

IZRADA PLATFORME ZA PRIBLIŽAVANJE STROJNOG UČENJA SREDNJOŠKOLSKOJ POPULACIJI

Plašć, Manuel

Undergraduate thesis / Završni rad

2023

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Algebra University College / Visoko učilište Algebra**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:225:325707>

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

Download date / Datum preuzimanja: **2024-11-09**



Repository / Repozitorij:

[Algebra University - Repository of Algebra University](#)



VISOKO UČILIŠTE ALGEBRA

ZAVRŠNI RAD

**IZRADA PLATFORME ZA PRIBLIŽAVANJE
STROJNOG UČENJA SREDNJOŠKOLSKOJ
POPULACIJI**

Manuel Plašč

Zagreb, rujan 2022.

Predgovor

Prije svega želio bih se zahvaliti svojim roditeljima koji su nesebično odvojili svoje vrijeme i svoj novac kako bi meni omogućili ugodan i bezbrižan život, a kasnije i mogućnost studiranja na Visokom učilištu Algebra.

Isto tako, želio bih se zahvaliti svim profesorima na Visokom učilištu Algebra koji su mi pomogli u sazrijevanju i razvoju mojih vještina.

Sažetak

Platforma će omogućiti korisnici da kroz zabavan i jednostavan način nauče o temama iz strojnog učenja. Korisnicima će biti prezentirana izabrana tema iz područja strojnog učenja i zatim će morati točno odgovoriti na pitanja iz izabrane teme kako bi mogli nastaviti na ostale teme. Nakon svakog uspješno završenog poglavlja korisnici će dobiti određenu sumu virtualne valute koju će moći iskoristiti u trgovini platforme. Trgovina platforme sadržavat će kozmetičke stavke koje će mijenjati izgled avatara korisnika.

Ključne riječi: strojno učenje, srednjoškolci, platforma za učenje

Sadržaj

1. Uvod	1
2. Projektni zadatak	2
2.1. Opis projekta	2
2.2. Igrifikacijski elementi.....	3
3. Izabrane teme strojnog učenja	4
3.1. Regresija	4
3.1.1. Jednostavna linearna regresija	4
3.1.2. Višestruka linearna regresija	7
3.1.3. Logistička regresija.....	9
3.2. Klasifikacija.....	12
3.2.1. Bayesov klasifikator	12
3.2.2. K-NN	15
3.2.3. Stablo odlučivanja	18
3.3. Grupiranje.....	20
3.3.1. Particionirano grupiranje	20
3.3.2. Grupiranje temeljeno na gustoći.....	22
3.3.3. Distribucijsko grupiranje	23
3.4. Umjetne neuronske mreže	25
3.5. Primjene strojnog učenja	27
4. Implementacija programskog rješenja.....	29
4.1. Arhitektura programskog rješenja	29
4.2. Dizajn programskog rješenja	30
5. Anketa zadovoljstva programskim rješenjem	43
5.1. Sadržaj ankete.....	43

5.2. Rezultati ankete i analiza	44
Zaključak	46
Popis kratica	49
Popis slika.....	50
Popis tablica.....	52
Popis kôdova	53
Literatura	54

1. Uvod

Strojno učenje i umjetna inteligencija nisu strani pojmovi za srednjoškolsku populaciju, no mnogi nisu sigurni što zapravo znače ove riječi i što se zapravo događa „iza scene“. Srednjoškolski obrazovni sustavi većinom nemaju strojno učenje kao obavezni predmet i zainteresirani srednjoškolci nemaju priliku produbiti svoje znanje u temi koja ih zanima. Možda im se strojno učenje čini zanimljivim iz razloga što ne znaju puno o samoj temi te ih privlači sama misterioznost teme. Isto tako u proteklih nekoliko godina često se vodi rasprava o umjetnoj inteligenciji i o zakonima koji bi je ograničili tako da strojno učenje dobiva dosta na publicitetu iz medija. Kad dolazi pitanje odabira profesije ili životnog puta, srednjoškolci su najugroženija grupa ljudi. Iako postoje različiti programi koje fakulteti provode da bi što bolje približili studijske programe, sama tema strojnog učenja u studijskim programima je relativno nova. Naravno, profesiju nije nemoguće promijeniti, no vrijeme prolazi i nema smisla trošiti dane na poslu koji jednostavno nije primjeren za određenog pojedinca. Strojno učenje i umjetna inteligencija nisu teme za svakoga i vrlo često ljudi nakon par godina uvide da to područje nije za njih, ali u tom trenutku nitko im ne može vratiti izgubljene godine.

Cilj ovog rada je izraditi interaktivnu platformu koja će srednjoškolcima, a i drugim dobnim skupinama omogućiti učenje osnovnih koncepata u strojnom učenju kroz jedan zabavan način. Platforma će omogućiti korisnicima, koji nemaju prethodnih znanja o strojnom učenju, da nauče na čemu se zasnivaju osnove strojnog učenja, a da se pri tome ne ulazi u dubinu algoritama korištenih u strojnom učenju te njihovih izračuna. Interaktivni elementi kao što su klizači i tablice omogućit će korisnicima promjene ulaznih podataka te će na taj način svatko moći eksperimentirati i učiti vlastitim tempom.

Rad je podijeljen na četiri poglavlja. U drugom poglavlju bit će objašnjen projektni zadatak te njegovi elementi. Kroz treće poglavlje objasnit će se teme strojnog učenja koje će platforma obrađivati. U četvrtom poglavlju bit će objašnjen programski kôd koji je korišten za izradu platforme, dizajn platforme te će se u petom poglavlju predstaviti analiza ankete zadovoljstva programskim rješenjem koju su ispunili korisnici sustava.

2. Projektni zadatak

2.1. Opis projekta

Primarni cilj predloženog sustava je da se kroz zabavan i lako shvatljiv način primarno srednjoškolcima, ali i ostalim dobnim grupama približi strojno učenje i umjetna inteligencija. Korisnici će kroz interaktivnu web platformu naučiti o osnovnim pojmovima i konceptima strojnog učenja.

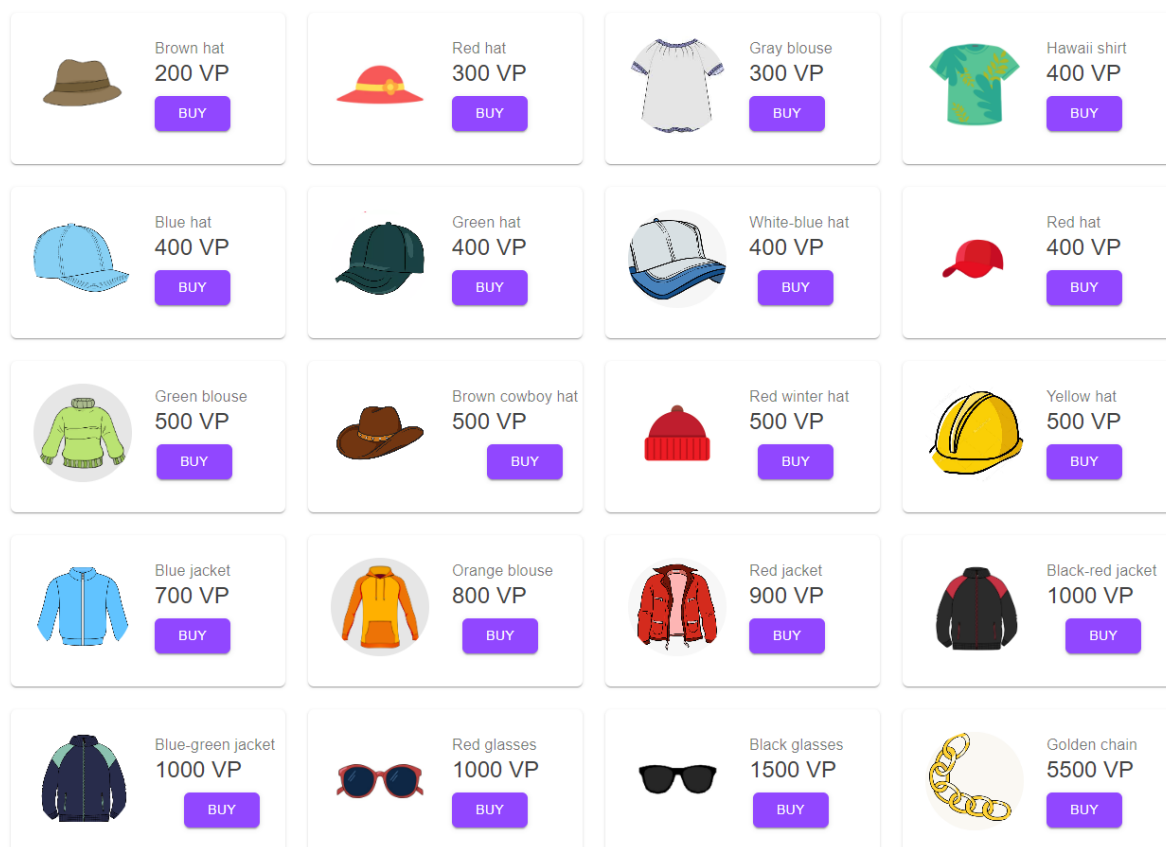
Platforma će se sastojati od dva ključna dijela, a to su lekcija i pitanje (dalje u tekstu kviz). Lekcije će biti grupirane u 3 poglavlja, a to su regresija, klasifikacija i grupiranje. Svako od poglavlja će zatim imati niz lekcija kroz koje će korisnici više naučiti o odabranoj temi. Nakon što korisnik odabere željeno poglavlje prikazat će mu se niz lekcija iz odabranog poglavlja. Ukoliko se korisnik tek registrirao sve lekcije osim prve bit će mu zaključane. Kako bi korisnik otključao sljedeću lekciju, prvo mora proći kroz prvu lekciju te točno odgovoriti na pitanje koje dolazi nakon svake lekcije. Ukoliko korisnik krivo odgovori na pitanje sustav ga automatski vraća na lekciju koja je vezana uz to pitanje. Po završetku kviza, ukoliko je korisnik točno odgovorio na pitanje, otključava se sljedeća lekcija. Korisnik se u bilo kojem trenutku može vratiti na prethodne lekcije te ih ponovno prolaziti.

VPoints je valuta platforme koju će korisnici moći iskoristiti u trgovini platforme za kupnju kozmetičkih stavki. Da bi dobili VPointse, korisnici moraju točno odgovoriti na kviz po završetku lekcije. Korisnici mogu samo jednom po lekciji osvojiti VPoints. To znači da korisnik ne može osvojiti VPointse ukoliko je uspješno prošao kviz te se kasnije ponovno vrati na isti. Uz to, postojat će sistem nagrada kako bi se motiviralo korisnike savladati zadane teme. Korisnici će nakon svakog uspješno završenog poglavlja dobiti određenu sumu virtualne valute koju će moći iskoristiti u trgovini platforme. Korisnici će u trgovini platforme moći kupiti razne odjevne dodatke koji će mijenjati izgled svog avatara.

Kako bismo imali metriku uspješnosti platforme, provest će se anketa među testnom grupom korisnika. Kroz anketu odgovorit će se na pitanja: jesu li teme objašnjene na kvalitetan način, što je dobro odrađeno na platformi, što se može poboljšati, je li ovaj način učenja lakši za svladavanje gradiva od tradicionalnog načina?

2.2. Igrifikacijski elementi

Platforma će sadržavati trgovinu u kojoj se mogu kupiti razne kozmetičke stvari koristeći VPointse. Razne stavke u trgovini omogućit će pojedincima koji su već prošli jedan dio ili cijela poglavlja da to na zanimljiv način pokažu drugim korisnicima. Na početku svi će korisnici dobiti iste avatare, no igranjem i kupovanjem stavki u internet trgovini moći će promijeniti izgled svojih avatara. Na osnovni avatar korisnici će moći dodati po jednu od kozmetičkih stavki, cijena će se mijenjati ovisno o tipu stavke. Isto tako, određene stavke u trgovini će se tek moći kupiti nakon što korisnik prođe kroz dovoljan broj lekcija. Kako bi avatari korisnika sa kupljenim kozmetički stavkama bili vidljivi svima, platforma će sadržavati top listu korisnika. Top lista korisnika će biti sortirana po broju uspješno završenih lekcija i prikazivat će korisnikovo ime uz avatara. Lekcija se smatra završenom kad korisnik uspješno odgovori na kviz nakon lekcije. Na slici Slika 2.1 prikazane su kozmetičke stavke iz trgovine platforme.



Slika 2.1 Primjeri stavki koje će se moći kupiti u trgovini platforme

3. Izabrane teme strojnog učenja

3.1. Regresija

3.1.1. Jednostavna linearna regresija

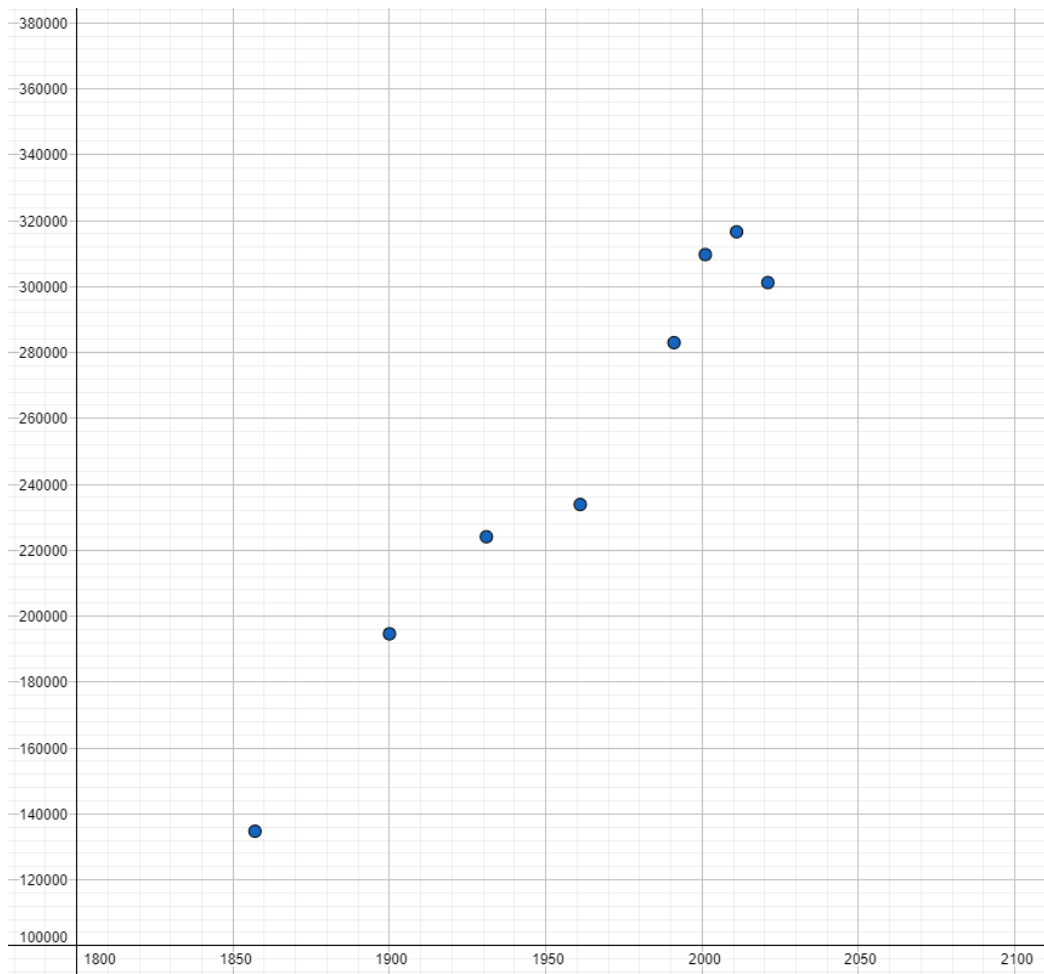
Kako je vidljivo iz [1], jednostavna linearna regresija je statistička tehnika koja nam omogućuje predviđanje vrijednosti zavisne varijable na temelju vrijednosti nezavisne varijable. Zavisna varijabla je varijabla koju pokušavamo predvidjeti, a nezavisna varijabla je varijabla koju koristimo za predviđanje. Ako želimo proučavati vezu između dvije varijable uzet ćemo uzorke podataka u obliku uređenih parova:

$$(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$$

Kao primjer za jednostavno linearnu regresiju koristit ćemo broj stanovnika u Zagrebačkoj županiji. Podaci su preuzeti sa [2] iz objavljenih rezultata popisa stanovništva te su upisani u tablicu Tablica 3.1. Uređene parove iz tablice zatim ćemo nacrtati u koordinatnom sustavu što možemo vidjeti na slici Slika 3.1.

	1857.	1900.	1931.	1961.	1991.	2001.	2011.	2021.
Zagrebačka županija	134 754	194 643	224 095	233 875	282 989	309 696	316 606	301 206

Tablica 3.1 Broj stanovnika u Zagrebačkoj županiji



Slika 3.1 Koordinatni sustav s ulaznim podacima

Za izradu jednostavnog linearnog regresijskog modela prvo moramo odabrati nezavisne i zavisne varijable. U ovom primjeru koristit ćemo godine kao nezavisnu varijablu, a broj stanovnika kao zavisnu varijablu. Zatim možemo stvoriti model linearne regresije koji izgleda ovako:

$$\text{broj stanovnika} = b_0 + b_1 * \text{godina}$$

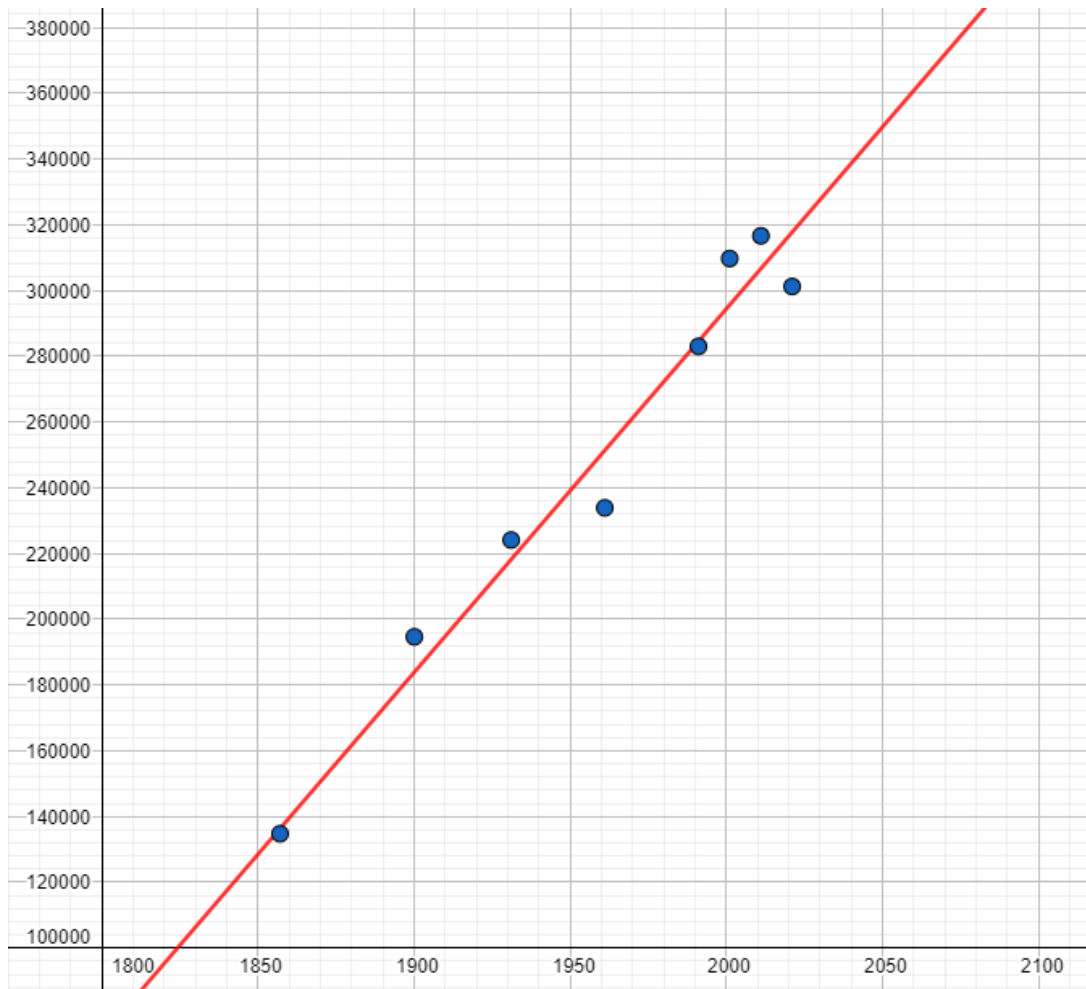
U ovoj jednadžbi, broj stanovnika je zavisna varijabla, godine su nezavisna varijabla, b_0 je presjek, a b_1 je nagib. Kako bismo olakšali razumijevanje linearne regresije možemo smatrati da je cilj ovog algoritma predvidjeti pravac, koji najviše odgovara našim ulaznim parovima, čija formula je oblika:

$$y = a * x + b$$

Nakon što imamo pravac, koji možemo vidjeti na slici Slika 3.2, možemo napraviti predviđanje broja stanovnika za neku godinu u budućnosti. Da bismo napravili predviđanje,

jednostavno uključimo godinu u jednadžbu. Na primjer, ako želimo predvidjeti broj stanovnika 2050. godine, stavit ćemo 2050 umjesto varijable godina, zatim bi jednadžba izgledala ovako:

$$\text{broj stanovnika} = b_0 + b_1 \cdot 2050$$



Slika 3.2 Koordinatni sustav s pravcem aproksimacije

Formula aproksimiranog pravca je:

$$y = 1\,079 * x - 1\,864\,987$$

Da bismo brzo i lako dobili jednadžbu pravca koristili smo online alat u koji smo upisali godine i broj stanovnika, a alat nam je na temelju unesenih vrijednosti generirao jednadžbu pravca. Radi lakšeg snalaženja dijelovi jednadžbe su obojani različitim bojama. Zelena boja predstavlja varijablu a u jednadžbi pravca koja odgovara varijabli b_1 u modelu linearne regresije, plava boja predstavlja varijablu x i u jednadžbi pravca i varijablu *godine* u modelu regresije te crvena boja predstavlja varijablu b u jednadžbi pravca te varijablu b_0 u modelu

regresije. Jedino što nam je preostalo je da zamijenimo b_0 sa presjekom pravca i b_1 sa nagibom pravca.

$$\text{broj stanovnika} = -1\,864\,987 + 1\,079 * 2\,050$$

$$\text{broj stanovnika} = 346\,963$$

Jednostavna linearna regresija nam omogućuje predviđanje vrijednosti na osnovu već prethodno upisanih vrijednosti. Na taj način prema prethodnim popisima stanovništva predvidjeli smo da će broj stanovnika 2050. godine iznositi 346 963.

3.1.2. Višestruka linearna regresija

Prema informacijama iz [3], višestruka linearna regresija je statistička tehnika koja se može koristiti za predviđanje budućih vrijednosti zavisne varijable, na temelju vrijednosti dviju ili više nezavisnih varijabli. Tehnika se koristi kada postoji linearni odnos između zavisne varijable i nezavisnih varijabli i kada odnos nije dobro prikazan jednom linijom. Višestruka linearna regresija može se koristiti za pronalaženje linije koja najbolje predstavlja odnos između zavisne varijable i nezavisnih varijabli. Formula koja opisuje višestruku linearnu regresiju je:

$$y = b_0 + b_1x_1 + b_2x_2 + b_3x_3 + \dots + b_px_p$$

Za primjer uzet ćemo vrijednost auta što će nam predstavljati zavisnu varijablu i za nezavisne varijable uzet ćemo starost auta i kilometražu. Prvo ćemo svaku nezavisnu varijablu posebno uvrstiti u koordinatni sustav, formule za izračun cijene bi bile:

$$\text{cijena} = b_0 + b_1 * \text{godina}$$

$$\text{cijena} = b_0 + b_1 * \text{kilometri}$$

Ovdje možemo primijetiti da su to zapravo dvije formule jednostavne linearne regresije, a kad kombiniramo dvije nezavisne varijable s istom zavisnom varijablom prema formuli višestruke linearne regresije ranije spomenute u radu dobivamo ovu formulu:

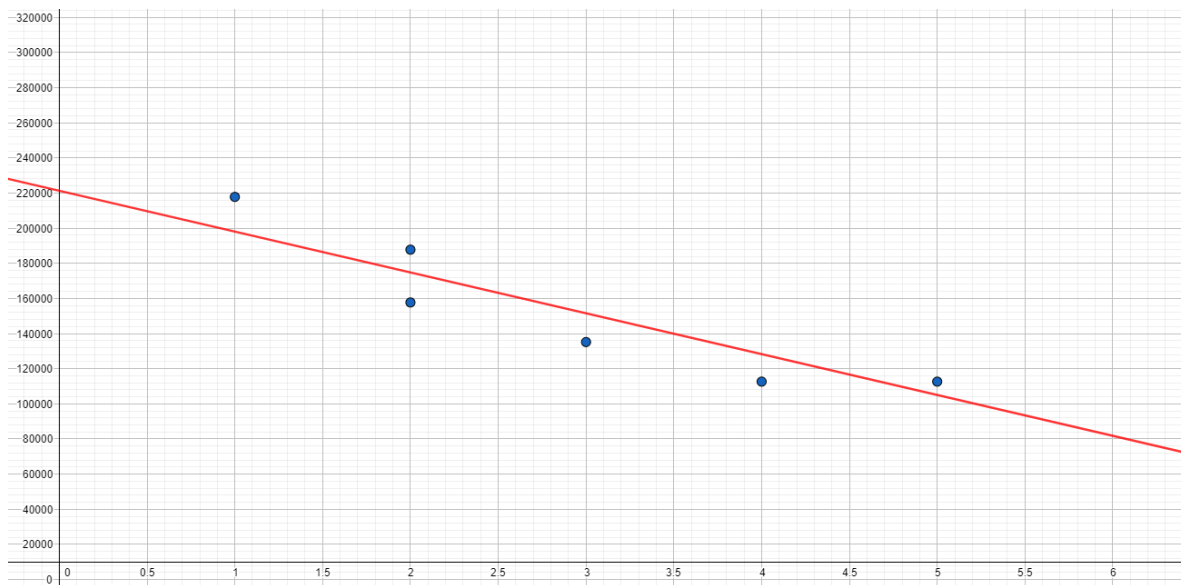
$$\text{cijena} = b_0 + b_1 * \text{godina} + b_2 * \text{kilometri}$$

Na temelju podataka iz tablice Tablica 3.2 izradit ćemo grafove kao na slikama 4 i 5 te ih upotrijebiti kako bismo predvidjeli cijenu automobila starog 7 godina koji je prošao 50 000 km.

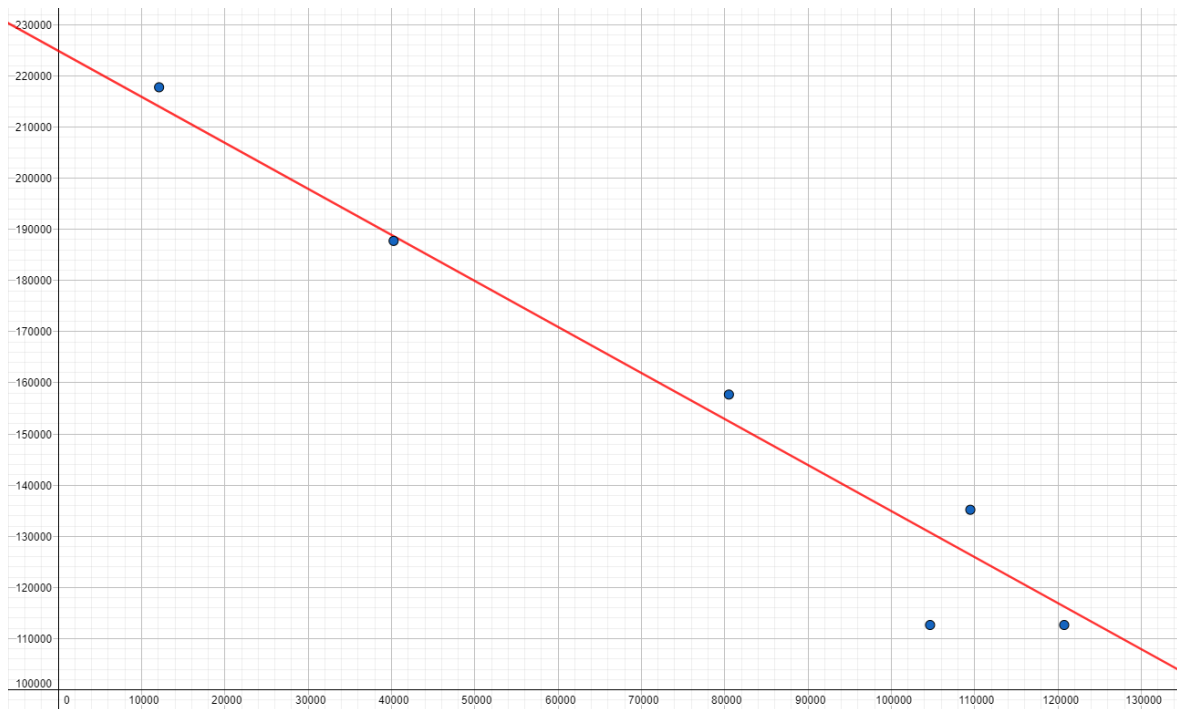
Cijena (kn)	Starost (godine)	Kilometraža (km)
-------------	------------------	------------------

217 790	1	28 980
187 750	2	40 250
157 710	2	80 500
135 180	3	109 480
112 650	4	120 750
112 650	5	104 650

Tablica 3.2 Cijena automobila s obzirom na starost i kilometražu



Slika 3.3 Koordinatni sustav s pravcem aproksimacije godine – cijena



Slika 3.4 Koordinatni sustav s pravcem aproksimacije kilometri – cijena

Formula pravca koji dobivamo višestrukom linearnom regresijom je:

$$cijena = 243\,769 - 11\,531 * godina - 0.7075 * kilometri$$

Da bismo brzo i lako dobili jednadžbu pravca koristili smo online alat u koji smo upisali godine, kilometre i cijenu, a alat nam je na temelju unesenih vrijednosti generirao jednadžbu višestruke linearne regresije. Jedino što nam je preostalo je unijeti starost od 7 godina i 50 000 umjesto kilometara

$$cijena = 243\,768 - 11\,531 * 7 - 0.7075 * 50\,000$$

$$cijena = 127\,676$$

3.1.3. Logistička regresija

Kako je navedeno u [4], logistička regresija je vrsta regresijske analize koja se koristi za predviđanje vjerojatnosti kategorički ovisne varijable. Kategorički ovisna varijabla je vrsta varijable čija vrijednost se može dodijeliti kategoriji. Na primjer, zanima nas je li učenik uspješno položio ispit. U ovom primjeru ne zanima nas koju je ocjenu učenik dobio. Učenik je uspješno položio ispit ako je više od 50% pitanja na ispitu točno odgovoreno. Odgovor na pitanje je li učenik položio test možemo kategorizirati u dvije kategorije:

- 50% i manje - nije položio ispit
- 51% i više – položio je ispit

Zavisna varijabla u logističkoj regresiji je binarna, što znači da može imati samo dvije vrijednosti, 0 ili 1. Koeficijenti modela logističke regresije mogu se tumačiti kao logaritamski izgledi zavisne varijable za jedinični porast nezavisne varijable.[4] Slična je linearnoj regresiji, ali se ishod predviđa pomoću logističke funkcije umjesto linearne funkcije. Formula krivulje korištene u logističkoj regresiji je:

$$f(z) = \frac{1}{1 + e^{-z}}$$

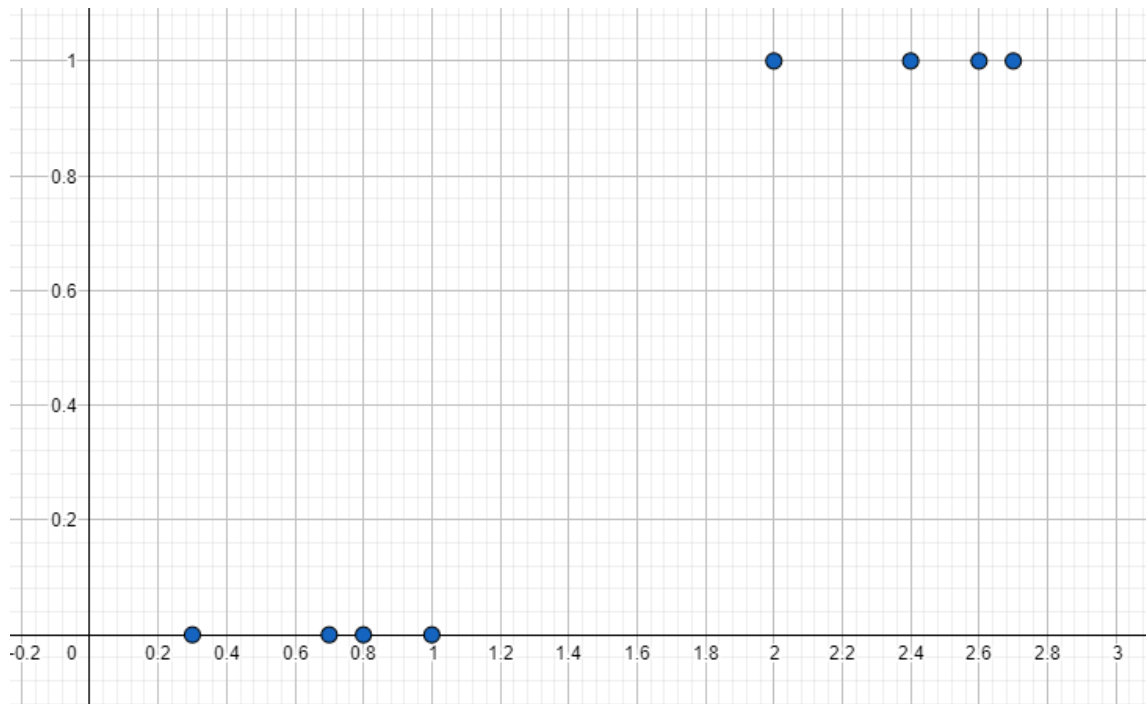
U ovoj formuli varijabla z predstavlja linearnu regresiju. Iz tog razloga formulu možemo modificirati i zamijeniti z sa formulom linearne regresije:

$$f(x) = \frac{1}{1 + e^{-(b_0 + b_1 * x)}}$$

Pošto logistička regresija može imati samo dva ishoda, a to su 0 ili 1, pokušat ćemo predvidjeti hoće li učenik proći ispit s obzirom na broj utrošenih sati na učenje. Ovdje se nećemo fokusirati na to koju je ocjenu učenik dobio, nego samo je li prošao ili pao ispit. U tablici Tablica 3.3 ćemo s 1 označiti učenike koji su prošli ispit, a s 0 one koji su pali ispit. U tablici su podaci o prolaznosti ispita s obzirom na vrijeme uloženo na učenje. Na slici Slika 3.5 podaci su prikazani u koordinatnom sustavu.

Vrijeme (sati)	Prolaznost (1 = prošao, 0 = pao)
0.7	0
1	0
0.3	0
0.8	0
2.4	1
2	1
2.6	1
2.7	1

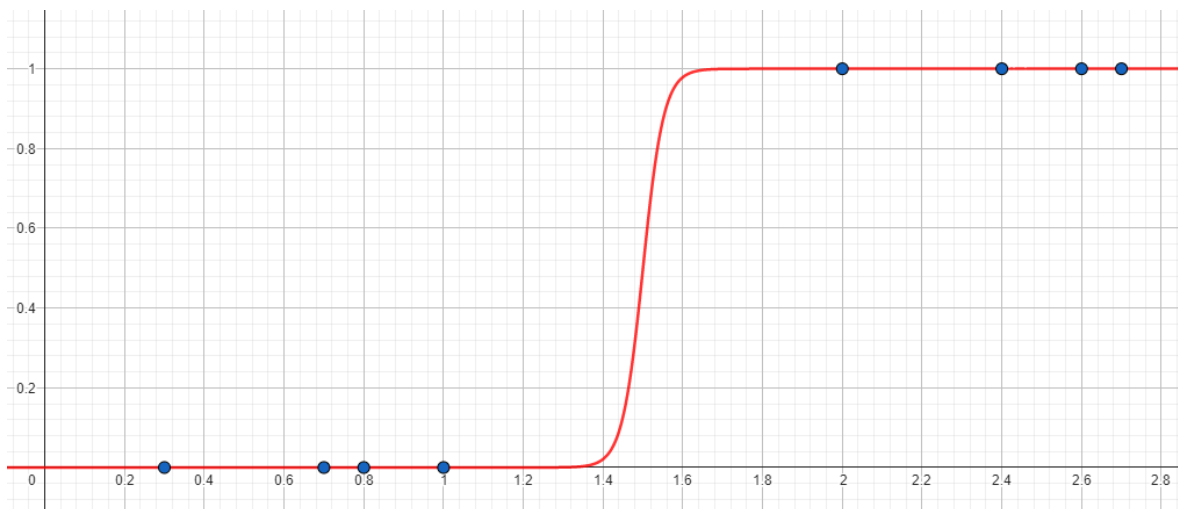
Tablica 3.3 Prikaz podataka o prolaznosti ispita



Slika 3.5 Koordinatni sustav sa podacima o prolaznosti ispita

Važno je da znamo s kakvim skupom podataka radimo jer u ovom slučaju je logistička regresija idealna za našu situaciju. Iz formule možemo pročitati da će naša funkcija izgledati kao slovo „S“ čije granice idu od 0 do 1 po y-osi. Da bismo brzo i lako dobili jednadžbu pravca koristili smo online alat u koji smo upisali vrijeme uloženo na učenje i prolaznost, a alat nam je na temelju unesenih vrijednosti generirao jednadžbu logističke regresije. Aproximirana funkcija vidljiva je na slici Slika 3.6 , a njezina formula je:

$$f(x) = \frac{1}{1 + e^{-(-57.51+38.33x)}}$$



Slika 3.6 Koordinatni sustav s funkcijom logističke regresije

Ukoliko želimo saznati hoće li učenik, koji je učio 3 sata, proći ispit samo moramo 3 sata uvrstiti u formulu:

$$y = \frac{1}{1 + e^{-(-57.51+38.33*3)}}$$

$$y = \frac{1}{1 + e^{-(57.48)}}$$

$$y = 1$$

Prema rezultatu formule možemo pretpostaviti da će učenik koji je učio 3 sata uspješno proći ispit. Što bi bilo da želimo predvidjeti hoće li učenik, koji je učio 1.55 sati (1 sat i 33 minute), proći ispit? Ako uvrstimo informacije u formulu:

$$y = \frac{1}{1 + e^{-(-57.51+38.33*1.55)}}$$

$$y = \frac{1}{1 + e^{-(1.9015)}}$$

$$y = 0.87$$

Dobili smo 0.87 što nije niti 1, niti 0, no to nije greška. S obzirom na ulazne podatke funkcija može biti strmija, ali uvijek će doći do problema ako je vrijednost predviđanja negdje u sredini s obzirom na ulazne podatke. Pošto smo dobili 0.87 možemo pretpostaviti, pošto je to više od 0.5, da će učenik ipak proći ispit. Na sličan način, kao što smo mi napravili pretpostavku, i računalo će napraviti pretpostavku, ali naravno puno je manja sigurnost u odnosu kad smo za rezultati dobili 1.

3.2. Klasifikacija

3.2.1. Bayesov klasifikator

Kako je vidljivo iz [5], Bayesov klasifikator algoritam je strojnog učenja koji koristi Bayesovo zaključivanje za predviđanje. Bayesova vjerojatnost je način kvantificiranja neizvjesnosti, temelji se na ideji da se vjerojatnost događaja temelji na njegovoj prethodnoj vjerojatnosti i dokazima koji su dostupni. Bayesov algoritam klasifikatora funkcionira kombiniranjem prethodne vjerojatnosti s dokazima kako bi se napravilo predviđanje. Prethodna vjerojatnost je vjerojatnost da će se događaj dogoditi bez ikakvih dokaza. Dokazi su informacije koje su dostupne o nekom događaju. Bayesov algoritam klasifikatora koristi

se u mnogim različitim područjima, kao što su medicina, financije i inženjerstvo. Formula za Bayesov klasifikator je:

$$P(c|x) = P(c) * \frac{P(x|c)}{P(x)}$$

gdje je c klasa, x je dokaz, $P(c)$ je prethodna vjerojatnost klase, $P(x|c)$ je vjerojatnost dokaza s obzirom na klasu, a $P(x)$ je prethodna vjerojatnost dokaz. Bayesov klasifikator daje predviđanja izračunavanjem zadnje vjerojatnosti svake klase s obzirom na dokaze. Klasa s najvećom zadnjom vjerojatnošću je predviđena klasa.

Recimo da želimo da računalo može klasificirati treba li čovjek uzeti kišobran sljedeći dan ako zna kakva će biti prognoza. U tablici Tablica 3.4 su podaci o tome jesmo li uzeli kišobran i kakvo je vrijeme bilo najavljeno za taj dan.

Prognoza	Kišobran
Kišovito	Da
Oblačno	Da
Sunčano	Ne
Kišovito	Da
Sunčano	Ne
Kišovito	Da
Oblačno	Ne
Oblačno	Da
Sunčano	Ne
Sunčano	Ne
Kišovito	Da
Kišovito	Ne
Sunčano	Da

Sunčano	Ne
Oblačno	Ne
Oblačno	Da
Sunčano	Ne
Kišovito	Da
Kišovito	Ne

Tablica 3.4 Prikaz podataka o prognozi i kišobranu

Da bismo mogli dalje nastaviti moramo napraviti frekvencijsku tablicu Tablica 3.5 koja će nam reći koliko se puta svaka situacija iz tablice Tablica 3.4 ponavlja.

Prognoza	Da	Ne
Sunčano	1	6
Oblačno	2	3
Kišovito	5	2
	8	11

Tablica 3.5 Tablica ovisnosti nošenja kišobrana o prognozi

Prije nego možemo računati koristeći formulu potrebno je još napraviti tablice Tablica 3.6 Tablica 3.7 vjerojatnosti događaja iz tablice Tablica 3.5

Prognoza	Vjerojatnost
Sunčano	$7/19 = 0.3684$
Oblačno	$5/19 = 0.2632$
Kišovito	$7/19 = 0.3684$

Tablica 3.6 Vjerojatnost prognoze prema ulaznim podacima

Kišobran?	Vjerojatnost
-----------	--------------

Da	8/19 = 0.4211
Ne	11/19 = 0.5789

Tablica 3.7 Vjerojatnost da ćemo trebati kišobran prema ulaznim podacima

Uzmimo za primjer rečenicu „Sutra će biti sunčano i trebam kišobran.“ U osnovnu formulu potrebno je uvrstiti informacije iz ove rečenice. Formula s podacima iz rečenice bi izgledala ovako:

$$P(c|x) = P(c) * \frac{P(x|c)}{P(x)}$$

$$P(Da|Sunčano) = P(Da) * \frac{P(Sunčano|Da)}{P(Sunčano)}$$

$$P(Da|Sunčano) = 0.4211 * \frac{0.125}{0.3684}$$

$$P(Da|Sunčano) = 0.1429$$

Prema izračunatim podacima vjerojatnost da će nam trebati kišobran ako će sutra biti sunčano je 0.1429 odnosno 14.29 %. Sad uzmimo za primjer „Sutra će biti kišovito i trebam kišobran“

$$P(Da|Kišovito) = P(Da) * \frac{P(Kišovito|Da)}{P(Kišovito)}$$

$$P(Da|Kišovito) = 0.4211 * \frac{0.625}{0.3684}$$

$$P(Da|Kišovito) = 0.7144$$

Algoritam smatra da je 71.44 % vjerojatnosti da će nam trebati kišobran ukoliko je sutrašnja prognoza kišovita. Što više podataka sustav ima, to lakše i preciznije može napraviti klasifikaciju.

3.2.2. K-NN

Prema informacijama iz [6], K-Algorithm najbližih susjeda (KNN) algoritam je strojnog učenja koji se koristi za klasifikaciju i regresiju. Algoritam funkcionira tako da pronalazi k najbližih susjeda dane podatkovne točke, a zatim koristi te susjede za predviđanje klase ili vrijednosti podatkovne točke. Za korištenje KNN-a za klasifikaciju, podaci se dijele na

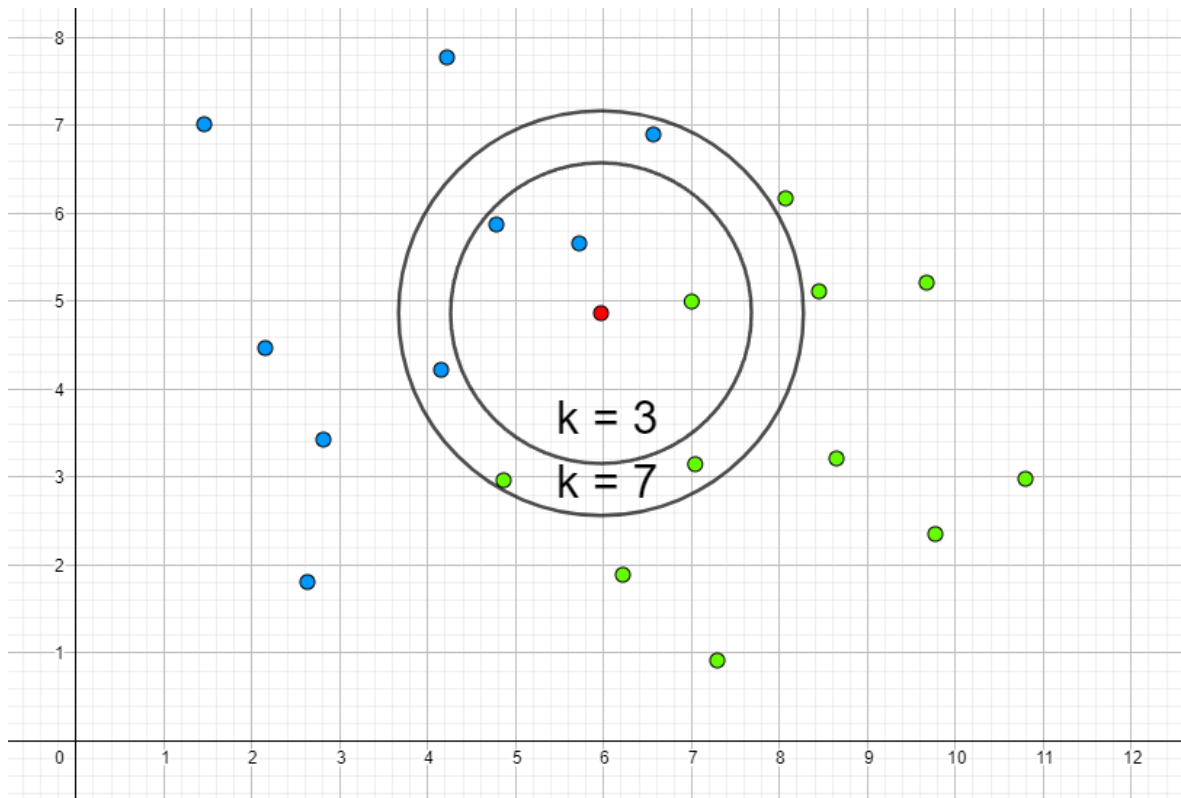
skupove za obuku i testove. Skup za obuku koristi se za obuku modela, a set za testiranje koristi se za procjenu modela.

Kako bi predvidio klasu podatkovne točke, KNN prvo izračunava udaljenost između podatkovne točke i svih točaka u skupu za obuku. Zatim pronalazi k točaka u skupu za obuku koje su najbliže podatkovnoj točki i koristi te točke za predviđanje klase podatkovne točke. KNN je relativno jednostavan algoritam, ali može biti vrlo učinkovit.[6]

Važno je odabrati dobru vrijednost za k , budući da će vrijednost koja je premala rezultirati prekomjernim uklapanjem, a vrijednost koja je prevelika će rezultirati nedostatkom. Objekt se klasificira većinom glasova njegovih susjeda, pri čemu se objekt dodjeljuje klasi koja je najčešća među njegovih k najbližih susjeda (k je pozitivan cijeli broj, obično mali). Ako je $k = 1$, tada se objekt jednostavno dodjeljuje klasi tog pojedinačnog najbližeg susjeda.

U regresiji K -najbližih susjeda, izlaz je vrijednost svojstva za objekt. Ova vrijednost je prosjek vrijednosti njegovih k najbližih susjeda.

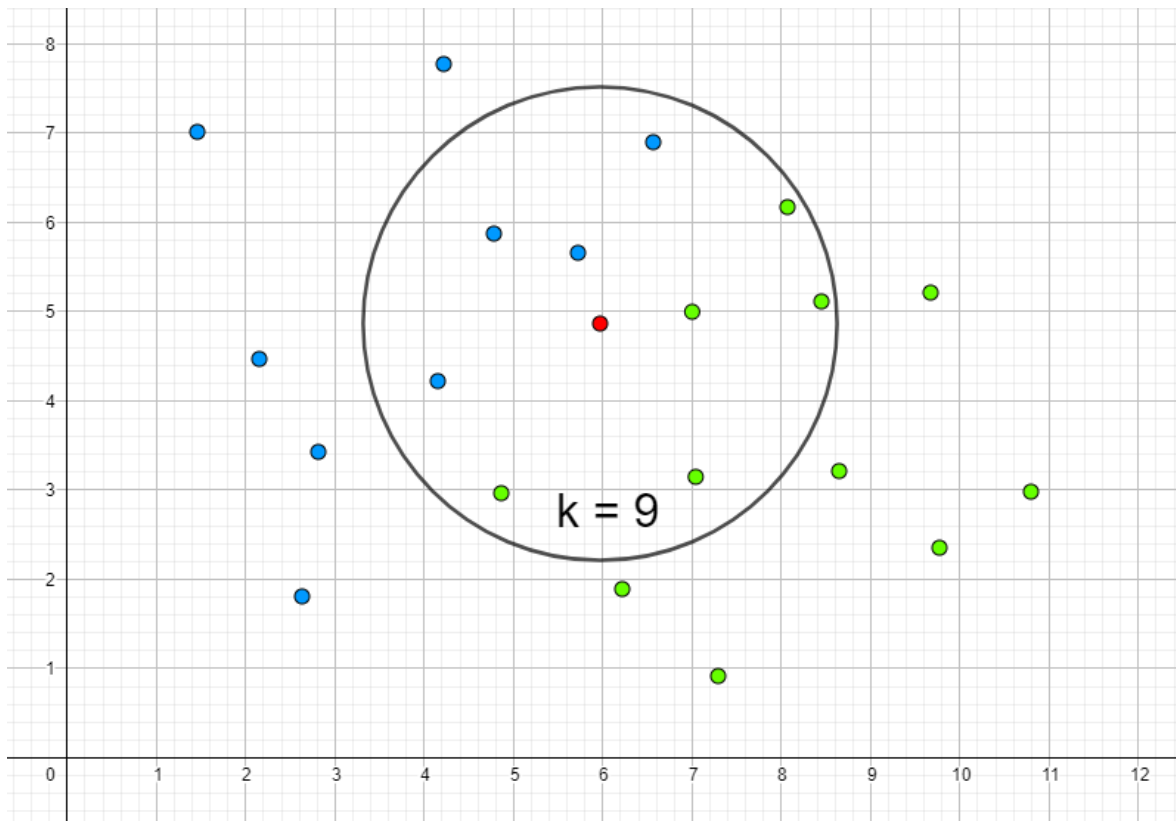
Kako bi lakše objasnili princip rada KNN algoritma, poslužiti ćemo se slikom koordinatnog sustava. Na slici Slika 3.7 su dvije grupe, jedna grupa je označena plavom bojom, a druga grupa je označena zelenom bojom. U koordinatni sustav dodat ćemo crveni objekt kojega želimo klasificirati, odnosno dodijeliti jednoj od grupa. Prvo moramo odabrati koliko će iznositi k , pravilo je ukoliko imamo paran broj grupa da odaberemo k neparan broj i obratno, ukoliko imamo neparan broj grupa odabiremo paran broj za k . Uzet ćemo $k = 3$ te pokušati klasificirati novo dodani objekt, a zatim ćemo probati sa $k = 7$.



Slika 3.7 Koordinatni sustav s dvije grupe objekata i pomoćnim kružnicama $k = 3$ i $k = 7$

Ukoliko odaberemo $k = 3$ to znači da uspoređujemo nepoznati objekt sa 3 najbliža susjeda. Od 3 susjeda 2 su plava i 1 je zeleni što znači da će crveni objekt biti klasificiran kao dio plave grupe. Na primjeru sa $k = 7$ od 7 najbližih susjeda 4 je plave boje te 3 je zelene boje. U ovom slučaju crveni objekt će isto biti klasificiran kao dio plave grupe. Za kraj, uzmimo primjer sa $k = 9$ koji je vidljiv na slici Slika 3.8 Iz slike možemo iščitati da od 9 najbližih susjeda 4 su plave boje te 5 zelene boje. Prema ovim podacima većina je susjeda zelene boje, stoga crveni kružić možemo dodijeliti zelenoj grupi.

Ovdje možemo primijetiti da za $k = 3$ i $k = 7$ dobili smo jedan rezultat, a za $k = 9$ smo dobili drugi rezultat. Ovo nam govori da je vrlo bitno koji se broj odabere za k te ukoliko odaberemo krivi broj postoji mogućnost krivog rezultata. Generalno pravilo je odabrati što manji broj, ali ne opet premali s obzirom na broj podataka. Naš skup podataka je imao 20 objekata te je naš idealni broj k neka manja brojka kao što su 3, 5 ili 7. Za primjere koji imaju više podataka odabrat ćemo veći k .



Slika 3.8 Koordinatni sustav s dvije grupe objekata i pomoćnom kružnicom $k = 9$

3.2.3. Stablo odlučivanja

Kako je navedeno u [7], algoritam stabla odlučivanja je algoritam koji se koristi za zadatke klasifikacije i regresije. Algoritam stvara stablo čvorova odluka, gdje svaki čvor predstavlja odluku koju je potrebno donijeti. Stablo se stvara počevši od korijenskog čvora, a zatim podijeli podatke u različite grupe na temelju vrijednosti značajki. Algoritam zatim nastavlja dijeliti podatke dok se svi podaci ne klasificiraju.

Stablo odlučivanja stvara se upotrebom skupa podataka za obuku za izradu modela koji se zatim može koristiti za predviđanje novih podataka. Postoji nekoliko različitih algoritama stabla odlučivanja, ali najčešći su CART (engl. *Classification and Regression Trees*) i ID3 (engl. *Iterative Dichotomiser 3*). CART je pohlepan algoritam koji stvara stablo dijeljenjem podataka na značajku koja daje najveće poboljšanje u točnosti. ID3 je rekurzivni algoritam koji stvara stablo rekurzivnim dijeljenjem podataka na značajku koja daje najveći dobitak informacija.[7]

Algoritam radi stvaranjem strukture nalik stablu, gdje svaki čvor predstavlja test na atributu, a svaka grana predstavlja ishod testa. Lišće stabla predstavlja oznake klase. Da bi klasificirao

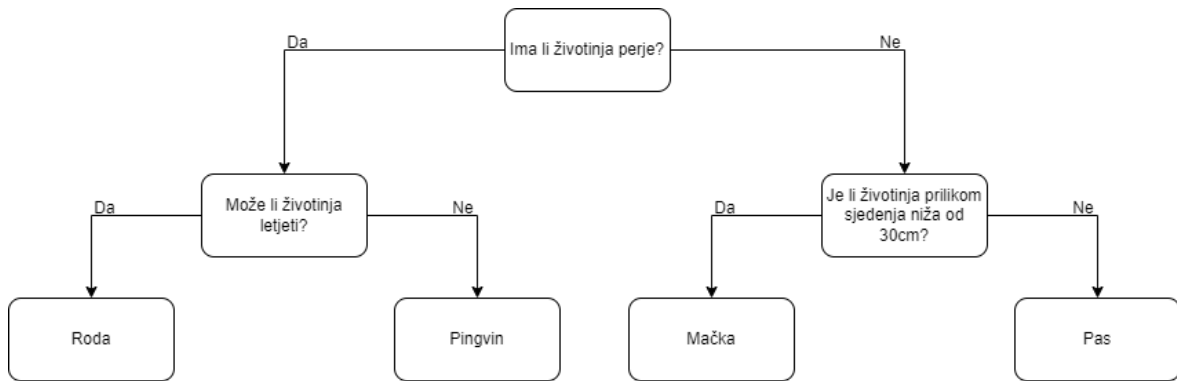
novu instancu, algoritam počinje od korijena stabla i ide niz stablo, praveći predviđanje na svakom čvoru, sve dok ne dođe do listnog čvora. Predviđanje je tada oznaka klase listnog čvora. Algoritam je jednostavan za razumijevanje i tumačenje, a može se koristiti za modeliranje složenih procesa odlučivanja.[7]

Za primjer imamo stablo odlučivanja na slici Slika 3.10. Ovim stablom odlučivanja možemo utvrditi o kojoj se životinji radi na temelju postavljenih pitanja. Na slici Slika 3.9 je nepoznata životinja koju želimo identificirati.



Slika 3.9 Nepoznata životinja za identifikaciju

Krećemo od pitanja „Ima li životinja perje?“, odgovor je „Ne“. Ovo je ishodišni čvor odnosno korijenski čvor stabla. Zatim se spuštamo niz stablo i dolazimo do novog čvora i taj postupak ponavljamo sve dok ne dođemo do lista stabla, odnosno do čvora koji nema pitanja, nego je zapravo naziv klase. Sljedeće pitanje je „Je li životinja kad prilikom sjedenja niža od 30 cm?“ i odgovor je „Da“. Nakon toga se spuštamo do lista te saznajemo da je nepoznata životinja „Mačka“. Ovo je vrlo pojednostavljeno stablo odlučivanja, ali na isti način funkcionira kad računalo pokušava klasificirati dvije ili više grupa podataka.



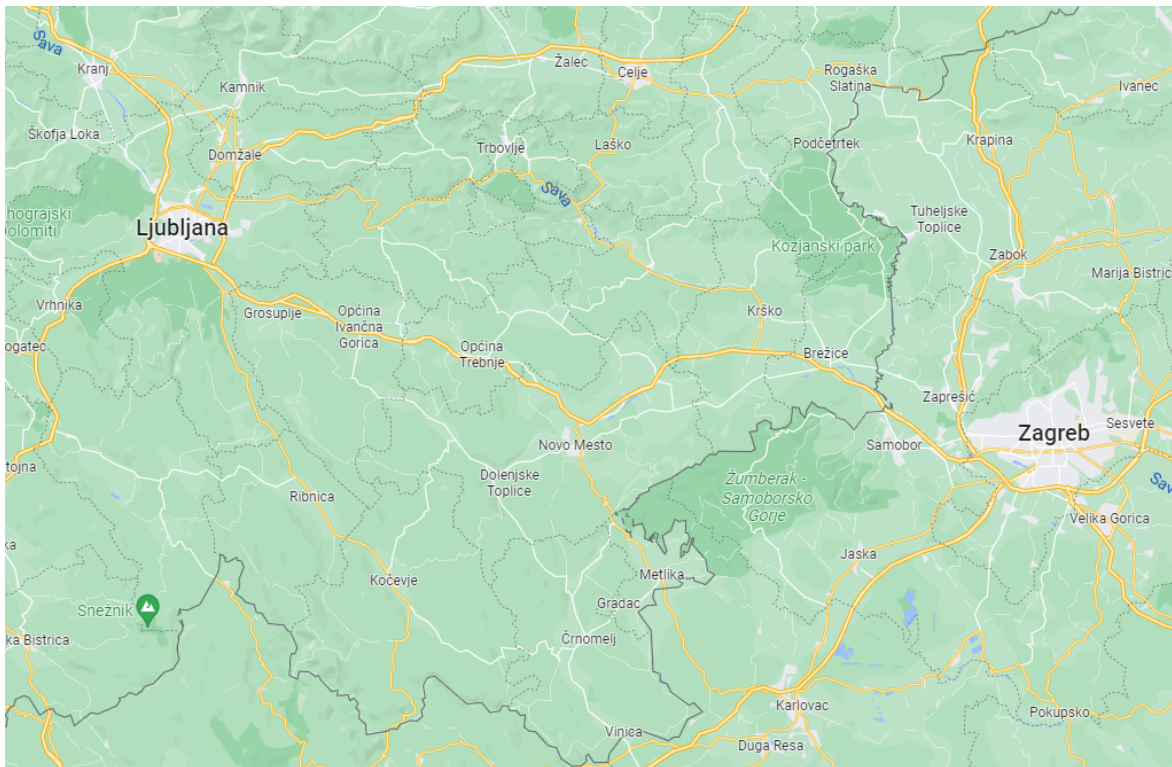
Slika 3.10 Stablo odlučivanja

3.3. Grupiranje

3.3.1. Particionirano grupiranje

Kako je navedeno u [8], particionirano grupiranje je tehnika grupiranja podataka koja se koristi za poboljšanje performansi operacija obrade podataka. Algoritam funkcionira tako da podatke podijeli u skupine prema određenim kriterijima, a zatim svaku skupinu zasebno obrađuje. To može dovesti do značajnih poboljšanja performansi, budući da se podaci mogu obrađivati paralelno, a cjelokupni proces može se dovršiti mnogo brže. Particija po grupiranju algoritam je grupiranja koji grupira podatkovne točke u particije na temelju njihove sličnosti. Algoritam se koristi za pronalaženje prirodnih grupiranja u podacima, kao što je grupiranje kupaca prema njihovim preferencijama ili grupiranje gena prema njihovoj funkciji. Algoritam funkcionira tako da prvo izračuna sličnost između svih parova podatkovnih točaka. Zatim grupira podatkovne točke koje su slične zajedno u particije. Algoritam se zaustavlja kada su sve podatkovne točke u particiji. Particija grupiranjem moćan je alat za pronalaženje strukture podataka. Može se koristiti za pronalaženje skupina kupaca sa sličnim preferencijama, skupina gena sa sličnim funkcijama ili bilo koje druge skupine točaka podataka sa sličnim karakteristikama.

Da bismo lakše objasnili na koji način radi particionirano grupiranje, za primjer ćemo uzeti kartu granice Hrvatske sa Slovenijom koja je vidljiva na slici Slika 3.11. Particionirano grupiranje grupira podatke po određenim kriterijima odnosno sličnostima koje podaci imaju međusobno.

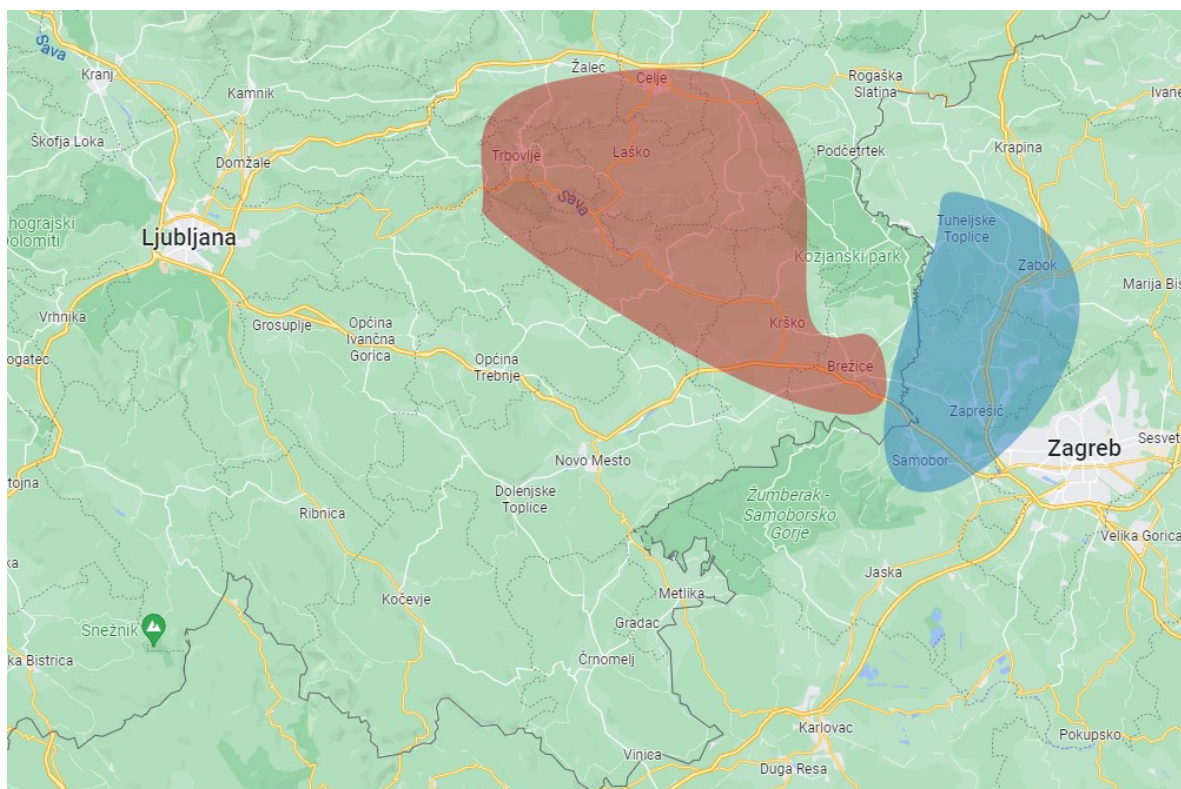


Slika 3.11 Karta Hrvatske i Slovenije preuzeta sa Google Maps

Ukoliko su naši ulazni podaci gradovi: Samobor, Zaprešić, Brežice, Krško, Zabok, Tuhejske Toplice, Trbovlje, Laško, Celje. Partitionirano grupiranje, ako mu damo za kriterij država u kojoj se gradovi nalaze te da moraju biti 2 grupe, grupirat će gradove na sljedeći način:

- Grupa 1: Samobor, Zaprešić, Zabok, Tuhejske Toplice
- Grupa 2: Brežice, Krško, Trbovlje, Laško, Celje

Na slici Slika 3.12 prikazane su grupe nastale partitioniranim grupiranjem po kriteriju pripadnosti državi.



Slika 3.12 Prikaz grupa nastalih particioniranim grupiranjem

3.3.2. Grupiranje temeljeno na gustoći

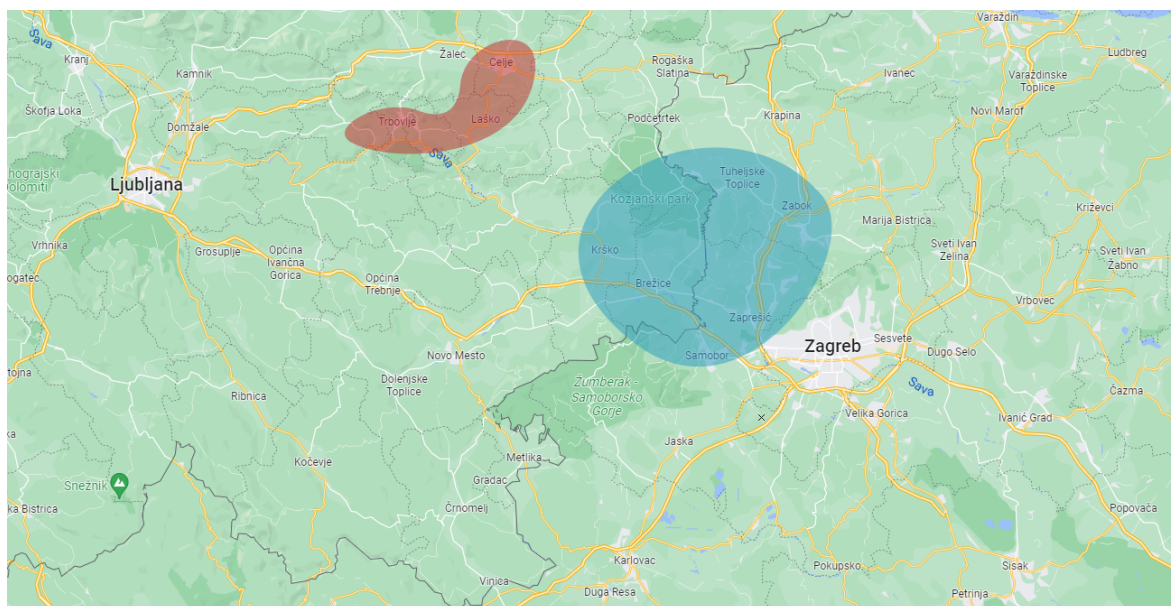
Prema informacijama iz [9], grupiranje temeljeno na gustoći je tehnika grupiranja podataka koja se koristi za pronalaženje područja visoke gustoće u skupovima podataka. Ova se tehnika koristi za pronalaženje grupa podatkovnih točaka koje su blizu jedna drugoj u prostoru. Ovaj se algoritam temelji na ideji gustoće, što je mjera koliko se točaka nalazi u određenom području. Algoritam grupiranja temeljen na gustoći funkcionira tako da prvo identificira podatkovnu točku koja se nalazi u području visoke gustoće. Ta se podatkovna točka zatim koristi za pronalaženje drugih podatkovnih točaka koje se nalaze u istom području visoke gustoće. Te se podatkovne točke zatim grupiraju zajedno. Ovaj se algoritam obično koristi za skupove podataka koji nisu dobro definirani i gdje grupe nije lako identificirati. Jedan od korištenijih algoritama za grupiranje temeljeno na gustoći je DBSCAN. Algoritam modelira grupe kao područja visoke gustoće odvojena područjima niske gustoće. DBSCAN je algoritam za grupiranje temeljen na gustoći koji je posebno prikladan za otkrivanje grupa proizvoljnog oblika. Algoritam funkcionira tako da identificira točke koje su u neposrednoj blizini jedna drugoj (tj. točke visoke gustoće) i grupira ih

zajedno u grupu. Točke koje nisu u neposrednoj blizini bilo koje druge točke smatraju se šumovima i ne dodjeljuju se grupi.

Za primjer koristi ćemo sliku Slika 3.11. Ulazni podaci će biti: Samobor, Zaprešić, Brežice, Krško, Zabok, Tuheljske Toplice, Trbovlje, Laško, Celje. Algoritam za grupiranje temeljeno na gustoći napraviti će dvije grupe:

- Grupa 1: Samobor, Zaprešić, Brežice, Krško, Zabok, Tuheljske Toplice
- Grupa 2: Trbovlje, Laško, Celje

Rezultati grupiranja su vidljivi na slici Slika 3.13. Možemo uočiti kako je algoritam grupirao one gradove koji se nalaze u blizini jedni drugih u odnosu na particionirano grupiranje koje je grupiralo gradove po državi u kojoj se gradovi nalaze.



Slika 3.13 Prikaz grupa nastalih grupiranjem temeljenom na gustoći

3.3.3. Distribucijsko grupiranje

Postoji nekoliko različitih algoritama koji se mogu koristiti za distribucijsko grupiranje, ali jedan od najčešćih je algoritam k-srednjih vrijednosti. Ovaj algoritam grupira podatkovne točke zajedno minimizirajući zbroj kvadrata unutar grupe. Kako je navedeno u [10], algoritam pokušava grupirati podatkovne točke zajedno tako da je zbroj kvadrata udaljenosti između podatkovnih točaka i grupnog težišta minimiziran. To se može učiniti nasumičnim pokretanjem težišta k skupine i zatim dodjeljivanjem svake podatkovne točke grupi s

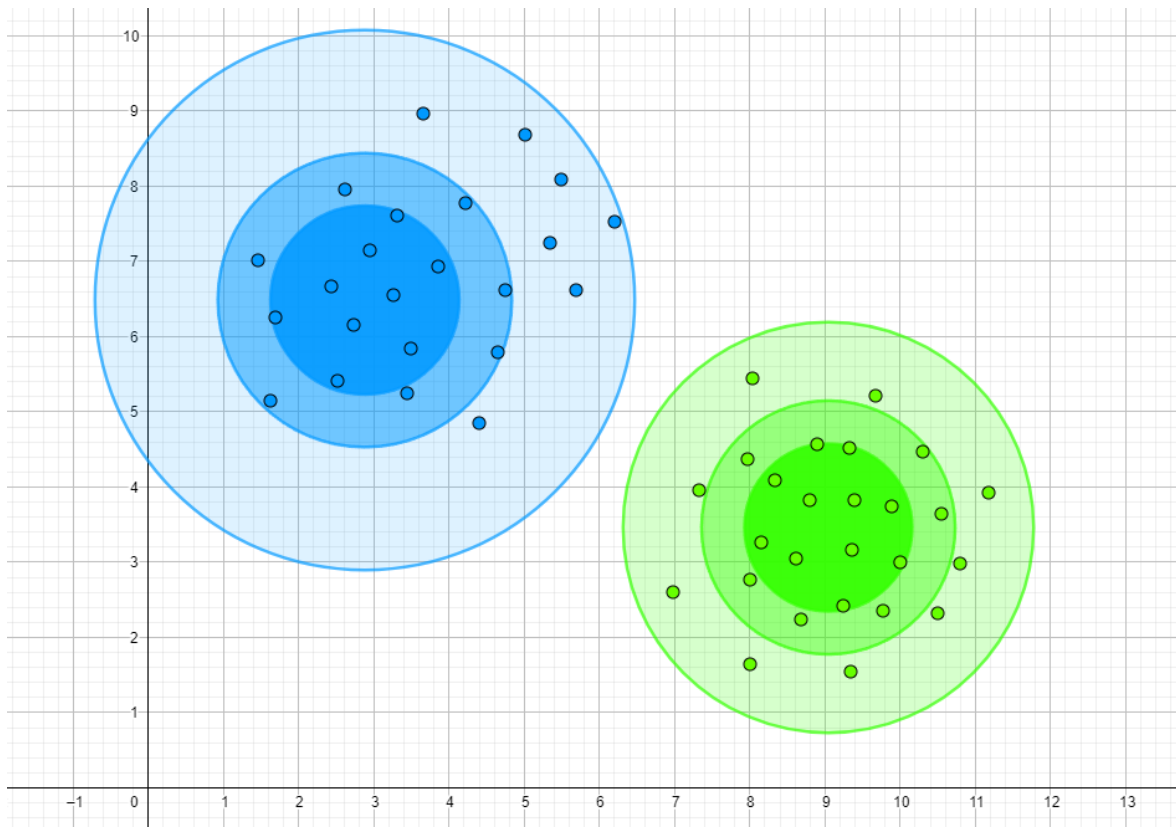
najbližim težištem. Težišne točke se zatim ažuriraju kako bi bile srednja vrijednost podatkovnih točaka u skupini. Ovaj se proces ponavlja dok se težišnice ne konvergiraju ili dok se ne postigne maksimalan broj ponavljanja. Gaussov model mješavine (GMM) jedna je od vrsta grupiranja temeljenog na distribuciji. Ovaj pristup grupiranja pretpostavlja da su podaci sastavljeni od distribucija, kao što su Gaussove distribucije. Distribucijsko grupiranje bi se trebalo koristiti samo kad smo sigurni da imamo distribucijsku podjelu ulaznih podataka, u suprotnom slučaju dobit ćemo krive rezultate.

Distribucijsko grupiranje radi na principu kolika je vjerojatnost da je neki objekt dio te grupe. Na slici Slika 3.14 prikazane su dvije grupe. Svaka grupa sadrži tri različite nijanse boje, svaka od tih nijansi reprezentira vjerojatnost pripadnosti grupi. Što je neki objekt dalje od središta grupe to je manja vjerojatnost da je objekt zapravo dio te grupe.

Velike tvrtke često koriste distribucijsko grupiranje kad žele grupirati kupce s obzirom na njihovu dob, primanja, ponašanje na internetu, potrošačke navike itd. Iz ovih informacija možemo napraviti 3 skupine i strategije za marketing tim grupama.

- Prva skupina: mladi kupci s malim primanjima – ovoj grupi kupaca će se nuditi proizvodi koji nemaju preveliku cijenu te su popularni u ovoj dobnoj skupini, to mogu npr. biti tenisice, hlače, majice
- Druga skupina: kupci srednje dobi sa srednjim primanjima – ovoj grupi kupaca nudit će se kućanski aparati, vozila, te skuplja odjeća i obuća
- Treća skupina: kupci starije dobi s većim primanjima – ovoj grupi kupaca nudit će se imovina, skuplja vozila, razne kućanske potrepštine

Algoritam distribucijskog grupiranja će svrstati kupce u ove tri skupine te im se na temelju njihove skupine mogu prikazivati automatizirani oglasi ili se te informacije mogu proslijediti agentima koji telefonom zatim nazivaju kupce i nude im proizvode koji su najčešće kupljeni u njihovoj skupini.



Slika 3.14 Koordinatni sustav s dvije grupe s prikazom pada vjerojatnosti ovisno o udaljavanju od središta grupe

3.4. Umjetne neuronske mreže

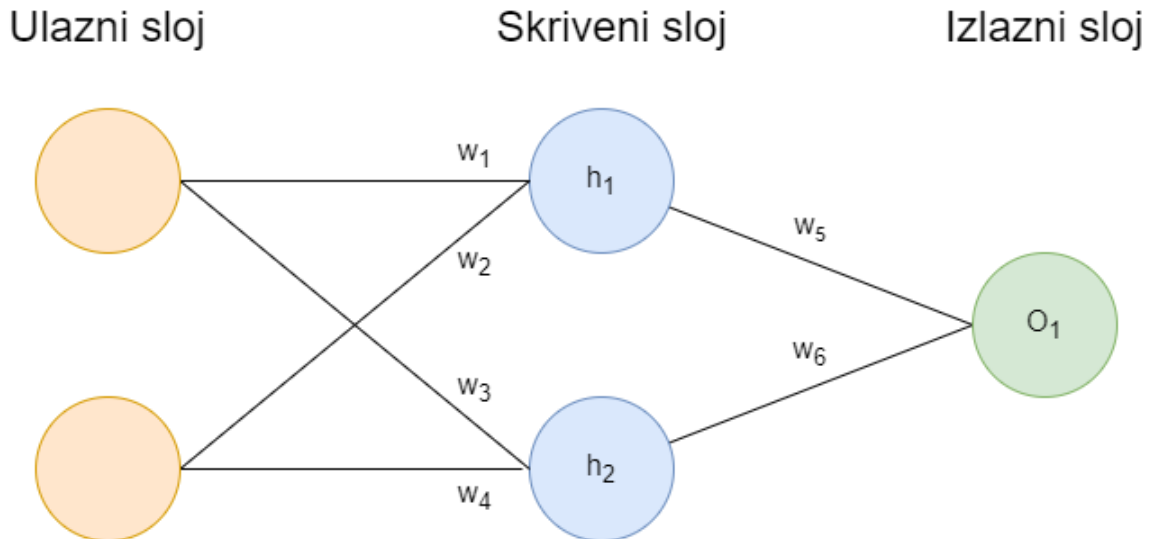
Kako je navedeno u [11], umjetne neuronske mreže računalni su sustavi dizajnirani za simulaciju rada ljudskog mozga. Ovi sustavi sastoje se od velikog broja međusobno povezanih procesorskih čvorova ili neurona koji mogu međusobno komunicirati. Čvorovi su obično organizirani u niz slojeva, pri čemu prvi sloj prima ulazne podatke iz vanjskog svijeta, a zadnji sloj proizvodi izlazne podatke. Veze između čvorova mogu se konfigurirati na razne načine, ovisno o zadatku za koji se mreža koristi. Čvorovi su organizirani u slojeve, pri čemu su čvorovi u svakom sloju u potpunosti povezani s čvorovima u prethodnom i sljedećem sloju što možemo vidjeti na slici Slika 3.15. Snaga veza između čvorova obično je predstavljena težinom, koja može biti pozitivna ili negativna. Izlaz čvora izračunava se zbrajanjem ponderiranih ulaza iz svih čvorova s kojima je povezan i primjenom nelinearne funkcije, poznate kao aktivacijska funkcija, na rezultat. Neuronske mreže se treniraju prilagođavanjem težine veza između čvorova na temelju rezultata skupa podataka za treniranje. Proces obuke obično uključuje ponavljanje niza prolaza naprijed i natrag kroz

podatke obuke. Tijekom prolaska naprijed unose se informacije iz seta podataka namijenjenih za vježbu. Neuronske mreže dobro su prilagođene zadacima koji su teški za tradicionalne računalne sustave, kao što je prepoznavanje uzoraka i klasifikacija podataka. Također su dobri u rukovanju podacima koji su šumoviti ili nepotpuni, što ih čini idealnim za aplikacije kao što su prepoznavanje slike i glasa. Umjetna neuronska mreža (ANN) je računalni sustav koji je inspiriran biološkim neuronskim mrežama i koji funkcionira na sličan način.

Jedan od poznatijih primjera korištenja neuronskih mreža je određivanje je li e-pošta neželjena (engl. *Spam*). Ovaj proces može se podijeliti u nekoliko koraka:

- Priprema podataka: Podaci za obuku i testiranje ANN-a obično bi bili skup podataka označenih e-poruka. Svaka e-pošta bila bi predstavljena kao niz značajki, to bi bile fraze i učestalost pojavljivanja određenih riječi u e-pošti. Vrijednosti oznaka bi bile dvije kategorije, a to su: e-pošta je neželjena i e-pošta nije neželjena. Ulazne podatke bismo podijelili na dvije skupine, skup za treniranje i skup za testiranje. Bitno je napomenuti da skupovi neće sadržavati iste primjere.
- Odabir arhitektura: Arhitektura ANN-a sastojala bi se od ulaznog sloja, jednog ili više skrivenih slojeva te izlaznog sloja. Ulazni sloj bi se sastojao od onoliko neurona koliko imamo različitih značajki koje predstavljaju e-poštu. Skriveni slojevi povezuju ulazni i izlazni sloj, važno je napomenuti da je svaki neuron skrivenog sloja povezan sa svim neuronima ulaznog i izlaznog sloja. Izlazni sloj bi imao jedan neuron s logističkom funkcijom koja bi na izlaz davala vrijednosti 1 ili 0, odnosno e-pošta je neželjena i e-pošta nije neželjena. Prikaz ovakve arhitekture vidljiv je na Slika 3.15.
- Treniranje: Proces treniranja se sastoji od prosljeđivanja podataka za treniranje u mrežu te prilagođavanje težina između slojeva s obzirom na razliku između predviđenog i stvarnog izlaza. Na Slika 3.15 težine su označene slovom w te indeksom, neuroni skrivenog sloja su označeni slovom h
- Evaluacija: Nakon treniranja mreža se testira sa novim, neviđenim podacima, to je skup za testiranje koji smo u prvom koraku odvojili. Radi se usporedba s predviđenim i stvarnim izlazom te se na temelju toga odlučuje je li mreža dobro trenirana ili je potrebno odraditi još treniranja.

- Podešavanja parametara: U koraku odabira arhitekture i treniranju postoji nekoliko parametara koji se mogu promijeniti kako bi se poboljšalo treniranje. Na primjer, može se promijeniti broj skrivenih slojeva ili broj neurona po sloju.



Slika 3.15 Prikaz međusobno povezanih neuronskih čvorova u 3 sloja

3.5. Primjene strojnog učenja

Neke popularne primjene strojnog učenja uključuju:

- **Autonomna vozila:** Strojno učenje koristi se za razvoj automobila koji se sami voze. Algoritmi se koriste za obradu podataka sa senzora i kamera za donošenje odluka o načinu navigacije.
- **Otkrivanje prijevare:** strojno učenje može se koristiti za prepoznavanje obrazaca lažnog ponašanja. To se može koristiti za označavanje potencijalne prijevarne aktivnosti i sprječavanje njezine pojave.
- **Predviđanje ponašanja potrošača:** strojno učenje može se koristiti za razvoj modela koji predviđaju što će potrošači vjerojatno učiniti. To se može koristiti za ciljanje marketinških kampanja i davanje preporuka proizvoda.
- **Prepoznavanje govora:** Strojno učenje koristi se za razvoj sustava za prepoznavanje govora. Algoritmi se koriste za obradu audio podataka i njihovo pretvaranje u tekst.
- **Prepoznavanje slika:** Strojno učenje koristi se za razvoj algoritama koji mogu identificirati objekte na slikama. To se može koristiti za aplikacije kao što su sigurnost i automobili koji se sami voze

Primjene strojnog učenja uključuju prediktivno modeliranje, klasifikaciju, regresiju i optimizaciju. Strojno učenje može se koristiti za stvaranje modela koji predviđaju ishode na temelju podataka, identificiraju obrasce u podacima i grupiraju podatke za daljnju analizu. Ti se modeli mogu koristiti za predviđanje budućih događaja ili za optimizaciju procesa donošenja odluka. Danas postoji mnogo načina na koje se strojno učenje koristi u stvarnom svijetu. Jedan od načina je kroz prediktivnu analitiku, koja se koristi za predviđanje budućih događaja, trendova i ponašanja. To se može koristiti u raznim područjima, kao što su marketing, zdravstvena skrb i financije.

4. Implementacija programskog rješenja

4.1. Arhitektura programskog rješenja

Za izradu programskog rješenja korišten je programski okvir ASP.NET Core 6. ASP.NET izgrađen je na temelju .NET 6. .NET je prijašnje bio nazvan .NET Core, te je bio kompatibilan sa Windows, Linux i macOS platformama dok je .NET Framework bio kompatibilan samo s Windows operacijskim sustavom. Danas postoji samo jedna verzija, a to je .NET koji je kompatibilan sa svim operacijskim sustavima te je kôd otvorenog izvora (engl. *Open-source*). Odabirom ASP.NET postoji mogućnost korištenja nekoliko programskih okvira (engl. *Framework*), kao što su WebForms, MVC Web App ili Blazor.[12]

Programsko rješenje izrađeno je u Blazor programskom okviru. Kako je navedeno u [13], Blazor ima nekoliko načina izvedbe, a to su Blazor Server, Blazor WebAssembly i kombinacija Blazor Server i Blazor WebAssembly. Blazor Server koristi SignalR za komunikaciju klijenta s poslužiteljem te se sve operacije obrađuje na poslužitelju, a klijent dobiva natrag HTML koji se samo treba prikazati unutar Internet preglednika. Kako bi se osigurala responzivnost stranice korisniku se šalju samo promjene u pojedinim komponentama, a ne svaki put cijeli HTML. Blazor WebAssembly sve operacije odrađuje na klijentskoj strani, unutar preglednika, no time smo ograničeni s podacima koje možemo koristiti.

Platforma koristi kombinaciju Blazor Server i Blazor WebAssemblya. Blazor Server koristi se kao *backend*, odnosno REST API sučelje, ali u isto vrijeme korisniku isporučuje Blazor WebAssembly, koji je u ulozi *frontend* aplikacije. Potrebno je napomenuti da se i *frontend* i *backend* programiraju u istom programskom jeziku, a to je C#. Blazor je omogućio C# *backend* developerima da se iskušaju u *frontendu* bez potrebe za učenjem novog programskog jezika.

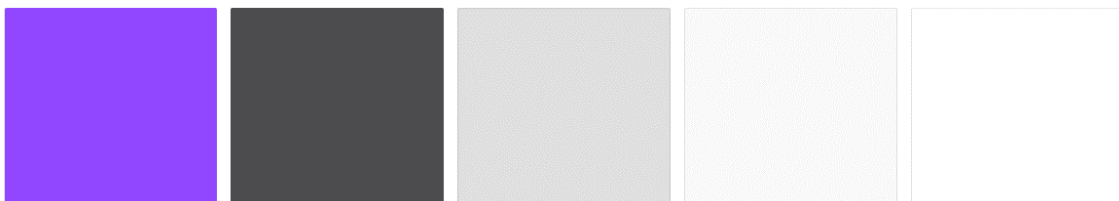
Za bazu podataka izabran je PostgreSQL. Prema informacijama iz [14] PostgreSQL je objektno-relacijski sustav baze podataka koji je otvorenog izvora te koristi i proširuje standardni SQL programski jezik. Činjenica da je PostgreSQL otvorenog izvora znači da iza njega postoji velika zajednica kojoj se možemo odabrati ukoliko imamo bilo kakvih

problema sa implementacijom PostgreSQL ili održavanjem istog. Ovdje dolazimo do još jedne arhitekturne odluke, a to je korištenje Dockera.

Prema informacijama iz [15], Docker je otvorena platforma za razvoj, isporuku i pokretanje aplikacija. Docker omogućuje odvajanje aplikacije od infrastrukture kako bi se omogućila brza isporuka programskog rješenja. Iskorištavanjem Dockerovog pristupa brzom isporuci, testiranju i implementaciji kôda, može se značajno smanjiti kašnjenje između pisanja kôda i njegovog pokretanja u proizvodnji. Docker pruža mogućnost pakiranja i pokretanja aplikacija u slabo izoliranim okruženjima koja se nazivaju spremnici (engl. *Container*). Izolacija i sigurnost omogućuju istovremeno pokretanje više spremnika na određenom računalu. Spremnici su lagani i sadrže sve što je potrebno za pokretanje aplikacije, nema potrebe za prethodnom instalacijom programa na glavno računalo. Lagano je dijeliti spremnike i može se omogućiti da svi u timu imaju pristup istom tipu spremnika, koji je za sve članove ispravno konfiguriran.

4.2. Dizajn programskog rješenja

Što se tiče dizajna, odabrana je paleta boja vidljiva na slici Slika 4.1 te je korišten MudBlazor programski okvir koji nam svojim komponentama omogućuje da na brz i jednostavan način stvorimo izgled i osjećaj poznatog Material Design razvijenog od strane Google. Uz MudBlazor korišten je i CSS za stilsko oblikovanje web stranice.



Slika 4.1 Paleta boja korištena u aplikaciji

Za potrebe aplikacije korišteno je više vrsti Blazor komponenti kako bi se postigao uniformni stil i lakoća održavanja. Osnovni elementi korisničkog sučelja (engl. *User interface*) su Blazor komponente. Mogu se koristiti već prethodno definirane komponente ili mogu se kreirati u potpunosti nove komponente. U kôdu Kôd 4.1 je prikaz jedne parametrizirane komponente, a u kôdu Kôd 4.2 je vidljiva upotreba te komponente.

```
<MudCard Elevation="0" Square="true">  
<MudCardHeader>
```

```

        <CardHeaderAvatar>
            <MudAvatar Image="@ImageUrl"/>
        </CardHeaderAvatar>
        <CardHeaderContent>
            <MudText Typo="Typo.body2">@UserName</MudText>
        </CardHeaderContent>
    </MudCardHeader>
</MudCard>

@code {

    [Parameter]
    public string ImageUrl { get; set; }

    [Parameter]
    public string UserName { get; set; }

}

```

Kôd 4.1 Primjer Blazor komponente

```

<PersonCard Class="mt-n2" ImageUrl="@userAccount.Avatar"
UserName="@context.User.Identity?.Name"/>

```

Kôd 4.2 Primjer uporabe parametrizirane Blazor komponente

Blazor nam omogućuje korištenje C# kôda unutar komponenti korisničkog sučelja korištenjem @ znaka ili ukoliko nam je potrebno više od jedne linije kôda možemo koristiti oznaku @code { }. Zatim, imamo Blazor stranice koje su po svojem obliku jednake Blazor komponentama uz dodatak @page direktive, jedna stranica može sadržavati više @page direktiva, u kôdu Kôd 4.3 prikazana je stranica za internet trgovinu. U @code dijelu prikazan je dohvat podataka sa Blazor WebAssemblya na Blazor Server, „ShopItems“ je ime metode koja se poziva na REST API sučelju.

```

@page "/shop"
@Inject HttpClient Http

<MudGrid>
    @foreach (var item in shopItems)

```

```

        {
            <ShopItem ItemImage="@item.Avatar"
                ItemName="@item.Name" ItemPrice="@item.Price"/>
        }
    </MudGrid>

@code {
    private ShopItem[]? shopItems;

    protected override async Task OnInitializedAsync()
    {
        shopItems = await
        Http.GetFromJsonAsync<ShopItem[]>("ShopItems");
    }
}

```

Kôd 4.3 Primjer Blazor stranice

Blazor raspored (engl. *Layout*) je zadnja komponenta koju ćemo spomenuti. Da bi neka komponenta bila tipa raspored mora naslijediti klasu `LayoutComponentBase`, a zatim naš kreirani raspored može biti opet naslijeđen od strane drugih rasporeda, u kôdu Kôd 4.4 prikazan je glavni raspored korišten na stranici. Kako bismo koristili raspored na nekom od naših stranica sve što je potrebno da na početak stranice stavimo `@layout MainLayout`

```

@inherits LayoutComponentBase

<MudThemeProvider Theme="_theme"/>
<MudDialogProvider/>
<MudSnackbarProvider/>
<MudLayout>
    <NavBar/>
    <MudMainContent>
        <MudContainer MaxWidth="MaxWidth.Large" Class="mt-4">
            @Body
        </MudContainer>
    </MudMainContent>
</MudLayout>

@code {
    private readonly MudBlazorMainTheme _theme = new();
}

```

```

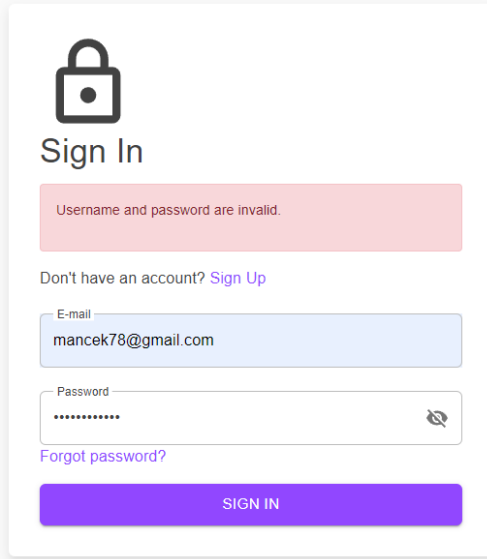
protected override void OnInitialized()
{
    StateHasChanged();
}
}

```

Kôd 4.4 Primjer kôda za Blazor raspored

Na slici Slika 4.2 prikazane su forme za registraciju i prijavu korisnika. Ukoliko dođe do greške, npr. korisnik je unio krivu lozinku pojavljuje se okvir upozorenja, iznad forme koja se ispunjava, sa informacijom o tome što je pogrešno napravljeno što je vidljivo na slici Slika 4.3.

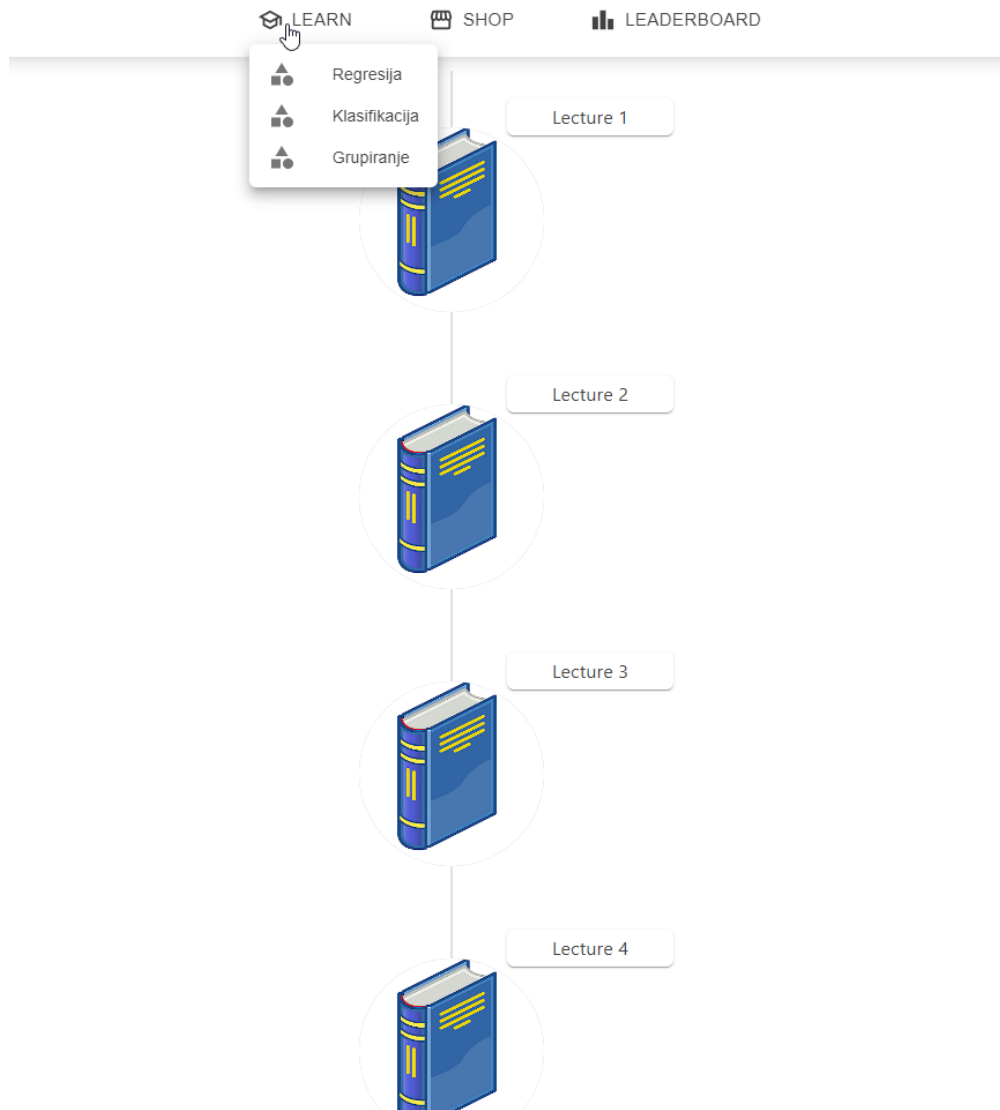
Slika 4.2 Prikaz forme za registraciju i prijavu korisnika



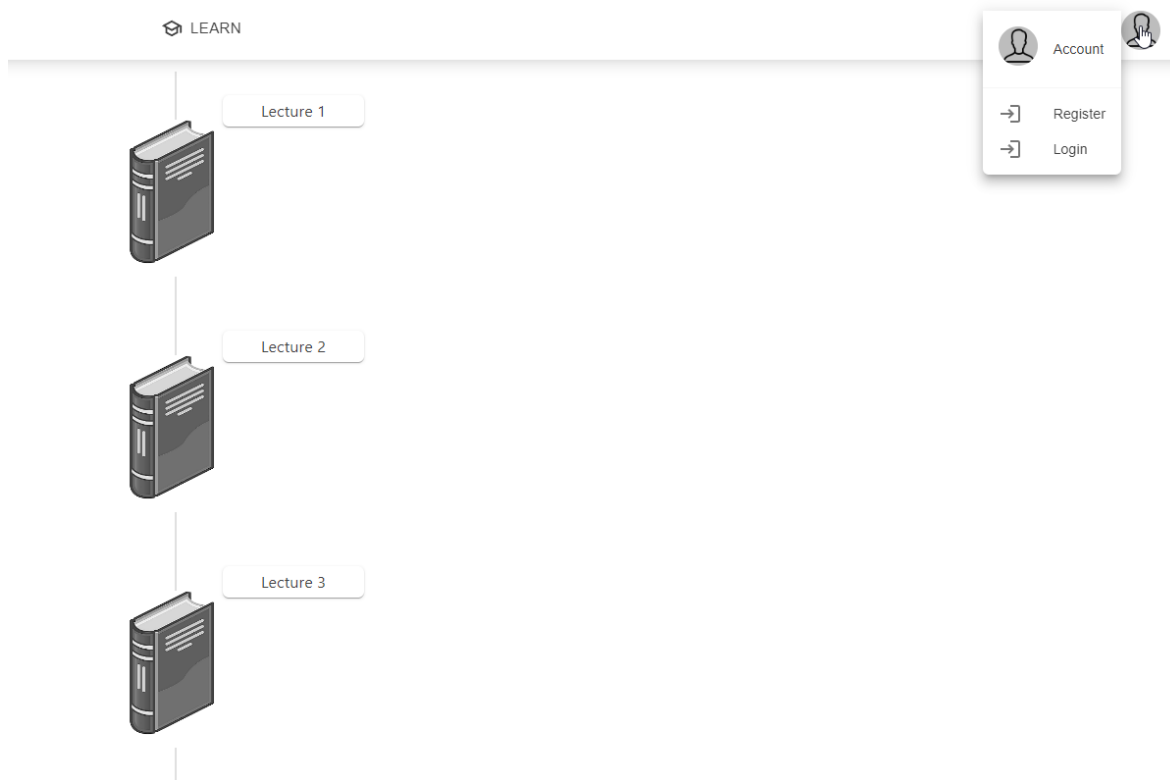
The image shows a 'Sign In' form with a lock icon at the top. Below the icon is the text 'Sign In'. A red error message box contains the text 'Username and password are invalid.'. Below this is a link 'Don't have an account? Sign Up'. There are two input fields: 'E-mail' with the value 'mancek78@gmail.com' and 'Password' with masked characters. A 'Forgot password?' link is below the password field. At the bottom is a purple 'SIGN IN' button.

Slika 4.3 Prikaz forme sa okvirom upozorenja i porukom greške

Prilikom posjeta platformi korisnicima će biti prikazana stranica vidljiva na slici Slika 4.4. Na navigacijskoj traci korisnici iz padajućeg izbornika mogu izabrati o kojem poglavlju žele učiti. Na navigacijskoj traci u desnom gornjem kutu korisnici imaju padajući izbornik koji ih vodi na stranicu za registraciju ili prijavu što je vidljivo na Slika 4.5. Ako je korisnik prethodno ulogiran padajući izbornik se mijenja te prikazuje korisnikov avatar te poveznicu na stranicu s podacima trenutno prijavljenog računa.



Slika 4.4 Početna stranica platforme




Slika 4.5 Prikaz padajućeg izbornika za registraciju i prijavu korisnika

Na slici Slika 4.6 vidi se forma koja se otvara prilikom odabira opcije Account. Na formi su prikazane informacije kao što su email adresa, količina VPointsa te slika avatara. Isto tako, pritiskom na karticu Change Password korisnici mogu mijenjati svoju trenutnu lozinku što je vidljivo na slici Slika 4.7.

Account

GENERAL SECURITY

Profile



mancek78@gmail.com

Profile Details

Virtual Points: 960

Email: mancek78@gmail.com


Dont worry, we shall not spam!

Slika 4.6 Prikaz forme za račun korisnika

Account

GENERAL SECURITY

Profile



mancek78@gmail.com

Change Password

Password*

Password Confirmation

Current Password

CHANGE PASSWORD

Slika 4.7 Prikaz forme za promjenu lozinke

Zadani prikaz početne stranice, prije odabira poglavlja, su lekcije iz regresije. Korisnik može kliknuti samo na lekcije čija slika nema crno-bijeli filter te mu se nakon toga otvara novi prozor sa lekcijom koja je vidljiva na slici Slika 4.8.

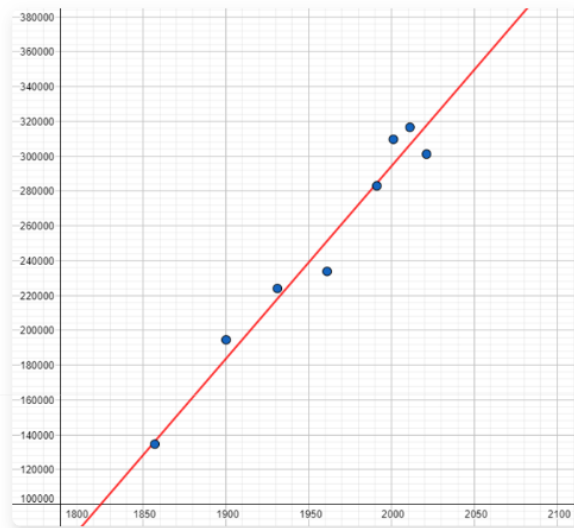
Jednostavna linearna regresija

Jednostavna linearna regresija koristi regresijski model odnosno formulu kako bi mogla predvidjeti vrijednosti zavisnih varijabli.

Kako bismo izradili jednostavni linearni regresijski model sve što nam je potrebno je odabrati zavisne i nezavisne varijable zadanog skupa podataka.

Za primjer uzet ćemo broj stanovnika nekog grada u različitim vremenskim periodima. U ovom primjeru koristit ćemo godine kao nezavisnu varijablu, a broj stanovnika kao zavisnu varijablu zato što želimo predvidjeti broj stanovnika u nekoj godini u budućnosti.

Grad	1995.	2005.	2010.	2014.
Dubai	689 420	1 321 450	1 874 730	2 656 450



Slika 4.8 Prikaz stranice s lekcijom

Kad je korisnik gotov s lekcijom te želi nastaviti mora kliknuti na gumb s natpisom „Done“. Klikom na gumb korisnika se preusmjerava na stranicu s pitanjem koja je vidljiva na slici Slika 4.9. Stranica omogućuje korisniku da *drag & drop* funkcionalnošću stavi točne riječi na pravo mjesto. Ukoliko korisnik krivo odgovori na pitanje preusmjeren je natrag na lekciju, a ako korisnik točno odgovori na pitanje preusmjeren je na početnu stranicu gdje može odabrati sljedeću lekciju.

Jednostavna linearna regresija

Imamo podatke o postotku napunjenosti bazena u određenom vremenu

Vrijeme	5:45	6:00	6:15	6:30	6:45
Postotak napunjenosti	0%	7%	16%	24%	30%

Koje ćemo vrijednosti uzeti za zavisnu a koje za nezavisnu varijablu ako želimo saznati koliko će bazen biti pun u 9:00

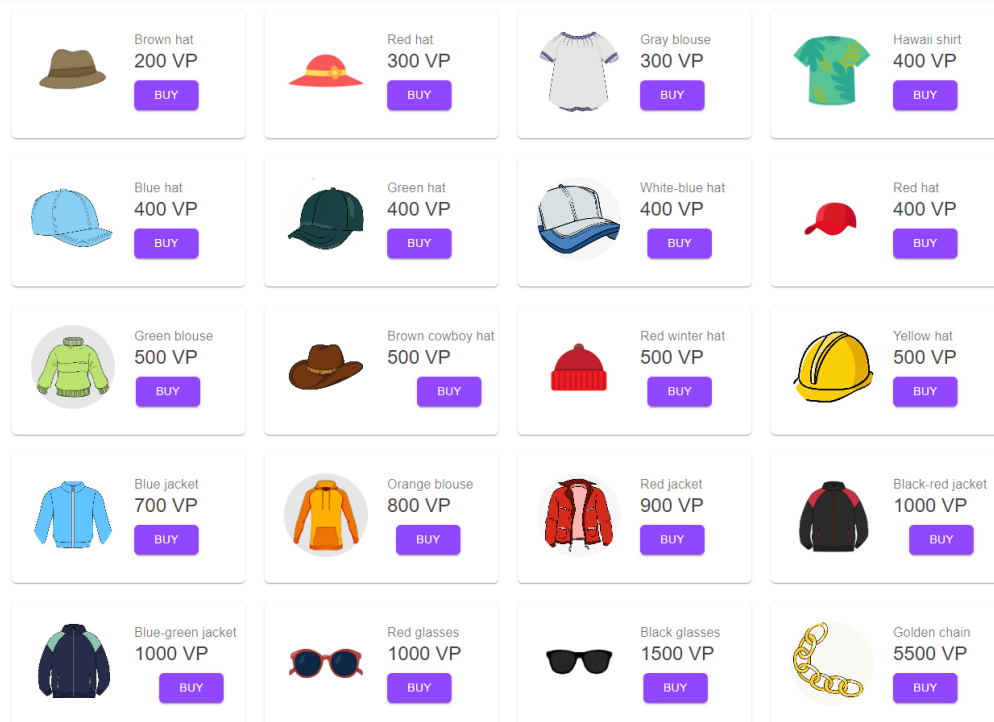
Zavisna varijabla je , a nezavisna varijabla je .

POSTOTAK NAPUNJENOSTI





VRIJEME

Slika 4.9 Prikaz stranice s pitanjem

Nakon svake uspješno odrađene lekcije korisnik dobiva određenu sumu VPointsa koje zatim može iskoristiti u trgovini platforme vidljivoj na slici Slika 4.10. Razne stavke u trgovini mijenjaju izgled korisnikovog avatara te su te promjene vidljive korisniku, ali i drugim korisnicima platforme kroz ljestvicu korisnika koji su prošli najviše lekcija što je vidljivo na slici Slika 4.11. Ljestvica pokazuje sve korisnike sortirane po broju uspješno završenih lekcija.



Slika 4.10 Prikaz stavki u trgovini platforme

Pozicija	Ime	Avatar	Level
1	Nikola		31
2	Darko		28
3	Karlo		16
4	Mirko		12

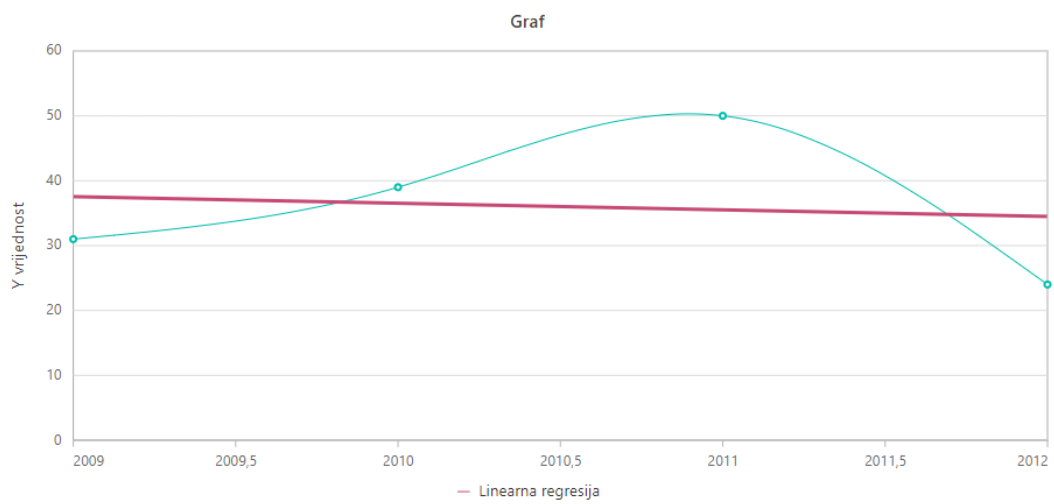
Slika 4.11 Prikaz ljestvice igrača

Svi prijavljeni korisnici imaju pristup igralištu (engl. *sandbox*) u kojem trenutno mogu koristiti opciju generiranja pravca linearne regresije što je vidljivo na Slika 4.12. Igralište ima tablicu u kojoj se mogu mijenjati x i y vrijednosti, zatim pritiskom na tipku „Generiraj“ generira se pravac linearne regresije. Ovo je idealan način za igranje s brojkama, a uz to korisnici mogu i naučiti kako funkcionira linearna regresija.

Playground

X vrijednost	Y vrijednost	
2009	31	✓
2010	39	
2011	50	
2012	24	

GENERIRAJ



Slika 4.12 Prikaz forme igrališta

5. Anketa zadovoljstva programskim rješenjem

Anketa o zadovoljstvu softverom je alat koji se može koristiti za mjerenje koliko su korisnici zadovoljni određenim softverskim programom. Anketa se može koristiti za prikupljanje podataka o različitim čimbenicima, uključujući sveukupno zadovoljstvo softverom, jednostavnost korištenja, funkcionalnost i podršku. Isto tako, anketa se može koristiti za prepoznavanje područja u kojima softver treba poboljšati i za praćenje zadovoljstva korisnika tijekom vremena.

5.1. Sadržaj ankete

Pošto sam prvi put radio obrazovnu platformu, odabrao sam skup pitanja na koja bih ja volio odgovoriti da testiram obrazovnu platformu, no isto tako sam pazio da odabrana pitanja imaju smisla te da mogu pomoći u daljnjem razvoju platforme. Odabrana pitanja bila su:

- Koliko ste zadovoljni objašnjenjem tema?
- Koliko je lakše savladati gradivo koristeći platformu u odnosu na tradicionalno učenje?
- Što se može poboljšati na platformi?

Prva dva pitanja služe kako bi se dobio opći dojam o izvedbi platforme i koliko je zapravo ostvaren cilj platforme dok kroz treće pitanje korisnici imaju slobodu napisati bilo što što bi moglo pomoći da platforma bude što bolja i korisnija korisnicima.

Korisnici aplikacije su kroz Google Obrasce odgovorili na pitanja. Google Obrasci su odabrani kao način provođenja ankete zato što se ankete mogu vrlo brzo i efikasno kreirati, da bi korisnici pristupili anketi dovoljno im je poslati samo poveznicu te rezultati ankete su prikazani na vrlo jednostavan, ali opet sveobuhvatan način. Korisnici koji su sudjelovali u anketi su iste dobne skupine sa malo ili ništa znanja o temama iz strojnog učenja. Postavljena anketa vidljiva je na slici Slika 5.1.

Koliko ste zadovoljni objašnjenjem tema?

Zadovoljan

Niti zadovoljan, niti nezadovoljan

Nezadovoljan

Što se može poboljšati na platformi?

Vaš odgovor _____

Koliko je lakše svladati gradivo koristeći platformu u odnosu na tradicionalno učenje?

Puno lakše

Niti lakše, niti teže

Teže

Slika 5.1 Anketa zadovoljstva korisnika


5.2. Rezultati ankete i analiza

Google Obrasci nam pružaju grafički prikaz odgovora anketiranih korisnika. Na slikama Slika 5.2 i Slika 5.3 prikazani su rezultati ankete.

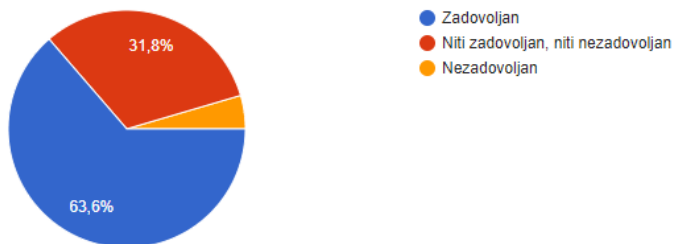
Na pitanje „Koliko ste zadovoljni objašnjenjem tema?“ 63,6 % korisnika je odgovorilo da su zadovoljni što nam govori da je malo više od polovice korisnika općenito zadovoljno opsegom te načinom objašnjenja tema, 31,8 % korisnika nije niti zadovoljno niti nezadovoljno te je 4,6 % korisnika izjavilo da nisu zadovoljni načinom ili opsegom objašnjenih tema.

Na pitanje „Koliko je lakše svladati gradivo koristeći platformu odnosu na tradicionalno učenje?“ 50 % korisnika je izjavilo da im je puno lakše učiti koristeći platformu, 31,8 % korisnika je reklo da nema razlike između učenja kroz platformu i učenja iz knjige ili bilježnice te 18,2 % korisnika smatra da je učenje kroz platformu zapravo teže od učenja iz knjige.

Koliko ste zadovoljni objašnjenjem tema?


 Kopiraj

22 odgovora

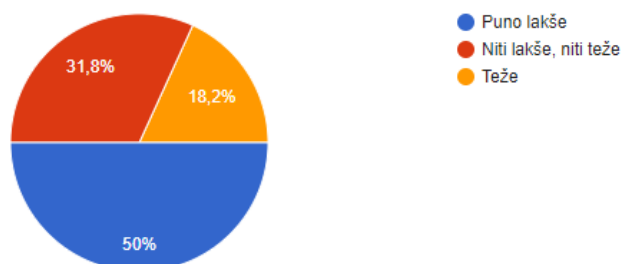


Slika 5.2 Rezultati ankete na pitanje „Koliko ste zadovoljni objašnjenjem tema?“

Koliko je lakše svladati gradivo koristeći platformu u odnosu na tradicionalno učenje?

 Kopiraj

22 odgovora



Slika 5.3 Rezultati ankete na pitanje „Koliko je lakše svladati gradivo koristeći platformu u odnosu na tradicionalno učenje?“

S obzirom na rezultate ankete većina je korisnika zadovoljna načinom na koji su objašnjene teme iz strojnog učenja te bi vjerojatno rado upotrebljavala platformu za učenje o temama iz strojnog učenja. U odnosu na tradicionalno učenje iz knjiga, polovica korisnika je rekla da im je lakše pratiti i naučiti nešto koristeći platformu, dok ostala polovica smatra da je jednako teško ili čak teže.

Zaključak

Cilj ovog rada bio je stvoriti platformu za učenje o temama iz strojnog učenja. Platforma je vrlo jednostavna i intuitivna za korištenje. Proces registracije je vrlo brz te zahtijeva samo email adresu i lozinku. Lekcije su objašnjene bez upotrebe previše tehničkih pojmova i izračuna te smatram da su primjerene dobnoj skupini koju cilja ova platforma, a to su srednjoškolci. Odgovori na pitanja na kraju lekcije su jednostavno i jednosmisleno napisani. Postavljena pitanja uvijek se vežu samo na jednu lekciju i to je lekcija netom prije pitanja. Glavni razlog zašto bi netko izabrao baš ovu platformu za učenje smatram da je trgovina platforme. Trgovina platforme pruža veliki broj stavki koje korisnici mogu iskoristiti za promjenu izgleda svojeg avatara. Isto tako, ljestvica najboljih korisnika stvara neku vrstu kompetitivne atmosfere. Kao što u video igrama želimo pokazati da smo bolji od drugih na sličan način ova platforma dopušta kompetitivnim korisnicima da pokažu svoju kompetitivnu stranu, a uz različite izgled avatar izgledat će super dok to rade.

Prema podacima iz provedene ankete vidljivo je da je većina ispitanika dobro prihvatila platformu te bi ju koristili u svojem svakodnevnom školstvu. Naravno, ima mjesta za napredak što se tiče dizajna platforme i same prezentacije informacija, ali možemo smatrati da je platforma ostvarila svoj cilj i da stvarno omogućuje srednjoškolcima da više nauče o temama iz strojnog učenja. S tim u mislima smatram da je vrijeme da se postavi novi cilj za platformu.

Platforma je primarno napravljena da pomogne srednjoškolcima približiti teme strojnog učenja, no sad platforma može ići u dva glavna smjera. Prvi smjer je proširenje platforme na ostale dobne skupine, prilagođavanje te dodavanje novih stvari koje bi mogle privući pozornost starijih dobnih skupina. Još ne znam koje bi to bile nove stvari i što bi to predstavljalo za platformu ako želim ići u ovom smjeru. Drugi smjer je proširenje i generalizacija platforme. Dodavanje administratorskog dijela platforme kroz koji bi učitelji mogli dodavati, brisati i mijenjati lekcije, pitanja i odgovore. Generalizacije platforme da se može koristiti za više tema, dodavanje više interaktivnih grafova, slika i ostalih elemenata koji bi pomogli korisnicima u učenju. Odvajanje trgovine platforme i sustav avatara u posebni sustav te bi se taj sustav nagrađivanja mogao koristiti za druge zadatke ili aplikacije. Platforma bi i dalje imala avatare, ali ne bi bila jedini način kupovanja stavki i promjene izgleda avatara. Trenutno na avatare je moguće primijeniti samo jednu zasebnu stavku u

jednom trenutku, zamisao je da se omogući primjena nekoliko stavki odjednom. Isto tako, zamisao je da se avatari „ožive“, da se naprave 3D modeli avatara, te se dodaju razne animacije na avatare koje bi još više popularizirale cijelu platformu i avatare.

Student vlastoručno potpisuje Završni rad na prvoj stranici ispred Predgovora s datumom i oznakom mjesta završetka rada te naznakom:

„Pod punom odgovornošću pismeno potvrđujem da je ovo moj autorski rad čiji niti jedan dio nije nastao kopiranjem ili plagiranjem tuđeg sadržaja. Prilikom izrade rada koristio sam tuđe materijale navedene u popisu literature, ali nisam kopirao niti jedan njihov dio, osim citata za koje sam naveo autora i izvor, te ih jasno označio znakovima navodnika. U slučaju da se u bilo kojem trenutku dokaže suprotno, spreman sam snositi sve posljedice uključivo i poništenje javne isprave stečene dijelom i na temelju ovoga rada“.

U Zagrebu, 03.04.2023.

Popis kratica

GMM	<i>Gaussian model mixture</i>	Gausov model
ANN	<i>artificial neural network</i>	umjetne neuronske mreže
.NET	Network Enabled Technologies	tehnologije omogućene mrežom
ASP.NET		Active Server Pages Network Enabled
Technologies		
HTML	HyperText Markup Language	jezik za označavanje hiperteksta
CSS	Cascading Style Sheets	kaskadni listovi stilova
REST	Representational State Transfer	prijenos reprezentativnog stanja
API	Application Programming Interface	programsko sučelje aplikacije
SQL	Structured Query Language	strukturirani upitni jezik
MVC	Model-View-Controller	model-pogled-kontroler
JWT	JSON Web Token	JSON mrežni žeton

Popis slika

Slika 2.1 Primjeri stavki koje će se moći kupiti u trgovini platforme	3
Slika 3.1 Koordinatni sustav s ulaznim podacima.....	5
Slika 3.2 Koordinatni sustav s pravcem aproksimacije.....	6
Slika 3.3 Koordinatni sustav s pravcem aproksimacije godine – cijena	8
Slika 3.4 Koordinatni sustav s pravcem aproksimacije kilometri – cijena.....	9
Slika 3.5 Koordinatni sustav sa podacima o prolaznosti ispita	11
Slika 3.6 Koordinatni sustav s funkcijom logističke regresije	11
Slika 3.7 Koordinatni sustav s dvije grupe objekata i pomoćnim kružnicama $k = 3$ i $k = 7$ 17	
Slika 3.8 Koordinatni sustav s dvije grupe objekata i pomoćnom kružnicom $k = 9$	18
Slika 3.9 Nepoznata životinja za identifikaciju	19
Slika 3.10 Stablo odlučivanja	20
Slika 3.11 Karta Hrvatske i Slovenije preuzeta sa Google Maps	21
Slika 3.12 Prikaz grupa nastalih particioniranim grupiranjem	22
Slika 3.13 Prikaz grupa nastalih grupiranjem temeljenom na gustoći	23
Slika 3.14 Koordinatni sustav s dvije grupe s prikazom pada vjerojatnosti ovisno o udaljavanju od središta grupe	25
Slika 3.15 Prikaz međusobno povezanih neuronskih čvorova u 3 sloja.....	27
Slika 4.1 Paleta boja korištena u aplikaciji.....	30
Slika 4.2 Prikaz forme za registraciju i prijavu korisnika	33
Slika 4.3 Prikaz forme sa okvirom upozorenja i porukom greške.....	34
Slika 4.4 Početna stranica platforme	35
Slika 4.5 Prikaz padajućeg izbornika za registraciju i prijavu korisnika.....	36
Slika 4.6 Prikaz forme za račun korisnika	37
Slika 4.7 Prikaz forme za promjenu lozinke	37

Slika 4.8 Prikaz stranice s lekcijom.....	38
Slika 4.9 Prikaz stranice s pitanjem.....	39
Slika 4.10 Prikaz stavki u trgovini platforme	40
Slika 4.11 Prikaz ljestvice igrača.....	41
Slika 4.12 Prikaz forme igrališta	42
Slika 5.1 Anketa zadovoljstva korisnika	44
Slika 5.2 Rezultati ankete na pitanje „Koliko ste zadovoljni objašnjenjem tema?“	45
Slika 5.3 Rezultati ankete na pitanje „Koliko je lakše svladati gradivo koristeći platformu u odnosu na tradicionalno učenje?“	45

Popis tablica

Tablica 3.1 Broj stanovnika u Zagrebačkoj županiji.....	4
Tablica 3.2 Cijena automobila s obzirom na starost i kilometražu	8
Tablica 3.3 Prikaz podataka o prolaznosti ispita	10
Tablica 3.4 Prikaz podataka o prognozi i kišobranu	14
Tablica 3.5 Tablica ovisnosti nošenja kišobrana o prognozi.....	14
Tablica 3.6 Vjerojatnost prognoze prema ulaznim podacima	14
Tablica 3.7 Vjerojatnost da ćemo trebati kišobran prema ulaznim podacima.....	15

Popis kôdova

Kôd 4.1 Primjer Blazor komponente	31
Kôd 4.2 Primjer uporabe parametrizirane Blazor komponente	31
Kôd 4.3 Primjer Blazor stranice	32
Kôd 4.4 Primjer kôda za Blazor raspored.....	33

Literatura

- [1] <https://online.stat.psu.edu/stat462/node/91>, 30. Kolovoza 2022.
- [2] <http://abcgeografija.com/teme/popis2021/>, 30. Kolovoza 2022.
- [3] <https://www.investopedia.com/terms/m/mlr.asp>, 30. Kolovoza 2022.
- [4] https://bookdown.org/jarneric/predavanja_smea/2-1-logisti-ka-regresija.html, 30. Kolovoza 2022.
- [5] <https://towardsdatascience.com/naive-bayes-classifier-explained-50F9723571ED>, 30. Kolovoza. 2022.
- [6] <https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning>, 30. Kolovoza 2022.
- [7] <https://towardsdatascience.com/decision-tree-classifier-explained-in-real-life-picking-a-vacation-destination-6226B2B60575>, 30. Kolovoza 2022.
- [8] <https://medium.com/analytics-vidhya/partitional-clustering-181D42049670>, 30. Kolovoza 2022.
- [9] <https://www.kdnuggets.com/2020/04/dbscan-clustering-algorithm-machine-learning.html>, 30. Kolovoza 2022.
- [10] <https://neptune.ai/blog/clustering-algorithms>, 30. Kolovoza 2022.
- [11] J Zupan - Acta Chimica Slovenica, 1994.
- [12] <https://docs.microsoft.com/en-us/aspnet/core/blazor/?View=aspnetcore-6.0>, 30. Kolovoza 2022.
- [13] <https://aws.amazon.com/what-is/net/>, 30. Kolovoza 2022.
- [14] <https://www.postgresql.org/about/>, 30. Kolovoza 2022.
- [15] <https://docs.docker.com/get-started/overview/>, 30. Kolovoza 2022.