Analiza podataka korištenjem metoda regresije i klasifikacije podataka

Lukić, Augustin

Undergraduate thesis / Završni rad

2023

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Organization and Informatics / Sveučilište u Zagrebu, Fakultet organizacije i informatike**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:211:491441

Rights / Prava: Attribution 3.0 Unported/Imenovanje 3.0

Download date / Datum preuzimanja: **2024-05-09**

SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE
VARAŽDIN

Repository / Repozitorij:

Faculty of Organization and Informatics - Digital Repository

zir.nsk.hr

DIGITALNI AKADEMSKI ARHIVI I REPOZITORIJI

SVEUČILIŠTE U ZAGREBU

FAKULTET ORGANIZACIJE I INFORMATIKE

V A R A Ž D I N

Augustin Lukić

Analiza podataka korištenjem metoda regresije i klasifikacije podataka

ZAVRŠNI RAD

Varaždin, 2023.

SVEUČILIŠTE U ZAGREBU

FAKULTET ORGANIZACIJE I INFORMATIKE

V A R A Ž D I N

Augustin Lukić

Matični broj: 0016141745

Studij: Informacijski sustavi

Analiza podataka korištenjem metoda regresije i klasifikacije podataka

ZAVRŠNI RAD

Mentor/Mentorica:

Prof. dr. sc. Jasminka Dobša

Varaždin, rujan 2023.

Augustin Lukić

Izjava o izvornosti

Izjavljujem da je moj završni/diplomski rad izvorni rezultat mojeg rada te da se u izradi istoga nisam koristio drugim izvorima osim onima koji su u njemu navedeni. Za izradu rada su korištene etički prikladne i prihvatljive metode i tehnike rada.

Autor/Autorica potvrdio/potvrdila prihvaćanjem odredbi u sustavu FOI-radovi

_____

Sažetak

Cilj ovog istraživanja bio je upoznati se s metodama regresije i klasifikacije podataka, ali i analizirati podatke na realnom skupu. Metode koje su opisane teorijski su i primijenjene na skupu podatka koristeći R programski jezik za analizu.

Ključne riječi: analiza podataka; klasifikacija podataka; regresijska analiza

Sadržaj

Sažetak ii
Sadržaj iii
1. Uvod 1
2. Metode i tehnike rada 2
3. Regresijska analiza 3
3.1. Korelacija 3
3.2. Regresijska analiza i njena povijest 8
3.3. Vrste regresijske analize 10
3.3.1. Jednostruka linearna regresija 11
3.3.1.1. Metoda najmanjih kvadrata 12
3.3.2. Višestruka linearna regresija 13
3.3.2.1. Metoda najmanjih kvadrata 14
3.4. Zahtjevi linearne regresije 17
4. Klasifikacija podataka 21
4.1. Metode i algoritmi 22
4.1.1. Logistička regresija 22
4.1.1.1. Model logističke regresije 23
4.1.1.2. Pogreška unakrsne entropije 24
4.1.1.3. Gradijentni spust za logističku regresiju 27
4.1.1.4. Logistička regresija standardni (grupni) gradijentni spust 27
4.1.1.5. Logistička regresija stohastički gradijentni spust 28
4.1.2. Regularizirana regresija 28
4.1.2.1. Višeklasna logistička regresija 31
4.1.3. Bayesov klasifikator 33
4.1.3.1. Bayesov klasifikator 35
4.1.3.2. Generativni modeli 37
4.1.3.3. Gaussov Bayesov klasifikator 37
4.1.4. Neparametarske metode 40
4.1.4.1. Algoritam k-NN 40
4.1.4.2. Stabla odlučivanja 41
5. Analiza u R programskom jeziku 42
5.1. R – uvod opis sučelja i instalacija 42
5.1.1. Instalacija i preuzimanje 42
5.2. Opis sučelja 43
5.2.1. Osnove rada u programu 43

5.2.2. Instaliranje i učitavanje paketa 44
5.2.3. Učitavanje paketa 44
5.2.4. Osnovna sintaksa i operatori 45
5.3. Učitavanje i priprema podataka 48
5.4. Opis skupa podataka 49
5.5. Regresijska analiza u R-u 49
5.6. Klasifikacija podataka u R-u 51
6. Zaključak 53
7. Popis literature 54
8. Popis slika 56
9. Popis tablica 57

1. Uvod

Analiza podataka je ključna komponenta u donošenju informiranih odluka u različitim područjima, od poslovanja i financija do zdravstva i raznih društvenih znanosti.

Metode regresije i klasifikacije podataka su dva vrlo moćna alata koji se koriste u analizi podataka. Cilj ovog rada jest istražiti teorijske koncepte i metode regresijske analize i klasifikacije podataka te zatim primijeniti te metode na analizi realnog skupa podataka koristeći R programski jezik i alate za statističku analizu.

Rad je strukturiran na sljedeći način: nakon uvoda, slijede poglavlja u kojem su opisani teorijski koncepti i metode regresijske analize i klasifikacije podataka. U trećem poglavlju je opisana regresija dok se četvrto poglavlje bavi klasifikacijom. Peto poglavlje opisuje kako se te metode mogu primijeniti u R programskom jeziku na realnom skupu podataka. Završno se donosi zaključak.

2. Metode i tehnike rada

Pri izradi rada korištena je stručna literatura o regresijskoj analizi i klasifikaciji podataka. Neki od najrelevantnijih autora na teme klasifikacije i regresije su referencirani, a koristi se podjednako literatura kako od domaćih tako i od stranih autora. Za analizu je potrebno odabrati relevantan skup podataka na kojem je moguća primjena metoda regresijske analize i klasifikacije podataka na skupu podataka. Jedan takav skup podataka koji je pogodan za analizu koristeći R programski jezik jest skup podataka Titanic koji dolazi u instalaciji zajedno s grafičkim sučeljem alata RStudio. Glavni razlog tog odabira jest taj što je taj skup podataka dostupan javno te ima dovoljan broj instanci za prikaz u analizi, a koji su bili jedni od glavnih kriterija pri izboru. Zatim slijedi procjena rezultata analize pomoću odgovarajućih statističkih metoda koje su obrađene kroz glavni dio rada. Metode linearne regresije korištene su za regresijsku analizu dok je za klasifikaciju korištena metoda logističke regresije. Naposlijetku se daje zaključak .o učinkovitosti i točnosti metoda regresijske analize i klasifikacije podataka.

3. Regresijska analiza

3.1. Korelacija

Ako promatramo različite društvene i prirodne pojave moguće je uočiti da između njih postoji određena povezanost. Primjerice, možemo uočiti da postoji povezanost između tjelesne visine i tjelesne mase, školskog uspjeha i razine intelektualnih sposobnosti. Godina starosti i krvnog tlaka. Aerobnog kapaciteta i brzine oporavka, maksimalne jakosti i uspješnosti u bacanju kugle itd. Ako izmjerimo neku skupinu ispitanika u dvije varijable postavlja se pitanje postoji li povezanost između tih dviju varijabli? Ako između njih postoji povezanost to znači da na temelju rezultata u prvoj varijabli možemo predviđati rezultate u drugoj varijabli. Onda se postavljaju još dva pitanja. Prvo, je li povezanost pozitivna ili negativna, odnosno prati li povećanje rezultata u prvoj varijabli, povećanje rezultata u drugoj varijabli ili smanjenje rezultata? A drugo, je li povezanost mala ili velika, odnosno kolika je? Odgovore na ta pitanja dobiju se korelacijskom analizom. [1, str. 160-179].

Jedno od pitanja moglo bi biti korelacija između dobi u kojoj dijete izgovori svoje prve rečenice i kasnijeg školskog uspjeha. Ako korelacijska analiza pokaže da su te dvije karakteristike povezane, možemo istražiti postoji li uzrok i veza. [2].

Ako postoji uzrok i veza, školski uspjeh bi se pomoću regresije mogao predvidjeti prema dobi u kojoj dijete izgovori svoje prve rečenice. Dakle, pitanje je sada, imamo li uzrok odnosa u našem primjeru, u ovom slučaju, odgovor je vrlo jednostavan jer imamo vremensku odvojenost. Ako postoji korelacija između dobi u kojoj dijete govori svoje prve rečenice i kasnije školskog uspjeha, jasno je da vrijeme u kojem dijete izgovori svoje prve rečenice utječe na kasniji školski uspjeh, a ne obratno.
Kasniji školski uspjeh ne može utjecati na dob u kojoj dijete izgovori prve rečenice, no nije uvijek tako lako utvrditi korelaciju.
Pogledajmo i sljedeći primjer. Recimo, na kraju nastave s učenicima napravimo test inteligencije i anketiramo ih o ocjeni u srednjoj školi.

Nakon što provedene ankete i analize svih podataka, sigurno će se pokazati da postoji korelacija između inteligencije i uspjeha u srednjoj školi. Ali sada dolazi veliko pitanje: postoji li uzrok veze? Naravno, sada bismo mogli jednostavno postaviti regresijski model i definirati inteligenciju kao nezavisnu varijablu, a ocjenu u srednjoj školi kao našu zavisnu

varijablu Zatim bismo mogli predvidjeti ocjene u srednjoj školi korištenjem rezultata testa inteligencije, ali to nije dokaz uzročnosti.

Uzročnost znači da postoji jasna veza i uzrok između dviju varijabli. Uobičajena pogreška u tumačenju statistike je da kada korelacija postoji, odmah se pretpostavlja da je ona uzrok i veza.
Postoje dva preduvjeta za korelaciju. Prvi preduvjet, postoji značajan odnos koji je značajna korelacija. Ovaj preduvjet je, naravno, vrlo lako testirati. Jednostavno uzmemo koeficijent korelacije i provjerimo razlikuje li se značajno od 0.
Drugi preduvjet može biti zadovoljen na dva načina. Prvo, zadovoljan je ako postoji vremenski poredak varijabli, dakle, varijabla *a* je prikupljena vremenski prije varijable *b*.
Nadalje, drugi uvjet može biti ispunjen, ako postoji teorijski utemeljena i prihvatljiva teorija, u kojem smjeru ide uzrok odnosa. Ako nijedno od to dvoje nije zadovoljeno, to znači da niti postoji vremenski poredak, niti se uzrok može opravdati ako nije teorijski utemeljen, tada možemo samo govorit o vezi, ali ne i uzročnosti.

Dakle, ne može se reći da varijabla *a* utječe na varijablu *b*, ili obrnuto. Pogledamo li ponovno naše primjere, u prvom primjeru, gdje je pitanje utječe li dob u kojoj dijete izgovori prve rečenice na kasniji školski uspjeh, tu je očito prisutna vremenska komponenta.

Mjerenje kada je prva rečenica izgovorena je očito prije mjerenja kasnijeg školskog uspjeha. Odnos ne može biti obrnut. Ako netko ima odličan školski uspjeh uz puno učenja i uloženog truda to naravno, nema nikakvog utjecaja na to kada su prve rečenice izgovorene.

S druge strane, ako istovremeno mjerimo inteligenciju i srednjoškolske ocjene, nemamo ovu vremensku komponentu. Ovdje treba tražiti utemeljenu teoriju koja varijabla utječe na koju varijablu.
Ako takva teorija ne postoji, ne može se utvrditi uzročnost. Sigurno postoji visoka korelacija između inteligencije i ocjene u srednjoj školi, no moguće je i da ljudi koji puno uče imaju dobre ocjene i stekli su svoje znanje kako bi dobro prošli na testu inteligencije. Dakle, u ovom slučaju, inteligencija i ocjene su određeni u isto vrijeme.
Naravno, to se može izbjeći polaganjem testa inteligencije prije početka škole. Tada možemo biti sigurni da škola nije imala utjecaja na test inteligencije. Dakako, ovo zahtijeva puno vremena i resursa jer rezultate dobijemo mnogo godina kasnije, a isti bi ljudi morali biti anketirani u dva različita vremenska perioda.

Korelacijska analiza pokazuje koliko rezultati u prvoj varijabli objašnjavaju rezultate u drugoj varijabli, odnosno rezultati tih dviju varijabli sukladno.

Francis Galton, poznat kao utemeljitelj korelacijske i regresijske analize, istraživao je kako nasljeđe utječe na razvoj ljudskih karakteristika. Galton je bio engleski antropolog čije se zasluge povezuju s razvojem ovih statističkih metoda. [1].

Pearson je razvio brojne postupke, među kojima i tzv. *produkt-moment koeficijent korelacije*. Koji se izračunava pomoću ove formule:

$$r = \frac{\sum_{i=1}^{n} (z_{xi} \cdot z_{yi})}{n}$$

Gdje je:

z_{xi} standardizirani rezultat ispitanika i u varijabli x.

z_{yi} standardizirani rezultat ispitanika i varijabli y.

Upravo se taj koeficijent danas najčešće koristi za izračunavanje povezanosti između dviju varijabli, a zovemo ga Pearsonov koeficijent korelacije ili samo koeficijent korelacije. Pearsonov koeficijent korelacije može se izračunati na različite načine, a jedan od njih je pomoću ove formule jer se umjesto standardiziranih rezultata koriste centrirani, odnosno odstupanja originalnog rezultata od aritmetičke sredine.:

$$r = \frac{\sum_{i=1}^{n} x_{ci} \cdot y_{ci}}{\sqrt{\sum_{i=1}^{n} x_{ci}^2} \cdot \sqrt{\sum_{i=}^{n} y_{ci}^2}}$$

A moguće ga je izračunati i iz originalnih rezultata ovom formulom:

$$r = \frac{\sum_{i=1}^{n} x_i y_i - \frac{\sum_{i=1}^{n} x_i \cdot \sum_{i=1}^{n} y_i}{n}}{\sqrt{\left(\sum_{i=1}^{n} y_i^2 - \frac{(\sum_{i=1}^{n} y_i)^2}{n}\right)} \cdot \sqrt{\left(\sum_{i=1}^{n} x_i^2 - \frac{(\sum_{i=1}^{n} x_i)^2}{n}\right)}}$$

Gdje je:

x_i – rezultat entiteta i u varijabli x.

y_i – rezultat entiteta i u varijabli y.

n – broj entiteta.

Tako izračunati koeficijent korelacije kreće se u intervalu od -1 do +1 i ako je koeficijent korelacije jednak 0, onda nema korelacije između dvije varijable. Ako je jednak +1, onda se radi o potpuno pozitivnoj korelaciji. Ako je jednak -1, onda se radi o potpuno negativnoj korelaciji. Ako je koeficijent korelacije veći od 0, a manji od + 1, onda se radi o nepotpuno pozitivnoj korelaciji i ako je koeficijent korelacije između 0 i -1, onda se radi o nepotpuno negativnoj korelaciji.

Korelacijski odnos između dvije varijable može se prikazati takozvanim korelacijskim dijagramom i to tako da se na osi x prikažu rezultati entiteta u prvoj varijabli, a na osi y u drugoj varijabli. Onda se točkama označe rezultati svih entiteta u tom takozvanom bivarijantnom koordinatnom sustavu. Ako svakom rezultatu u jednoj varijabli može odgovarati bilo kakav dobar, prosječan, loš rezultat u drugoj varijabli onda te dvije varijable nisu u korelaciji. Jer rezultati jedne varijable kreću se potpuno neovisno od rezultata druge varijable. Što ukazuje na nepostojanje bilo kakve korelativne veze između promatranih varijabli.
Ako možete točno predviđati rezultate u drugoj varijabli na temelju rezultata nekog ispitanika u jednoj varijabli, tj. ako svako povećanje ili smanjenje rezultata u jednoj varijabli prate proporcionalno povećanje ili smanjenje rezultata u drugoj varijabli, tada govorimo o potpunoj pozitivnoj korelaciji. To znači da osobe koje postižu iznadprosječne rezultate u varijabli x obično imaju iznadprosječne rezultate i u varijabli y, dok osobe s ispodprosječnim rezultatima u varijabli x obično imaju ispodprosječne rezultate i u varijabli y.

Ako svako povećanje rezultata u jednoj varijabli prati isto toliko smanjenje rezultata u drugoj varijabli, odnosno ako postoji potpuno obrnuta proporcionalna veza između dviju varijabli, onda se takva korelacija naziva potpuno negativnom korelacijom. U takvom scenariju, osobe s iznadprosječnim rezultatima u varijabli x obično imaju ekvivalentno ispodprosječne rezultate u varijabli y, dok osobe s ispodprosječnim rezultatima u varijabli x obično pokazuju iznadprosječne rezultate u varijabli y.

Ovo ukazuje na to da većim rezultatima u jednoj varijabli obično odgovaraju veći rezultati u drugoj varijabli, ali nije strogo definiran odnos jedan prema jedan jer se rezultati kreću unutar određenog intervala. Ovakav tip odnosa se naziva nepotpunom pozitivnom korelacijom.

U zaključku, korelaciju često interpretiramo korištenjem koeficijenta determinacije, koji predstavlja postotak zajedničkog varijabiliteta između dviju varijabli. Također, važno je napomenuti da se koeficijent korelacije može koristiti za prikazivanje odnosa između varijabli u geometrijskom smislu, gdje različite vrijednosti korelacije odgovaraju različitim kutovima između vektora. Ako je korelacija između varijabli jednaka 0 tada su dva normirana vektora pod kutom od 90 stupnjeva.
Ako je korelacija između varijabli i vektora maksimalna i pozitivna, dakle +1, tada je kut između dvaju vektora 0 stupnjeva.
Ako je korelacija između varijabli i vektora maksimalna i negativna, dakle -1, tada kut između dvaju vektora iznosi 180 stupnjeva.

A ako je korelacija nepotpuna pozitivna, (0< r <+1), tada je kut između dvaju vektora veći od 0 stupnjeva, a manjih od 90 stupnjeva.
A ako je korelacija nepotpuna negativna, (-1< r <0), tada je kut između dvaju vektora veći od 90 stupnjeva, a manjih od 180 stupnjeva. [1].

Naposlijetku dolazimo do problema testiranja statističke značajnosti koeficijenta korelacije, s obzirom da je koeficijent korelacije kao i ostali statistički pokazatelji, najčešće izračunat na uzorku ispitanika koji je odabran iz neke populacije.
Dakle, čim se nešto računa na uzorku ispitanika, postavlja se pitanje je li taj koeficijent dobiven na uzorku uistinu toliki i u populaciji? Zbog toga, u slučaju izračunavanja, odnosno testiranja statičke značajnosti koeficijenta korelacije postavljamo dvije hipoteze nultu i alternativnu.
Nulta hipoteza glasi, korelacija nije statistički značajna uz pogrešku *p*.
Dok alternativna glasi, korelacija je statistički značajna uz pogrešku *p*.
p je najčešće pogreška 0,01 odnosno 1% ili 0,05 što bi odgovaralo pogrešci od 5%. Testiranje statističke značajnosti koeficijenta korelacije vršimo prema ovoj formuli:

$$t = r \cdot \sqrt{\frac{n-2}{1-r^2}}$$

Gdje je:
t - vrijednost tog testa koja ima Studentovu t distribuciju,
r - koeficijent korelacije,
n - broj entiteta.
Iz formule je vidljivo da t vrijednost ovisi o broju ispitanika i veličini koeficijenta korelacije. Vrijednost t je veća što je veća korelacija i što je veći broj entiteta u uzorku. Izračunata t vrijednost usporedi se s kritičnom t vrijednošću koja se utvrdi uz pomoć statističkih tablica ili se izračuna pomoću statističkih funkcija koji se nalaze u okviru statističkih aplikacija. Uz određeni broj stupnjeva slobode *df*=*n*-2, i uz određenu pogrešku statičkog zaključka najčešće 1 ili 5%.
Ako je izračunata t vrijednost veća od kritične t vrijednosti, odbacujemo nultu hipotezu i prihvaćamo alternativnu, uz zaključak da je koeficijent korelacije statistički značajan uz pogrešku 0,05 odnosno 0,01. Zašto? Zbog toga što je vjerojatnost da se tako veliki koeficijent korelacije dogodi slučajno, dakle kad je korelacija u populaciji 0, manji od 0,05 odnosno 5 %, ako je izračunata t vrijednost manja od tablične, odnosno kritične t vrijednosti. U tom slučaju prihvaćamo nultu hipotezu i zaključujemo da koeficijent korelacije nije statistički značajan. To naravno ne znači da korelacija u populaciji mora biti jednaka 0,

već da se ne može sa 99 ili 95% sigurnosti tvrditi da je korelacije u populaciji različita od 0.

3.2. Regresijska analiza i njena povijest

U razvoju regresijske analize značajnu ulogu imaju radovi Gaussa, Galtona, Pearsona i Yulea. Pearson i Galton su razvili korelacijsku analizu proučavajući nasljedne osobine. Upotrijebili su regresijsku jednadžbu za utvrđivanje funkcionalnog odnosa između tjelesne visine očeva i tjelesne visine sinova, što je bila prva primjena regresijske analize u antropološkim istraživanjima, dok je Gauss 70 godina prije njih već koristio regresijsku analizu za određivanje orbite planetu. Yulea je pak razvio svoj pristup korelaciji i time je postavio temelje višestruke regresijske analize. Sam naziv regresijska analiza potječe od Francisa Galtona koji je proučavajući odnos između nekih osobina ljudi, primijetio da, primjerice, izrazito visoki očevi imaju visoku djecu, ali ne toliko visoku kao njihovi očevi, a da izrazito niski očevi imaju nisku djecu, ali nešto višu od očeva. Ovu tendenciju da ekstremne vrijednosti koje postoji u jednoj generaciji, u sljedećoj se pomaknu prema prosjeku Galton je nazvao zakonom o regresiji. [3]. Nakon čega se za matematičku funkciju kojom se uspostavlja funkcionalna veza između varijabli, počeo koristiti naziv regresijska funkcija, a cijeli postupak regresijska analiza. Regresijska analiza je matematičko statistička metoda u kojim se identificira prikladna funkcionalna veza između jedne varijable koja zavisi od drugih, i jedne ili više varijabli koje ih predviđaju. Zavisna ili kriterijska varijabla je varijabla čija se varijabilnost objašnjava nezavisnim varijablama. Zavisne ili prediktorske varijable su varijable na temelju kojih se objašnjava varijabli te zavisne varijable. Regresijska analiza se najčešće koristi u svrhu utvrđivanja utjecaja jedne varijable ili skupa varijabli na neku kriterijsku varijablu. Na primjer, utvrđivanje utjecaja antropoloških karakteristika na uspješnost u nekoj aktivnosti. I za utvrđivanje trenda razvoja rezultata u nekom sportu, primjerice, utvrđivanje trenda razvoja najboljih rezultata u nekoj atletskoj disciplini na svjetskim prvenstvima ili olimpijskim igrama. Funkcionalna veza između prediktorskih varijabli i kriterijske varijable su pomoću odgovarajuće regresijske jednadžbe. Opći oblik regresijske jednadžbe izgleda ovako:

$$y = f(x_1, x_2, \dots, x_m) + e$$

Regresijske modele moguće je generalno podijeliti na temelju dva kriterija, prema broju nezavisnih varijabli možemo ih podijeliti na jednostruke ili jednostavne regresijske modele i na višestruke, odnosno multiple regresijske modele kojoj pripada i logistička regresija. Prema odnosu između zavisnih i nezavisnih varijabli moguće ih je podijeliti na linearne regresijske modele kao što su polinom prvog stupnja (pravac) i nelinearne regresijske modele, kao što su

primjerice logaritamska funkcija, eksponencijalna polinomi drugog, trećeg Ili općenito *n*-tog stupnja. [1], [4].
Kako bi se stekao dobar uvid najbolje je krenuti s primjerima. Započnimo s mjerenjem utjecaja jedne ili više varijabli na drugu. U kontekstu istraživačkog rada moglo bi nas zanimati što utječe na dječju sposobnost koncentracije. Zanima nas možemo li dokazati da postoje parametri koji su pozitivno ili negativno utjecali na dječju sposobnost koncentracije.

Drugi primjer bi bio da želimo istražiti ima li stupanj obrazovanja roditelja i mjesto stanovanja utjecaj na budući, stupanj obrazovanja djece. Ovo je područje vrlo istraživano i ima veliku primjenu u društvenim i ekonomskim znanostima.

Drugo područje koje koristi regresiju za predviđanja više je orijentirano na primjenu da izvučemo najviše iz popunjenosti kapaciteta bolnice. Moglo bi nas zanimati koliko dugo pacijent ostaje u bolnici. Dakle, na temelju karakteristika potencijalnog pacijenta kao što su starosna dob, razlog za boravak i postojeća stanja, želimo znati koliko će dugo ta osoba vjerojatno ostati u bolnici na temelju ovog predviđanja.

Zamislimo još jedan primjer, kao vlasnik internetske trgovine. Jako vas zanima koji će proizvod neka osoba najvjerojatnije kupiti. Želite posjetitelju predložiti proizvod kako biste povećali prodaju internetske trgovine.
Tu dolazi do izražaja regresija. Dakle, treba znati da postoje različite vrste regresijske analize, ali ih nije teško razumjeti. Kada govorimo o regresijskoj analizi, razlikujemo jednostruku linearnu, višestruku linearnu i logističku regresiju.

3.3. Vrste regresijske analize

U jednostrukoj linearnoj regresiji koristi se samo jedna nezavisna varijabla za zaključivanje zavisne varijable. U primjeru, gdje želimo predvidjeti plaću osobe. Koristimo samo jednu varijablu, na primjer, tjedno radno vrijeme ili starosnu dob osobe.

U višestrukoj linearnoj regresiji za predviđanje zavisne varijable koristi se nekoliko neovisnih varijabli. Dakle, koristite stupanj obrazovanja, tjedno radno vrijeme i starosna dob osobe kako bi se predvidjela njezina plaća. Dakle, razlika između jednostavne i višestruke regresije je u tome što se u jednom slučaju koristi samo jedna nezavisna varijabla, a u drugom slučaju, koristimo nekoliko varijabli.
Obje vrste regresije imaju zajedničko to što je zavisna varijabla metrika. Metričke varijable su, na primjer, plaća osobe, veličina cipela ili potrošnja električne energije.

Za razliku od toga, logistička regresija se koristi kada imamo kategoričku zavisnu varijablu, na primjer, kada želimo saznati je li osoba u opasnosti od preopterećenja ili ne. Kad god imamo da i ne odgovore, koristimo logističku regresiju. Dakle, u linearnim regresijama, zavisna varijabla je metrička u logističkim regresijama je kategorička zavisna varijabla. Kada god je zavisna varijabla, da ili ne, koristi se logistička regresija. Tako, na primjer, je li osoba zdrava ili je bolesna?

U svim slučajevima nezavisne varijable mogu biti nominalne, ordinalne ili metričke. U sva tri slučaja u jednostavnoj linearnoj, u višestrukoj linearnoj i u logističkoj regresiji, ta je zavisna varijabla metrička u linearnom slučaju dok je nominalna ili ordinalna u slučaju logističke regresiju. Metričke odnosno kvantitativne ili brojčane varijable, nominalne odnosno kvalitativne ili nenumeričke varijable i ordinalne odnosno redoslijedne ili uređene varijable su vrste varijabli prema nivou mjerenja. Metričke varijable su one koje imaju numeričku vrijednost i mogu se mjeriti na intervalnoj ili racionalnoj skali. Primjeri metričkih varijabli su visina, težina, temperatura, vrijeme i sl. Nominalne varijable su one koje imaju kategoričku vrijednost i ne mogu se poredati po veličini. One služe samo za označavanje ili klasifikaciju objekata. Primjeri

nominalnih varijabli su spol, nacionalnost, boja očiju i sl. Ordinalne varijable su one koje imaju kategoričku vrijednost i mogu se poredati po veličini, ali ne pokazuju pravi numerički odnos između kategorija. One služe za rangiranje ili ocjenjivanje objekata. Primjeri ordinalnih varijabli su ocjene, ocjene stupnja zadovoljstva, preferencije i sl. [5], [6].

Važno je napomenuti da u slučaju nominalnih ili ordinalnih nezavisnih varijabli, imaju samo dvije karakteristike. Kao što su spol s muškim i ženskim. Ako varijable imaju više od dvije karakteristike, onda se moraju formirati takozvane indikatorske varijable (eng. *dummv* varijable).

3.3.1. Jednostruka linearna regresija

Za početak ćemo najprije upoznati jednostruku regresijsku analizu i to linearni model. Prvi problem koji moramo riješiti pri uspostavi funkcionalne veze između dvije varijable je odabir adekvatnog modela jednostruke regresije, odnosno koja je to funkcija koja najbolje objašnjava relaciju između dviju varijabli. Taj problem rješavamo pomoću korelacijskog dijagrama jer se na osi *x* nalaze rezultati nezavisnih ili prediktorskih varijabli, a na osi *y* rezultati tih entiteta u zavisnoj ili kriterijskoj varijabli. Ako je raspodjela točkica takva da upućuje na linearan odnos između varijabli tada biramo jednostavni linearni model, odnosno pravac. Jedno pitanje na koje možete odgovoriti regresijskom analizom je što utječe na plaću osobe? Da bismo to učinili, možemo uzeti u obzir razinu obrazovanja, tjedno radno vrijeme i starosnu dob osobe. [7].

S druge strane, također možemo predvidjeti plaću osobe na temelju stupnja razine obrazovanja i postignuća, tjednog radnog vremena i starosne dobi.

Cilj jednostruke linearne regresije je predvidjeti vrijednost zavisne varijable na temelju nezavisne varijable. Što je veći linearni odnos između nezavisne varijable i zavisne varijable rezultat predviđanja će nam biti točniji.
Stoga se regresijska analiza koristi u dvije svrhe. Jedna svrha je mjerenje utjecaja jedne ili više varijabli na drugu varijablu, a druga svrha je predviđanje varijable pomoću jedne ili više drugih varijabli. Jednostavna linearna regresija koristi samo jednu nezavisnu varijablu za predviđanje zavisne varijable, u primjeru, gdje želimo predvidjeti plaću osobe, koristimo također samo jednu varijablu kao na primjer tjedno radno vrijeme ili starosna dob osobe.

Kako ćemo uspostaviti vezu između jedne nezavisne prediktorske varijable i jedne zavisne ili kriterijske? Jednostavno linearnom regresijskom analizom utvrđuje se linearna veza između

jedne nezavisne, odnosno prediktorske i jedne zavisne ili kriterijske varijable, pri čemu regresijska jednadžba ima sljedeći oblik:

$$y_i = b_0 + b_1 x + e_i$$
$$i = 1,2,\dots,n$$

Gdje je:

y_i - rezultat entiteta u kriterijskoj varijabli,

b_0 i b_1 - regresijski koeficijenti,

x_i - rezultat entiteta prediktorskih varijabli

e_i - rezidualna vrijednost odnosno odstupanje prognoziranog rezultata od originalnog izmjerenog rezultata.

Proširenjem modela na multiplu regresiju se dobije, niz od *n* regresijskih jednadžbi koje možemo napisati u matričnom obliku na sljedeći način.

$$\begin{vmatrix} y_1 \\ \vdots \\ y_n \end{vmatrix} = \begin{vmatrix} 1 & x_1 \\ \vdots & \vdots \\ 1 & x_n \end{vmatrix} \cdot \begin{vmatrix} b_0 \\ b_1 \end{vmatrix} + \begin{vmatrix} e_1 \\ \vdots \\ e_n \end{vmatrix}$$

Gdje je:

y - vektor n rezultata entiteta u zavisnoj varijabli,

x - matrica reda n × k rezultata entiteta u nezavisnoj varijabli,

b - vektor regresijskih koeficijenata,

e - vektor rezidualnih rezultata n entiteta.

Dakle, problem se svodi na izračunavanje nepoznatog vektora regresijskih koeficijenata b i vektora rezidualnih vrijednosti e. Koeficijenti regresijskog pravca utvrđuju se metodom najmanjih kvadrata.

3.3.1.1. Metoda najmanjih kvadrata

Metoda najmanjih kvadrata temelji se na uvjetu da je suma kvadrata rezidualnih vrijednosti, odnosno odstupanja prognoziranih od stvarnih rezultata u kriterijskoj varijabli minimalna. Gdje je y_i rezultat entiteta i u kriterijskoj varijabli, y_i' je prognozirani rezultat entiteta i u kriterijskoj varijabli, a e rezidualna vrijednost entiteta i, dakle metodom najmanjih kvadrata izračunaju se regresijski koeficijenti b_0 i b_1. Regresijski koeficijent b_0 predstavlja odsječak na osi zavisne varijable, odnosno predstavlja vrijednost zavisne varijable, ukoliko je vrijednost nezavisne varijable jednaka 0. Koeficijent b_1 u regresiji odražava nagib pravca, tj. pokazuje koliko se, u prosjeku, zavisna varijabla mijenja u linearnoj vezi za svaki jedinični porast vrijednosti nezavisne varijable. [1], [6].

Rezidualni vektor se jednostavno izračuna tako da od originalnih, odnosno stvarnih rezultata koje smo dobili mjerenjem, oduzmemo prognozirane rezultate, one koje smo dobili pomoću izračunatih regresijskih koeficijenata. Rezidualne vrijednosti normalno su distribuirane oko regresijskog pravca sa konstantnom varijancom, odnosno standardnom devijacijom za sve vrijednosti nezavisne, odnosno prediktorske varijable.

A varijanca, odnosno standardna devijacija rezidualnih vrijednosti izračuna se ovom formulom:

$$\sigma_e^2 = \frac{r_{ss}}{df} = \frac{\sum_{i=1}^{n}(y_i - y_i')^2}{n-2}$$

Na kraju se još izračuna koeficijent korelacije između kriterijske i prediktorske varijable kojim se izračunava veličina, odnosno jačina njihove linearne povezanosti. U slučaju regresijske analize koeficijent korelacije, dakle mjera jačine funkcionalnog odnosa između prediktorske i kriterijske varijable. Ako je koeficijent korelacije jednak 0, onda to znači da nezavisna varijabla nema nikakav utjecaj na kriterijsku varijablu, a ako je koeficijent korelacije jednak 1, onda to znači da cjelokupan varijabilitet kriterijske varijable možemo pripisati utjecaju nezavisne ili prediktorske varijable.

$$b_0 = \frac{\sum_{i=1}^{n} y_i \sum_{i=1}^{n} x_i^2 - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} x_i y_i}{n \sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i\right)^2}$$

$$b_1 = \frac{n \sum_{i=1}^{n} x_i y_i - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i}{n \sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i\right)^2}$$

3.3.2. Višestruka linearna regresija

U različitim istraživanjima znatno se češće utvrđuje funkcionalna veza između više nezavisnih ili prediktorskih varijabli i jedne zavisne ili kriterijske varijable. Višestruka linearna regresija često se koristi u empirijskim društvenim istraživanjima kao i u istraživanju tržišta. U oba je područja od velikog interesa saznati kakav utjecaj različiti čimbenici imaju na određenu varijablu. [4].

Primjerice, veliki broj istraživanja proveden je s ciljem utvrđivanja relacija između antropoloških obilježja i uspjeha u pojedinim aktivnostima. Cilj regresije je procijeniti jednu varijablu na temelju nekoliko drugih varijabli. Funkcionalnu zavisnost dviju ili više nezavisnih varijabli i jedne zavisne varijable moguće je utvrditi višestrukom linearnom regresijskom analizom, pri čemu regresijska jednadžba ima sljedeći oblik:

$$y = b_0 + b_1 x_1 + b_2 x_2 + \cdots + b_m x_m + e$$

Gdje je:

y - zavisna varijabla

$x_1 \dots x_m$ - nezavisne varijable

$b_0 \dots b_m$ - regresijski koeficijenti

e - rezidualna varijabla koja označava pogrešku prognoze.

Problem se svodi na izračunavanje regresijskih koeficijenata uz uvjet da rezidualne vrijednosti budu što je moguće manje. Za rješavanje ovog problema potrebno je imati veći broj entiteta od broja nezavisnih varijabli kako bi se dobio dovoljan broj linearnih jednadžbi za izračunavanje regresijskih koeficijenata. Ako *n* entiteta opišemo jednom zavisnom i sa *m* nezavisnih varijabli gdje je broj entiteta veći od broja nezavisnih varijabli, dobije se sustav od *m* linearnih jednadžbi. Koji možemo zapisati i u matričnom obliku na sljedeći način.

$$\begin{vmatrix} y_1 \\ \vdots \\ y_n \end{vmatrix} = \begin{vmatrix} 1 & x_{11} & \cdots & x_{1m} \\ \vdots & \vdots & \cdots & \vdots \\ 1 & x_{n1} & \cdots & x_{nm} \end{vmatrix} . \begin{vmatrix} b_0 \\ b_1 \\ \vdots \\ b_m \end{vmatrix} + \begin{vmatrix} e_1 \\ \vdots \\ e_n \end{vmatrix}$$

Gdje je:

y - vektor rezultata entiteta u zavisnoj varijabli,

x - matrica rezultata entiteta u nezavisnim varijablama,

b - vektor regresijskih koeficijenata

e – vektor rezidualnih vrijednosti.

3.3.2.1. Metoda najmanjih kvadrata

Metoda najmanjih kvadrata izračunaju se regresijski koeficijenti za jednadžbu $y = xb + e$ uz uvjet da suma kvadrata rezidualnih vrijednosti, odnosno odstupanja izmjerenih vrijednosti od prognoziranih bude minimalna. Prvi korak u rješavanju ovog problema je množenje formule $y = xb$ / x^Ts transponiranom matricom x, a transponirana matrica x dobije se tako da se reci zamjene stupcima, a stupci recima. Nakon toga dobivenu formulu pomnožimo sa inverzom matrice $x^T x$ kako slijedi

$x^T y = x^T \times b$ / $(x^T x)^{-1}$ i time dobijemo ovu formulu:

$$b = (x^T x)^{-1} x^T y$$

S obzirom da priloženi umnožak daje matricu identiteta, a množenje bilo koje matrice matricom identiteta ostavlja matricu nepromijenjenu, uostalom kao i broj jedan u skalarnoj algebri, dobije se vektor regresijskih koeficijenata. Vektor regresijskih koeficijenata ima m +1 regresijskih koeficijenata. Gdje je:

m – broj nezavisnih varijabli

b_0 – regresijski koeficijent

Predstavljena je vrijednost zavisne varijable kad su vrijednosti nezavisnih varijabli nula.

b_1 do b_m su regresijski koeficijenti i oni predstavljaju iznos promjena vrijednosti zavisne varijable za porast vrijednosti jedinično od određene nezavisne varijable. Pri tome se treba držati uvjeta da su vrijednosti preostalih nezavisnih varijabli uvijek iste.

Dakle, ako se vrijednost neke nezavisne varijable poveća za 1, uz uvjet da su ostale vrijednosti nezavisnih varijabli ostale iste, odnosno ostale konstantne, vrijednost zavisne varijable povećat će se u prosjeku za vrijednost odgovarajućeg regresijskog koeficijenta koji pripada toj varijabli. Rezidualni vektor e dobije se oduzimanjem prognoziranih rezultata entiteta od izmjerenih $e = y - y'$. A standardna pogreška prognoze, odnosno standardna devijacija izmjerenih rezultata u odnosu na prognozirane, izračuna se ovom formulom.

$$\sigma_e = \sqrt{\frac{\sum_{i=1}^{n} \left(y_i - y_i'\right)^2}{n - (m+1)}}$$

$$k = \beta_1 z_1 + \beta_2 z_2 + \cdots + \beta_m z_m + \varepsilon$$

$$k = z\beta + \varepsilon$$

Ako se rezultati dobiveni mjerenjem n entiteta u zavisnoj varijabli i m nezavisnih varijabli standardiziraju, odnosno transformiraju u z vrijednosti. Onda dobijemo takozvani standardizirani oblik linearne regresijske jednadžbe gdje je k vektor standardiziranih rezultata entiteta u zavisnoj varijabli. $\underline{Z}$ je matrica standardiziranih rezultata entiteta u nezavisnim varijablama. β je vektor standardiziranih regresijskih koeficijenata, a ε je vektor standardiziranih rezidualnih vrijednosti. Istim postupkom za izračunavanje regresijskih koeficijenata pod modelom najmanjih kvadrata, računamo i standardizirane regresijske koeficijente, uz dodatak dijeljenja sa brojem entiteta. Time dobijemo vektor korelacija zavisne i nezavisnih varijabli i matricu interkorelacije nezavisnih varijabli. Dakle, sad imamo da je $r = R\beta$ te ako tu formulu pomnožimo sa inverzom matrice R kako slijedi $r = R\beta \;/\; R^{-1}$ dakle matrice korelacija između nezavisnih varijabli dobijemo vektor standardiziranih regresijskih koeficijenata. Standardizirani regresijski koeficijenti reprezentiraju veličinu promjene zavisne varijable izraženu u jedinicama standardne devijacije za svaki jedinični porast standardizirane

vrijednosti određene nezavisne varijable, pretpostavljajući da ostale nezavisne varijable ostanu nepromijenjene. S obzirom da su standardizirani regresijski koeficijenti izračunati na standardiziranim varijablama, oni predstavljaju relativan doprinos svake nezavisne varijable prognozi zavisne varijable pa ih možemo smatrati koeficijentima utjecaja određene nezavisne varijable na zavisnu. Koeficijent determinacije multiple korelacije izračunava se prema formuli.

$$\rho^2 = \sum_{i=1}^{m} \beta_i r_i = \sum_{i=1}^{m} p_i$$

A onda se multipla korelacija izračuna kao drugi korijen iz koeficijenta determinacije $\rho = \sqrt{\rho^2}$. Multipla korelacija zapravo predstavlja mjeru povezanosti skupa nezavisnih varijabli i zavisne varijable. I ako je vrijednost multiple korelacije jednaka 0, tada skupom nezavisnih varijabli nije moguće predviđati zavisnu varijablu. Ako je multipla korelacija jednaka 1, dakle ima maksimalnu vrijednost, tada je cjelokupan varijablitet zavisne varijable moguće pripisati utjecaju nezavisnih varijabli, odnosno rezultate u zavisnoj varijabli moguće je točno predviđati pomoću nezavisnih varijabli. Dakle, multipla korelacija pokazuje kolika je vrijednost svih nezavisnih varijabli u predviđanju, odnosno objašnjavanju neke zavisne varijable.

Nakon što smo izračunali sve bitne pokazatelje regresijskog modela, testiramo njihovu statističku značajnost. Prvo se testira multipla korelacija budući je ona mjera doprinosa svih nezavisnih varijabli objašnjenju zavisne varijable i u tu svrhu postavljamo dvije hipoteze. Nultu hipotezu koja glasi multipla korelacija nije statistički značajna uz pogrešku p i alternativnu hipotezu koja glasi multipla korelacija statistički je značajna uz pogrešku p, koja je najčešće 1 ili 5 posto. Testiranje postavljenih hipoteza vrši se pomoću F-testa prema tablici. Izračunata F vrijednost uspoređuje se sa kritičnom F vrijednosti koja se dobije uz pomoć F distribucije za određeni broj stupnjeva slobode i pogrešku statističkog zaključka p. Ako je izračunata F vrijednost manja od kritične f vrijednosti, prihvaća se nulta hipoteza i zaključuje da nema statistički značajne povezanosti između analiziranog skupa nezavisnih varijabli i zavisne varijable. Ako je izračunata F vrijednost veća od kritične, odbacujemo nultu hipotezu, prihvaćamo alternativnu i zaključujemo da postoji statistički značajna povezanost između skupa nezavisnih varijabli i zavisne varijable uz pogrešku p. Za testiranje statističke značajnosti pojedinog regresijskog koeficijenta postavljaju se također nulta i alternativna hipoteza. Nulta hipoteza glasi, regresijski koeficijent, dakle određeni b_j nije statistički značajan uz pogrešku p i alternativna koja glasi regresijski koeficijent b_j statistički je značajan uz pogrešku p.

Testiranje statističke značajnosti regresijskih koeficijenata vrši se tako da se određeni regresijski koeficijent b_j podijeli sa svojom pripadajućom standardnom pogreškom.

$$t = \frac{b_j}{\sigma_{b_j}} \qquad \sigma_{b_j} = \frac{\sigma_e}{\sqrt{\sum_{t=1}^{n} (x_i - \bar{x})^2}}$$

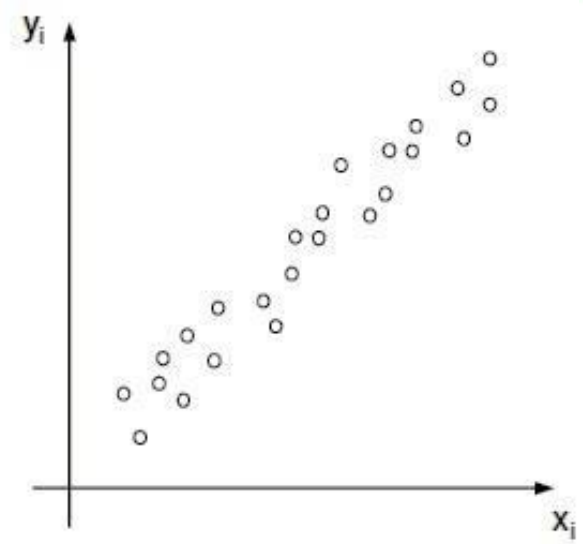
Ako je apsolutna |t| vrijednost manja od kritične t vrijednosti prihvaćamo nultu hipotezu. Odnosno, ako je apsolutna $|t|$ vrijednost veća od kritične t vrijednosti odbacujemo nultu hipotezu i prihvaćamo alternativnu uz pogrešku p. Kritična t vrijednost dobiva se uz pomoć studentove t distribucije, zavisno o broju stupnjeva slobode i pogrešci statističkog zaključka. [8].

3.4. Zahtjevi linearne regresije

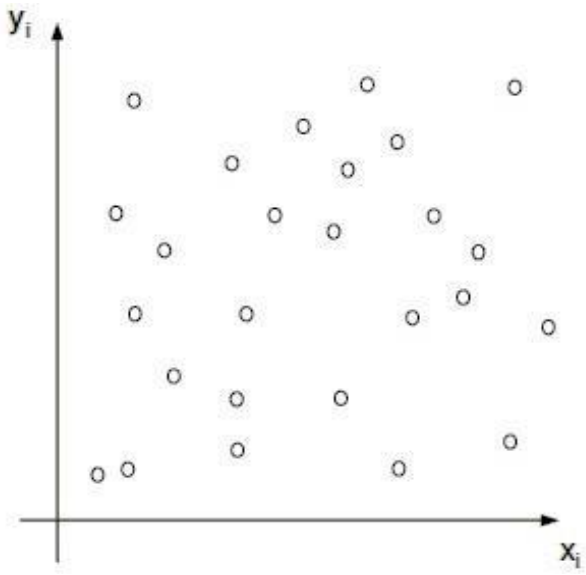
Zahtjevi za linearnu regresiju su sljedeći: [9].

- Mora postojati linearni odnos između zavisnih i nezavisnih varijabli.
- Greška epsilon mora biti normalno raspoređena.
- Ne smije biti multikolinarnosti, ako bi postojala multikolinarnost, to bi značilo da su regresijski koeficijenti nestabilni.
- Ne smije biti heteroskedastičnosti, odnosno heterogenost varijance, to znači da varijanca reziduala mora biti konstantna u predviđenim vrijednostima.

U linearnoj regresiji kroz podatke se povlači ravna linija. Ako je relacija nelinearna, ravna linija ne može ispuniti ovaj zahtjev. Pogledajmo ove dvije slike. U prvom grafikonu vidimo gotovo savršen linearni odnos između zavisne i nezavisne varijable.



[10]. Slika 1: Pozitivna veza



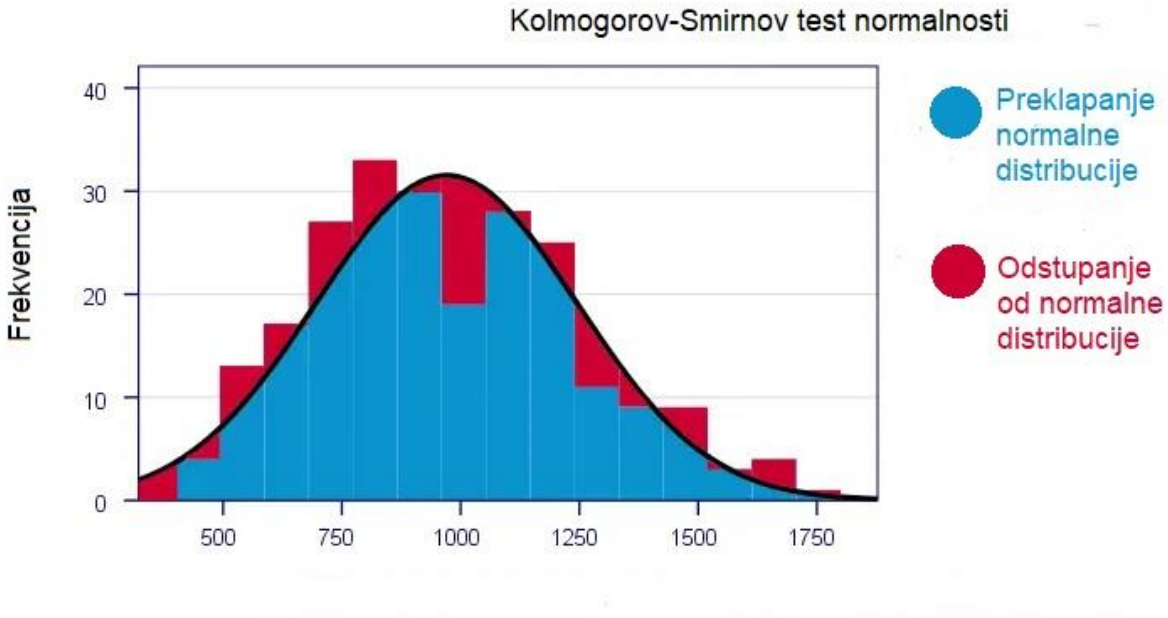
Slika 2: Nema veze među entitetima

Ovdje je moguće i regresijsku liniju postaviti u graf na razuman način. U drugom slučaju, vidimo da ne postoji veza između zavisne i nezavisne varijable. Stoga nije moguće smisleno postaviti regresijsku liniju kroz točke.

Budući da ne postoji linearni odnos u ovom slučaju. Koeficijenti regresijskog modela ne mogu se smisleno interpretirati ili se mogu pojaviti pogreške u predviđanju koje su veće od očekivanih. Stoga je važno na početku provjeriti postoji li linearna veza između zavisne i nezavisne varijable.

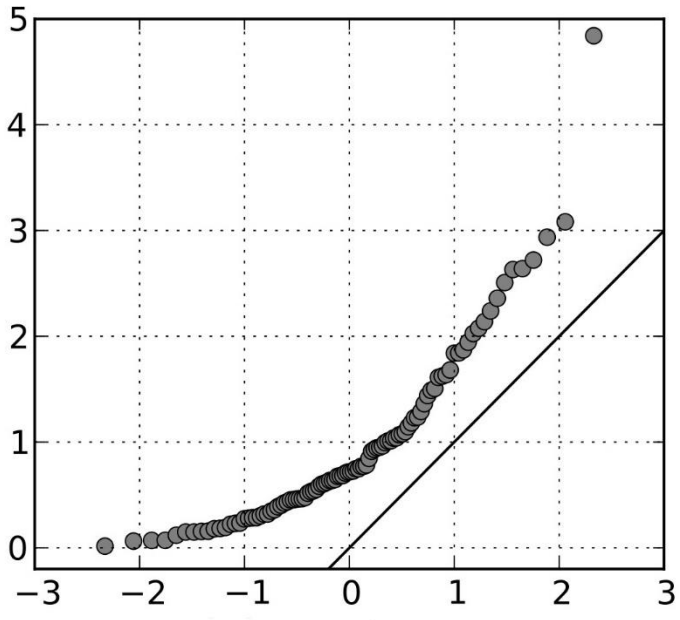
Ovo se obično radi grafički. Drugi uvjet je da greška epsilon mora biti normalno raspoređena, kako bi se to provjerilo, postoje dva načina. Jedan je analitički, a drugi grafički način. Na analitički način možemo koristiti Kolmogorov-Smirnov test ili Shapiro-Wilk test.
Ako je p-vrijednost testiranja veća od 0,05, nema odstupanja podataka od normalne distribucije. Dakle, možemo pretpostaviti da su podaci normalno distribuirani. Međutim, ti se analitički testovi koriste rjeđe jer je teže uvijek potvrditi normalnu distribuciju u malim uzorcima. I vrlo brzo postaju značajni u velikim uzorcima.

Dakle, odbacuju nultu hipotezu da se podaci normalno distribuiraju u velikim uzorcima, vrlo brzo. Zbog toga se češće koristi grafička opcija. Kako bismo saznali postoji li normalna raspodjela pogreške, kada gledamo grafičku opciju, možemo pogledati histogram:



[5] Slika 3: Kolmogorov-Smirnov test normalnosti

Zatim, bolje je pogledati tzv. QQ dijagram (eng. Quantile-Quantile Plot).



Slika 4: Quantile-quantile plot dijagram

Kada pogledamo QQ dijagram, moramo saznati jesu li podatkovne točke blizu linije, što su podatkovne točke bliže liniji. Bolja je normalna distribucija podataka. Sada dolazimo do trećeg zahtjeva, a to je homoskedastičnost. Zahtjev za linearnu regresiju je da reziduali imaju konstantnu varijancu.

Budući da naš regresijski model nikada točno ne predviđa, naša zavisna varijabla u praksi uvijek imamo određenu pogrešku. Sada možemo iscrtati zavisnu varijablu na x-osi i strelicu na y-osi. Pogreška bi se sada trebala ravnomjerno raspršiti po cijelom rasponu. Ako je to slučaj, prisutna je homogenost varijance, odnosno homoskedastičnost.

U slučaju heteroskedastičnosti imamo različitu varijancu pogreške ovisno o rasponu vrijednosti zavisne varijable. Ako je to slučaj, zahtjev za linearnu regresiju nije ispunjen.
Dakle, prvo pitanje, što je multikolinearnost? Multikolinearnost znači da su dvije ili više neovisnih varijabli međusobno snažno povezane.

Problem kod multikolinearnosti je taj što se učinak pojedinačnih varijabli ne može jasno razdvojiti. Pogledajmo regresijsku jednadžbu. Opet, ovdje imamo zavisnu varijablu i nezavisnu varijablu s odgovarajućim koeficijentima. Na primjer, ako postoji visoka korelacija između x_1 i x_2, ili ako su ove dvije varijable gotovo jednake, tada je prilično teško odrediti b_1 i b_2.

Ako su obje varijable potpuno jednake, regresijski model ne zna kako odrediti b_1 i b_2. To znači da regresijski model postaje nestabilan. Ako sada želimo koristiti regresijski model za predviđanje, to nije važno. Ako postoji multikolinearnost u predviđanju, jedino nas zanima koliko je dobro predviđanje, ali nas ne zanima koliki je utjecaj odgovarajućih varijabli. Međutim, ako se regresijski model koristi za mjerenje utjecaja nezavisna varijabla na zavisnu varijablu ne smije postojati multikolinearnost a ako postoji, koeficijenti se ne mogu smisleno interpretirati.

Dakle, sljedeće pitanje je, kako sada možemo otkriti multikolinarnost? Ako ponovno pogledamo regresijsku jednadžbu, imamo varijablu x_1 x_2 i na varijablu x_k. Sada želimo znati je li x_1 posve identičan bilo kojoj drugoj varijabli ili kombinaciji drugih varijabli.

Da bismo to učinili, jednostavno postavljamo regresijski model u ovom novom regresijskom modelu, uzimamo x_1 kao novu zavisnu varijablu, ako sada možemo vrlo dobro predvidjeti x_1 iz drugih nezavisnih varijabli, više nam ne treba x_1 jer umjesto toga možemo koristiti druge varijable, ako bismo sada koristili sve varijable, moglo bi biti da regresijski model postane vrlo nestabilan.

Sada bismo to mogli učiniti za sve ostale varijable. Dakle, sada procjenjujemo x_2 koristeći druge varijable. I procjenjujemo x_k pomoću druge varijable. U ovom slučaju imamo k novih regresijskih modela. Za svaki od ovih regresijskih modela izračunavamo toleranciju i faktor inflacije varijance.

Tolerancija se dobiva $T = 1 - R^2$ gdje je R^2 koeficijent determinacije ili objašnjenje varijance, Faktor inflacije varijance je $FIV = \frac{1}{1-R^2}$. Multikolinearnost može postojati ako je tolerancija manja od 0,1, ako pogledamo faktor inflacije varijance, može postojati multikolinarnost, ukoliko je faktor inflacije varijance veći od 10. [datatab.net/youtube video]

4. Klasifikacija podataka

Nadzirano učenje može se primijeniti za rješavanje dviju različitih vrsta problema, odnosno, za obavljanje klasifikacije ili regresije. Kod klasifikacije primjeru pridružujemo klasu kojoj taj primjer pripada. Kod regresije primjeru pridružujemo neku kontinuiranu vrijednost. Razlika tih dvaju postupaka leži u tome je li ciljna varijabla diskretna ili nominalna, odnosno klasifikacija ili je kontinuirana, odnosno regresija. Razmotrimo klasifikaciju.
Klasifikacija podataka je jedna od najčešćih primjena strojnog učenja. Cilj klasifikacije je razvrstati podatke u različite kategorije ili klase na temelju njihovih karakteristika. Klasifikacija se koristi u mnogim područjima, kao što su medicina, financije, marketing i sl. [11], [12], [13].

4.1. Metode i algoritmi

Osnovna podjela klasifikacijskih postupaka jest na generativne i diskriminativne modele. Razlikuju se po tome kako modeliramo pripadnost primjera klasi. Analogno vrijedi i za modele regresije. Neki tipični generativni modeli su Bayesov klasifikator, mješavina Gaussovih distribucija (eng. *Gaussian mixture model*, GMM), latentna Dirichletova alokacija (eng. *latent Dirichlet allocation*, LDA), Bayesove mreže i skriveni Markovljev model (eng. *hidden Markov models*, HMM). Tipični primjeri neprobabilističkih diskriminativnih modela jesu perceptron, višeslojni perceptron (eng. *multilayer perceptron*, MLP), stroj s potpornim vektorima (eng. *support vector machine*, SVM), stabla odlučivanja, k-najbližih susjeda (KNN) i linearna diskriminantna analiza (LDA). Neovisno o podjeli, s obzirom na odnos između broja primjera za učenje i broja parametara modela, nadzirani postupci mogu se podijeliti na parametarske i neparametarske modele.
U nastavku su opisani neki od najčešće korištenih. [14], [19].

4.1.1. Logistička regresija

Jedan od probabilističkih diskriminativnih modela jest logistička regresija. Iako naziv sugerira da se radi o regresiji, radi se ipak o klasifikacijskom algoritmu. Model je diskriminativan, ali za razliku od diskriminativnih modela, daje izlaz koji ima vjerojatnosno tumačenje. Logistička regresija se često koristi u problemima klasifikacije u kojima je potrebno donijeti odluku na temelju dvije moguće vrijednosti. [8], [14].

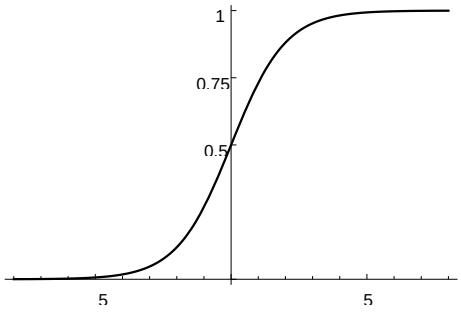
4.1.1.1. Model logističke regresije

4.1.1.2.

Logistička funkcija, također poznata kao sigmoidna funkcija
Ili jednostavno sigmoida. Ta je funkcija definirana ovako:

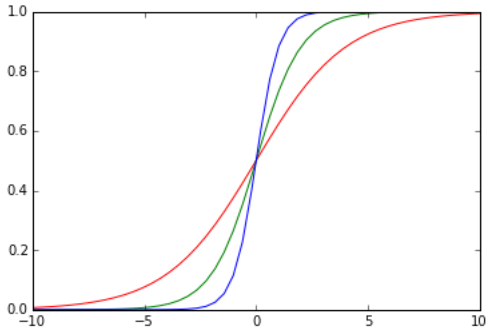
$$\sigma(a) = \frac{1}{1 + \exp(-a)}$$

Ovako izgleda graf logističke funkcije.



[14] Slika 5: Logistička funkcija [14]

Kut nagiba sigmoide može biti prilagođen množenjem ulaznih vrijednosti sa odgovarajućim faktorom.



[14] Slika 6: promjena nagiba krivulje ovisno o faktoru[14]

Plavom krivuljom prikazan je graf funkcije $\sigma(2a)$, crvenom graf funkcije $\sigma(0{,}5a)$, a zelenom graf funkcije $\sigma(a)$. Možemo vidjeti da se množenjem ulaza sigmoide konstantom većom od 1 nagib sigmoide povećava, odnosno postaje strmija. Sigmoida ima tri važne karakteristike koji ju čine pogodnom. Prvo, vrijednosti gnječi na otvoreni interval (0,1), što znači da ćemo vrijednosti sigmoide moći interpretirati kao vjerojatnosti. Drugo, oblikom je slična funkciji praga, što znači da će davati vrijednost blizu 1 primjerima iz jedne klase, a vrijednost blizu 0 primjerima iz druge klase. Treće, funkcija je derivabilna, što će nam omogućiti da izvedemo učinkovit postupak optimizacije.

Model logističke regresije izgleda ovako.

$$h(x; w) = \sigma\left(w^T \phi(x)\right) = \frac{1}{1+\exp\left(-w^T \phi(x)\right)}$$

gdje je uvrštena funkcija preslikavanja iz ulaznog prostora u prostor značajki, ϕ, kako bi mogli ostvariti nelinearnost granice između klasa. Model je funkcija h koja je parametrizirana težinama w definirana kao sigmoida omotana oko skalarnog produkta vektora težina i vektora značajki. Ovo je binarni klasifikacijski model. Granica između dviju klasa definirana je hiperravninom u prostoru značajki za koju vrijedi $h(x; w) = 0{,}5$.

Model logističke regresije svim primjerima dodjeljuje vrijednosti iz intervala (0,1) koje se dobiju primjenom logističke funkcije na skalarni produkt vektora težina i vektora značajki. Primjeri koji su daleko od granice imat će vrijednosti blizu 1, odnosno blizu 0, u odnosu na kojoj su strani od hiperravnine, ali nikada ne izvan tog intervala. Možemo interpretirati vrijednost hipoteze h(x) kao vjerojatnost da primjer pripada pozitivnoj klasi, tj. klasi gdje je y=1,4,1,1,3.

4.1.1.3. Pogreška unakrsne entropije

Funkcija pogreške je očekivanje funkcije gubitka. Prvi korak jest da odaberemo distribuciju kojom bismo modelirali distribuciju oznaka y za zadani primjer x uslijed postojanja šuma, tj. distribuciju za $P(y|x)$. Kod logističke regresije, budući da provoditmo klasifikaciju, y nije broj, nego je diskretna vrijednost 0 ili 1. U vjerojatnosnom smislu, to je onda binarna slučajna varijabla, odnosno varijabla koja može poprimiti vrijednost $y = 1$ (primjer pripada klasi) ili $y = 0$ (primjer ne pripada klasi) s nekom vjerojatnošću μ odnosno 1 - μ. U teoriji vjerojatnosti se takva varijabla naziva Bernoullijeva varijabla. Kod logističke regresije oznaku y se za primjer x modelirat ćemo Bernoullijevom distribucijom. Modeliranjem oznake kao slučajne varijable uvažavamo činjenicu da na oznaku utječe šum, tj. da je oznaka koju u skupu D opažamo samo jedna moguća realizacija slučajne varijable. Distribucija Bernoullijeve varijable definirana izgleda ovako:

$$P(y|\mu) = \{\mu \; ako \; y = 1; \; \mu_{-1 \; inače}\}$$

gdje μ definira vjerojatnost nastupanja događaja, odnosno vjerojatnost $P(y|\mu) = 1$. U nastavku trebati računati derivacije izraza koje sadrže vjerojatnosti Bernoullijeve varijable, a ako–inače izraze ne znamo derivirati. Potrebno je onda da ako–inače izraz napišemo kao njemu ekvivalentan algebarski izraz, ali s operacijama koje znamo derivirati. To možemo lako napraviti, ovako:

$$P(y|\mu) = \mu^y \cdot (1-\mu)^{1-y}$$

Vrijednost μ zapravo je vjerojatnost da je primjer klasificiran u klasu $y = 1$. Tu vjerojatnost nam daje hipoteza, odnosno izlaz sigmoidne funkcije. Drugim riječima, vjerojatnost da je primjer x označen kao $y = 1$ je:

$$P(y = 1|x) = \mu = h(x; w)$$

Općenito, za zadani primjer x, vjerojatnost njegove oznake (bilo $y = 1$ ili $y = 0$) prema definiciji za Bernoullijevu distribuciju jednaka je:

$$P(y|x) = h(x; w)^y (1 - h(x; w))^{1-y}$$

Umjesto μ uvrsti se izlaz modela $h(x; w)$, jer je izlaz modela jednak vjerojatnosti da je primjer klasificiran u klasu $y = 1$.

Sada je potrebno definirati logaritam vjerojatnosti oznaka, uz pretpostavku Bernoullijeve distribucije za varijablu oznake y. Ta vjerojatnost se tretira kao funkcija parametra w, i nju po tom parametru želimo maksimizirati. Zatim, negativna vrijednost te funkcije jednaka je funkciji pogreške, koju onda želimo minimizirati.

Ukupna vjerojatnost skupa oznaka je:

$$\boldsymbol{P}_{(y|x)} = \prod_{i=1}^{N} \boldsymbol{p}\left(\boldsymbol{y}^{(i)} \middle| \boldsymbol{x}^{(i)}\right)$$

Ukupna, odnosno zajednička izglednost značajnih podataka definirana je kao umnožak po svim primjerima od 1 do $\boldsymbol{N}$, pojedinačnih vjeroajatnosti za oznake $\boldsymbol{y}^{(i)}$ ako je ulazni primjer $\boldsymbol{x}^{(i)}$.

$$\ln \boldsymbol{P}_{(y|x)} = \ln \prod_{i=1}^{N} \boldsymbol{p}\left(\boldsymbol{y}^{(i)} \middle| \boldsymbol{x}^{(i)}\right)$$

$$= \sum_{i=1}^{N} \ln p\left(y^{(i)} \middle| x^{(i)}\right)$$

$$= \sum_{i=1}^{N} \ln(h(x; w)^y (1 - h(x; w))^{1-y})$$

$$= \sum_{i=1}^{N} (-y^{(i)} \ln h(x; w) + (1 - y^{(i)} \ln \left(1 - h(x; w)\right))$$

$$E(w|D) = \sum_{i=1}^{N} (-y^{(i)} \ln h(x^{(i)}; w) + (1 - y^{(i)} \ln \left(1 - h\left(x^{(i)}; w\right)\right))$$

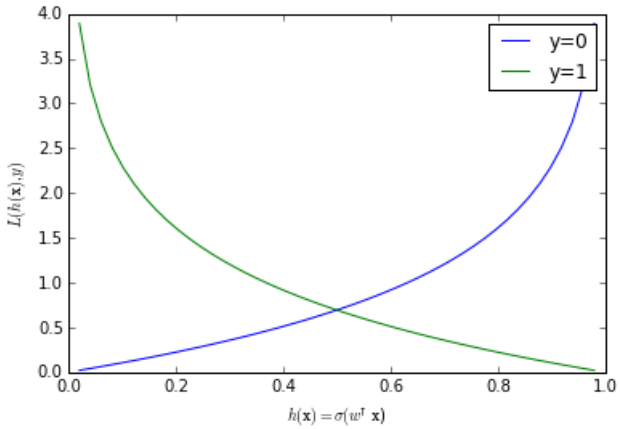
Empirijska pogreška jednaka je negativnoj vrijednosti funkcije, a kako ne bi ovisila o broju primjera skalira se sa faktorom $\frac{1}{N}$.

$$E(w|D) = \frac{1}{N} \sum_{i=1}^{N} (-y^{(i)} \ln h(x^{(i)}; w) + (1 - y^{(i)} \ln \left(1 - h\left(x^{(i)}; w\right)\right))$$

Ova pogreška naziva se unakrsna entropija (eng. *cross-entropy error*), a iz nje možemo izravno dobiti funkciju gubitka koja izgleda ovako:

$$L\big(y, h(x)\big) = -y \ln h(x) - (1-y) \ln\big(1 - h(x)\big)$$

Funkcija gubitka L je funkcija dvije varijable te se može prikazati u dva slučaja, kada je $y = 1$ i kada je $y = 0$. Graf će izgledati ovako:



Slika 7: Graf funkcije gubitka [14]

Ako je $y = 1$, a na ovome grafu ta je krivulja označena zelenom bojom, možemo ustanoviti da će gubitak biti 0 ili vrlo blizu te vrijednosti u situacijama kada je izlaz modela $h(x) = 1$ ili blizu te vrijednosti, odnosno klasifikacija se smatra ispravnom ako je izlaz modela visok u slučaju da je $y = 1$ gubitak je malen. Suprotno plava krivulja pokazuje slučaj za oznaku $y = 0$ u tom slučaju gubitak će biti najmanji ako je i izlaz modela $h(x) = 0$ ili vrlo blizu 0. U slučaju da je $y = 1$, a izlaz modela $h(x) = 0$ imat ćemo vrlo visok gubitak. Slično je I za oznaku primjera $y = 0$, a izlaz modela $h(x) = 1$ i u tom slučaju imamo vrlo velik gubitak. Gubitci su izjednačeni u točki 0,5 u kojem se primjeri klasificiraju u jednu ili drugu klasu ovisno o tome na kojem se dijelu hiperravnine nalaze. Gubitaka nema samo onda kada je primjer savršeno klasificiran, a to je za $y = 1$ $h(x) = 1$, odnosno za $y = 0$ $h(x) = 0$. U svim ostalim slučajevima postoji određeni gubitak.

$$L\big(w^T \phi(x)\big) = \ln\big(1 + \exp(-w^T \phi(x) y)\big)$$

Funkcija $L\big(w^T \phi(x)\big)$ je logistička funkcija gubitka (eng. Logistic loss). Zbog vrijednosti funkcije koja za vrijednosti koje se nalaze točno na granici $w^T \phi(x) = 0$ iznosi $\ln\big(1 + \exp(-w^T \phi(x) y)\big) = \ln 2$ treba napraviti korekciju kako bi se funkcija mogla usporediti s ostalim funkcijama gubitka na grafu čija je vrijednost $L = 1$. To skaliranje suštinski ne utječe puno već samo skalira gubitak za konstantan faktor radi kasnije lakše usporedbe.

$$L\left(w^T\phi(x)\right) = \frac{1}{\ln 2}\ln\left(1+\exp(-w^T\phi(x)y)\right)$$

4.1.1.4. Gradijentni spust za logističku regresiju

Kako je funkcija pogreške prosjek funkcija gubitka po svim primjerima, onda je i gradijent funkcije pogreške jednostavno prosjek gradijenata funkcije gubitka: [15].

$$\nabla_w E(w;D) = \frac{1}{N}\sum_{i=1}^{N}(-y^{(i)}\ln h\left(x^{(i)};w\right) - \left(1-y^{(i)}\right)\ln h\left(x^{(i)};w\right)))$$

$$\nabla_w E(w;D) = \frac{1}{N}\sum_{i=1}\nabla_w L(y^{(i)}, h\left(x^{(i)};w\right))$$

Za stohastički gradijentni spust trebat će nam samo ∇L, dok ćemo za standardni, odnosno grupni gradijentni spust ukupni gradijent dobiti zbrajanjem gradijenata za pojedinačne primjere, prema gornjoj formuli. Izračunajmo onda najprije gradijent funkcije gubitka. Funkcija gubitka je: [15].

$$L\left(y, h(x)\right) = -y\ln h(x) - (1-y)\ln\left(1-h(x)\right)$$

A gradijent te funkcije po težinama w je:

$$\nabla_w L(y;h(x)) = \left(-\frac{1}{h(x)} + \frac{1-y}{1-h(x)}\right)h(x)\left(1-h(x)\phi(x)\right)$$

$$= (h(x)-y)\phi(x)$$

Ovo je dovoljno za stohastički gradijentni spust. Ako želimo raditi standardni (grupni) gradijentni spust, onda gradijent funkcije pogreške po težinama w izračunavamo kao:

$$\nabla_w E(w) = \sum_{i=1}(h\left(x^{(i)}\right) - y^{(i)})\phi(x)$$

gdje smo faktor $\frac{1}{N}$ ispustili, jer ga možemo uključiti u stopu učenja. Sada možemo konstruirati algoritam optimizacije težina gradijentnim spustom. [14], [15], [16].

4.1.1.5. Logistička regresija standardni (grupni) gradijentni spust

Pseudo kod izgleda ovako: najprije kreće od toga da vektor težina w inicijalizira kao nulvektor te zatim do konvergencije ponavlja sljedeće. U vektor ∇_w skupljat će se vektor

smjera spuštanja početno inicijaliziran kao nulvektor. Zatim prolazi kroz sve primjere od 1 do njih N i računa izlaz modela h tako da izračuna sigmoidu skalarnog produkta vektora značajki x i vektora težina w za primjer i. Zatim se krećemo suprotno od smjera gradijenta funkcije gubitka te na taj način u vektoru ∇_w gradimo vektor u smjeru spuštanja. Nakon što prođe kroz svih N primjera želimo naći optimalnu stopu učenja η što se ostvaruje linijskim pretraživanjem u smjeru vektora ∇_w. Konačno radi se ažuriranje vektora težina w u smjeru spuštanja pomnoženo sa stopom učenja η. [17].

1: $w \leftarrow (0, 0, \ldots, 0)$
2: ponavljaj do konvergencije
3: $\nabla_w \leftarrow (0, 0, \ldots, 0)$
4: za $i = 1, \ldots, N$
5: $h \leftarrow \sigma(w^T \phi(x^{(i)}))$
6: $\nabla_w \leftarrow \nabla_w - (h - y^{(i)}) \phi(x^{(i)})$
7: η optimum linijskim pretraživanjem u smjeru ∇_w
8: $w \leftarrow w + \eta \nabla_w$

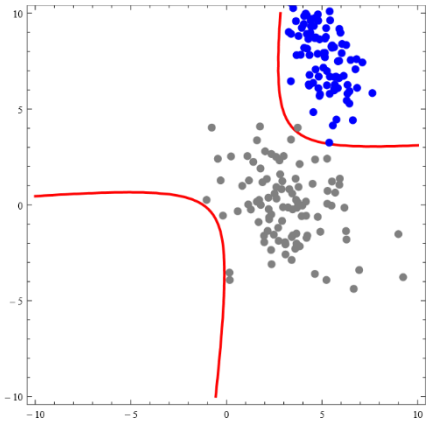
4.1.1.6. Logistička regresija stohastički gradijentni spust

Stohastički gradijentni spust, u usporedbi sa standardnim pristupom, razlikuje se po tome što se ažuriranje težina izvršava unutar petlje koja prolazi kroz sve primjere podataka. Ovaj pristup ne zahtijeva linijsko pretraživanje. Prije svake epohe, primjeri se slučajno permutiraju kako bi se izbjeglo ažuriranje težina u istom redoslijedu primjera, što bi smanjilo stohastičnost gradijentnog spusta. [15], [16], [17].

1: $w \leftarrow (0, 0, \ldots, 0)$
2: ponavljaj do konvergencije
3: slučajno permutiraj primjere u D
4: za $i = 1, \ldots, N$
5: $h \leftarrow \sigma(w^T \phi(x^{(i)}))$
6: $\nabla_w \leftarrow -(h - y^{(i)}) \phi(x^{(i)})$
7: $w \leftarrow w + \eta \nabla_w$

4.1.2. Regularizirana regresija

Algoritam logističke regresije nije imun na problem presloženog modela. Evo primjera presloženog modela logističke regresije:



[14] Slika 8: Granica između klasa u dvodimenzijskom ulaznom prostoru

Na slici je prikazana granica između klasa u dvodimenzijskome ulaznom prostoru što nam pokazuju crvene krivulje, dobivena algoritmom logističke regresije. Ovaj model je očito presložen, jer dobivena hipoteza gotovo pa savršeno klasificira sve primjere iz skupa za učenje, međutim granica je više nelinearna nego što bi to trebala biti. Konkretno, donji desni dio ulaznoga prostora proglašen je regijom plavih primjera, premda u tom prostoru nema niti jedan plavi primjer. Jedan od načina da spriječimo prenaučenost, ili barem smanjimo mogućnost prenaučenosti, jest regularizacija. Regularizacijom težine modela potiskujemo prema nuli, što ostvarujemo dodavanjem regularizacijskog izraza u empirijsku pogrešku definiranog tako da kažnjava hipoteze s velikim težinama.
Kod logističke regresije, možemo govoriti o višestrukom učinku regularizacije:

- Ako je model nelinearan, regularizacijom se sprječava prenaučenost uslijed pretjerane nelinearnosti kao u gornjem primjeru;
- Ako imamo puno značajki, regularizacijom se efektivno smanjuje broj značajki jer se težine potiskuju prema nuli, čime aw sprječava prenaučenost uslijed previše značajki;
- Specifično za logističku regresiju: Ako je problem linearno odvojiv, sprječava se "otvrdnjivanje" sigmoide, odnosno povećanje njezina nagiba, koje bi inače opet dovelo do prenaučenosti [14], [18].

L2 regularizirana logistička regresija

$$\nabla_w E_R(w;D) = \sum_{i=1}^{N}(-y^{(i)} \ln h\left(x^{(i)};w\right) - \left(1-y^{(i)}\right) \ln h\left(x^{(i)};w\right)) + \frac{\lambda}{2} w^T w$$

$$\nabla_w E_R(w;D) = \sum_{i=1}^{N} (h\left(x^{(i)}\right) - y^{(i)})\phi(x)) + \lambda w)$$

Gradijent funkcije po vektoru težina w:

$$w \leftarrow\ w(1-\lambda\eta) - \eta \sum_{i=1}^{N} (h\left(x^{(i)}\right) - y^{(i)})x^{(i)}$$

$$w \leftarrow\ w - \eta(\sum_{i=1}^{N} (-y^{(i)} \ln h\left(x^{(i)};w\right) - \left(1-y^{(i)}\right) \ln h\left(x^{(i)};w\right))) + \lambda w)$$

Pseudokod L2-regularizacije

Pseudokod izgleda ovako: najprije kreće od toga da vektore pomaka i težina (∇_{w_0}, ∇_w)inicijalizira kao nulvektor te zatim do konvergencije ponavlja u petlji sljedeće. Za svaki primjer 1,…,N računa izlaz modela h. Zatim računa grešku za taj primjer i za težinu vektora w i $w_0$radimo ažuriranje vektora spuštanja, nakon što smo taj vektor izgradili zbrajanjem kroz sve primjere. Nakon što smo dobili taj vektor tražimo optimum stope η linijskim pretraživanjem u smjeru ∇_w. Zatim radimo korak pri čemu radimo prigušenje težina za vector w, ali ne radimo prigušenje za vector w_0 jer se ona ne regularizira.

```
1: w ← (0, 0,….,0)
2: ponavljaj do konvergencije
3:     (∇_{w_0}, ∇_w) ← (0, 0,….,0)
4:     za i =1,…,N
5:           h ← σ(w^T φ(x^(i)))
6:           (∇_{w_0}, ∇_w) ← ∇_w-(h-y^(i)) φ(x))
7:     η optimum linijskim pretraživanjem u smjeru ∇_w
8:     w_0 ← w_0 + η∇_{w_0}
9:     w ← w(1-ηλ)+η∇_w
```

Pseudokod za stohastički gradijentni spust

Razlika ovdje jest da ažuriranje težina radimo unutar petlje koja ide po svim primjerima do N. Također razlika je kako bi se povećala stohastičnost radi se permutacija primjera u svakoj iteraciji primjera algoritma stohastičkog gradijentnog spusta.

```
1: w ← (0, 0,….,0)
2: ponavljaj do konvergencije
3:    slučajno permutiraj primjere u D
4:    za i =1,…,N
5:          h ← σ(w^T φ(x̃^(i)))
6:          (∇_w0, ∇_w) ← ∇_w -(h-y^(i)) φ(x))
7:          w_0 ← w_0 + η∇_w0
8:          w ← w(1 − ηλ) + η∇_w
```

4.1.2.1. Višeklasna logistička regresija

Ako imamo više od dvije klase, tj. *K* > 2, potrebno je napraviti multinomijalnu logističku regresiju (MNR, nekad MLR), koju također zovemo i klasifikator maksimalne entropije (eng. *maximum entropy classifier*).

Model

Koristeći poseban vektor w_k težina za svaku od K klasa, potom se skalarni produkt $w_k^T x$ obrađuje kroz odgovarajuću aktivacijsku funkciju, koja osigurava da se vjerojatnosti za sve klase normaliziraju tako da njihova suma bude jednaka 1. Funkcija koja radi baš to naziva se funkcija softmax. Za neki ulazni primjer x, funkcija softmax uzima vrijednosti $w_k^T x$ za svaku od *K* klasa, dakle *K*-dimenzijski vektor, te ih preslikava u *K*-dimenzijski vektor čije se komponente zbrajaju u 1. [13].
Formalno funkcija softmax : $\mathbb{R}^n \rightarrow \mathbb{R}^n$, gdje je *k*-ta komponenta izlaznog vektora jednaka je:

$$softmax_k(x_{1,\dots,}x_k) = \frac{\exp(x_k)}{\Sigma_j \exp(x_j)}$$

Funkcija softmax radi dvije stvari: normalizira sve vrijednosti tako da u zbroju budu 1, ali također pojačava veće vrijednosti i smanjuje manje vrijednosti. Funkcija ima naziv softmax jer odgovara funkciji max, ali je „meka“ u smislu da je, za razliku od funkcije max, neprekidna i diferencijabilna. Model h multinomijalne logističke regresije definirat ćemo kao skup modela th_ku$_k$, gdje je svaki model h_k zadužen za *k*-tu od ukupno *K* klasa. Svaki model h_k definirat ćemo tako da daje vjerojatnost da primjer x pripada klasi *k*, pomoću funkcije softmax:

$$h_k(x; W) = \frac{\exp\left(w_k^T \phi(x)\right)}{\Sigma_j \exp(w_j^T \phi(x))} = P(y = k|x, W)$$

Gdje je $W = (w_1, \dots, w_k)$ matrica sastavljena od *K* vektora težina w_k. Model h_k za klasu k kroz funkciju softmax uzima u obzir izlaze sve ostale $(k-1)$ modele za preostale klase.

Funkcija pogreške

Binarna logistička regresija je izvedena iz definicije negativnog logaritma vjerojatnosti oznaka. Oznake su bile Bernoullijeve varijable $y \in \{0, 1\}$. Zbog toga što izlaz kod ove vrste regresije može imati više od dvije vrijednosti (K > 2) preporučeno je prijeći na kategoričku varijablu koja se ujedno zove i multinoullijeva varijabla.

Takva varijabla se prikazuje kao vektor indikatorskih varijabli, odnosno vektor binarnih varijabli:
$y = (y_1, y_2, \cdots, y_k)^T$
U ovom izrazu $y_k = 1$ ukoliko je ishod varijable k, dok je u svakom drugom ishodu $y_k = 0$. Na primjer: y = (0, 0, 1, 0) označava da je multinoullijeva varijabla poprimila treće stanje od četiri mogućih stanja. Zatim vrijedi da je $\sum_k y_k = 1$. Pritom označimo vrijednost P$(y_k = 1)$ sa μ_k. Poslije toga je potrebno definirati distribuciju te varijable. Zbog toga što distribucija Bernoullijeve varijable ima samo dvije vrijednosti kao što je već spomenuto, ista se definira preko parametra μ na sljedeći način:
$P(y|\mu) = \mu^y(1-y)^{1-\gamma}$
Moguće je poopćiti ovaj izraz na $K \geq 2$. Prije svega, potrebno je imati K parametara pa će biti definiran vektor parametara:
$\mu = (\mu_1, \ldots, \mu_k)$
U ovom izrazu uza parametre μ_k vrijedi $\sum_k \mu_k = 1$ i $\mu_k \geq 0$. To zaključujemo jer pretpostavljamo da ti parametri predstavljaju vjerojatnosti. Zatim, prema analogiji s Bernoullijevom distribucijom, distribucija za kategoričku varijablu se definira kao

$$P(y|\mu) = \prod_{k=1}^{K} \mu_k^{y_k}$$

Iz toga se može zaključiti da kao i kod binarne logističke regresije vjerojatnost da primjer *x* pripada klasi *k* i putem toga moguće je zaključiti sljedeći izraz:

$$h_k(x; W) = \mu_k = P(y = k|x, W) = \frac{\exp\left(w_k^T \phi(x)\right)}{\Sigma_j \exp(w_j^T \phi(x))}$$

Putem tih izraza moguće je napisati logaritam vjerojatnosti oznaka iz *D*:

$$\ln P(y|X) = \ln \prod_{k=1}^{N} P(y^i|x)$$

$$= \ln \prod_{k=1}^{N} \prod_{k=1}^{K} \mu_k^{y_k^i}$$

$$= \ln \prod_{k=1}^{N} \prod_{k=1}^{K} h_k(x^i; W)^{y_k^i}$$

$$= \sum_{i=1}^{N} \sum_{k=1}^{K} y_k^i \ln h_k(x^i; W)$$

Funkcija pogreške koju je potrebno minimizirati je negativan logaritam vjerojatnosti oznaka:

$$E(W|D) = -\sum_{i=1}^{N} \sum_{k=1}^{K} y_k^i \ln h_k(x^i; W)$$

Iz toga je moguće zaključiti da smo došli do poopćenja pogreške unakrsne entropije na *K* klasa. Ujedno je moguće i iz ovoga iščitati da je funkcija gubitka jednaka:

$$L(y, h_k(x)) = -\sum_{k=1}^{K} y_k^i \ln h_k(x^i; W)$$

Isti je način razmišljanja kao i kod binarne logističke regresije koja govori da ako je oznaka y_k^i nekog primjera *i* za klasu *k* jednaka 1 onda je potrebno da za predikciju modela bude visoka vjerojatnost blizu 1. Jer tada $\log h(\mathrm{x}) \approx 0$ i tad će gubitak biti nula. Ukoliko model za primjer čija je oznaka jednaka jedan daje vrijednost blizu nula, logaritam će tad biti velik negativan broj. Gubitak će biti velik jer će njegova negacija biti velik broj.

Nije moguće minimizirati $E(W|D)$ u zatvorenoj formi što je isto kao i kod binarne logističke regresije. Zbog toga je potrebno raditi iterativnu optimizaciju. Za gradijentni spust je moguće prikazati da je gradijent funkcije pogreške jednak:

$$\nabla_{wk} E(W|D) = \sum_{i=1}^{N} (h_k(x^{(i)}; W) - y_k^i)\phi(x^{(i)})$$

Ovo je gradijent po težinama posebno za klasu *k*. Svrha ovoga je da je tad moguće ažurirati težine za svaku klasu posebno. Iz toga je moguće izvesti stohastički gradijentni spust jer ažuriramo težine za svaki primjer i za svaku klasu.

4.1.3. Bayesov klasifikator

Bayesov klasifikator je statistički algoritam koji se temelji na Bayesovom teoremu. Ovaj algoritam se koristi za klasifikaciju podataka pomoću vjerojatnosti. Bayesov klasifikator se često koristi u problemima klasifikacije teksta, kao što su kategorizacija e-pošte i prepoznavanje autora teksta. Bayesov klasifikator je najjednostavniji probabilistički model koji je ujedno i generativni model. Ovaj algoritam se koristi za klasifikaciju podataka pomoću vjerojatnosti.

Pravila vjerojatnosti

Prije nego krenemo s opisom Bayesovog klasifikatora potrebno je sjetiti se teorije vjerojatnosti, a koja je prisutna kod probabilističkih modela. Cijela algebra teorije vjerojatnosti svodi se na dva jednostavna pravila: pravilo zbroja i pravilo umnoška.

Pravilo zbroja je:

$$P(x) = \sum_{y} P(x, y)$$

$$P(x, y) = \sum_{z} P(x, y, z)$$

Vjerojatnost $P(x, y)$je vjerojatnost zajedničke realizacije x i y. U strojnom učenju, to je vjerojatnost da primjer x ima oznaku y. Tu vjerojatnost nazivamo zajednička ili združena vjerojatnost (eng. *joint probability*). Pravilo zbroja nam govori da iz zajedničke vjerojatnosti možemo dobiti vjerojatnost pojedinačnih varijabli ili podskupa varijabli. Ta vjerojatnost se onda zove marginalna vjerojatnost. Zove se marginalna jer se dobiva marginalizacijom, odnosno izračunom vjerojatnosti podskupa varijabli. Marginalizacija, naravno, vrijedi i kada imamo više varijabli. Tada možemo marginalizirati po, npr., samo jednoj varijabli:

$$P(x) = \sum_{y} \sum_{z} P(x, y, z)$$

Drugo pravilo je pravilo umnoška:

Uvjetna vjerojatnost definirana je kao:

$$P(y|x) = \frac{P(x, y)}{P(x)} ili\ P(x|y) = \frac{P(x, y)}{P(y)}$$

To je vjerojatnost vrijednosti y, ako znamo da je ostvarena vrijednost x (ili obrnuto, za drugu formulu). Pravilo umnoška samo je drugi pogled na definiciju uvjetne vjerojatnosti:

$$P(x, y) = P(y|x)P(x) = P(x|y)\ P(y)$$

Ovo su ta dva osnovna pravila. Oba pravila vrijede i ako vjerojatnost P zamijenimo s gustoćom vjerojatnosti *p*. Sve što ćemo mi raditi u nastavku svodit će se na pametnu primjenu ova dva pravila. Sada ćemo primijeniti ova dva pravila kako bismo dobili ključno pravilo koje je potrebno za Bayesov klasifikator, a to je Bayesovo pravilo.

Bayesovo pravilo:

$$P(x, y) = P(y|x)P(x) = P(x|y)P(y)\ /P(x)$$

$$P(y|x) = \frac{P(x|y)P(y)}{P(x)}$$

Ovo pravilo omogućava da preokrenemo smjer zaključivanja: iz vjerojatnosti $P(y|x)$ možemo zaključiti o vjerojatnosti $P(y|x)$. Daje nam mogućnost zaključivanja od podataka

prema klasi, tj. od posljedice prema uzroku. Sada kad imamo potrebnu podlogu možemo početi s Bayesovim klasifikatorom. [20, str. 94).

4.1.3.1. Bayesov klasifikator

$$P(y|x) = \frac{p(x|y)P(y)}{p(x)} = \frac{p(x,y)}{p(x)}$$

Ovdje nam se pojavljuju vjerojatnosti veliko "P" i gustoće vjerojatnosti malo "p", ovisno o tome je li varijabla diskretna ili kontinuirana. Konkretno, varijabla y je diskretna oznaka klase, dok je vektor primjera x općenito vektor značajki, diskretnih ili kontinuiranih, pa pišemo gustoću vjerojatnosti jer je to općenitiji slučaj. To što množimo gustoću vjerojatnosti p s vjerojatnošću P nije nikakav problem jer će rezultat biti gustoća vjerojatnosti. [21].

Vjerojatnost $P(y|x)$ nazivamo aposteriorna vjerojatnost oznake (eng. *posterior*). To je vjerojatnost oznake y za zadani primjer x. I to je ono što nas zapravo zanima: koliko je vjerojatno da primjer x pripada klasi y. Gustoću vjerojatnosti $p(y|x)$ nazivamo izglednost klase (eng. *class likelihood*). To je gustoća vjerojatnosti primjera uz zadanu klasu, odnosno distribucija primjera unutar dotične klase. Vjerojatnost $P(y)$ naziva se apriorna vjerojatnost klase (eng. *class prior*). To je vjerojatnost klase neovisno o primjerima. Konačno, gustoća vjerojatnosti $p(x)$ je gustoća vjerojatnosti primjera neovisno o klasi. [20, str. 403).

$$p(x) = \sum_{y} p(x,y) = \sum_{y} p(x|y)P(y)$$

Faktorizaciju smo napravili tako da smo najprije primijenili pravilo zbroja po svim mogućim oznakama klase, a zatim pravilo umnoška. Ovako modeliran p(x) je jednostavniji jer se zasebno modelira izglednost klase, a zasebno apriorna vjerojatnost klase. Te dvije distribucije mogu se modelirati jednostavnim teorijskim distribucijama. [13].

Model Bayesovog klasifikatora *h*:

$$h_j(x;\theta) = P(y = j|x) = \frac{p(x|y)P(y)}{\Sigma_{y'} p(x|y')P(y')}$$

Objasnite što računamo ovom formulom i sve oznake u formuli. Ovako definirana hipoteza daje nam vjerojatnost klasifikacije u klasu j. Bayesov klasifikator može raditi s više klasa. Imamo po jednu hipotezu h_j za svaku od K klasa.

Ako samo želimo odrediti oznaku primjera, onda ćemo ga klasificirati u klasu čija je vjerojatnost najveća. To je tzv. maksimum aposteriori hipoteza (MAP). [14], [20]. Model tada definiramo kao:

$$h(x;\theta) = \underset{y}{argmax}\ p(x|y)P(y)$$

Vektor θ je vektor parametara apriorne distribucije i izglednosti klase. Koji su to točno parametri i koliko ih ukupno ima ovisi o tome koje smo distribucije odabrali. Sto se tiče apriorne vjerojatnosti klase, ako je klasifikacija binarna (K=2), koristit ćemo Bernoullijevu distribuciju, dok ćemo za višeklasnu (K >2) klasifikaciju koristiti kategoričku (multinulijevu) distribuciju. Sto se izglednosti klase tiče, ako su značajke diskretne, koristit ćemo Bernoullijevu ili kategoričku distribuciju, ovisno o tome je li značajka binarna ili viševrijednosna. Za kontinuirane značajke koristit ćemo Gaussovu (normalnu) distribuciju. Zapravo, budući da ćemo uvijek imati više od jedne značajke, koristit ćemo multivarijatnu Gaussovu distribuciju.

Bayesov klasifikator je parametarski model, a to znači da broj parametara modela ne ovisi o broju primjera. No, budući da se ovdje konkretno radi o probabilističkom modelu, to također znači da model pretpostavlja da se primjeri x i oznake y pokoravaju nekoj teorijskoj vjerojatnosnoj distribuciji. Svakako, količina parametara u tim distribucijama nije vezana uz broj primjera, pa prema tome, ukupan broj parametara u modelu također nije povezan s brojem primjera.

Bayesov klasifikator ćemo trenirati kroz procjenu parametara. Za procjenu parametara upotrijebit ćemo MLE (eng. maximum likelihood estimator) procjenitelje ili MAP (eng. maximum a posteriori probability) procjenitelje. Postupak optimizacije može biti usmjeren na maksimizaciju izglednosti parametara (MLE) ili maksimizaciju aposteriorne vjerojatnosti parametara (MAP). Sukladno tome, funkcija pogreške bit će jednostavno negativna izglednost parametara (MLE) ili negativna aposteriorna vjerojatnost parametara (MAP).

$$P(y|x) = \frac{p(x,y)}{p(x)} = \frac{p(x|y)P(y)}{p(x)}$$

4.1.3.2. Generativni modeli

Generativni modeli modeliraju zajedničku vjerojatnost, odnosno zajedničku gustoću vjerojatnosti $p(x, y)$. Na temelju te vjerojatnosti može se vrlo jednostavno, primjenom Bayesovog pravila, izračunati aposteriorna vjerojatnost $P(y|x)$, tj. vjerojatnost da primjer x pripada klasi y. Ovakav pristup nazivamo generativnim jer modelira postupak generiranja (nastanka) podataka. Želimo li opisati način nastanka označenih primjera $D=\{(x^{(i)}, y^{(i)})\}_i$, koji se ravnaju po zajedničkoj distribuciji $p(x, y)$, možemo reći da primjeri nastaju stohastičkim procesom koji se sastoji od dva koraka. U prvome koraku odabrana je oznaka y po distribuciji $P(y)$, a u drugome je koraku za odabrani y odabran primjer x po distribuciji $p(x|y)$. Takva priča, koja objašnjava stohastički postupak generiranja podataka, naziva se generativna priča (eng. *generative story*). [21].

Kod Bayesovog klasifikatora generativna je priča dosta kratka, jer se sastoji od svega dva koraka, ali općenito, kod složenijih generativnih modela, priča može biti složenija. Takvi složeniji generativni modeli su, npr., Bayesove mreže, mješavina Gaussovih distribucija (eng. *Gaussian mixture model, GMM*) i dr.

Generativna priča također se može upotrijebiti za generiranje sintetičkih podataka. Jednom kada je model naučen, možemo slijediti generativnu priču kako bismo uzorkovali primjere iz zajedničke distribucije. Za svaki primjer metode Bayesovog klasifikatora, prvo bi bila uzorkovana klasa y za distribuciju P(y), poslije toga bi bio uzorkovan primjer iz distribucije P(x V y) što bi značilo da ta distribucija odgovara izglednosti klase y. [19].

4.1.3.3. Gaussov Bayesov klasifikator

Sada ćemo definirati Bayesov klasifikator za kontinuirane značajke, tzv. Gaussov Bayesov klasifikator. Kod tog modela primjer x predstavljen je kao vektor brojeva, tj. značajke su numeričke, što znači da ćemo izglednosti klasa $P(x|y)$ modelirati Gaussovom distribucijom. Izglednost svake klase modeliramo kao jednu zasebnu multivarijatnu Gaussovu distribuciju. Srednja vrijednost te distribucije, dakle vektor μ, predstavlja prototipni primjer te klase, i taj primjer ima najveću gustoću vjerojatnosti. Primjeri koji su udaljeni od središta su manje vjerojatni da pripadaju toj klasi. Idealno, svi primjeri koji pripadaju ovoj klasi bi bili jednaki

prototipnom primjeru, međutim zbog šuma to nije slučaj. Dakle, Gaussova distribucija ovdje modelira odstupanje od ideala uslijed šuma.

Univarijantni Gaussov Bayesov klasifikator

Kada je prostor primjera jednodimenzijski, tj. kada imamo samo jednu značajku. U tom slučaju izglednost klase $P(x|y)$ modeliramo univarijatnom Gaussovom distribucijom.

$$x|y \sim N(\mu_{j,}\sigma_j^2)$$

Ako nas zanima samo pouzdanost za svaku klasu j, a ne vjerojatnost, možemo definirati model čiji je izlaz jednak zajedničkoj vjerojatnosti:

$$h_j(x|\theta_j) = -\ln \sigma_j - \frac{(x-\mu_j)^2}{2\sigma_j^2} + \ln P(y=j)$$

Gdje je vektor parametara $\theta_j = (\mu_j \ \mu_j' \ \sigma_j)$ pri čemu μ_j' označava parametar distribucije $P(y=1)$, koji je jednak apriornoj vjerojatnosti klase $y=1$. Ovdje se radi o poznatim distribucijama (Gaussova i Bernoullijeve). Njihove parametre možemo procijeniti pomoću procjenitelja MLE: [13].

$$\hat{\mu}_j = \frac{1}{N_j}\sum_{i=1}^{N} 1\{y^{(i)}=j\}x^{(i)}$$

$$\hat{\sigma}_j^2 = \frac{1}{N_j}\sum_{i=1}^{N} 1\{y^{(i)}=j\}(x^{(i)}-\hat{\mu}_j)^2$$

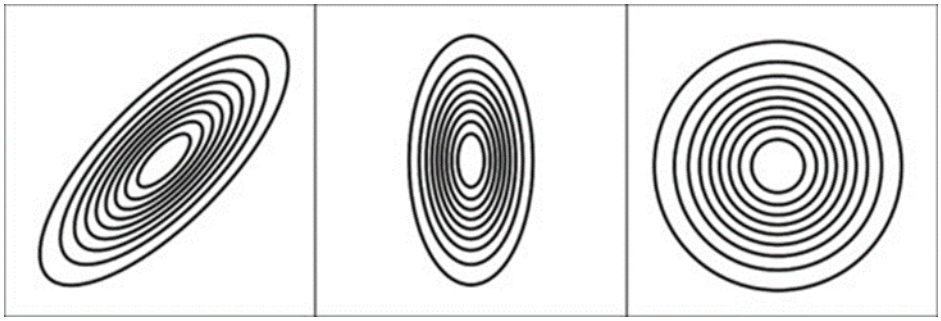
$$P(y=j) = \hat{\mu}_j' = \frac{1}{N}\sum_{j=1}^{N} 1\{y^{(i)}=j\} = \frac{N_j}{N}$$

Multivarijantni Gaussov Bayesov klasifikator

Univarijantni slučaj je manje realističan a puno izgledniji je slučaj da je primjer **x** vektor realnih brojeva. Tada izglednost svake klase *j* modeliramo multivarijatnom Gaussovom distribucijom:

$$p(x|y=j) = \frac{1}{(2\pi)^{n/2}|\sum_j|^{1/2}} \exp(-\frac{1}{2}(x-\mu_j)^T \sum^{-1} (x-\mu_j))$$

Prema tome, $\sum_j$, je matrica kovarijacije a μ_j vektor srednjih vrijednosti za klasu *j*. Interpretacija μ i $\sum$ je analogna kao i za jednodimenzijski slučaj: μ_j je prototipna vrijednost primjera u klasi *j*, dok je $\sum_j$količina suma i korelacija između izvora šuma unutar klase *j*. Prema tome kako se definira kovarijacijska matrica, moguće je na različite načine modelirati korelaciju između izvora šuma. Na primjer:



$\sum$ $\sum$ = diag(σ_j^2) $\sum = \sigma^2 I$

[13] Slika 9: Primjer modela

Postavlja se pitanje: koliko ovaj model ukupno ima parametara? Ukupan broj parametara ovog modela je broj parametara za koji je potrebno pohraniti vrijednosti kovarijacijske matrice za svaku klasu i vrijednosti vektora prototipa primjera za svaku klasu. Iz modela možemo zaključiti da je ukupna ovisnost broja parametara O(n^2). Dakle, ovdje možemo reći da postoji kvadratna ovisnost parametara o broju značajki zbog kovarijacijske matrice. Za procjenu parametara možemo koristiti MLE procjene za srednju vrijednost klase $\hat{\mu}_j$ te zbrajamo sve primjere koji pripadaju klasi j.

$$\hat{\mu}_j = \frac{1}{N}\sum_{j=1}^{N} 1\{y^{(i)} = j\} = \frac{N_j}{N}$$

Putem ovoga, apriornu vjerojatnost klase j dobivamo kao relativnu frekvenciju, odnosno broj primjera koji pripadaju klasi j podijeljen sa ukupnim brojem primjera u skupu za učenje. [13].

4.1.4. Neparametarske metode

Neparametarske metode se mogu opisati putem tri ključne svojstva:

1. Broj parametara se prilagođava na osnovu dostupnih primjera.
2. Izvode lokalne aproksimacije hipoteza blizu pohranjenih primjera.
3. Dodatna karakteristika, koja će biti dodana naknadno.

Umjesto toga ti modeli rade lokalne aproksimacije hipoteze u lokalnom prostoru na temelju sličnosti, odnosno udaljenosti od postojećih primjera. [13].

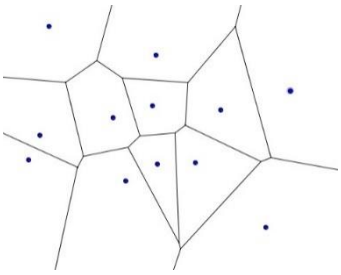
4.1.4.1. Algoritam k najbližigh susjeda (engl. k nearest neighbours,k-NN)

Algoritam k-NN j je neparametarski klasifikacijski algoritam. Predikcija na temelju većinske oznake *k* najbližih susjeda (*nearest neighbors*). Ukratko, algoritam radi tako da najprije pohrani sve primjere iz značajnog skupa podataka te onda kada treba klasificirati novi primjer jednostavno dohvati k najbližih susjeda tog primjera i primjer klasificira u većinsku klasu. Postoji funkcija NNparametrizirana s k, gdje je k broj susjeda koja preslikava neki primjer iz ulaznog prosotra x u podskup ulaznog prostora x, odnosno skup primjera preslikava u partitativni skup 2^x po nekoj mjeri udaljenosti.
Formalno model izgleda ovako:

$$h_{(x)} = \underset{j\epsilon\{0,\dots,K-1\}}{argmax} \sum_{(x^{(i)},y^{(i)})\epsilon NN_k(x)} 1\{y^{(i)} = j\}$$

Gdje je j oznaka klase koja može biti $\{0, \dots, K-1\}$. Zatim se treba maksimizirati broj glasova za tu klasu od primjera $(x^{(i)}, y^{(i)})$ u susjedstvu (x). Gleda se je li oznaka $y^{(i)} = j$. Ako je to slučaj skuplja se jedan glas za tu klasu j. U konačnici se gleda koja je klasa pobijedila, u slučaju izjednačenja broja glasova razriješavaju se na neki konzistentan način. Ovaj model radi s više klasa, dakle k klasa. Između ostalog za ovaj model nije potrebna funkcija gubitka niti optimizacijski postupak. Ovo je jedan vrlo jednostavan algoritam te je zato izrazito popularan u praksi.

Vrijednost k je hiperparametar algoritma koji određuje koliko susjeda se uzima u obzir. Ako je $k = 1$ onda će se primjer klasificirati u klasu jednog njegovog najbližeg susjeda. U tom slučaju ulazni prostor efektivno je particioniran Voronoijevim dijagramom koji izgleda ovako. [22].



[13] Slika 10: Voronoijev dijagram

Općenito, svaka ćelija je poliedar koji okružuje svaki primjer za učenje te će se novi primjer klasificirati prema tome u čiju ćeliju upadne. Budući da je k hiperparametar modela, on određuje njegovu složenost. Upravo je iz tog razloga lako prenačiti model. Općenito, što je više primjera n trebat će i veći k, drugim rječima trebat će uzeti u obzir više susjeda ako se želi sprječiti prenaučenost modela. S druge strane, preveliki k dat će prejednostavni model. [13].

Težinski k-NN

Težinski *k*-NN – utjecaj primjera ovisi o udaljenosti/sličnosti

Formalno izgleda ovako:

$$h_{(x)} = \underset{j\epsilon\{0,\dots,K-1\}}{argmax} \sum_{i=1}^{N} k\left(x^{(i)}, x\right) 1\left\{y^{(i)} = j\right\}$$

Gdje se računa vrijednost jezgrene funkcije između primjera $x^{(i)}$ te ulaznog primjera x i tu vrijednost množimo s indikacijom da je $y^{(i)} = j$. Odnosno, drugim rječima kada god neki primjer glasa za klasu j, taj će se glas otežati ovisno o vrijednosti jezgrene funkcije. Mogu se koristiti različite mjere udaljenosti. [13].

4.1.4.2. Stabla odlučivanja

Stabla odlučivanja su još jedan popularan algoritam za klasifikaciju.

Klasificira se kao neparametarski model jer se broj parametara (odnosno broj razina ili složenost modela) povećava s porastom broja primjera. Ulazni prostor se podjednako razdvaja na lokalne regije, tj. dva potprostora, ovisno o tome je li ispunjen određeni uvjet ili ne. Ovaj algoritam koristi stablo da bi predvidio ishod na temelju ulaznih podataka. Stablo odlučivanja se često koristi u problemima klasifikacije u kojima je potrebno donijeti odluku na temelju niza uvjeta. Problemi i nedostaci kod ovog modela jesu sklonost prenaučenosti i često su modeli dosta nestabilni.

Ove metode i algoritmi su samo neki od primjera koji se koriste za klasifikaciju podataka. Odabir odgovarajuće metode ovisi o vrsti problema i vrsti podataka koji se obrađuju. [23].

5. Analiza u programskom jeziku R

5.1. R – uvod opis sučelja i instalacija

R je programski jezik i softversko okruženje za statističku analizu, grafičko predstavljanja podataka, pravljenje izvještaja. Kreirali su ga Ross Ihaka i Robert Gentleman sa Sveučilišta Aukland iz Novog Zelanda. Početna verzija je objavljena 1993., zatim od 1997. godine razvija ga R Core Team koji se bavi razvojem ovog programskog jezika. R je interpreterski jezik, a što ga opisuje su jednostavan i efikasan programski jezik koji podržava uvjetno izvršavanje, cikluse, korisnički definirane rekurzivne funkcije i olakšan ulaz i izlaz. Efikasan rad s podacima i njihovo spremanje. Podržava operatore za rad sa nizovima, listama, vektorima i matricama. Omogućava grafičko predstavljanje podataka, tako da je R jedan od najpopularnijih i najviše korištenih programskih jezika za statističke proračune i analize. [24], [25].

5.1.1. Instalacija i preuzimanje

Instalacija se vrši u dva koraka. Prvi korak potrebno je preuzeti odgovarajući R instalacijski paket za željeni operacijski sustav. Verzije programa su gotovo identične za sve operacijske sustave. Zatim drugi korak, ukoliko želimo biti više produktivni u R-u, dostupan je i besplatan skup integriranih alata pod nazivom RStudio koji se može preuzeti s istog izvora. RStudio je dostupan na svim platformama kao i R instalacijski paket. [25].

5.2. Opis sučelja

RStudio sučelje, podijeljen je na 4 dijela.

Prvi do je editor R skripte odnosno koda. U editoru je moguće kreiranje, otvaranje, uređivanje datoteke koji sadrže skripte. Ako na primjer, želite da kreirate novu R skriptu, treba odabrati *File->New file->R script* ili lakši način klikom na +, zatim *R script* ili kombinacijom tipki Ctrl + Shift + N te nam se otvori nova skripta.

Drugi dio je R konzola. Prozor sa konzolom je mjesto na kojem upisujemo što želimo napraviti u R-u. U ovom dijelu se također prikazuje rezultat naredbi. U njemu možemo vidjeti što se događa nakon izvršenog programskog koda. Možete direktno unositi naredbe u konzolu, no one će biti izgubljene nakon zatvaranja sesije. Sve što unesete u tu konzolu nestaje nakon što završite s radom i izađete iz programa, pa se stoga preporučuje da svaki put naredbe zapisujete u skriptu. Skriptu koju spremite možete kasnije pregledavati, koristiti i nadopunjavati prema svojim potrebama.

Treći dio ***Environment/History/Connection*** je gupiran u tri dijela. ***Environment*** tab podrazumijeva listu svih aktivnih R objekata, vrijednosti, funkcije ili svega ostalog što je kreirana tijekom R sesije. ***History*** tab čuva povijest svih prethodnih naredbi.

Connection tab predstavlja poveznicu za naše poveznice.

Posljednji dio sadrži grupu ***Files/Plots/Packages/Help. Files*** tab prikazuje datoteke u okviru radnog direktorija. ***Plot*** tab prikazuje sve vizualizacije koje su kreirane tijekom R sesije. One se mogu po potrebi sačuvati kao *.jpeg* ili u *.pdf* formatu klikom na export. ***Packages*** tab prikazuje sve vanjske R biblioteke odnosno pakete koji su instalirani na našem lokalnom sustavu. Ukoliko su biblioteke označene onda će se i učitati pri pokretanju programa, a ako nisu označene, onda ih je potrebno označiti kako bi ih koristili. Također, moguće je klikom na *Packages->Install packages* instalirati biblioteku koju želimo.

RStudio projekti nudi korisnu opciu imena ***Projects***. Ona omogućava korisnicima jednostavne i lake promjene projekta u njegovim okvirima. Svaki projekt ima različite radne direktorije i datoteke. Korištenje projekata se svakako preporučuje zbog bolje organizacije datoteka i skripti na kojima radimo. Valja napomenuti da nakon svakog pokretanja kreće se s definiranim radnim direktorijem. Upisivanjem *getwd()* u Editor možemo provjeriti trenutni radni direktorij.

5.2.1. Osnove rada u programu

Postoje dva osnovna načina izvršavanja instrukcija u R programskom jeziku. Prvi način je pisanje u konzoli, piše se jedna po jedna instrukcija i tako se izvršava. Drugi način je pisanje

skripti, odnosno cijeli program u datoteku koja se sprema lokalno i kao takav se izvršava. Konzola se prepoznaje po znaku **>** (predstavlja prompt).
Pokretanje instrukcija možemo napraviti kombinacijom tipki Ctrl + Enter ili klikom na *Run*. [26].

5.2.2. Instaliranje i učitavanje paketa

U ovom ulomku ću pojasniti instalaciju i učitavanje paketa. U rujnu 2018. godine broj R paketa je premašio brojku od trinaest tisuća. Instalaciji određenih paketa se može pristupiti kroz GUI (Graphic User Interface), odnosno grafičko korisničko sučelje RStudio-a. Neophodno je iz kartice ***Packages*** izabrati gumb *Install* ili *Tools->Install packages*. Nakon toga upisati ime paketa i postaviti opciju *install dependencies* uključeno. Instalacija paketa se također može izvršiti i pomoću funkcije instal pak iz *install.packages(„NazivPaketa“)*. Može se reći da su paketi u R-u sličnih bibliotekama u *C*, *C++* ili *Python-u* . Paketi u R-u sadrže funkcije, primjere i testne podatke. Paketi se prilikom uobičajene instalacije preuzimaju sa **CRAN** repozitorija, međutim moguća je instalacija s drugih web stranica putem URL-a ili sa *GitHub*-a. U tome može poslužiti paket devtools. Instalacija s *GitHub*-a može biti korisna kada se radi o upotrebi funkcija koja su u razvoju. S obzirom da autori paketa svoje radne verzije često postavljaju tamo prije nego što dostignu potpunu stabilnost, nakon čega se dopunjuju na **CRAN** repozitoriju. Grupu paketa određenog područja možemo instalirati pomoću paketa *ctv* i naredbe *install.views(„NazivGrupe“)*. Neke od dostupnih grupa paketa za specijalizirana područja su

- Strojno učenje
- Računarstvo visokih performansi
- Kliničke studije
- Genetika
- Numerička matematika
- Prostorne analize
- Društvene znanosti

Za cijelu listu paketa grupirane prema specijaliziranim područjima mogu se pronaći na https://cran.r-project.org/ pod Task Views. [26].

5.2.3. Učitavanje paketa

Poslije instalacije paketa i pokretanja nove sesije u R-u neophodno je učitati paket. Nakon učitavanja dostupne postoje određene funkcije, podaci i upute sadržane u tom paketu.

Učitavanje paketa je neophodno jer bi u suprotnom svi instalirani paketi prilikom svakog pokretanja R-a bili učitavani, što bi zahtijevalo dodatno vrijeme i resurse.
Postavljanje paketa odvojenih od osnovne instalacije R-a daje i mogućnost zamjene pojedinačnih paketa novim verzijama bez ponovne instalacije. Paketi se učitavaju pomoću naredbe *library(„NazivPaketa“)*. Svaki put prilikom pokretanja sesije neophodno je učitati podatke, odnosno učitati pakete. Naziv paketa može biti naveden sa i bez znakova navoda. Kako bi izvršili isključivanje nekog od paketa, dovoljno je pokrenuti funkciju *detach(„package:nazivpaketa“).* Isključivanje paketa je nekada potrebno kada se više paketa istovremeno koristi u okviru jednog projekta, a imaju funkcije istog imena, a različite namjene. Međutim i to možemo zaobići istovremenim pozivanjem paketa i njegove pripadajuće funkcije *NazivPaketa::NazivFunkcije*.
Koje pakete imamo instalirane u R-u provjeravamo pokretanjem funkcije *library()*. [26].

5.2.4. Osnovna sintaksa i operatori

R razlikuje velika i mala slova (case-sensitive), što je slučaj sa većinom programskih jezika baziranih na ***UNIX***-u. Nazivi ne smiju počinjati brojevima. Najčešći znakovi korišteni u R-u su točka-zarez, #, vitičaste zagrade i znak +. Kratko pojašnjenje svakog od njih: točka-zarez služi za odvajanje naredbi u okviru jednog reda, ako se naredbe nalaze u različitim redovima, nije ih potrebno razdvajati točka-zarezom. # označavaju početak komentara koji obuhvaća sve napisano do kraja reda koji se ne izvršava. Vitičaste zagrade grupiraju izraze u blokove, dok znak + R sam ispisuje ako naredba nije stala u jedan red. Neke od korisnih općih naredbi su:

Tablica 1: Opće naredbe u R-u

Naziv	Opis
NazivObjekta	navedemo samo ime objekta, prikazuje sadržaj objekta
length(NazivObjekta)	Ispisuje dužinu željenog objekta
atributes(NazivObjekta)	Daje nam osnovne informacije o objektu
class(NazivObjekta)	Ispisuje klasu objekta
str(NazivObjekta)	Prikazuje internu strukturu R objekta
c(NazivObjekta1, NazivObjekta2…)	Kombinira objekte u vektor
rbind(NazivObjekta1, NazivObjekta2…)	Kombinira objekte u redove
cbind(NazivObjekta1, NazivObjekta2…)	Kombinira objekte u stupce
ls()	Izlistava dostupne objekte
rm(NazivObjekta)	Briše objekt
novi_objekt<-edit(NazivObjekta)	Mijenja i sprema sadržaj u novi objekt
fix(NazivObjekta)	Mijenja postojeći objekt

(Izvor: Venables i Smith, 2004)

Operatori su simboli koji upućuju program da izvrši određene matematičke ili logičke operacije. R programski jezik bogat operatorima. Posjeduje aritmetičke operatore, relacijske, logičke, operatore dodjele i ostale operatore [26].

Tablica 2: Operatori u R-u

Operator	Opis
+	Zbrajanje
-	Oduzimanje
*	Množenje
/	Dijeljenje
^ ili **	Eksponent
%%	Ostatak pri dijeljenju dva broja (vektora) - mod
%/%	Količnik pri dijeljenju dva broja (vektora) - div
x & y	x i y

(Izvor: Venables i Smith, 2004)

Tablica 3: Još operatora u R-u

Operator	Opis
<	Manje
<=	Manje ili jednako
>	Veće
>=	Veće ili jednako
==	Jednako
!=	Različito
!x	Nije x
x | y (x || y)	x ili y (kraća forma uzima sve elemente vektora, duža samo prvi element)
x & y (x && y)	x i y (kraća forma uzima sve elemente vektora, duža samo prvi element
isTRUE(x)	Testira je li x istinit
S i @	Ekstrakcija komponenti
::	Pristup u određenom imenskom prostoru

(Izvor: Venables i Smith, 2004)

Tablica 4: Tablica operatora dodjele

Operator	Opis
<- ili = ili <<-	Lijevi operator dodjele
-> ili ->>	Desni operator dodjele

(Izvor: Venables i Smith, 2004)

Tablica 5: Tablica ostalih operatora

Operator		Opis
:		Dvotočka dodjeljuje niz brojeva pri kreiranju vektora
%in%		Koristi se za identifikaciju pripada li neki element vektoru
%*%		Koristi se pri množenju matrica

(Izvor: Venables i Smith, 2004)

5.3. Učitavanje i priprema podataka

Prvo je potrebno preuzeti skup podataka. U ovom primjeru je bio korišten skup podataka „train“ na primjeru putnika povijesnog broda Titanica. Bit će skinute datoteke tipa .csv što ukazuje na nekakvu strukturiranu tablicu podataka kao neke već poznate Excel tablice. Tad je potrebno ući u progam RStudio. Poslije toga je potrebno kliknuti na alatnu traku file i odabrati opciju import dataset te odabrati opciju From text(basic). Onda je potrebno pronaći train skup podataka u kojoj se nalaze svi podatci o putnicima Titanic broda te kliknuti open. Automatski će se pozvati funkcija View(“train”) poslije toga otvorit će se tablica u view obliku u kojoj možemo vidjeti sve unose u tablicu. To je kraj pripreme podataka.

5.4. Opis skupa podataka

Skup podataka Titanic u programskom okruženju R sadrži informacije o sudionicima na prvoj plovidbi broda Titanic 1912 koji je potonuo. Ovaj skup podataka je dostupan u mnogim formatima, a u R-u se može dohvatiti pozivom funkcije Data("Titanic").

Postoje i drugi paketi koji sadrže ovaj skup podataka, kao što je paket titanic. Ovaj paket sadrži podatke o preživljavanju putnika, njihovoj cijeni karte, spolu, dobi i drugim karakteristikama. Podaci su formatirani za strojno učenje i sadrže uzorak za treniranje, uzorak za testiranje i dva dodatna skupa podataka koji se mogu koristiti za dublju analizu strojnog učenja.

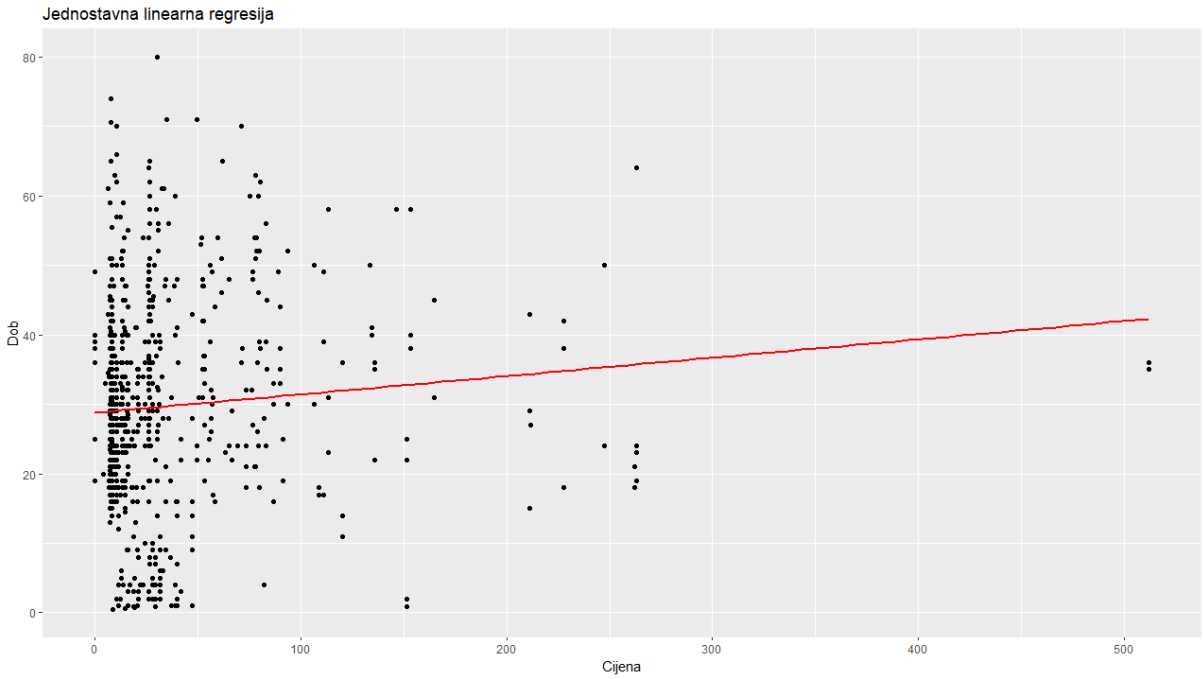
5.5. Regresijska analiza u R-u

Jednostavna linearna regresija u R-u se može dobiti pozivanjem sljedeće funkcije:

```
# Jednostavna linearna regresija
simple_model <- lm(Dob ~ Cijena, data = train)
```

Uz pomoć paketa ggplot2 moguće je postići vizualizaciju podataka u obliku grafa. Potrebno je odabrati x i y os za graf te koristiti metodu „lm" (eng. Linear model). Uz pomoć funkcije ggtitle dodaje se naslov samom grafu a nazivlja x i y osi ggplot2 nasljeđuje iz samog skupa podataka.

```
# Jednostavna linearna regresija
ggplot(train, aes(x = Cijena, y = Dob)) +
geom_point() +
geom_smooth(method = "lm", se = FALSE, color = "red") +
ggtitle("Jednostavna linearna regresija")
```



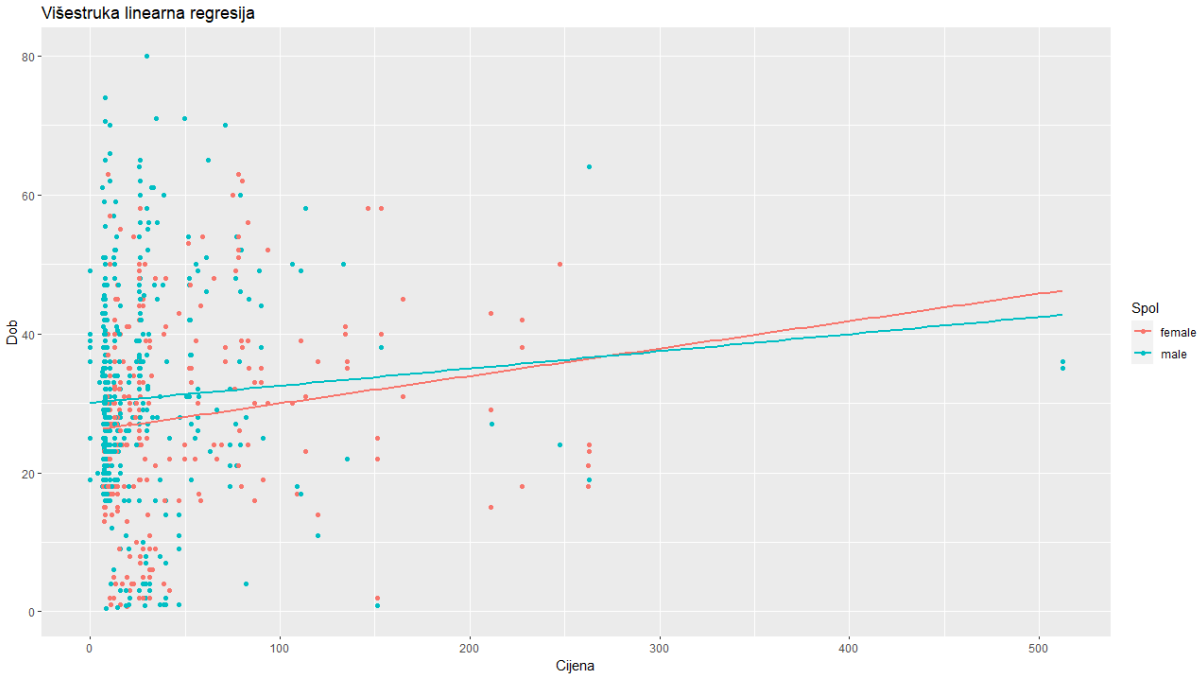
Slika 11: Jednostavna linearna regresija kreirana u R programskom jeziku (ggplot2)

Višestruka linearna regresija u programskom kodu R u programu RStudio se postiže sljedećim linijama koda i funkcije lm (eng. Linear model):

```
# Višestruka linearna regresija
multiple_model <- lm(Dob ~ Cijena + Pclass + Spol, data = train)
```

Vizualizacija višestruke linearne regresije u programskom kodu R u programu RStudio se postiže sljedećim linijama koda uz pomoć funkcije aes (eng. Aesthetic mappings):

```
#Višestruka linearna regresija
ggplot(train, aes(x = Cijena, y = Dob, color = Spol)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE) +
  ggtitle("Višestruka linearna regresija")
```



Slika 12: Višestruka linearna regresija kreirana u R programskom jeziku (ggplot2)

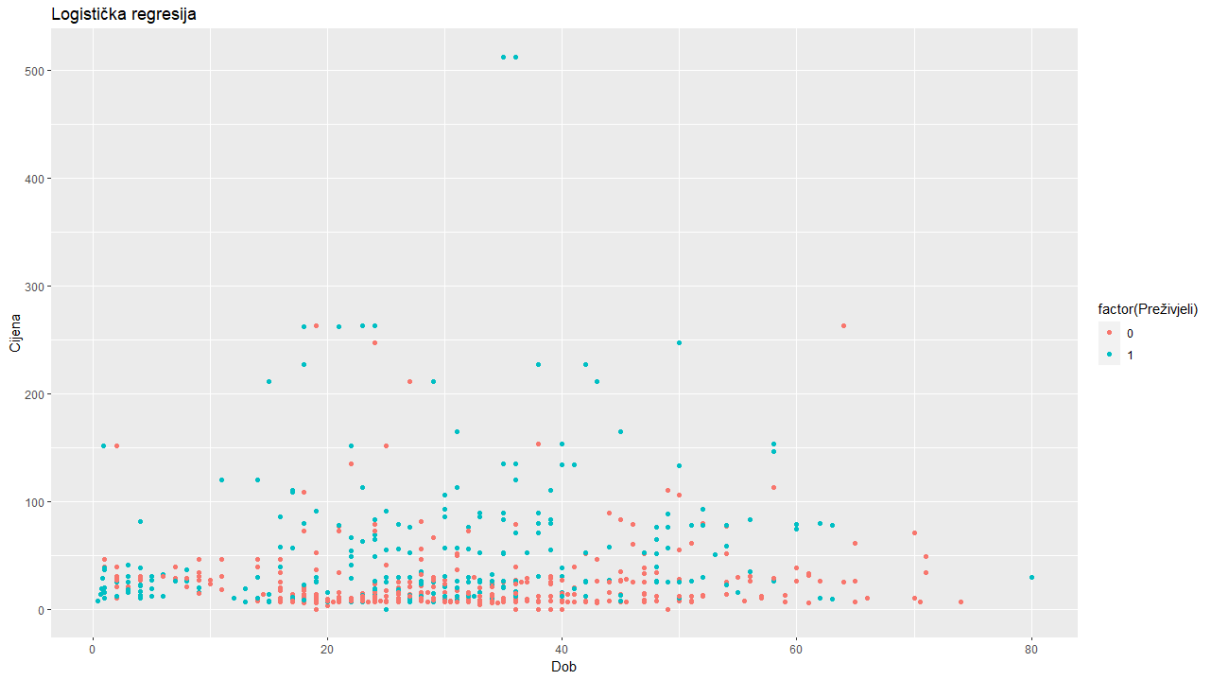
5.6. Klasifikacija podataka u R-u

Logistička regresija u programskom jeziku R u programu RStudio se postiže sljedećim linijama koda uz pomoć funkcije glm (eng. Generalized linear model):

```
# Logistička regresija
logistic_model <- glm(Preživjeli ~ Dob + Cijena + Pclass + Spol, data = train,
family = "binomial")
```

Vizualizacija logističke regresije u programskom kodu R u programu RStudio se postiže sljedećim linijama koda uz pomoć biblioteke ggplot2 i funkcije aes (eng. Aesthetic mappings):

```
# Logistička regresija
ggplot(train, aes(x = Dob, y = Cijena, color = factor(Preživjeli))) +
  geom_point() +
  ggtitle("Logistička regresija")
```



Slika 13: Logistička regresija kreirana u R programskom jeziku (ggplot2)

6. Zaključak

U ovom radu istražili smo teorijske koncepte i metode regresijske analize i klasifikacije podataka te smo te metode primijenili na analizi realnog skupa podataka. Bile su nabrojane različite vrste regresijske analize a u radu je bio fokus na linearnu jednostruku i višestruku regresijsku analizu. Opisan je utjecaj jedne ili više prediktorskih varijabli na ciljnu odnosno kriterijsku varijablu što se prikazivalo na grafu kao regresijska linija. Ovaj rad sadrži uvid u čimbenike koji utječu na regresiju i kako se oni dobiju izračunom. Zatim je opisana i klasifikacija u kojoj je naglasak bio na metodi logističke regresije. Logistička regresija je detaljno opisana kao klasifikator te je objašnjeno kako izgleda model logističke regresije, koja je njena funkcija pogreške, što je pogreška unakrsne entropije te kako izgleda gradijentni spust za optimizaciju logističke regresije. Objašnjena je višeklasna logistička regresija. Prikazane su metode klasifikacije, njihova svrha te je objašnjena metoda Bayesovog klasifikatora. Krajnje, dan je uvid u vrstu klasifikacije koja grupira neparametarske metode kao što su stablo odlučivanja i k-NN algoritam. Također smo napisali nekoliko implementacija za programski jezik R i vizualizirali ih uz pomoć gotovih implementacija u obliku paketa za R. U tim implementacijama postoje gotovi paketi koji sadrže funkcije linearne jednostavne, višestruke regresije i logističke regresije za klasifikaciju. Uz pomoć paketa ggplot2 smo putem implementacije kreirali detaljne grafove. Bio je korišten skup podataka o prvoj plovidbi broda Titanic te je isti skup podataka bio vizualiziran kao primjer. Iz ovog možemo reći da je R kvalitetan alat za analizu podataka i vizualizaciju istih.

Naša analiza pokazala je da su metode regresije i klasifikacije podataka odlični alati za predviđanje zavisne varijable na temelju nezavisnih varijabli. Također, identificirali smo ključne čimbenike koji utječu na rezultate regresijske analize i klasifikacije podataka. Ovaj rad pruža koristan uvid u primjenu metoda regresije i klasifikacije podataka u analizi realnih skupova podataka, te može poslužiti za daljnja istraživanja u ovom području.

7. Popis literature

[1] D. Dizdar (2006.) *Kvantitativne metode* [Na internetu]. Dostupno: https://km.com.hr/wp-content/uploads/2018/04/Kvantitativne_metode.pdf#page=162 [pristupano 2.6.2023.].

[2] "Korelacija," (bez dat.). u *Wikipedia the Free Encyclopedia.* Dostupno: https://hr.wikipedia.org/wiki/Korelacija, [pristupano 3.5.2023.]

[3] "Linearna regresija," (bez dat.). u *Wikipedia the Free Encyclopedia.* Dostupno: https://hr.wikipedia.org/wiki/Linearna_regresija, [pristupano 4.5.2023.]

[4] N. Šunjo, *Višestruka regresija i prognoza ukupne akcijske prodaje* [Diplomski rad]. Prirodoslovno-matematički fakultet, Sveučilište u Zagrebu, Zagreb, 2014..

[5] DATAtab e.U. (2023.), "DATAtab: Online Statistics Calculator" [Na internetu]. Dostupno: https://datatab.net [pristupano 10.8.2023.]

[6] J. Šnajder, "Regresija", nastavni materijali na predmetu Strojno učenje 1 [Moodle], Sveučilište u Zagrebu, Fakultet elektrotehnike i računarstva, Zagreb, 2021

[7] DATAtab e.U. (2.2.2021.) "Regression Analysis: An introduction to Linear and Logistic Regression," *Youtube* [Video datoteka]. Dostupno: https://www.youtube.com/watch?v=FLJ0yYetywE&list=PL-p9JpwN5NNHaP9BngSyJyfk879ydH3BU [pristupano: 10.7.2023.].

[8] OpenIntro statistics (bez dat.) "Linear and logistic regression" [Na internetu]. Dostupno: https://github.com/OpenIntroStat/openintro-statistics-slides/blob/master/Chp%208/chp8.pdf [pristupano 17.6.2023.].

[9] I. Lulić, *Uporaba metode regresijske analize u rješavanju problema vezanih za inženjersku praksu* [Završni rad]. Fakultet strojarstva i brodogradnje, Sveučilište u Zagrebu, Zagreb, 2014.

[10] A. Štambuk, M. Biljan-August, "Regresijska i korelacijska analiza", Rijeka, 2013.

[11] K. P. Murphy, *Machine Learning: A Probabilistic Perspective, Cambridge, Massachusetts*, Engleska. 2012.

[12] „Nadzirano učenje," (bez dat.). u *Wikipedia the Free Encyclopedia. Dostupno:* https://bs.wikipedia.org/wiki/Nadzirano_u%C4%8Denje, [pristupano 25.8.2023.]

[13] „Supervised learning," (bez dat.). u Wikipedia the Free Encyclopedia. Dostupno: https://en.wikipedia.org/wiki/Supervised_learning, [pristupano 27.8.2023.]

[14] J. Šnajder, B. B. Dalbelo "Strojno učenje", nastavni materijali na predmetu Strojno učenje 1 [Moodle], Sveučilište u Zagrebu, Fakultet elektrotehnike i računarstva, Zagreb, 2022.

[15] L. Bottou, „Curiously fast convergence of some stochastic gradient descent algorithms,“ *SLDS 2009 conference,* Pariz, 2009.

[16] M. Gürbüzbalaban, A. Ozdaglar I P. Parrilo. „Why random reshuffling beats stochastic gradient descent,“ *Mathematical programming,* 2019. [Na internetu]. Dostupno: https://doi.org/10.1007/s10107-019-01440-w [pristupano: 29.8.2023.]

[17] A. Defazio, F. Bach i S. Lacoste-Julien, „Saga: A fast incremental gradient method with support for non-strongly convex composite objectives“, Engleska, 2014.

[18] S. Boyd, S. P. Boyd i L. Vandenberghe, „Convex optimization“, Engleska, 2004.

[19] I. H. Witten, E. Frank, i M. A. Hall, *Data Mining: Practical Machine Learning Tools and Techniques,* 3. izd., SAD: Morgan Kaufmann Books. 2011.

[20] J. Dobša, “Vjerojatnost i slučajne varijable“, nastavni materijali na predmetu Statistika [Moodle], Sveučilište u Zagrebu, Fakultet organizacije i informatike, Varaždin, 2017.

[21] J. Dobson i A. G. Barnett, „An introduction to generalized linear models“, SAD, 2018.

[22] V. Grbavac, *Rudarenje podataka kako metoda upravljanja znanjem* [Završni rad]. Fakultet organizacije i informatike, Varaždin, Dostupno: https://dabar.srce.hr/islandora/object/foi:4076 [pristupano 28.6.2023.]

[23] G. Vinković, *Oktrivanje znanja u bazama podataka* [Završni rad]. Sveučilište Jurja Dobrile u Puli, Pula, 2017. Dostupno: https://dabar.srce.hr/islandora/object/unipu%3A2359 [pristupano 2.6.2023.]

[24] “R programming language,” (bez dat.). u Wikipedia the Free Encyclopedia. Dostupno: https://en.wikipedia.org/wiki/R_(programming_language) [pristupano 15.6.2023.]

[25] W. N. Venerables, D. M. Smith i sur. “Uvod u korištenje R-a”. [Na internetu] Dostupno: https://cran.r-project.org/doc/contrib/Kasum+Legovic-UvodUr.pdf [pristupano 15.6.2023.]

[26] R. Gentleman i R. Ihaka, “The R Project for Statistical Computing”. [Na internetu] Dostupno: https://www.r-project.org/ [pristupano 16.6.2023.]

8. Popis slika

Slika 1: Pozitivna veza....17
Slika 2: Nema veze među entitetima....17
Slika 3: Kolmogorov-Smirnov test normalnosti....19
Slika 4: Quantile-quantile plot dijagram....20
Slika 5: Logistička funkcija....23
Slika 6: promjena nagiba krivulje ovisno o faktoru....23
Slika 7: Graf funkcije gubitka....26
Slika 8: Granica između klasa u dvodimenzijskom ulaznom prostoru....29
Slika 9: Primjer modela....39
Slika 10: Voronoijev dijagram....41
Slika 11: Jednostavna linearna regresija kreirana u R programskom jeziku (ggplot2)....50
Slika 12: Višestruka linearna regresija kreirana u R programskom jeziku (ggplot2)....51
Slika 13: Logistička regresija kreirana u R programskom jeziku (ggplot2)....52

9. Popis tablica

Tablica 1: Opće naredbe u R-u 46
Tablica 2: Operatori u R-u 46
Tablica 3: Još operatora u R-u 47
Tablica 4: Tablica operatora dodjele 47
Tablica 5: Tablica ostalih operatora 47