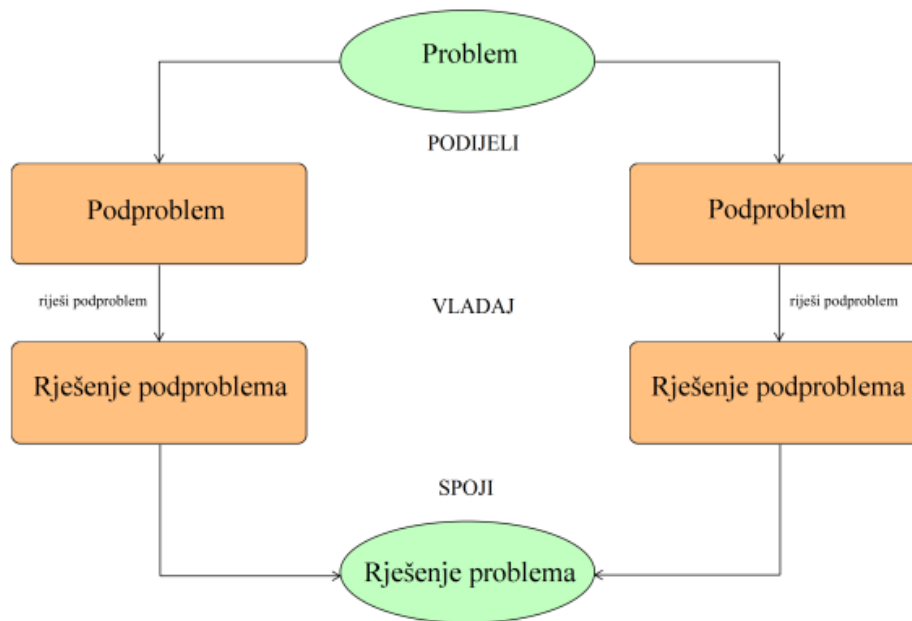
Sposobnost primjene općih pravila na određene situacije poznata je kao deduktivno zaključivanje. Postoji jasna razlika između deduktivnog razmišljanja i logičkog razmišljanja [6].

Dedukcija se odnosi na proces dolaska do zaključka na temelju određenih premisa. Logika je formalno područje koje pokušava karakterizirati specifične oblike argumentacije kao valjane ili nevaljane. Drugim riječima, logika pruža normativne okvire za ispravne dedukcije, norme koje se često eksplicitno koriste u akademskim kontekstima za ocjenjivanje adekvatnosti deduktivnog razmišljanja. Važnost logičkih normi za opisivanje argumenata je toliko intuitivno jasna da je izraz "nije logično" uobičajen način opisivanja osobe čiji se argumenti smatraju lošima [4].

1.1.3. Dekonstrukcija

Kada softverski inženjeri istražuju uzročno-posljedične veze, kao što je ranije spomenuto, koriste i jednu od najčešćih strategija za rješavanje problema – dekompoziciju. To podrazumijeva razlaganje problema na manje, pojedinačne dijelove. Ove tehnike dekompozicije često se nazivaju strategijama "podijeli i vladaj". Cilj je podijeliti složeni

problem na manje pod probleme, a zatim rješavati svaki od njih zasebno kako bi se postiglo rješenje cijelog problema.



Slika 1.1. Strategija "podijeli i vladaj"

Neka od razmišljanja iza mudrosti strategije "podijeli i vladaj" objašnjena su u radu psihologa George Miller-a. Bit Millerovog rada je da je ljudski mozak prilično ograničen u svojim sposobnostima razmišljanja i da je za neke vrste mišljenja ljudsko pamćenje ograničeno na 7 ± 2 stavke [5].

U praktičnom smislu, to znači da su mnogi problemi previše složeni za ljudski um, osim ako se ne pristupi u obliku jednostavnijih parcijalnih problema.

1.1.4. Apstrakcija

Iz svojih latinskih korijena, riječ apstrakcija bi u osnovi trebala značiti "nešto što je izvučeno ili odvojeno." Oxford English Dictionary opisuje apstrakciju kao "radnju razmatranja nečega u apstraktnom obliku, neovisno o njegovim asocijacijama ili atributima; proces izoliranja svojstava ili karakteristika zajedničkih za niz različitih objekata, događaja itd., bez pozivanja na specifična svojstva pojedinačnih primjera ili slučajeva."

White i Mitchelmore nadogradili su pojam apstrakcije kao konstruktivni proces s dvostrukim djelovanjem "prepoznavanja sličnosti i ignoriranja razlika."

Ovo podsjeća na Lockeovo gledište o apstrakciji kao dvosmjernom procesu, gdje se neka obilježja zadržavaju, dok se druga ignoriraju. Ova jednostavna karakterizacija odjekuje kod mnogih računalnih znanstvenika, na primjer:

- "Apstrakcija je eliminacija nebitnog i pojačavanje bitnog" [9]
- "Proces apstrakcije – odlučivanje koje detalje trebamo istaknuti, a koje možemo ignorirati" [4]
- "Apstrakcija se definira kao sposobnost određivanja koji su aspekti važni, a koji nisu" [10]

Vjerojatno se osnovno značenje apstrakcije odnosi se na proces prepoznavanja zajedničkih obilježja u različitim primjerima te stvaranje ili definiranje nove kategorije koja grupira sve objekte s takvim obilježjima – uključujući time i primjere [7].

Prema Piagetu, postoje četiri faze kognitivnog razvoja čovjeka: senzomotorna (0-2 g.), predoperacijska (2.-7. g), konkretnih operacija (7.-11.g) i formalnih operacija (>12 g) [7]. Odlazak u školu uobičajeno se događa u fazi konkretnih operacija gdje je dijete u stanju logički pristupiti rješavanju problema. Vrijeme ulaska u fazu formalnih operacija varira, dobar dio ljudi nikada ne dođe do ove faze koja je, zapravo, sposobnost apstraktnog mišljenja. Očito je da je ljudima prirodno razmišljati od konkretnog prema apstraktnom [8].

Aktivnosti apstrakcije namijenjene osnovnoškolskom/srednjoškolskom uzrastu često ciljaju ovu perspektivu: na primjer, istraživanjem sličnosti i traženjem algoritamskih (posebno iterativnih ili rekurzivnih) obrazaca. Na primjer, pri uvođenju ponavljanja u Logo ili Scratch. Nakon što nauče crtati linije i naredbe za naprijed i desno, učenici dobivaju zadatak crtanja kvadrata, što obično rezultira kodiranjem slijeda od četiri para komandi naprijed i desno. Sada, učenici se pozivaju da prepoznaju ponavljajući obrazac i da uoče korisnost primjene naredbe ponavljanja (4 puta). Na temelju toga, razina apstrakcije bi se kasnije mogla dodatno podići usmjeravajući učenike na crtanje različitih pravilnih poligona s različitim brojem i/ili veličinama stranica.

Računalno razmišljanje može se učinkovito poučavati kroz niz pažljivo osmišljenih aktivnosti koje ne zahtijevaju upotrebu računala. Ove aktivnosti, poznate kao *CS Unplugged*, dizajnirane su za učenje osnovnih koncepata računalne znanosti bez digitalnih uređaja i mogu se prilagoditi različitim dobnim skupinama učenika. Jedan od prvih primjera ovih aktivnosti usmjeren je na poučavanje apstrakcije, a namijenjen je učenicima osnovne škole.

U ovoj aktivnosti, učenici primaju imenice napisane na karticama i brzo crtaju skice koje drugi učenici trebaju pogoditi. Ovaj proces pomaže učenicima da se usmjere na bitne informacije i zanemare suvišne detalje, čime razvijaju vještine apstraktnog razmišljanja.

Računalno razmišljanje postaje temeljna vještina u 21. stoljeću, pružajući nam alate i koncepte potrebne za suočavanje s izazovima suvremenog društva. Integracija ovog pristupa u obrazovni sustav ključna je za pripremu budućih generacija za uspjeh u svijetu koji se brzo mijenja. Prema pedagogu Karlu Fischu, profesoru srednje škole u SAD i autoru motivirajućih prezentacija o promjenama koje 21 stoljeće donosi „Trenutno pripremamo studente za poslove koji još ne postoje, za koje će koristiti tehnologije koje još nisu izumljene da bi rješavali probleme za koje još niti ne znamo da su problemi.“

1.2. Teorija testova

Teorija testiranja je disciplina koja proučava metode za provođenje testiranja radi postizanja pouzdanih i valjanih rezultata. Test se može proučavati iz različitih perspektiva, a stavke u testu mogu se ocjenjivati prema različitim teorijama. Dvije takve teorije su klasična teorija testova (engl. Classical Test Theory, skraćeno CTT) i teorija odgovora na stavke (engl. Item Response Theory, skraćeno IRT).

Ove teorije predstavljaju dva glavna okvira koji se koriste za razvoj, evaluaciju i proučavanje testnih stavki. Ovi okviri temelje se na različitim pretpostavkama i koriste različite statističke pristupe. Ne bave se samo razvojem, evaluacijom ili određivanjem pouzdanosti i valjanosti testova, već i cjelokupnim poboljšanjem kvalitete testnih stavki [9].

CTT je dominirao poljem testiranja veći dio 20. stoljeća i bio je ključni okvir za razvoj i analizu testova. Temelji se na osnovnim konceptima pouzdanosti, valjanosti i standardnih grešaka mjerenja. Pouzdanost testa u CTT-u odnosi se na dosljednost rezultata pri ponovnom provođenju testa, uz koeficijente kao što su pouzdanost ponovljenog testiranja i unutarnja konzistentnost. Dok pouzdanost mjeri koliko su rezultati konzistentni, valjanost procjenjuje koliko test uspješno mjeri ono što je namijenjen da mjeri. CTT također nudi alate za analizu različitih aspekata testova, uključujući postotak točnih odgovora, prosječne ocjene i distribuciju rezultata.

S razvojem računalnih tehnologija i naprednih statističkih metoda, pojavila se teorija odgovora na stavke, koja pruža detaljniji i precizniji uvid u karakteristike testova i sposobnosti ispitanika. Dok CTT fokusira na ocjenu testa u cjelini, IRT analizira svaki odgovor na stavke pojedinačno, omogućujući preciznije procjene sposobnosti ispitanika i težine stavki. IRT modelira vjerojatnost da će ispitanik točno odgovoriti na stavku, uzimajući u obzir specifične karakteristike stavki, poput težine i diskriminacije, te sposobnost ispitanika. Ovaj pristup pruža bogatije informacije o tome kako svaka stavka doprinosi ukupnom mjerenju i kako različite razine sposobnosti utječu na odgovore.

Dodatno, IRT podržava adaptivno testiranje, gdje se težina stavki prilagođava sposobnostima ispitanika u stvarnom vremenu, što poboljšava preciznost mjerenja i optimizira trajanje testiranja. Ovaj moderni okvir stoga nadopunjuje i proširuje ulogu koju je CTT imao, pružajući napredne metode za razvoj, analizu i prilagodbu testova.

1.2.1. Klasična teorija testiranja

Klasična teorija testiranja nastala je početkom 20. stoljeća, kao rezultat tri izuzetna postignuća prethodnih 150 godina: prepoznavanje prisutnosti pogrešaka u mjerenjima, shvaćanje tih pogrešaka kao nasumičnih varijabli te koncept korelacije i način njenog indeksiranja. Godine 1904. Charles Spearman pokazao je kako ispraviti koeficijent korelacije uzrokovan pogreškom mjerenja i kako dobiti indeks pouzdanosti potreban za tu ispravku. Spearmanovo prikazivanje označilo je početak teorije klasičnih testova. Naknadno su okvir teorije klasičnih testova razrađivali i usavršavali Spearman, George Udny Yule, Truman Lee Kelley i drugi [8].

Nakon što ispitanik završi test, dobiveni rezultat naziva se bruto, opaženi ili sirovi rezultat. Termin "bruto" označava da ovaj rezultat uključuje nekoliko komponenti. Prema klasičnoj teoriji testiranja, bruto rezultat sastoji se od dva dijela: pravog rezultata i komponente pogreške.

Na primjer, ako kandidat postigne 16 od 20 na testu numeričkog razmišljanja, njegov opaženi rezultat je 16. Međutim, nijedna psiho metrijska procjena nije potpuno pouzdana jer pogreške uvijek utječu na rezultat, što znači da se opaženi rezultat razlikuje od pravog rezultata. Pravi rezultat odražava stvarnu razinu numeričkog razmišljanja kandidata, koja je u praksi nepoznata. Razlika između opaženog i pravog rezultata ovisi o razini pogreške povezanoj s procjenom; nepouzdanе procjene imaju veću razinu pogreške [7].

Pravi rezultat predstavlja stvarnu vrijednost karakteristike koju nastojimo izmjeriti. U idealnim uvjetima, bez pogrešaka u mjerenju, dobili bismo upravo pravi rezultat. Međutim, zbog neizbježnih pogrešaka u mjerenju, ono što zapravo dobivamo je bruto rezultat, koji uključuje i pogrešku mjerenja. Ako bismo isti test ponovili mnogo puta s istim ispitanicima, prosječan rezultat tih ponovljenih mjerenja za svakog ispitanika bio bi njihov pravi rezultat, budući da bi se slučajne pogreške međusobno poništile. Klasična teorija testiranja polazi od pretpostavke da se karakteristika koja se mjeri ostaje stabilna kroz vrijeme.

Indeks težine u klasičnoj teoriji testiranja predstavlja udio ispitanika koji točno odgovaraju na zadatak. Iako se ponekad naziva p-vrijednost, ovaj termin može biti zbunjujući jer se može pomiješati s p-vrijednošću u statističkom testiranju, koja ima drugo značenje. Za politomijske zadatke, težina zadatka je prosječni rezultat, što može značiti da paradoksalno teži zadatak može imati niži indeks težine u CTT-u.

Visoka diskriminacija znači da zadatak bolje razlikuje ispitanike s različitim razinama konstrukta. Dakle, visoka diskriminacija je poželjna jer omogućuje bolju diferencijaciju između ispitanika koji razumiju testirani materijal i onih koji ga ne razumiju, ili između onih s pozitivnim i negativnim stavovima.

Metoda ekstremnih grupa koristi se za izračunavanje indeksa diskriminacije u tri koraka. Prvi korak uključuje razdvajanje ispitanika u gornju i donju grupu prema njihovim ukupnim rezultatima na skali, obično birajući gornjih i donjih 27% rezultata. U drugom koraku analizira se udio ispitanika u svakoj grupi koji odgovara na određenu stavku ili njezinu kategoriju. U trećem koraku izračunava se indeks diskriminacije oduzimanjem udjela u donjoj grupi od udjela u gornjoj grupi.

Uspoređivanjem indeksa diskriminacije različitih stavki može se utvrditi koje stavke najbolje razlikuju grupe. Viši indeks diskriminacije ukazuje na bolju sposobnost stavke da razlikuje grupe. Prema Truman Kelleyevoj "27% uzorka" veličini grupe, vrijednosti od 0,4 i više smatraju se visokima, dok se vrijednosti ispod 0,2 smatraju niskima [9].

Najčešće korištena procjena unutarnje konzistentnosti je statistika 'Cronbachova Alpha', koja se kreće od 0 do 1. Unutarnja konzistentnost opisuje u kojoj mjeri sve stavke u testu mjere isti koncept ili konstrukt, te je stoga povezana s međusobnom povezanošću stavki unutar testa. Rezultati od 0,7 ili više općenito ukazuju na zadovoljavajuću razinu pouzdanosti. Više razine pouzdanosti ukazuju na niže razine pogreške, što znači veću podudarnost između pravog rezultata i opaženog rezultata. Međutim, niske razine pouzdanosti ukazuju na veće razine pogreške, što znači da se opaženi rezultat može značajno razlikovati od pravog rezultata, čineći tako rezultate nevaljanima [13].

Standardna pogreška srednje vrijednosti predstavlja mjerenje preciznosti procjene srednje vrijednosti uzorka. Veća veličina uzorka obično rezultira manjom standardnom pogreškom, što znači da je procjena srednje vrijednosti preciznija.

Standardna pogreška srednje vrijednosti može pružiti grubu procjenu intervala pouzdanosti u kojem će se s određenom vjerojatnošću nalaziti stvarna srednja vrijednost populacije. Da bi se izračunao 95%-tni interval pouzdanosti, trebamo još pomnožiti s 1,96. To je faktor koji dolazi iz normalne distribucije i označava da će u 95% slučajeva stvarna srednja vrijednost populacije biti unutar ovog intervala. Ovaj interval daje procjenu raspona unutar kojeg će se s visokom vjerojatnošću nalaziti stvarna srednja vrijednost populacije.

U klasičnoj teoriji testiranja, prosječan rezultat zadatka ovisi o razini vještine ili osobini populacije ispitanika. Ako bi test polagala druga populacija s višom razinom osobine, težine zadataka bi bile veće, što znači da bi lakši zadaci imali veće indekse težine.

U CTT-u vrijede aksiomi da je aritmetička sredina pogreške nula, odnosno da bi opaženi rezultat sudionika u slučaju beskonačnog broja primjena testa bio jednak pravom rezultatu. Varijanca pogreške nije povezana s varijancom pravih rezultata, kao ni s varijancom pogreške u paralelnim mjerenjima [18].

Dakle, osnovne pretpostavke od kojih polazi klasična teorija testova su: [8]

1. Predmet mjerenja je stabilan kroz vrijeme (inače pravi rezultat ne bi mogao biti isti).
2. Pogreške mjerenja su slučajne (i mijenjaju rezultat slučajno).
3. Bruto rezultat je suma pravog rezultata i komponente pogreške.

1.2.2. Moderna teorija testiranja

Moderna teorija testiranja predstavlja značajan napredak u području psihometrije i mjerenja u društvenim znanostima. Ovaj pristup razvio se kao odgovor na ograničenja klasične teorije testova i donosi niz sofisticiranih metoda za razvoj, analizu i interpretaciju testova u različitim domenama, uključujući obrazovanje, psihologiju, medicinu i društvene znanosti.

Ključni trenutak u razvoju moderne teorije testiranja bio je rad Frederica Lorda i Melvina Novicka, koji su 1968. godine objavili knjigu "Statistical Theories of Mental Test Scores". Ova knjiga postavila je temelje za razvoj Item Response Theory i drugih modernih pristupa testiranju. Moderna teorija testiranja karakterizirana je sljedećim obilježjima:

- Preciznije mjerenje: Moderne metode omogućuju točniju procjenu sposobnosti ispitanika i karakteristika test pitanja. Prema Hambletonu i Jonesu, "moderne metode teorije testiranja pružaju temelj za rješavanje mjernih problema koji su bili teški ili nemogući za adekvatno rješavanje u okviru klasične teorije testova".ⁱ
- Fleksibilnost: Nove metode mogu se prilagoditi različitim vrstama podataka i testnih formata, uključujući dihotomne, politomne i kontinuirane odgovore.
- Informativnost: Pružaju detaljnije informacije o testu i ispitanicima. Na primjer, IRT pruža informacije o tome koliko dobro svako pitanje mjeri sposobnost na različitim razinama [4].

- Efikasnost: Omogućuju razvoj kraćih testova koji zadržavaju ili čak povećavaju preciznost mjerenja. Ovo je posebno vidljivo u razvoju adaptivnih testova [5].
- Invarijantnost: Mnogi moderni pristupi, posebno IRT, omogućuju procjenu parametara pitanja neovisno o uzorku ispitanika, i obrnuto [6].

1.2.2.1 Item Response Theory

Riječ „teorija“ se ovdje koristi u smislu paradigme koja nastoji objasniti sve činjenice s kojima se može suočiti [19]. Ne možemo reći da je to teorija u tradicionalnom smislu jer ne objašnjava zašto osoba daje određeni odgovor na pojedinu stavku niti kako osoba odlučuje što odgovoriti. Umjesto toga, IRT je poput teorije statističke procjene.

IRT je psihometrijski okvir za analizu kategorijskih podataka iz testova i upitnika s ciljem mjerenja latentnih osobina [12]. Procjenjuje parametre za stavke i pojedince, s primjenama u obrazovnom testiranju, psihološkoj procjeni i drugim područjima. IRT modeli mogu biti unidimenzionalni ili multidimenzionalni, i uključuju varijante za dihotomne i polinomne podatke. Ovaj pristup nudi prednosti kao što su poboljšana točnost bodovanja i adaptivna administracija testa [15].

Dakle, IRT je statistički modeliran pristup koji se koristi za analizu rezultata testova koji se temelji na pretpostavci da vjerojatnost točnog odgovora na stavku ovisi o latentnoj osobini ispitanika i svojstvima stavke. IRT pruža detaljniji uvid u karakteristike stavki testa, kao što su težina, diskriminativnost i donja granica, te omogućuje procjenu sposobnosti ispitanika na temelju njihovih odgovora na testu.

U temelju IRT-a leže tri osnovna koncepta: latentna osobina, stavke testa i parametri modela. Latentna osobina predstavlja neopipljivu osobinu koja se mjeri testom, poput sposobnosti ili znanja. Stavke testa su pojedinačni zadaci ili pitanja koja se koriste za procjenu latentne osobine. Parametri modela su vrijednosti koje opisuju svojstva stavki i odnose se na njihovu težinu, diskriminativnost i donju granicu.

1.2.2.2 Latentna osobina

Latentna osobina predstavlja neopipljivu karakteristiku koju mjerimo pomoću testa. U teoriji odgovora na stavke često se pretpostavlja da ispitanik posjeduje ovu latentnu, nevidljivu osobinu (poznatu i kao sposobnost) koju nije moguće izravno proučavati. Svrha IRT-a je razviti modele koji povezuju ovu latentnu osobinu s vidljivim karakteristikama ispitanika, osobito s njihovim sposobnostima da točno odgovore na niz pitanja u testu [16].

Latentna varijabla može se konceptualizirati kao kontinuirana, kategorizirana ili kao kombinacija oboje. Kada je varijabla konceptualizirana kao kontinuirana, tada je uporaba CTT-a ili IRT-a prikladna matematička tehnika. Međutim, ako je varijabla konceptualizirana kao kategorizirana, tada bi analiza latentnih klasa mogla biti najprikladnija psihometrijska metoda za korištenje [16].

Svaki zadatak na testu ocijenjen je prema težini i diskriminativnosti. Viša težina i diskriminativnost sugeriraju da je zadatak teži i bolje razlikuje među ispitanicima.

Kada ispitanici odgovore na test, njihovi se odgovori koriste za procjenu njihovih latentnih osobina pomoću logističke funkcije. Logistička funkcija, u kontekstu modeliranja rezultata testova, omogućuje nam da predvidimo vjerojatnost da će ispitanik dati određeni odgovor na testni zadatak na temelju svoje latentne osobine. Neka θ predstavlja latentnu osobinu ispitanika. Svaka stavka testa i ima pridruženu karakteristiku težine b_i . Da bismo izračunali vjerojatnost točnog odgovora na stavku i za ispitanika s latentnom osobinom θ , koristimo logističku funkciju gdje je a_i parametar diskriminativnosti stavke i , a b_i je parametar težine stavke i .

$$P_i(\theta) = \frac{1}{1 + e^{-a_i(\theta - b_i)}}$$

1.2.2.3 Stavke Testa

Stavka testa odnosi se na pojedinačno pitanje ili zadatak unutar testa ili upitnika. Svaka stavka dizajnirana je za mjerenje određenog aspekta latentne osobine koju test procjenjuje. Stavke testa su ključne komponente jer zajedno pružaju podatke koji se koriste za procjenu latentne osobine ispitanika. Kvaliteta i karakteristike svake stavke, kao što su težina i diskriminativnost, utječu na to koliko dobro test može razlikovati ispitanike s različitim razinama latentne osobine.

1.2.2.4 Parametri Modela

Parametri modela određuju karakteristike stavki testa. Za jednoparametarski model (engl. One Parameter Logistic Model, skraćeno 1PL), parametri se procjenjuju samo za težinu stavki. Dvoparametarski model (engl. Two Parameter Logistic Model, skraćeno 2PL), dodaje parametar diskriminativnosti. Dok troparametarski model (engl. Three Parameter Logistic Model, skraćeno 3PL), dodaje i parametar donje granice vjerojatnosti točnog odgovora.

Težina stavke, diskriminativnost stavke i donja granica su ključni parametri modela IRT-a. Težina stavke određuje poziciju na latentnom kontinuumu potrebnu za postizanje određene vjerojatnosti točnog odgovora. Diskriminativnost stavke mjeri koliko dobro stavka razlikuje između ispitanika s različitim sposobnostima, dok donja granica predstavlja vjerojatnost odgovora na stavku kod ispitanika s ekstremno niskom sposobnošću.

1.2.2.5 Krivulje karakteristike

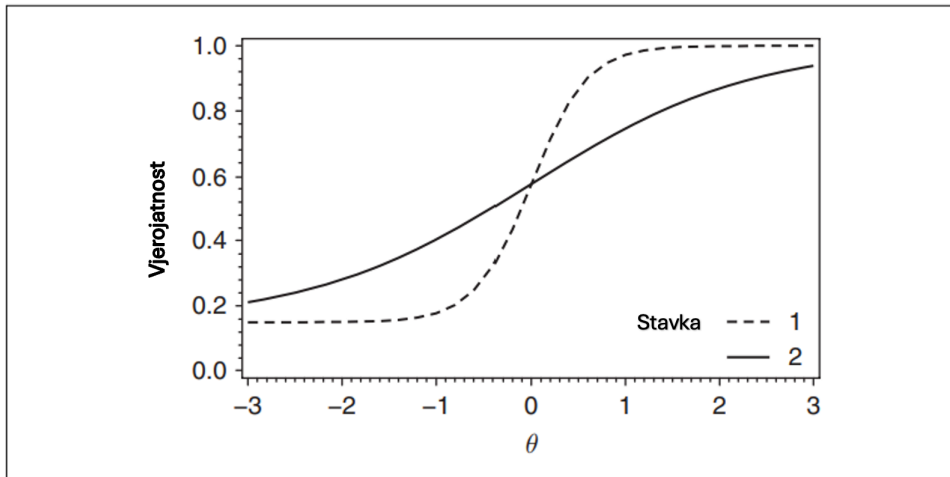
Krivulja karakteristika stavke (engl. Item Characteristic Curve, skraćeno ICC) je grafički prikaz koji ilustrira odnos između vjerojatnosti točnog odgovora na stavku i latentne osobine ispitanika. To je esencijalni alat u teoriji odgovora na stavku koji omogućuje razumijevanje ponašanja pojedinačnih stavki testa. ICC se konstruira tako da se na x-osi označi latentna osobina ispitanika, dok se na y-osi nalazi vjerojatnost točnog odgovora na stavku. Svaka stavka u testu ima svoju ICC krivulju koja opisuje kako se vjerojatnost točnog odgovora mijenja s promjenom latentne osobine.

ICC krivulje prikazuju odnos između vjerojatnosti odgovora na točno određeno pitanje i latentne osobine ispitanika koja se mjeri. Na horizontalnoj osi prikazuje se latentna osobina, dok se na vertikalnoj osi nalazi vjerojatnost točnog odgovora na pitanje.

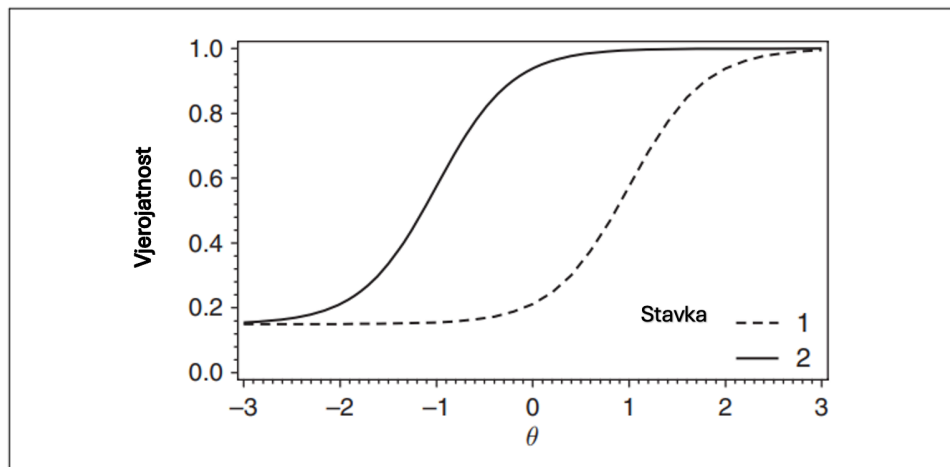
ICC krivulje koriste se za procjenu kako se stavke ponašaju u odnosu na latentnu osobinu. Stavke s višim poteškoćama imaju niže vjerojatnosti točnih odgovora za ispitanike s nižom latentnom osobinom, dok stavke s nižim poteškoćama imaju više vjerojatnosti točnih odgovora za istu grupu ispitanika. Ove krivulje omogućuju uvid u diferencijalnu funkciju stavki, pomažući identificirati stavke koje su relevantne za mjerenje latentne osobine.

Ključni koncepti koji se mogu izvući iz ICC uključuju:

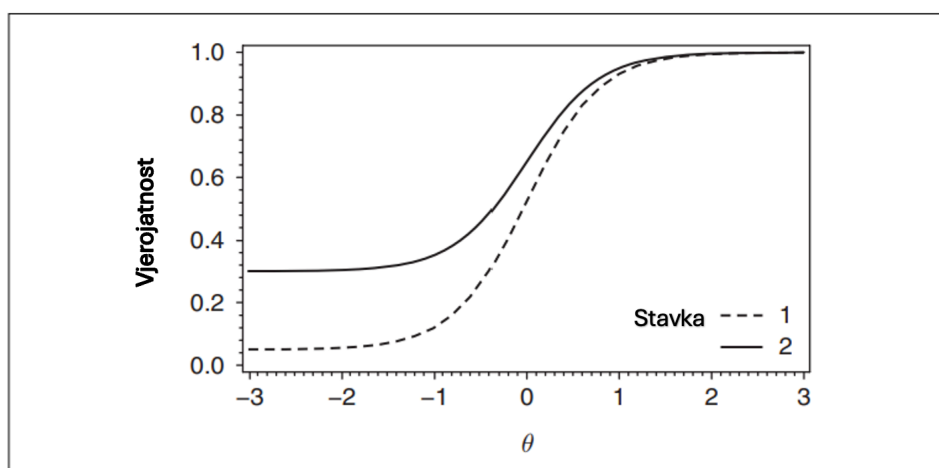
1. Različite krivulje ICC mogu ukazivati na različitu diskriminativnost stavki. Stavke s većom diskriminativnošću imaju krivulje koje se brže uzdižu prema maksimalnoj vjerojatnosti točnog odgovora.
2. Položaj i nagib krivulje ICC određeni su težinom stavke. Veće težine stavke rezultiraju krivuljama s većom nagibom, što znači da je za postizanje visokih vjerojatnosti točnog odgovora potrebno više latentne osobine.
3. Početna točka krivulje ICC na y-osi predstavlja donju granicu vjerojatnosti točnog odgovora. Ova točka odražava minimalnu vjerojatnost točnog odgovora na stavku kod ispitanika s ekstremno niskim sposobnostima [15].



Slika 1.2. Stavke s različitim a parametrom. Stavka 1 je više diskriminantna.



Slika 1.3. Stavke s različitim b parametrom. Stavka 1 je teža.



Slika 1.4. Stavke s različitim c parametrom. Stavka 2 ima veći c parametar.

Osobe na donjem kraju latentnog kontinuuma mogu se očekivati da imaju visoku vjerojatnost davanja odgovora 0. Na primjer, ispitanici s niskom razinom matematičke sposobnosti mogu se očekivati da netočno odgovore na pitanje iz topologije na ispitu iz matematike. Ako taj ispit koristi format višestrukog izbora, neki od ovih ispitanika s niskom sposobnošću mogu odabrati točan odgovor jednostavno pogađanjem. U takvim slučajevima, funkcija odgovora na zadatak ima donju asimptotu koja možda nije asimptotska s 0 već s nekom vrijednošću različitom od nule. 3PL model uzima u obzir ovu donju asimptotu koja nije jednaka nuli [18]. Matematička formula za 3PL model je:

$$P(\theta) = c_i + (1 - c_i) \frac{e^{1.7a_i(\theta - b_i)}}{1 + e^{1.7a_i(\theta - b_i)}} \quad (1)$$

Primjena eksponencijalne funkcije u logističkoj funkciji ključna je za dobivanje sigmoidne krivulje koja modelira vjerojatnosti. Koeficijent 1.7 koristi se kako bi se logistička funkcija skalirala na način koji je u skladu s kumulativnom normalnom funkcijom (koja prati normalnu distribuciju). To je u povijesti bio preferirani model za teoriju odgovora na stavke prije nego su logističke funkcije postale popularne.

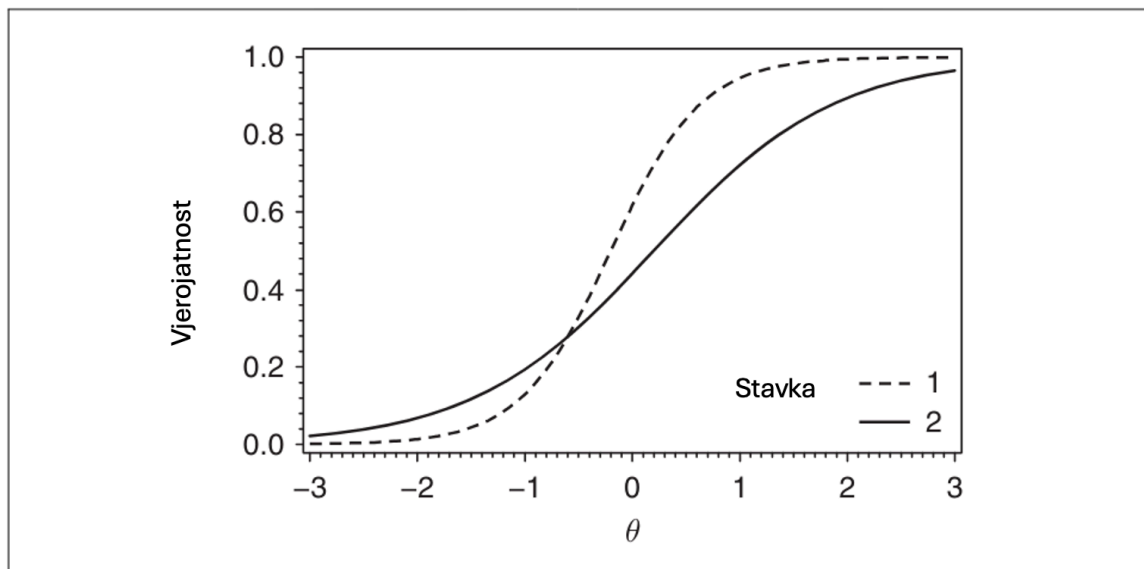
Kumulativna normalna funkcija zahtijeva matematičku integraciju za izračunavanje vjerojatnosti točnog odgovora, što može biti složenije. Logističke funkcije su jednostavnije za rad jer ne zahtijevaju integraciju; vjerojatnosti se mogu izračunati izravno.

Dvoparametarski i jednoparametarski IRT modeli su posebni slučajevi, odnosno ograničene verzije, 3PL modela. Ograničiti model znači fiksirati vrijednost jednog ili više parametara.

Stavke mogu varirati u svojoj sposobnosti da diskriminiraju ispitanike smještene na različitim točkama duž kontinuuma varijable. Stoga, korištenje IRT modela koji hvata informacije o diskriminaciji može biti korisno za procjenu položaja osobe. Dvoparametarski model je jedan od takvih modela jer omogućava da se stavke razlikuju ne samo u svojim položajima, već i u svojoj sposobnosti da razlikuju osobe smještene na različitim točkama kontinuuma. Najočitija manifestacija stavki s različitim parametrima diskriminacije je da se njihove odgovarajuće krivulje presijecaju negdje na kontinuumu. Iako parametar diskriminacije teoretski može varirati od $-\infty$ do ∞ , stavke s pozitivnim vrijednostima između približno 0,8 i 2,5 smatraju se dobrim vrijednostima.

Za dvoparametarski logistički model, vrijednost donje asimptote je fiksirana na nulu pa tako model postaje:

$$P(\theta) = \frac{e^{1.7a_i(\theta-b_i)}}{1 + e^{1.7a_i(\theta-b_i)}} \quad (2)$$

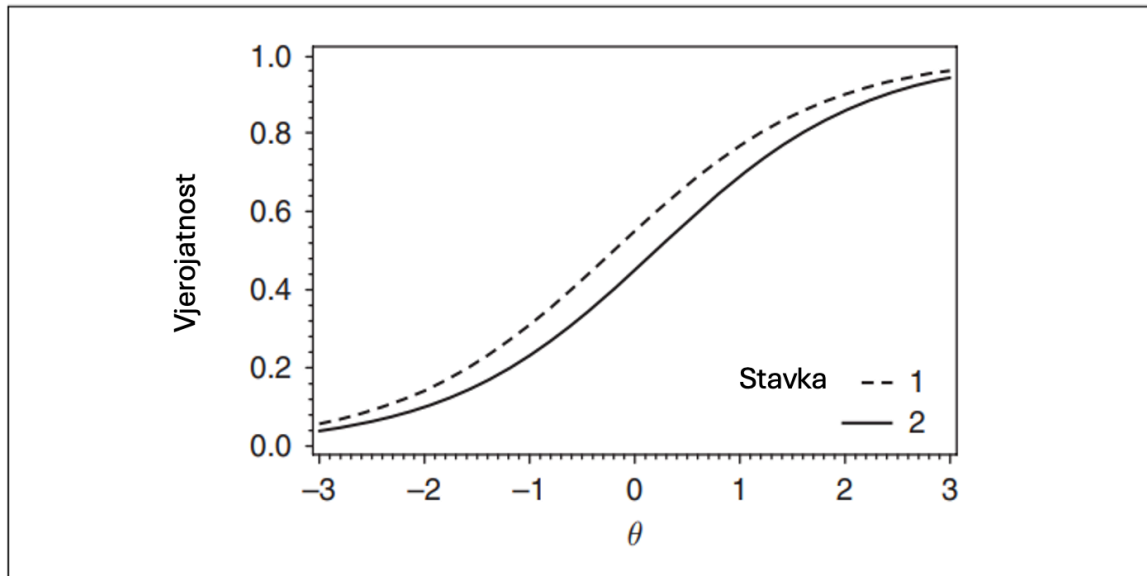


Slika 1.5. 2PL model sa različitim a i b parametrima

Za jednoparametarski logistički model, vrijednost c je fiksirana na nulu i a je fiksiran na istu vrijednost za sve stavke. Tako model postaje:

$$P(\theta) = \frac{e^{1.7a(\theta-b_i)}}{1 + e^{1.7a(\theta-b_i)}} \quad (3)$$

Primijetimo da više nema indeksa na a-parametru, jer je isti za sve stavke.



Slika 1.6. 1PL model, stavke sa jednakim a parametrom i različitim b parametrom

Nadalje, spomenimo još neke grafičke prikaze. Krivulja informativnosti stavke (engl. Item Information Curve, skraćeno IIC) je grafički prikaz koji pokazuje koliko informacija određena stavka pruža o sposobnosti ispitanika na različitim razinama latentne osobine (npr. sposobnosti, znanja, stavova) unutar IRT.

Ključne karakteristike Item Information Curve:

1. Oblik krivulje – IIC je obično zvonolikog oblika. To znači da stavka pruža najviše informacija na određenom rasponu sposobnosti (na vrhu krivulje), dok izvan tog raspona (prema ekstremno visokim ili niskim sposobnostima), stavka pruža manje informacija.
 - Vrhunac krivulje: Na ovoj točki stavka najbolje razlikuje između ispitanika različitih sposobnosti. Na primjer, ako je pitanje srednje težine, najbolje će procijeniti osobe sa srednjom razinom sposobnosti.
 - Rubovi krivulje: Pri vrlo niskim ili vrlo visokim sposobnostima, stavka pruža malo informacija jer svi ispitanici u tim područjima uglavnom odgovaraju na isti način (npr. svi točno ili svi netočno).
2. Širina krivulje – Širina IIC-a pokazuje raspon sposobnosti na kojem stavka pruža korisne informacije. Šira krivulja znači da stavka pruža informacije o širem rasponu

sposobnosti, dok uža krivulja znači da je stavka preciznija samo u određenom užem rasponu.

S druge strane, krivulja informativnosti testa (engl. Test Information Curve, skraćeno TIC) prikazuje ukupnu količinu informacija koju cijeli test pruža o sposobnostima ispitanika. Ova krivulja se sastavlja kao zbroj informacija koje pružaju svi zadaci u testu i pokazuje koliko je test uspješan u procjeni sposobnosti ispitanika na različitim razinama. Slično kao i kod Item Information Curve, Test Information Curve omogućuje identifikaciju raspona sposobnosti za koji test pruža najviše informacija i pomaže u optimizaciji dizajna testa.

1.2.3. Rasch model

Georg Rasch bio je danski matematičar i statističar, rođen 1901. godine i preminuo 1980. godine. Poznat je po svom značajnom doprinosu psihometriji i statistici. Rasch je posebno poznat po razvoju Rasch modela, koji je ključan za analizu podataka u psihometriji.

Raschov model je temeljni model mjerenja u teoriji odgovora na stavke koji omogućuje univerzalnost mjerenja i linearno skaliranje [20]. On pretvara podatke s ordinalne razine sirovih rezultata u intervalne mjere, izražavajući težinu stavki i sposobnost osobe u logitima, koji su jedinični intervali. Model pretpostavlja unidimenzionalnost i nepromjenjivu težinu stavki među svim osobama. Široko se koristi u različitim područjima, uključujući medicinske pretrage vida, procjene kvalitete života i u obrazovanju [17].

Matematički gledano, 1PL model i Raschov model su ekvivalentni. Vrijednosti iz jednog modela mogu se transformirati u drugi primjenom odgovarajućeg skaliranja [20]. Kako je Raschov model razvijen neovisno o 1PL modelu i nije samo specifičan slučaj dvoparametarskog modela. Mnogi korisnici Raschovog modela radije izbjegavaju upotrebu oznake IRT za ovaj model.

Raschov model je prvobitno specificiran u odnosu na omjer izgleda $vjerojatnost/(1-vjerojatnost)$ ili logaritam omjera izgleda (prirodni logaritam omjera izgleda, također nazvan logiti), ali se sada često specificira u odnosu na vjerojatnost. Kada je specificiran na ovaj način, model je isti kao jednadžba za 1PL, osim što se općenito izostavlja konstanta 1.7, a parametar a se izbacuje iz modela, efektivno ga fiksirajući na 1. Jedina posljedica toga je promjena veličine mjernih jedinica. Notacijski sustav je također obično drugačiji, koristeći d umjesto b i b umjesto y [16].

Očito, to nema nikakav utjecaj na matematičku funkciju, ali trebamo biti svjesni promjene simbola. Tipičan način specificiranja Raschovog modela stoga bi bio:

$$P(\beta) = \frac{e^{(\beta-\delta_i)}}{1 + e^{(\beta-\delta_i)}} \quad (4)$$

Izraženo riječima, formula (4) kaže da je vjerojatnost odgovora 1 na stavku i funkcija udaljenosti između osobe smještene na β i stavke smještene na δ . (Tehnički, govorimo o vjerojatnosti nasumično odabrane osobe iz onih smještenih na β .) Desna strana preslikava (potencijalno beskonačnu) udaljenost između položaja osobe i položaja stavke na skalu vjerojatnosti $[0, 1]$. Odgovor 1 jednostavno označava da se događaj dogodio ili da smo zabilježili uspjeh. Teoretski raspon položaja stavki δ , kao i položaja osoba β , kreće se od $-\infty$ do ∞ . Međutim, tipični položaji stavki i osoba obično padaju unutar raspona od -3 do 3 . U testiranju sposobnosti položaji stavki nazivaju se težinama stavki. Općenito, stavke smještene nešto ispod $0,0$ nazivaju se lakim stavkama (npr. ispod $-2,0$), dok se stavke nešto iznad $0,0$ nazivaju teškim stavkama (npr. iznad $2,0$). Općenito, stavke koje se smatraju lakima su one koje osobe s niskim sposobnostima imaju tendenciju točno odgovoriti. S druge strane, teže stavke su one koje osobe s visokim sposobnostima obično točno riješe. Stavke oko $0,0$ smatraju se stavkama prosječne težine [16].

Kao primjer korištenja Raschovog modela za predviđanje ponašanja pri odgovaranju, pretpostavimo da imamo stavku koji se nalazi na razini 1 (tj. $\delta = 1$) i damo je pojedincima smještenima na razini 0 (tj. $\beta = 0$). Prema formuli (4)

$$p = \frac{e^{(0-1)}}{1 + e^{(0-1)}} = 0.2689$$

vjerojatnost točnog odgovora za nasumično odabranu osobu iz ove skupine iznosila bi 0.2689. To znači da je vjerojatnost da će nasumično odabrana osoba smještena na razini 0 točno odgovoriti na ovaj zadatak samo 0.2689. Ova vrijednost vjerojatnosti ne bi trebala biti iznenađujuća, s obzirom na to da je zadatak smješten iznad (ili desno od) položaja osobe, što znači da zadatak zahtijeva veću matematičku sposobnost nego što osoba posjeduje. Ustvari, veće su šanse da će osoba smještena na razini 0 netočno odgovoriti na ovaj zadatak, jer je vjerojatnost netočnog odgovora $1 - 0.2689 = 0.7311$. Još jedan način tumačenja naših vjerojatnosti je njihovo pretvaranje u omjer za točan odgovor na zadatak. Na primjer, pretvaranjem ovih vjerojatnosti u omjer, dolazimo do zaključka da su omjeri za odgovor 1

otprilike 1 prema 2.7, odnosno, gotovo tri puta je vjerojatnije da će osoba netočno odgovoriti na zadatak nego što će točno odgovoriti [20].

Raschov model ima nekoliko svojstava koja ga čine praktičnim u analizi. Posebno, posjeduje svojstvo dovoljne statistike, što znači da je broj točnih odgovora na testu dovoljan za procjenu latentne osobine ispitanika. To implicira da ispitanici s istim brojem točnih odgovora imaju istu procjenu latentne osobine, bez obzira na to koje su specifične stavke točno odgovorili. Ovo svojstvo pojednostavljuje procjenu latentnih osobina i čini rezultate testa transparentnima.

Važno je napomenuti da, iako je broj točnih odgovora dovoljan za procjenu latentne osobine, transformacija između broja točnih odgovora i latentne osobine nije linearna. Razlike u broju točnih odgovora ne uvijek odražavaju iste promjene u latentnoj osobini. Na primjer, razlika između 10 i 11 točnih odgovora ne mora biti jednaka razlici između 18 i 19 točnih odgovora na temelju latentne osobine.

Kroz ova svojstva, Raschov model pruža jednostavan i transparentan način procjene latentnih osobina ispitanika na temelju njihovih rezultata na testu, olakšavajući interpretaciju rezultata i donošenje relevantnih odluka.

Krivulja koju generira model iz formule (4) poznata je kao karakteristična krivulja zadatka, krivulja zadatka ili funkcija odgovora na zadatak (engl. Item Response Function, skraćeno IRF) [30]. U Raschovom modelu položaj zadatka, označen s δ , definira se kao točka infleksije, tj. srednja točka IRF-a, gdje se mijenja smjer nagiba funkcije. S obzirom na to da IRF u Raschovom modelu ima donju granicu od 0 i gornju granicu od 1, srednja točka se nalazi na vrijednosti 0,50. To znači da položaj zadatka u Raschovom modelu odgovara točki na skali gdje je vjerojatnost odgovora 1 jednaka 50%. Ove krivulje pružaju jasan vizualni uvid u povezanost latentnih osobina ispitanika i njihovih rezultata na testu, omogućujući dublje razumijevanje kako se performanse mijenjaju ovisno o različitim razinama latentne osobine.

1.2.4. Person Item Map

Person Item Map je grafički prikaz koji se koristi u okviru IRT-a kako bi se vizualizirala povezanost između sposobnosti ispitanika i težine stavki u testu. Poznata još kao i Wright Map, ovaj graf sastoji se od dva dijela:

- Gornji dio grafa ili histogram:
Ovaj dio prikazuje raspodjelu parametara osoba (Person Parameter Distribution). Riječ je o histogramu koji prikazuje kako su ispitanici raspoređeni duž latentne dimenzije, tj. njihove računalne sposobnosti.
- Donji dio grafa ili scatter plot:
Ovaj dio prikazuje položaj stavki u odnosu na latentnu dimenziju. Svaka točka predstavlja jedan zadatak ili pitanje, a njihova pozicija na horizontalnoj osi prikazuje težinu stavke (odnosno koliko je teško to pitanje u latentnoj dimenziji).

Zajedno, ovi grafovi služe za usporedbu distribucije ispitanika i težine stavki u testu, što je često korišteno u IRT analizi. Ukratko, Person Item Map omogućuje nam vizualizaciju odnosa između stavki i osoba na latentnoj dimenziji, što nam pomaže u razumijevanju strukture podataka i interpretaciji rezultata. Stavke su raspoređene duž horizontalne osi prema njihovoj težini. Stavke koje su lakše za rješavanje nalaze se bliže lijevom rubu, dok su one teže za rješavanje smještene bliže desnom rubu.

1.2.5. Partial Credit Model

Rasch model je donekle umjetan s obzirom na to da smo dihotomizirali odgovore. U praksi vjerojatno ne bismo nametnuli takvu transformaciju našim podacima i umjesto toga bismo analizirali podatke iz testa kao politomijske odgovore od samog početka procjene skale. Pokazati ćemo kako se politomijski podaci mogu modelirati i istraživati u eRm paketu.

Mnoge funkcije i mogućnosti jednostavno se proširuju s Rasch modela na politomijske modele (model parcijalnih kredita i model ocjenske skale). Fitting modela parcijalnih kredita provodi se istim metodama koje se koriste u Rasch modelu. Jedina razlika je što sada koristimo model djelomičnog kredita (engl. Partial Credit Model, skraćeno PCM).

Na taj način dobivamo vrijednosti procjena parametara težine za svaku kategoriju unutar svakog pitanja, standardnu pogrešku koja mjeri nesigurnosti za procjenu težine. Stavke s pozitivnim parametrima težine su teže (teži odgovori za postizanje). Dakle, parametri težine za svaku kategoriju pokazuju koliko je teško dobiti određeni odgovor na to pitanje. Osim toga imamo i vrijednosti za intervale pouzdanosti gdje širi intervale pouzdanosti ukazuju na veću nesigurnost u procjeni parametara, dok užiji intervale ukazuju na veću preciznost.

Item fit statistika u modelu djelomičnog kredita je način procjene koliko dobro određena stavka odgovara modelu, odnosno koliko su stvarni odgovori ispitanika u skladu s

predviđanjima modela. PCM je ekstenzija Raschovog modela koja se koristi kada stavke imaju više kategorija odgovora.

Postoji nekoliko ključnih statistika koje se koriste za procjenu item fit-a:

1. Infit (Information-weighted fit)

- Infit mjeri koliko su stvarni odgovori u skladu s predviđanjima modela, uz veću težinu odgovora koji su bliže sposobnosti ispitanika.
- Dobro prilagođena stavka imat će infit vrijednost blizu 1. Vrijednosti iznad 1 ukazuju na to da postoji više odstupanja nego što model predviđa (previše neočekivanih odgovora), dok vrijednosti ispod 1 ukazuju na previše dosljednosti u odgovorima (previše uredni podaci).

2. Outfit (Outlier-sensitive fit)

- Outfit je sličan infitu, ali je osjetljiviji na odgovore koji su daleko od sposobnosti ispitanika (tj. odgovore koji su u nepredviđenim ekstremima).
- Kao i infit, idealna outfit vrijednost je 1. Vrijednosti daleko od 1 ukazuju na to da stavka možda ne odgovara dobro predviđanjima modela.

3. Chi-square fit statistika

- PCM može koristiti chi-square test za provjeru fit-a stavki. Chi-square statistika uspoređuje stvarne odgovore s onima koje bi model predvidio i daje uvid u to koliko su odstupanja značajna.

2. Metodologija istraživanja

2.1. Vrsta istraživanja

U ovom istraživanju, primijenila sam kvantitativni pristup zbog potrebe za preciznim i objektivnim mjerenjem razine računalnog razmišljanja (engl. Computational Thinking – skraćeno CT).

Korištenjem kvantitativnih metoda, omogućila sam detaljnu analizu učinkovitosti testa u procjeni različitih aspekata CT-a, bez oslanjanja na subjektivne procjene ili interpretacije.

Kvantitativni pristup bio je neophodan za pružanje objektivnih i mjerljivih rezultata, što je omogućilo preciznu analizu i identifikaciju područja koja zahtijevaju poboljšanja. Ovaj pristup pruža čvrste temelje za daljnja istraživanja i razvoj testa, omogućujući kontinuirano usavršavanje i optimizaciju instrumenta za procjenu računalnog razmišljanja.

2.2. Predmet i cilj istraživanja

Predmet ovog istraživanja je izrada i validacija testa za mjerenje CT-a neovisno o programiranju i programskim jezicima. Cilj istraživanja je razviti pouzdan i valjan instrument za procjenu CT-a, koristeći IRT i Raschov model za analizu. Ovim istraživanjem želimo utvrditi koliko učinkovito test mjeri kompetencije CT-a i identificirati područja koja zahtijevaju poboljšanja. Ovo istraživanje predstavlja početnu točku u razvoju testa, a daljnje analize će biti potrebne za kontinuirano poboljšavanje i prilagođavanje pitanja kako bi se postigla optimalna preciznost i valjanost mjerenja.

1. Koje stavke testa računalnog razmišljanja trebaju reviziju za poboljšanje unutarnje dosljednosti, valjanosti i preciznosti, te koliko su pouzdane u procjeni različitih razina kompetencija?
2. Postoje li značajne razlike u postignućima na testu računalnog razmišljanja s obzirom na spol, dob, iskustvo ili obrazovni status ispitanika?

2.3. Instrumenti

Za prikupljanje podataka korišten je test izrađen u aplikaciji Google Forms. Test se sastoji od 13 zadataka višestrukog izbora, koji su dizajnirani kako bi procijene različite aspekte računalnog razmišljanja. Svaki zadatak ima pet ponuđenih odgovora, od kojih je jedan točan. Dakle, ukupan raspon bodova iznosi od 0 do 13. U prilogu na kraju rada nalazi se cijeli test.

Ovaj test za mjerenje računalnog razmišljanja proizašao je iz prethodnog pilot istraživanja malog opsega, koje je provedeno na uzorku od 6 studenata. Cilj pilot istraživanja bio je ispitati funkcionalnost i primjenjivost testa prije konačne primjene. Na temelju povratnih informacija dobivenih od sudionika, neke stavke su revidirane kako bi bile jasnije i razumljivije. Konkretno, pojedina pitanja su preformulirana kako bi se smanjila mogućnost pogrešnog tumačenja i povećala preciznost mjerenja računalnog razmišljanja.

Zadaci su odabrani kako bi se osiguralo da pokrivaju širok spektar vještina vezanih za CT.

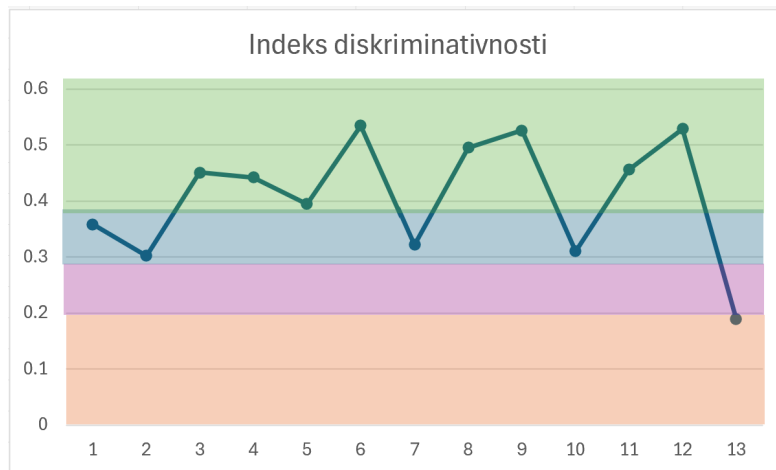
Prva tri pitanja su iz [31], četvrto i osmo pitanje inspirirana su zadacima iz *Computational Thinking Challenge Solutions Guide 2019*, dok je peto pitanje iz [32], pitanje 7 inspirirano je zadatkom s Matematičkog klockana 2023. godine (natjecanje za juniore), a pitanje 11 preuzeto je s Matematičkog klockana 2020. godine (natjecanje za kadete). Pitanja 9, 10, 12 i 13 preuzeta su iz [33].

Testiranje je provedeno u razdoblju od 1. svibnja do 15. lipnja 2024. godine, obuhvaćajući šest tjedana kako bi se omogućilo dovoljno vremena za provedbu testova u školama.

Nakon što su sudionici završili test, podaci su automatski prikupljeni i pohranjeni u obliku CSV datoteke. Ovi podaci su zatim preuzeti i analizirani u statističkom softveru R.

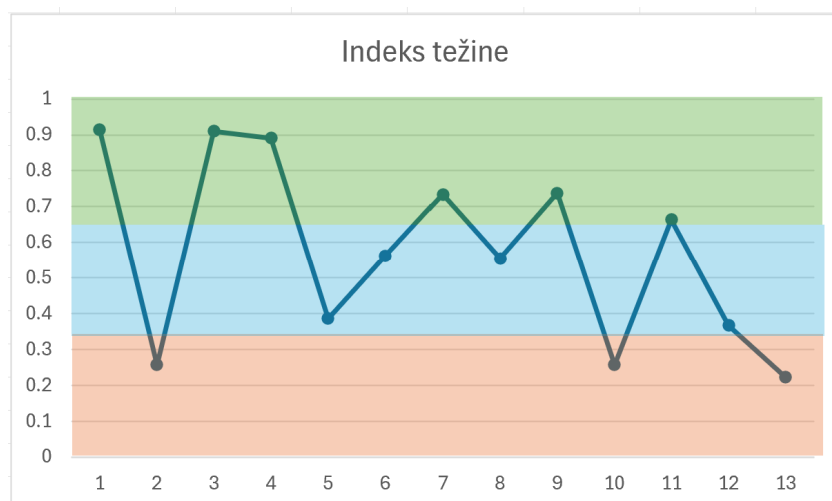
Sada ćemo analizirati ključne metrike testa, kao što su indeks težine stavki i indeks diskriminativnosti.

Rekli smo kako indeks diskriminativnosti mjeri koliko dobro pojedina stavka testa razlikuje ispitanike s različitim sposobnosti. U osnovi, on pokazuje koliko je stavak učinkovit u razlikovanju između onih koji su na visokoj razini kompetencije i onih koji su na niskoj razini. Zaključujemo da stavke 3, 4, 5, 6, 8, 9, 11, 12 dobro razlikuju sposobnosti ispitanika. Posljednja stavka ne razlikuje dobro ispitanike s različitim razinama sposobnosti. Drugim riječima, i oni s visokim i oni s niskim razinama kompetencije imaju slične rezultate na toj stavci.



Slika 2.1. Indeks diskriminativnosti

Promotrimo i indeks težine koji mjeri koliko je određenu stavku teško riješiti ispitanicima. Osvrnimo se opet na posljednju stavku, većina ispitanika ima poteškoća s točnim odgovaranjem na tu stavku. Dakle, bilo bi korisno u budućnosti razmotriti prilagodbu te stavke kako bi se postigla bolja ravnoteža između težine i diskriminativnosti.



Slika 2.2. Indeks težine

Analiza rezultata otkriva važne informacije o performansama ispitanika na pojedinačnim pitanjima testa. Tablica prikazuje prosječne vrijednosti i standardne devijacije za svako pitanje, zajedno s točnošću u postocima. Ovi podaci pružaju uvid u razinu težine pojedinačnih zadataka, varijabilnost odgovora ispitanika te opću uspješnost u rješavanju testa. Identifikacija pitanja s niskom točnošću (Pitanje 2, Pitanje 10, Pitanje 13) može ukazati na potencijalne probleme u razumijevanju ili primjeni određenih koncepata.

	Prosječna vrij.	S.Devijacija	Točnost
Q1	0.9140625	0.2808208	91.40625
Q2	0.2578125	0.4382871	25.78125
Q3	0.9101562	0.2865179	91.01562
Q4	0.8906250	0.3127205	89.06250
Q5	0.3867188	0.4879523	38.67188
Q6	0.5625000	0.4970501	56.25000
Q7	0.7343750	0.4425306	73.43750
Q8	0.5546875	0.4979738	55.46875
Q9	0.7382812	0.4404314	73.82812
Q10	0.2578125	0.4382871	25.78125
Q11	0.6640625	0.4732424	66.40625
Q12	0.3671875	0.4829825	36.71875
Q13	0.2226562	0.4168443	22.26562

Tablica 2.1. Uspješnost po pitanjima.

Na primjer, pitanje 13 je prikazano na slici. Problem može izgledati složeno, posebno pod testnim uvjetima gdje sudionici mogu osjećati pritisak koji otežava logičko razmišljanje i analizu problema. Također, nedostatak prakse može objasniti lošu rješivost. Ako studenti nisu ranije radili s ovakvim tipom logičkih problema, možda će im nedostajati iskustvo potrebno za brzo i efikasno rješavanje.

Pitanje 13. Dobili ste 9 novčića iste vrijednosti, ali jedan od njih je lažan. Lažni novčić je lakši od ostalih.

Za vaganje novčića imate vagu kao na slici, a svako vaganje može rezultirati "vaga se naginje ulijevo", "vaga se naginje udesno" ili "vaga ostaje stabilna".

Pretpostavljajući da važete najmanji potreban broj puta, koliko vaganja je potrebno da biste identificirali lažni novčić?



- A) 1 vaganje
- B) 2 vaganja
- C) 3 vaganja
- D) 9 vaganja
- E) Ne znam

Slika 2.3. Pitanje 13

2.4. Sudionici

Istraživanje je provedeno u različitim srednjim školama, na fakultetu te metodom lavine. Uzorak je činilo ukupno 256 sudionika, od kojih su 131 bili muškarci, 115 žene, dok se 10 sudionika nije izjasnilo o spolu.

Prema prikupljenim podacima, test računalnog razmišljanja riješilo je 61 učenika iz III. gimnazije. U Elektrotehničkoj školi zabilježeno je 16 riješenih testova. U V. gimnaziji ukupno je riješeno 70 testova. Škola za dizajn, grafiku i održivu gradnju zabilježila je 34 riješena testa. Na Prirodoslovno-matematičkom fakultetu riješeno je 50 testova, dok je izvan navedenih ustanova zabilježeno još 25 riješenih testova. Dakle, ukupno je 181 srednjoškolac riješio test računalnog razmišljanja, 50 studenata i 25 ostalih.

Što se tiče obrazovanja u programiranju, 20 sudionika (7,8%) izjavilo je da nikada nisu obrađivali teme programiranja u školi. Većina sudionika (68,0%; N = 174) imala je 1 - 4 godine obrazovanja u programiranju. Nadalje, 58 sudionika (22,7%) imalo je 4 - 8 godina obrazovanja u programiranju, dok su 3 sudionika (1,2%) imala više od 8 godina obrazovanja u programiranju.

2.5. Postupak

Odobrenje za provedbu studije dobiveno je od Etičkog odbora Prirodoslovno-matematičkog fakulteta Sveučilišta u Splitu.

Konačni test proveden je u srednjim školama i na fakultetu tijekom nastave informatike. Sudionicima je dodijeljeno do 30 minuta za rješavanje 13 zadataka, što je omogućilo dovoljno vremena za razmatranje i odgovaranje na svaki zadatak.

Svaki zadatak kasnije je u analizi ocjenjivan dihotomski: bodovao se kao točan (1) ili netočan (0). Sudionici su morali odgovoriti na svako pitanje kako bi prešli na sljedeće, čime je osigurana potpuna pokrivenost svih stavki testa.

Tijekom provedbe, svim sudionicima bilo je dopušteno koristiti papir i olovku za bilješke kako bi se izbjeglo moguće kognitivno opterećenje tijekom rješavanja zadataka. Nakon završetka testiranja, odgovori sudionika prikupljeni su u obliku CSV datoteke kako bi se omogućila daljnja analiza i obrada podataka u statističkim programima.

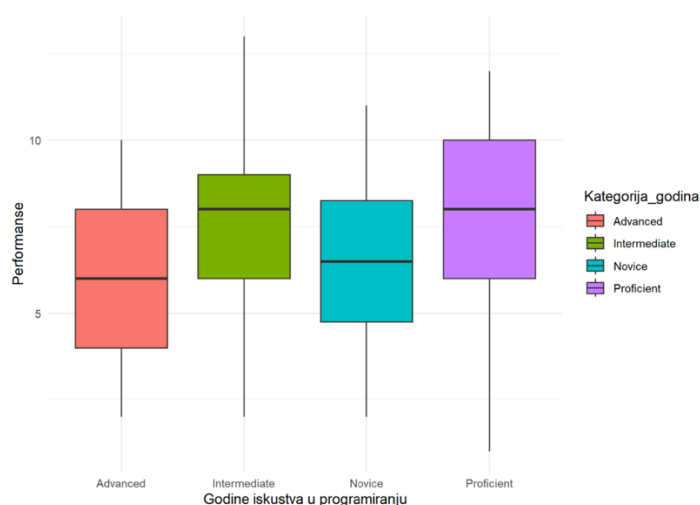
Analiza stavki provedena je iz koristeći Raschov model, koji pruža detaljniju analizu svake pojedinačne stavke testa. Izbor Raschovog modela za analizu podataka opravdan je nekoliko razloga. Prije svega, Raschov model omogućuje obradu latentnih konstrukcija poput računalnog razmišljanja, koje se ne mogu izravno promatrati. Također, IRT metodologija, koja uključuje Raschov model, pokazala se kao robusnija i pouzdanija u razvoju i validaciji procjena temeljenih na kompetencijama. Kao početna faza razvoja novih procjena, korištenje Raschovog modela omogućuje procjenu valjanosti i preciznosti testa, čime se omogućuju potrebna poboljšanja ili izmjene stavki za buduću upotrebu.

3. Rezultati

3.1. Deskriptivna statistika

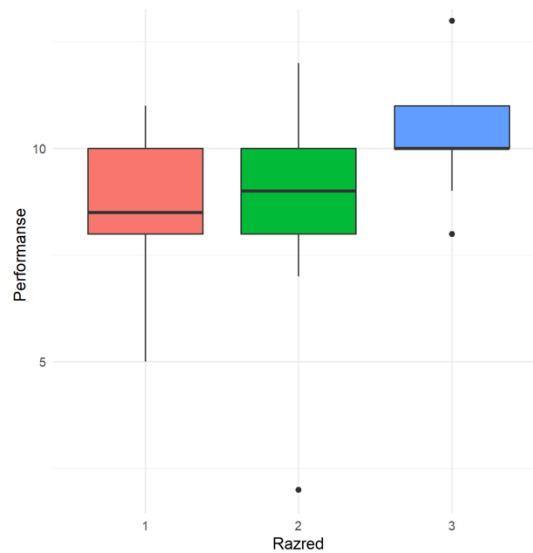
Grafikon prikazuje kako se performanse u programiranju mijenjaju s godinama iskustva. Na x-osi su kategorije godina iskustva: početnik, srednji, napredni i stručnjak, dok y-osa prikazuje ukupan broj bodova na testu. Na temelju rezultata testa, možemo izvući sljedeće zaključke za svaku grupu:

- Početnici (Novice): Raspon performansi je između 5 i 7, s medijanom oko 6. Ova grupa ima najniži medijan performansi, što je očekivano s obzirom na nedostatak iskustva.
- Srednji (Intermediate): Performanse se kreću između 6 i 9, s medijanom oko 8. Raspodjela je asimetrična s duljim donjim dijelom što sugerira prisutnost nekoliko nižih vrijednosti. Ova grupa ima najviši medijan performansi, što sugerira da programeri s srednjim nivoom iskustva postižu bolje rezultate u usporedbi s ostalim grupama.
- Napredni (Advanced): Performanse u ovoj grupi variraju od oko 4 do 8, s medijanom oko 6. Izduženi oblik ukazuje na postojanje ekstremnih vrijednosti.
- Stručnjaci (Proficient): Stručnjaci imaju najviše i najkonzistentnije rezultate. Njihove performanse su blizu maksimalnih bodova, što ukazuje na visoku razinu stručnosti i iskustva.



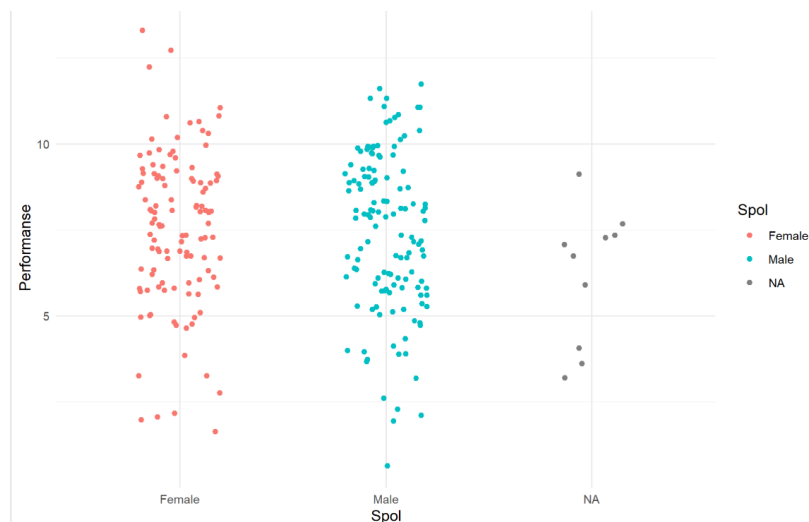
Slika 3.1. Raspodjela performansi prema godinama iskustva u programiranju

Taj trend postaje očit i kada analiziramo podatke unutar jedne škole. Na primjer, u III. gimnaziji pregledom rezultata po razredima vidimo da učenici s napredovanjem kroz razrede postižu sve bolje rezultate na testovima. Ovaj napredak možemo objasniti stjecanjem sve većeg znanja i vještina s godinama, što se pozitivno odražava na njihovim rezultatima.



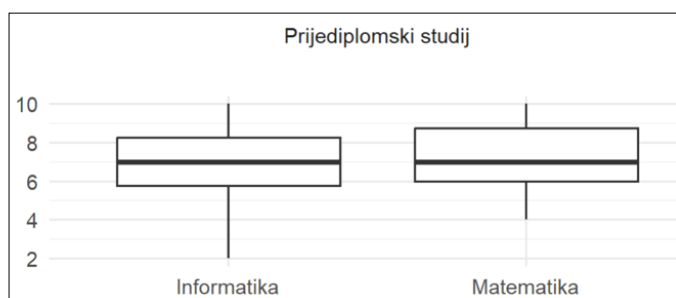
Slika 3.2. Raspodjela performansi po razredima u III. gimnaziji Split

Nadalje, promotrimo rezultate po spolu. Uočavamo da su performanse po spolu prilično slično raspodijeljene, iako naravno unutar svake skupine postoje pojedinci s različitim razinama performansi.

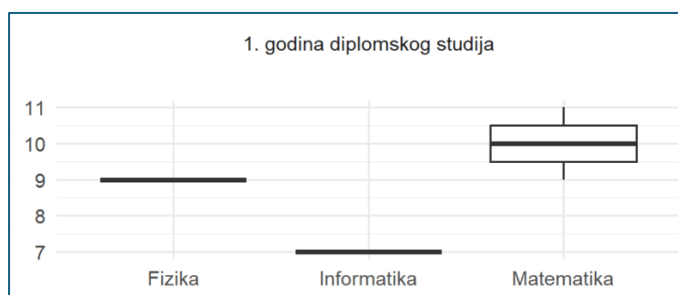


Slika 3.3. Performanse po spolu

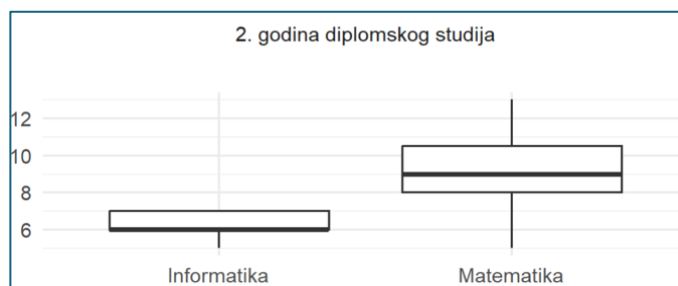
Pogledajmo sada usporedbu između studenata na različitim obrazovnim razinama: prijediplomskoj, prvoj godini diplomskog studija i drugoj godini diplomskog studija. Želimo istražiti postoje li razlike u njihovim rezultatima na testu.



Slika 3.4. Rezultati na prijediplomskoj razini



Slika 3.5. Rezultati na prvoj godini diplomskog studija



Slika 3.6. Rezultati na drugoj godini diplomskoj studija

3.2. Korelacija rezultata testa i odabranih varijabli

Nadalje, ispitivana je konvergentna valjanost koja predstavlja koeficijente korelacije između različitih varijabli. Konvergentna valjanost potvrđena je značajnim i pozitivnim odnosom između ukupne uspješnosti na testu i godina učenja programiranja. Odnosno, postoji statistički značajna i pozitivna veza između uspješnosti na testu i godina učenja programiranja sudionika ($t = 2.43$, $df = 254$, p -vrijednost = 0.015). Procjena Pearsonovog koeficijenta korelacije iznosi 0.15, što sugerira da postoji slaba pozitivna linearna veza između ovih varijabli.

```
data: test_performance and years
t = 2.4397, df = 254, p-value = 0.01538
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.0292575 0.2689296
sample estimates:
      cor
0.1513167
```

Slika 3.7. Veza uspješnosti na testu i godina učenja programiranja

Međutim, korelacija između uspješnosti na testu i samoprocjene znanja programiranja sposobnosti sudionika, iako pozitivna, nije statistički značajna ($r = 0.03$, $p = 0.6034$). Ova veza je vrlo slaba, s procijenjenim Pearsonovim koeficijentom korelacije od 0.0326237.

```
Pearson's product-moment correlation

data: test_performance and ability
t = 0.52021, df = 254, p-value = 0.6034
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
-0.09033962 0.15460731
sample estimates:
      cor
0.0326237
```

Slika 3.8. Veza uspješnosti na testu i samoprocjene znanja

Ovi rezultati sugeriraju da postoji određena povezanost između uspješnosti na testu i godina učenja programiranja sudionika, ali da ta povezanost nije jaka. S druge strane, veza između uspješnosti na testu i procjene sposobnosti nije statistički značajna.

	correlation	p-value
test_performance vs. ability	0.0326237	0.15131672
test_performance vs. years	0.6033681	0.01538495

Slika 3.9. Povezanost uspješnosti na testu, godina učenja i procjene sposobnost

3.3. Rasch Model

Prije primjene Raschovog modela, pretpostavka unidimenzionalnosti testirana je pomoću Martin-Löf testa. Ovaj test koristi LR-vrijednost (likelihood ratio) za procjenu unidimenzionalnosti podataka, koja se zatim uspoređuje s χ^2 raspodjelom kako bi se utvrdila značajnost testa. Stupnjevi slobode u χ^2 raspodjeli predstavljaju broj parametara procijenjenih u modelu.

U ovom slučaju, p-vrijednost od 0.6 ukazuje na to da ne postoji dovoljno dokaza za odbacivanje nulte hipoteze o unidimenzionalnosti. To sugerira da podaci podržavaju pretpostavku unidimenzionalnosti.

```
Martin-Loef-Test (split criterion: median)
LR-value: 38.099
Chi-square df: 41
p-value: 0.6
```

Slika 3.10. Provjera unidimenzionalnosti

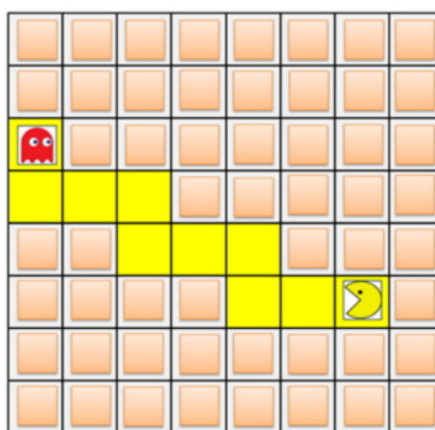
Za analizu podataka Raschovim modelom korišten je eRm paket u programskom jeziku R. Rezultati procjene parametara težine (η) i lakoće (β) za svako pitanje pružili su dublji uvid u relativnu težinu i lakoću svake stavke u testu. Parametri težine daju informaciju o relativnoj sposobnosti pitanja da razlikuju između sudionika s različitim latentnim osobinama, dok parametri lakoće ukazuju na relativnu težinu odgovaranja na pojedina pitanja.

```
Item (Category) Difficulty Parameters (eta): with 0.95 CI:
      Estimate Std. Error lower CI upper CI
Pitanje2_kod      1.677      0.147      1.388      1.965
Pitanje3_kod     -2.124      0.218     -2.551     -1.698
Pitanje4_kod     -1.886      0.201     -2.280     -1.491
Pitanje5_kod      1.016      0.134      0.753      1.280
Pitanje6_kod      0.210      0.132     -0.050      0.470
Pitanje7_kod     -0.665      0.147     -0.954     -0.377
Pitanje8_kod      0.246      0.132     -0.013      0.505
Pitanje9_kod     -0.688      0.148     -0.978     -0.399
Pitanje10_kod     1.677      0.147      1.388      1.965
Pitanje11_kod    -0.282      0.139     -0.553     -0.010
Pitanje12_kod     1.109      0.135      0.844      1.375
Pitanje13_kod     1.887      0.154      1.586      2.189
```

Slika 3.11. Procjena parametara težine

Na primjer, procjena težine za pitanje 2 iznosi 1.677, što sugerira da je ovo pitanje relativno teže za sudionike s nižim latentnim osobinama u usporedbi s drugim pitanjima. Također, pitanje 13 je isto bilo među težima. Najlakša pitanja su bila pitanja 3 i 4.

Pitanje 2. Koliko puta treba ponoviti navedene instrukcije da pac-man dođe do duha? *



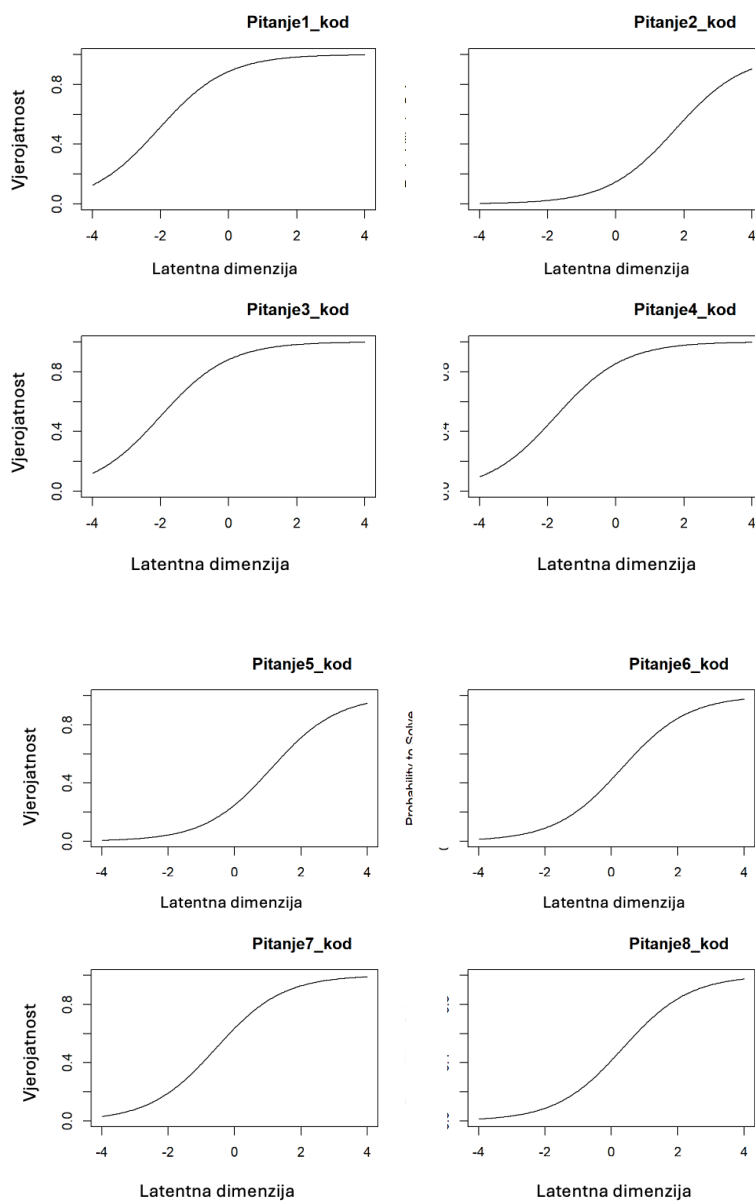
- 2
- 3
- 4
- 5
- Ne može doći do duha

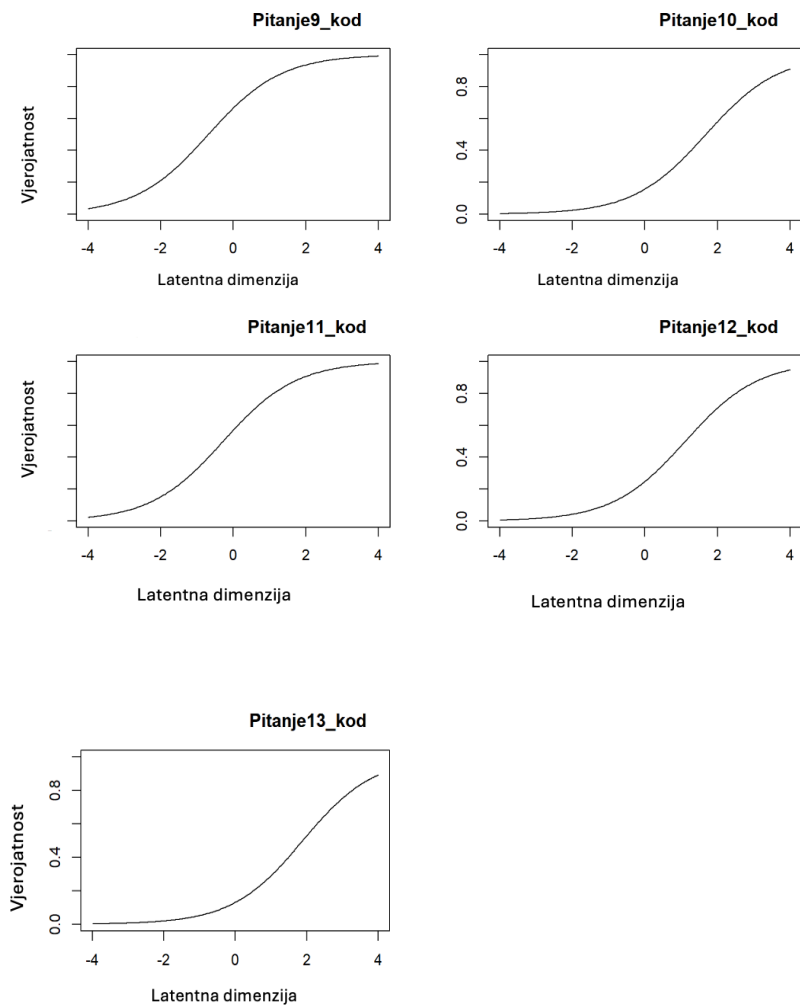
Slika 3.12. Pitanje 2

3.4. Item Characteristic Curve

Koristimo eRm paket u programu R za generiranje ICC krivulja. Prikazujemo ICC krivulje za svaku stavku u našem skupu podataka i interpretiramo rezultate kako bismo bolje razumjeli ponašanje stavki u odnosu na latentnu osobinu koja se mjeri.

Ispitanici s nižom latentnom sposobnošću (lijevo na x-osi) imaju nižu vjerojatnost za točno rješavanje teških pitanja. Ispitanici s višom latentnom sposobnošću (desno na x-osi) imaju veću vjerojatnost za točno rješavanje svih pitanja, uključujući i teška.





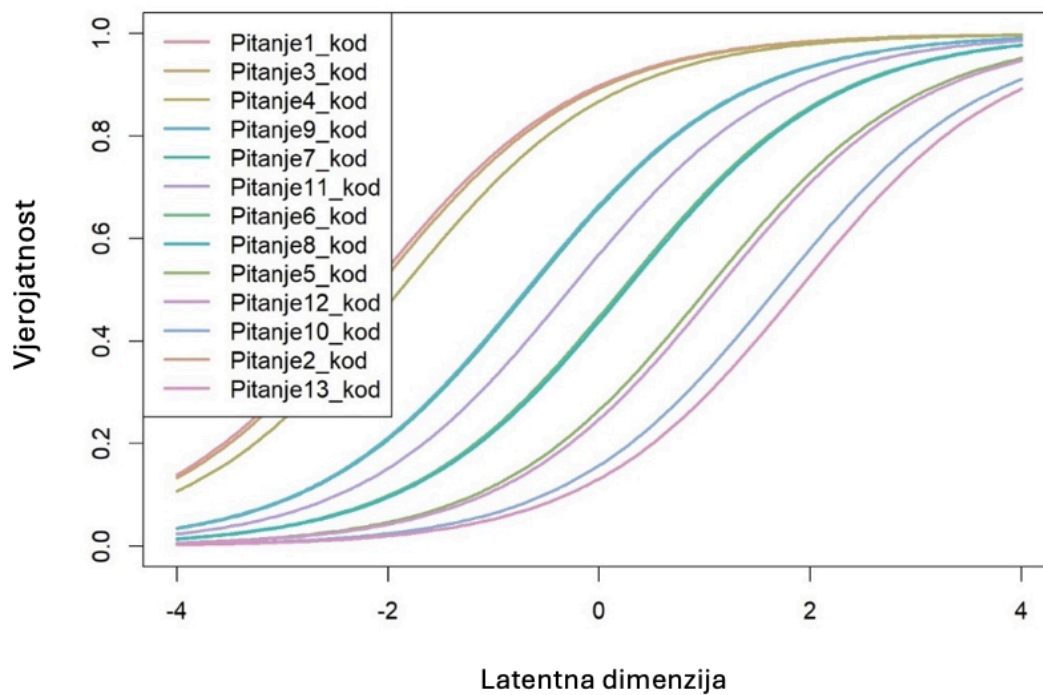
Slika 3.13. Krivulje karakteristika pojedinačnih stavki

Iako na slici 3.13. možemo promatrajući oblik krivulje za svako pitanje uočiti koja su pitanja bila laka, a koja teška, lakše je usporediti sve stavke zajedno na slici 3.14. Ova slika prikazuje krivulje karakteristika za svih 13 stavki testa, omogućujući bolju usporedbu i preglednost svih stavki u cjelini.

Stavke na gornjem lijevom dijelu (npr. Pitanja 1, 3, 4) predstavljaju lake stavke jer je vjerojatnost ispravnog odgovora visoka za ispitanike s nižim performansama. Krivulje u sredini (npr. Pitanja 9,7,11) predstavljaju stavke srednje težine. Krivulje na donjem desnom dijelu (npr. Pitanja 10, 13) predstavljaju teške stavke; vjerojatnost ispravnog odgovora je niska za većinu ispitanika i povećava se tek kada se postigne viša razina performansi.

Dakle, Pitanje3 i Pitanje4 imaju krivulje najviše ulijevo, što potvrđuje da su ta pitanja bila najlakša, što je u skladu s procjenama težine. Isto tako, Pitanje2 i Pitanje13 imaju krivulje najviše udesno, potvrđujući da su ta pitanja bila najteža.

Većina krivulja prelazi kroz središnji dio grafa (latentna dimenzija oko 0), što sugerira da test sadrži pitanja koja su prikladna za prosječnu sposobnost ispitanika. Nema puno krivulja koje su izrazito lijevo ili desno, što znači da test nije izrazito težak ili lagan u cjelini, već ima dobar raspon težine.



Slika 3.14. Krivulje karakteristika svih stavki

Nadalje, na slici 3.14. navodim primjer pitanja srednje razine težine, poput pitanja 9. Riječ je o pitanju gdje je trebalo predvidjeti kako će tablica izgledati nakon zamjene stupaca.

Pitanje 9. Prikazana je tablica s redcima i stupcima. U tablici su neka polja crna, a neka bijela. Zamislite da prvo zamijenimo stupac 4 sa stupcem 2. Nakon toga zamijenimo redak 2 s redkom 4. Koji će biti rezultat?

		Stupac			
		1	2	3	4
Red	1		■		
	2			■	
	3	■			■
	4		■		■

A)

		1 2 3 4			
1					
2				■	
3	■			■	
4			■		

B)

		1 2 3 4			
1			■		■
2		■			
3	■				
4			■		

C)

		1 2 3 4			
1		■		■	
2					
3	■				■
4		■		■	

D)

		1 2 3 4			
1		■		■	
2			■		
3	■				
4		■		■	

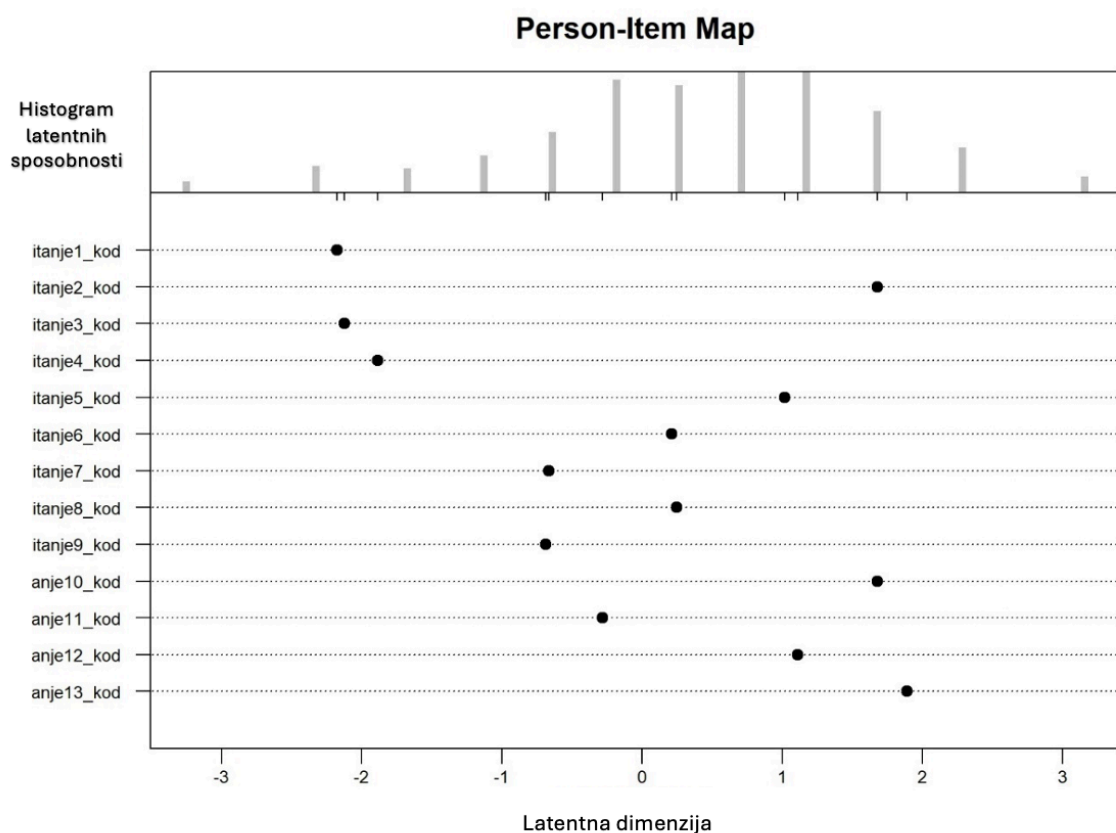
E) Ne znam.

Slika 3.15. Pitanje 9

3.5. Person Item Map

Na slici 3.16, histogram prikazuje distribuciju sposobnosti ispitanika. Iz grafičkog prikaza vidimo da većina ispitanika ima sposobnosti koncentrirane oko latentne dimenzije 0, s većim brojem ispitanika koji imaju sposobnosti između -1 i 2. Postoji manji broj ispitanika s vrlo niskom (ispod -2) ili vrlo visokom sposobnošću (iznad 2).

Pitanja 3 i 4 imaju najniže težine (-2.124 i -1.886, respektivno), što potvrđuje da su najlakša. Pitanja 2 i 13 imaju najviše težine (1.677 i 1.887), što ih čini najtežima. Ostala pitanja imaju težine raspoređene između ovih ekstrema, s većinom težina između -1 i 2.



Slika 3.16. Person Item Map

Prikazana mapa na slici sugerira da su pitanja 2, 10 i 13 bila točno odgovorena samo od strane ispitanika smještenih na vrhu distribucije θ latentne vrijednosti, dok su stavke 1,3 i 4 točno odgovorene od gotovo svih ispitanika osim vrlo malog broja koji su smješteni na dnu distribucije θ vrijednosti.

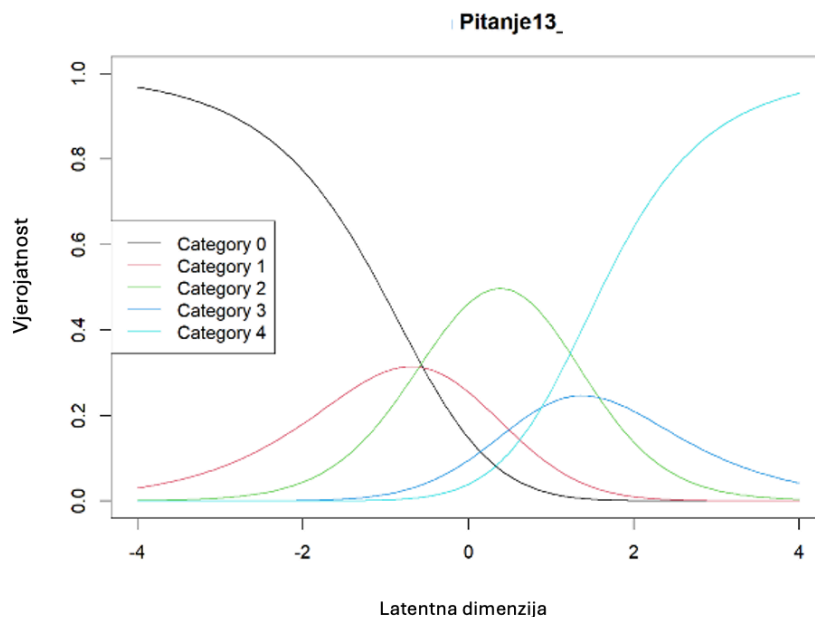
3.6. Partial Credit Model

Prikazane su procjene parametara težine za svaku kategoriju unutar svakog pitanja, zajedno sa standardnim pogreškama koje mjere nesigurnost u procjeni težine. Stavke s pozitivnim parametrima težine su teže za postizanje. Parametri težine pokazuju koliko je teško dobiti određeni odgovor, dok intervali pouzdanosti ukazuju na preciznost procjene; širi intervali znače veću nesigurnost, a uži veću preciznost. Na primjer, kod pitanja 3 bilo je vrlo lako postići točan odgovor

Item (Category)	Difficulty Parameters (eta):	Estimate	Std. Error	lower CI	upper CI
Pitanje1_numeric.c2		0.680	0.457	-0.216	1.576
Pitanje1_numeric.c3		1.842	0.641	0.586	3.098
Pitanje2_numeric.c1		-3.267	0.668	-4.577	-1.958
Pitanje2_numeric.c2		-0.825	0.736	-2.268	0.617
Pitanje2_numeric.c3		1.327	1.153	-0.932	3.586
Pitanje2_numeric.c4		-3.712	0.607	-4.902	-2.523
Pitanje3_numeric.c1		-1.295	0.748	-2.762	0.172
Pitanje3_numeric.c2		-1.116	0.735	-2.556	0.325
Pitanje3_numeric.c3		-4.196	0.651	-5.471	-2.921

Slika 3.17. Partial Credit Model za prvih nekoliko pitanja

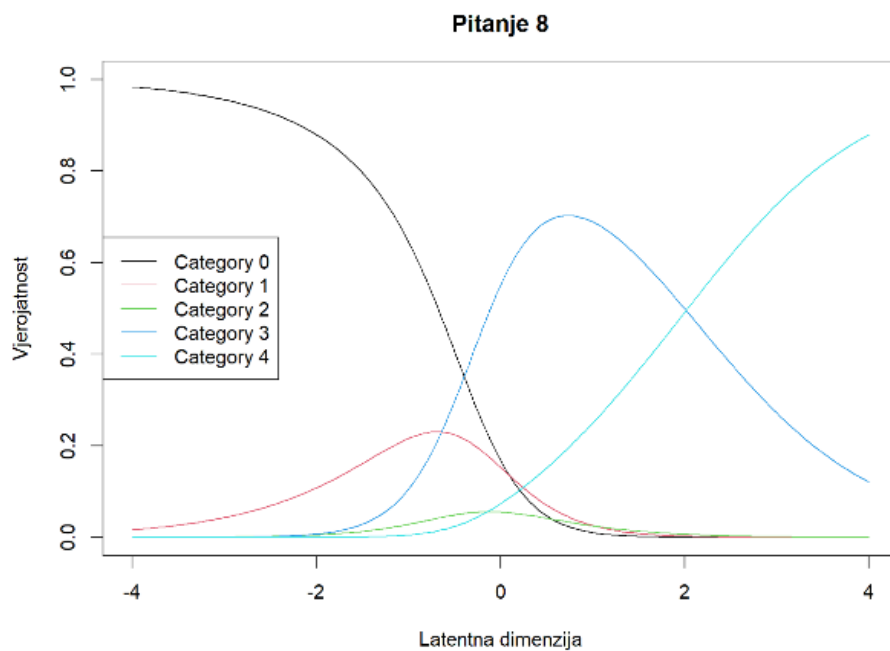
Karakteristične krivulje za model parcijalnih kredita prikazane su zasebno korištenjem iste funkcije plotICC. Odabrali smo ICC za pitanja 8 i 13. Stavka 13 pruža primjer stavke s neuređenim kategorijama, što može ukazivati na potrebu za revizijom stavke.



Slika 3.18. Partial Credit Model - Pitanje 13

Međutim, čini se da se stavka 8 ponaša mnogo bolje u usporedbi, jer je svaka kategorija odgovora najvjerojatniji odgovor za određeni raspon latentne dimenzije. ICC-ovi pružaju opću ideju o prilagođenosti specifičnih stavki PCM-u, ali detaljnije informacije dobivaju se sa statistikom prilagođenosti stavki.

Pitanje 8 ima dobro razdvojene kategorije, svaka s definiranim rasponom latentne dimenzije za najvjerojatniji odgovor, dok je pitanje 13 problematično jer krivulje za kategorije 1, 2 i 3 pokazuju preklapanje i nedostatak jasno definiranih dominantnih raspona na latentnoj dimenziji, sugerirajući potrebu za revizijom.



Slika 3.19. Partial Credit Model - Pitanje 8

3.7. Item Fit Statistic

Outfit MSQ i Infit MSQ vrijednosti bliske 1 ukazuju na dobro uklapanje u model. Vrijednosti znatno iznad ili ispod 1 ukazuju na moguća odstupanja. Pitanja 3 i 4 imaju niske vrijednosti Infit MSQ i Outfit MSQ, što ukazuje na to da su previše predvidljiva. To znači da su se ispitanici vrlo slično ponašali na tim pitanjima, što može smanjiti njihovu sposobnost razlikovanja između različitih sposobnosti ili znanja.

Itemfit Statistics:

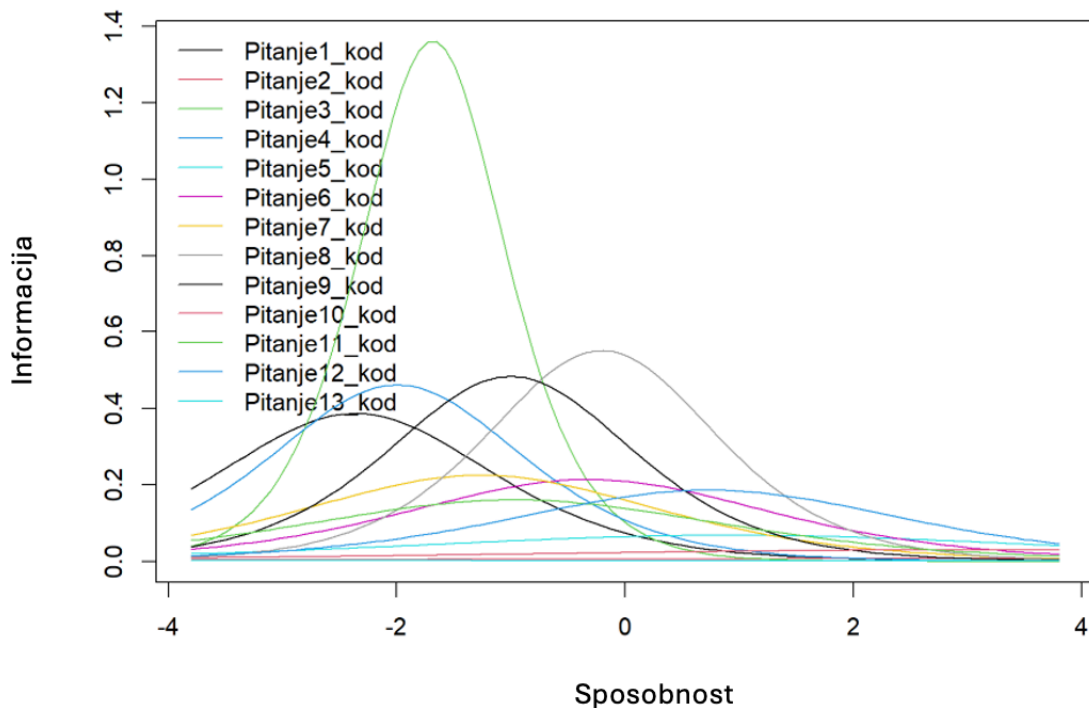
	Chisq	df	p-value	Outfit MSQ	Infit MSQ	Outfit t	Infit t	Discrim
Pitanje1_numeric	256.246	255	0.466	1.001	0.926	0.085	-0.232	-0.110
Pitanje2_numeric	253.243	255	0.519	0.989	0.923	-0.078	-1.233	-0.149
Pitanje3_numeric	163.991	252	1.000	0.648	0.725	-1.375	-1.250	0.224
Pitanje4_numeric	192.932	255	0.999	0.754	0.679	-0.943	-1.940	0.221
Pitanje5_numeric	203.890	239	0.952	0.850	0.857	-1.830	-1.839	0.145
Pitanje6_numeric	222.384	255	0.931	0.869	0.869	-1.443	-1.438	0.225
Pitanje7_numeric	265.424	255	0.314	1.037	0.984	0.357	-0.153	-0.215
Pitanje8_numeric	224.661	229	0.569	0.977	0.939	-0.201	-0.715	0.074
Pitanje9_numeric	199.031	255	0.996	0.777	0.830	-1.600	-1.298	0.332
Pitanje10_numeric	214.912	255	0.968	0.839	0.831	-1.961	-2.072	0.381
Pitanje11_numeric	254.190	249	0.397	1.017	1.010	0.205	0.148	-0.036
Pitanje13_numeric	210.306	255	0.981	0.822	0.824	-2.185	-2.152	0.301

Slika 3.20. Item Fit Statistike

t-vrijednosti su standardizirane vrijednosti koje pokazuju koliko je Infit i Outfit odstupanje značajno. Pitanje 10 ima visoke t-vrijednosti za Outfit t (-1.961) i Infit t (-2.072), što ukazuje na to da je previše nepredvidljivo. To znači da se ispitanici nisu dosljedno ponašali na tom pitanju, što može smanjiti njegovu sposobnost razlikovanja između različitih sposobnosti ili znanja.

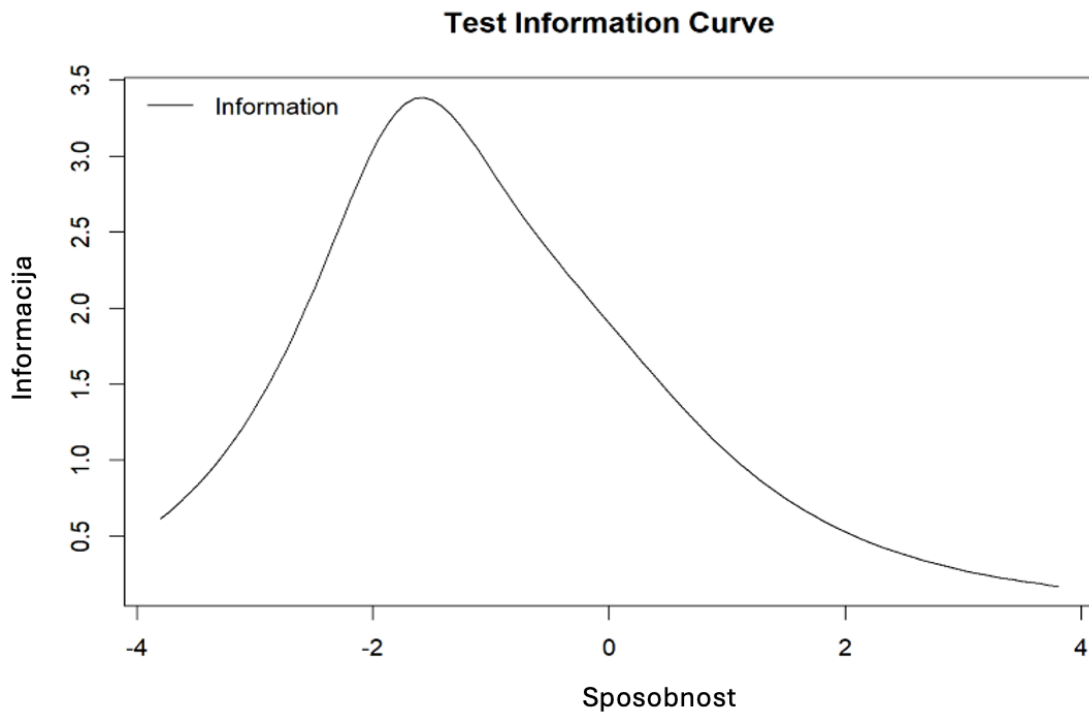
3.8. Item Information Curve

Svako pitanje ima svoj vrhunac informacijske vrijednosti, koja pokazuje koliko informacija pitanje daje o ispitaniku s određenom sposobnošću. Pitanje 4 ima najvišu informaciju kod sposobnosti blizu -1.5. To znači da ovo pitanje najbolje razlikuje ispitanike s tom razinom sposobnosti. Većina pitanja pruža najviše informacija za sposobnosti između -3 i 0. Dakle, treba dodati pitanja koja su informativnija za sposobnosti iznad 0 i ispod -3 kako bi se poboljšala sveukupna informativnost testa za širi raspon sposobnosti.



Slika 3.21. Item Information Curve

Test ima najvišu informaciju kod sposobnosti blizu -1. To znači da je test najinformativniji za ispitanike s tom razinom sposobnosti. Informacijska vrijednost opada za više i niže razine sposobnosti, što može ukazivati na smanjenje preciznosti testa za te ekstremnije vrijednosti.



Slika 3.22. Test Information Curve

3.9. Ograničenja istraživanja

Kao što je često slučaj, ovo istraživanje susreće se s nekoliko izazova koji su važni za istaknuti kako bismo pravilno razumjeli dobivene rezultate i razmotrili njihovu primjenjivost na širem spektru. Prvo ograničenje odnosi se na činjenicu da su svi ispitanici izabrani iz splitskih škola i fakulteta, što može ograničiti generalizaciju rezultata na opću populaciju. Drugo, veličina uzorka od 256 ispitanika, iako solidna, može utjecati na statističku snagu i preciznost dobivenih rezultata. Nadalje, bitno je napomenuti da su svi ispitanici sudjelovali dobrovoljno, što može imati utjecaj na njihovu motivaciju i odgovore, te posljedično na valjanost dobivenih rezultata.

4. Zaključak

Analiza rezultata pruža važne uvide u performanse ispitanika na testu računalnog razmišljanja. Identificirani su ključni aspekti performansi na pojedinačnim pitanjima, kao i statistički relevantni odnosi među varijablama. Nalazi sugeriraju da su neka pitanja bila izazovnija od drugih, što može ukazivati na potrebu za poboljšanjem razumijevanja određenih koncepata među sudionicima. Na primjer, pitanje 13 ističe se kao teže, što može biti rezultat nedostatka prakse ili nedostatka iskustva s takvim vrstama problema.

Također, rezultati pokazuju određene statističke veze između uspješnosti na testu i varijabli poput godina učenja programiranja, pri čemu veća iskustva pokazuju pozitivnu korelaciju s boljim performansama. Međutim, nije pronađena statistički značajna veza između uspješnosti na testu i samoprocjene znanja programiranja.

Dodatno, primjena Rasch modela omogućila je detaljniju analizu svakog pitanja, pružajući dublji uvid u relativnu težinu svake stavke u testu. Prepoznata su pitanja koja nisu dovoljno teška ili pitanja koja su previše predvidljiva. Na primjer, pitanja 3 i 4 su previše predvidljiva.

Također, pokazalo se da u test treba uključiti pitanja koja su informativnija za različite razine sposobnosti. Grafički prikazi su dodatno olakšali interpretaciju rezultata, posebno kroz krivulje karakteristika stavki, koje su omogućile procjenu ponašanja stavki u odnosu na latentnu osobinu.

Iako ovo istraživanje pruža važne uvide u performanse na testu računalnog razmišljanja, postoje i određena ograničenja, poput ograničenog uzorka i nedostatka reprezentativnosti. Stoga se preporučuje daljnje istraživanje s većim uzorkom i uključivanjem različitih demografskih skupina radi dobivanja sveobuhvatnijih saznanja.

U cijelosti, ova studija doprinosi razumijevanju performansi na testu računalnog razmišljanja i pruža smjernice za daljnja istraživanja u području obrazovanja računalnog razmišljanja. Obzirom na kompleksnost ovog područja, kontinuirano istraživanje i razvoj metodologija za procjenu CT-a su od vitalnog značaja za unapređenje obrazovanja i pripremu studenata za suvremeno društvo.

Literatura

- [1] J. M. Wing, »Computational thinking,« *Communications of the ACM*, pp. Svezak 49, broj 3, 33–35, Ožujak 2006.
- [2] K. Kasalo, »Poučavanje umjetne inteligencije na primjeru problema klasifikacije slika,« 2023.
- [3] A. Yadav, C. Stephenson i H. Hong, »Computational thinking for teacher education,« 2017.
- [4] J. M. Wing, »Computational Thinking and Thinking About Computing,« *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, svez. 366, br. 1881, pp. 3717-3725, 2008.
- [5] P. S. Wang, u *From computing to computational thinking*, CRC Press, 2016, pp. 93-95.
- [6] W. F. (. Overton, »Reasoning, necessity, and logic: Developmental perspectives,« Lawrence Erlbaum Associates, Inc., 1990.
- [7] H. Markovits, »The Development of Deductive Reasoning,« u *The Nature of Reasoning*, Cambridge University Press, 2004, pp. 313-339.
- [8] G. Miller, »The magical number seven, plus or minus two: Some limits on our capacity for processing information,« *The Psychological Review*, svez. 63, pp. 81-97, 1956.
- [9] R. C. Martin, »Agile Software Development: Principles, Patterns, and Practices,« Prentice Hall PTR, United States, 2003.
- [10] o. L. J. Tolboom, W. J. Rijke, L. Bollen i T. H. S. Eysink, »Computational thinking in primary school: An examination of abstraction and decomposition in different age groups,« Vilniaus Universiteto Leidykla, 2018.

- [11] C. Miroló, C. Izu, V. Lonati i E. Scapin, »Abstraction in Computer Science Education: An overview,« *Informatics in Education*, svez. 20, pp. 615-639, 2021.
- [12] N. Bubica, M. Mladenović i I. Boljat, »Programiranje kao alat za razvoj apstraktnog mišljenja,« 2013.
- [13] O. A. Awopeju i Afolabi, E. R. I., »Comparative Analysis of Classical Test Theory and Item Response Theory Based Item Parameter Estimates of Senior School Certificate Mathematics Examination,« *European Scientific Journal*, svez. 12, br. 28, 2016.
- [14] R. E. Traub, »Classical Test Theory in Historical Perspective,« u *Educational Measurement: Issues and Practice*, svez. 16, 1997.
- [15] D. Husremović, u *Osnove psihometrije za studente psihologije*, 2016, pp. 48-50.
- [16] R. L. Ebel, »Procedures for the Analysis of Classroom Tests,« *Educational and Psychological Measurement*, svez. 14, br. 2, pp. 352-364, 1954.
- [17] M. Tavakol i R. Dennick, »Making sense of Cronbach's alpha,« *International Journal of Medical Education*, svez. 2, pp. 53-55, 2011.
- [18] M. R. Novick, »The axioms and principal results of classical test theory,« *Journal of Mathematical Psychology*, svez. 3, br. 1, 1966.
- [19] T. S. Kuhn, »Logic of Discovery or Psychology of Research?,« Cambridge University Press, 1970.
- [20] P. Mair, »Item Response Theory,« u *Modern Psychometrics with R. Use R!*, Springer, 2018.
- [21] A. Xinming i Y. F. Yung, »Item Response Theory: What It Is and How You Can Use the IRT Procedure to Apply I,« 2014.
- [22] A. A. Bichi i R. Talib, »Item Response Theory: An Introduction to Latent Trait Models to Test and Item Development,« *International Journal of Evaluation and Research in Education*, svez. 7, br. 2, pp. 142-151, 2018.

- [23] R. J. de Ayala, u *The Theory and Practice of Item Response Theory*, Guilford Press, 2009, pp. 7-9.
- [24] C. DeMars, u *Item response theory*, Oxford University Press, Inc., 2010, pp. 9-15.
- [25] R. J. de Ayala, »The Three-Parameter Model,« u *The Theory and Practice of Item Response Theory*, Guilford Press, 2009, pp. 123-135.
- [26] P. R. Kelley i C. F. Schumacher, »The Rasch Model,« *Evaluation & the Health Professions*, svez. 7, pp. 443-454, 1984.
- [27] T. Mallinson, »Why Measurement Matters for Measuring Patient Vision Outcomes,« *Optometry and Vision Science*, svez. 84, p. E675–E682, 2007.
- [28] R. J. De Ayala, »The One-Parameter Model,« u *The Theory and Practice of Item Response Theory.*, Guilford Press, 2009, pp. 11-38.
- [29] C. DeMars, u *Item Response Theory* , Oxford University Press, 2010, pp. 15-20.
- [30] F. Lord, »A theory of test scores,« *Psychometric Monographs*, 1952.
- [31] M. Román-González, »Computational thinking test: design guidelines and content validation,« Barcelona, 2015.
- [32] R. P. Y. Lai, »Beyond Programming: A Computer-Based Assessment of Computational Thinking Competency,« *ACM Transactions on Computing Education (TOCE)*, svez. 22, br. 2, 2021.
- [33] M. Lafuente, O. Lévêque, C. Hardebolle i J. D. Zufferey, »Assessing Computational Thinking: Development and Validation of the Algorithmic Thinking Test for Adults,« 2021.

Popis slika i tablica

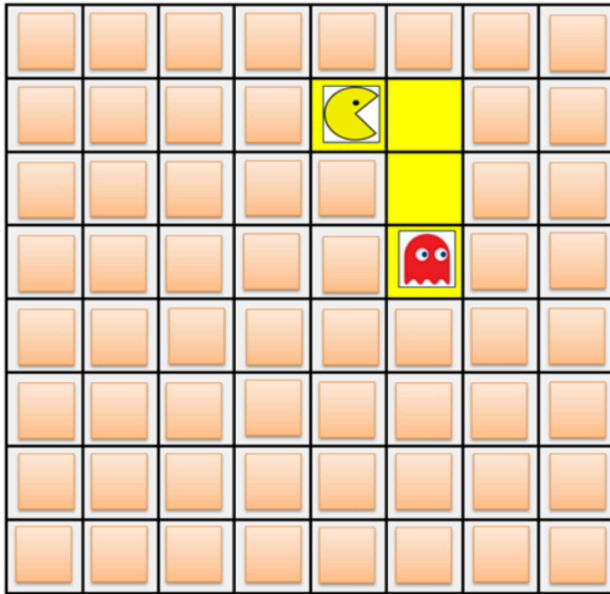
Slika 1.1. Strategija "podijeli i vladaj"	6
Slika 1.2. Stavke s različitim a parametrom. Stavka 1 je više diskriminantna.....	16
Slika 1.3. Stavke s različitim b parametrom. Stavka 1 je teža.....	16
Slika 1.4. Stavke s različitim c parametrom. Stavka 2 ima veći c parametar.....	16
Slika 1.5. 2PL model sa različitim a i b parametrima.....	18
Slika 1.6. 1PL model, stavke sa jednakim a parametrom i različitim b parametrom	19
Slika 2.1. Indeks diskriminativnosti	27
Slika 2.2. Indeks težine.....	27
Slika 2.3. Pitanje 13.....	29
Slika 3.1. Raspodjela performansi prema godinama iskustva u programiranju	31
Slika 3.2. Raspodjela performansi po razredima u III. gimnaziji Split	32
Slika 3.3. Performanse po spolu	32
Slika 3.4. Rezultati na prijediplomskoj razini	33
Slika 3.5. Rezultati na prvoj godini diplomskog studija.....	33
Slika 3.6. Rezultati na drugoj godini diplomskoj studija	33
Slika 3.7. Veza uspješnosti na testu i godina učenja programiranja.....	34
Slika 3.8. Veza uspješnosti na testu i samoprocjene znanja.....	34
Slika 3.9. Povezanost uspješnosti na testu, godina učenja i procjene sposobnost.....	34
Slika 3.10. Provjera unidimenzionalnosti.....	35
Slika 3.11. Procjena parametara težine.....	35
Slika 3.12. Pitanje 2.....	36
Slika 3.13. Krivulje karakteristika pojedinačnih stavki.....	38
Slika 3.14. Krivulje karakteristika svih stavki.....	39
Slika 3.15. Pitanje 9.....	40

Slika 3.16. Person Item Map	41
Slika 3.17. Partial Credit Model za prvih nekoliko pitanja	42
Slika 3.18. Partial Credit Model - Pitanje 13.....	42
Slika 3.19. Partial Credit Model - Pitanje 8.....	43
Slika 3.20. Item Fit Statistike	44
Slika 3.21. Item Information Curve.....	45
Slika 3.22. Test Information Curve	46
Tablica 1.1. Primjeri uzročno-posljedičnih veza	5
Tablica 2.1. Uspješnost po pitanjima.....	28

Prilozi

Prilog 1. Test računalnog razmišljanja

Pitanje 1. Koje instrukcije označavaju istaknuti put kojim će pac-man doći do duha? *



Označite samo jedan oval.

idi naprijed

okreni udesno ↻

idi naprijed

idi naprijed

okreni udesno ↻

idi naprijed

idi naprijed

A)

B)

idi naprijed

okreni ulijevo ↻

idi naprijed

idi naprijed

idi naprijed

okreni ulijevo ↻

idi naprijed

C)

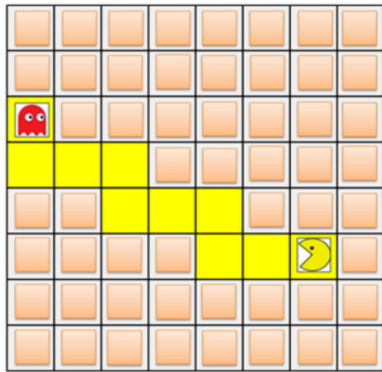
D)

E) Ništa od navedenog

Pitanje 2. Koliko puta treba ponoviti navedene instrukcije da pac-man dođe do duha? *

```

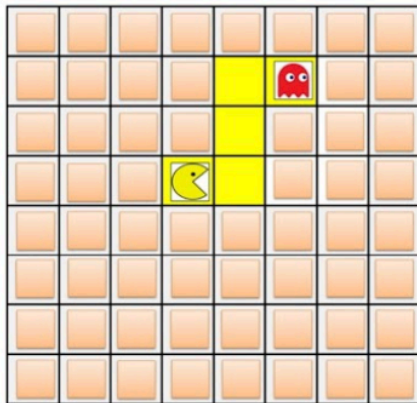
idi naprijed
idi naprijed
okreni udesno ↻
idi naprijed
    
```



Označite samo jedan oval.

- 2
- 3
- 4
- 5
- Ne može doći do duha

Pitanje 3. Pac-man istaknutim putem želi doći do duha. S desne strane su prikazani njegovi koraci. U kojem koraku je greška? *



```

idi naprijed → Korak A
okreni ulijevo ↻ → Korak B
idi naprijed
idi naprijed → Korak C
okreni ulijevo ↻ → Korak D
idi naprijed
    
```

Označite samo jedan oval.

- A
- B
- C
- D
- Nema greške

Pitanje 4. Ana ima rolu trake. Izrezala je dva komada trake i dobila prikazane komade. *
 Koji će oblik Ana dobiti kada izreže sljedeći komad?



Označite samo jedan oval.



A)

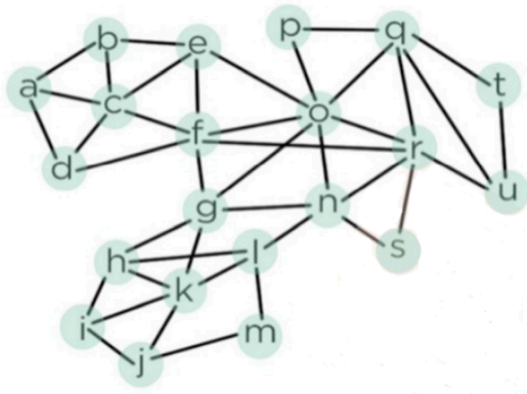
B)



C)

D)

Pitanje 5. Prikazana je karta zračnih luka i veza među njima. Širi se virus, koje tri zračne luke bi zatvorio/la da osiguraš najmanju zarazu?



Označite samo jedan oval.

- A) o n r
- B) f o n
- C) g o r
- D) o g k
- E) Ništa od navedenog

Pitanje 6. Hana ima vrt u kojem svake noći procvjetaju tri nova cvijeta, a jedan uvene. *

Danas u vrtu cvjetaju četiri cvijeta. Kada će Hana moći odjednom ubrati 15 cvjetova u vrtu?



Označite samo jedan oval.

- A) Nakon 4 dana
- B) Nakon 5 dana
- C) Nakon 6 dana
- D) Nakon 7 dana
- E) Ništa od navedenog

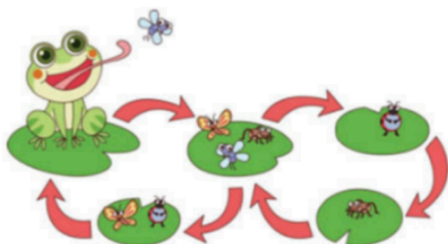
Pitanje 7. Jučer je bio ponedjeljak. Koji dan je četiri dana nakon sutra? *

Označite samo jedan oval.

- A) Ponedjeljak
- B) Nedjelja
- C) Petak
- D) Subota
- E) Ne znam.

Pitanje 8. Žabica može samo skakati između lopočevih listova duž crvenih strelica. Kada skoči, pojede bubicu gdje sleti. Može pojesti samo jednu bubicu svaki put.

Koliko najmanje puta treba skočiti da bi pojela sve bubice na lopočevim listovima?



Označite samo jedan oval.

- A) 10
- B) 9
- C) 5
- D) 8
- E) Ništa od navedenog

Pitanje 9. Prikazana je tablica s redcima i stupcima. U tablici su neka polja crna, a neka bijela. Zamislite da prvo zamijenimo stupac 4 sa stupcem 2. Nakon toga zamijenimo redak 2 s redkom 4. Koji će biti rezultat?

		Stupac			
		1	2	3	4
Red	1		■		
	2			■	
	3	■			■
	4		■		■

Označite samo jedan oval.

		1	2	3	4
1					
2				■	
3	■	■		■	
4			■		

A)

		1	2	3	4
1					■
2			■		■
3	■	■			
4			■		

B)

		1	2	3	4
1			■		■
2					
3	■				
4		■			■

C)

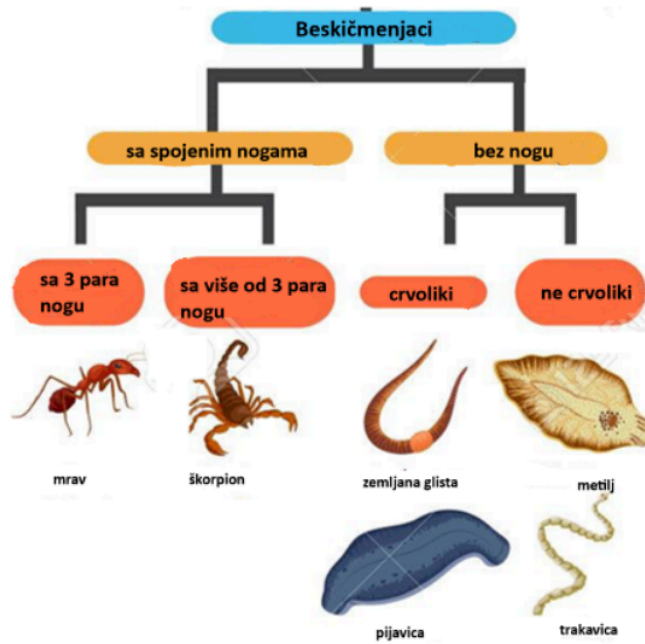
		1	2	3	4
1			■		■
2				■	
3	■				
4		■			■

D)

Pitanje 10. Zamisli da netko slučajno odabere jednu od ovih životinja, a ti trebaš pogoditi o kojoj životinji je riječ. Smiješ postavljati samo pitanja na koja se može odgovoriti s "da" ili "ne". *

Primjer dobrog pitanja je, na primjer: "Ima li zamišljena životinja noge?"

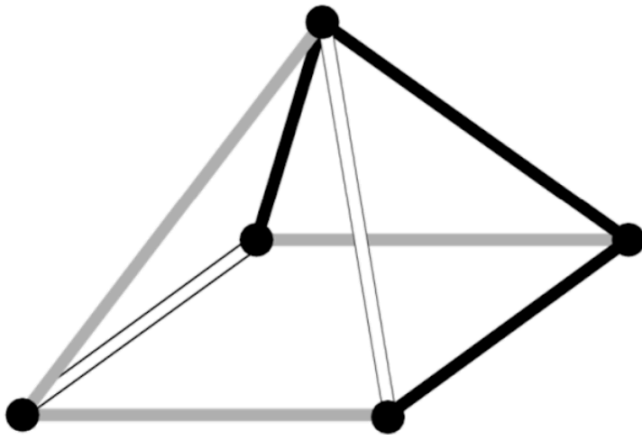
Zamislimo da koristiš optimalnu strategiju, gdje postavljaš najmanji broj pitanja kako bi pogodio životinju. Koji je prosječan broj pitanja koje trebaš postaviti?



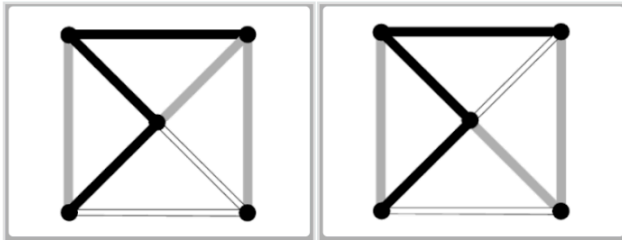
Označite samo jedan oval.

- A) 2.3
- B) 2.6
- C) 3
- D) 6
- E) Ne znam

Pitanje 11. Što bismo vidjeli kada bismo objekt sa slike pogledali odozgo? *

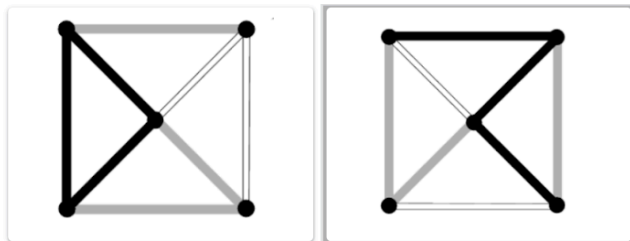


Označite samo jedan oval.



A)

B)

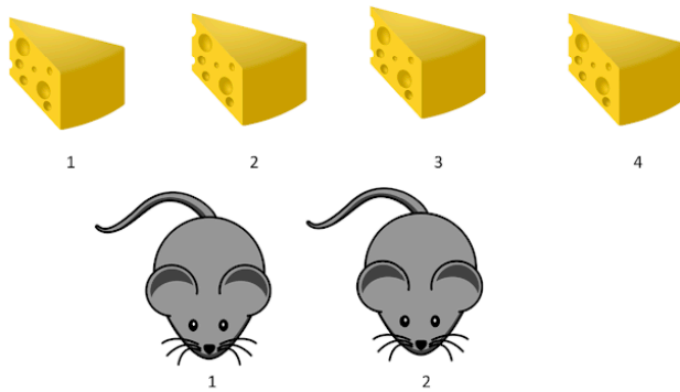


C)

D)

E) Ne znam.

Pitanje 12. Imate četiri različita sira, a jedan od njih sadrži bakteriju bez mirisa i okusa. * Taj sir uzrokuje simptome trovanja hranom 24 sata nakon konzumiranja. Želite saznati koji je sir zaražen, pa odlučite koristiti miševe koji će probati sireve. Međutim, u vašem laboratoriju imate samo dva miša, a rok za pronalazak zaraženog sira je točno 24 sata od sada. Možete dodijeliti svakom mišu koliko god i koji god sir želite. Kako biste postupili, kojem mišu bi dodijelili koji sir ili sireve?



Odaberite sve točne odgovore.

	Sir 1	Sir 2	Sir 3	Sir 4	Ne znam
Miš 1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Miš 2	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Pitanje 13. Dobili ste 9 novčića iste vrijednosti, ali jedan od njih je lažan. Lažni novčić * je lakši od ostalih.

Za vaganje novčića imate vagu kao na slici, a svako vaganje može rezultirati "vaga se naginje ulijevo", "vaga se naginje udesno" ili "vaga ostaje stabilna".

Pretpostavljajući da važete najmanji potreban broj puta, koliko vaganja je potrebno da biste identificirali lažni novčić?



Označite samo jedan oval.

- A) 1 vaganje
- B) 2 vaganja
- C) 3 vaganja
- D) 9 vaganja
- E) Ne znam

Skraćenice

STEM	Science, Technology, Engineering, and Mathematics
PISA	Programme for International Student Assessment
CT	Computational thinking
CTT	Classical Test Theory
IRT	Item Response Theory
1PL	One Parameter Logistic Model
2PL	Two Parameter Logistic Model
3PL	Three Parameter Logistic Model
ICC	Item Characteristic Curve
IIC	Item Information Curve
TIC	Test Information Curve
PCM	Partial Credit Model
IRF	Item Response Function
