Univerzitet u Nišu

Elektronski fakultet

SEMINARSKI RAD

Redukcija dimenzionalnosti

Predmet: Prikupljanje i predobrada podataka za mašinsko učenje

Mentor: Student:

Doc. dr Aleksandar Stanimirović Milica Rangelov 1690

Sadržaj

1. Uvod..... 3
2. Problemi sa velikim brojem ulaznih promenljivih..... 4
3. Redukcija dimenzionalnosti..... 5
4. Metode za redukciju dimenzionalnosti..... 8
4.1 Metode Linearne algebra..... 8
4.1.1 Non-negative Matrix factorization..... 8
4.1.2 PCA..... 9
4.1.3 SVD..... 13
4.1.4 LDA..... 15
4.2 Manifold Learning Metode (Non-linear)..... 16
4.2.1 Isomap..... 17
4.2.2 t-SNE..... 18
4.2.3 LLE..... 19
4.3 Odabir osobina (Feature selection)..... 20
4.3.1 Selekcija na osnovu varijanse..... 21
4.3.2 Selekcija na osnovu važnosti osobina..... 21
4.3.3 Selekcija na osnovu korelacione matrice..... 22
4.3.4 Rekurzivno uklanjanje osobina..... 22
4.3.5 Lasso Regresija..... 23

1. Uvod

Da bi se razumeo sam proces redukcije ili smanjivanja dimenzionalnosti odgovarajućeg skupa podataka, pre svega potrebno je jasno definisati pojam dimenzionalnosti. Dimenzionalnost određenog skupa podataka odnosi se na broj ulaznih promenljivih ili kako se još nazivaju "*feautire-a*", to jest osobina podataka.

Kada se vrši analiza i prikupljanje skupa podataka, veoma često se dobija pristup velikom broju osobina. Osobine omogućavaju da detaljno opisivanje podataka, ali u koliko ih ima previše mogu napraviti dosta problema. Pre svega, u koliko imamo previše osobina koje je potrebno prikupiti za svaki podatak, može se doći do situacije u kojoj se ne može prikupiti dovoljno primeraka i observacija koje mogu pokriti i najmanji deo konfiguracija modela. Kao posledica ovoga, odgovarajući algoritam neće imati dovoljno podataka za rad i učenje. Takođe, model teško može obaviti proces predikcije u koliko ima veliki broj osobina sa kojima mora raditi. Ovo se drugačije može definisati i kao "kletva" velike dimenzionalnosti.

Kako bi se izbegla "kletva" velike dimenzionalnosti skupa podataka, potrebno je uzeti u obzir da nisu sve osobine od podjednakog značaja. To ujedno predstavlja i cilj same redukcije dimenzionalnosti. Sam proces podrazumeva izvlačenje osobina koje su od veće značaja. Ovo podrazumeva da se iz inicijalnog skupa osobina kreira novi koji će zadržati sve najbitnije informacije, a odnaciti one manjeg značaja. Drugačije rečeno, sa malim gubitkom u skupu podataka i dalje se dobija mogućnost dobre predikcije rezultata. Uglavnom, proces smanjenja dimenzionalnosti koristi se zarad prikaza podataka. Pored toga, skup podataka za regresiju i klasifikaciju se može uprostiti kako bi poboljšao proces predikcije.

I ako sve ovo zvuči savršeno, svaki proces poseduje odgovarajuće mane. U okviru procesa redukcije dimenzionalnosti, problem se ogleda u tome što novi skup osobina neće biti čitljiv čoveku. Sam skup podataka će biti jasan modelu koji se trenira, ali će ljudskom oku izgledati kao skup sa nasumičnim brojevima. Ovaj problem može se rešiti ne redukcijom podataka već selekcijom odgovarajućih osobina.

U ostatku dokumenta, obradiće se upravo navedeni problemi. Deteljno će se opisati porches identifikacije samog problema i pronalaženja potencijalnog rešenja. Pored toga, obratiće se uslovi u kojima je moguće primeniti sama rešenja, na čemu se zasnivaju i koje su prednosti i mane.

2. Problemi sa velikim brojem ulaznih promenljivih [1]

Da bi se bolje razume problem koji izaziva velika količina ulaznih promenljivih potrebno je na što adekvatniji način opisati sam skup podataka i konkretno definisati koje su njegove osobine. Najjednostavnije je zamisliti podatke kao jedan fajl u Excel-u. Same ulazne promenljive predstavljaju kolone, dok podaci su sami redovi u okviru dokumenta. Same kolone predstavljaju ulaze koji se prosleđuju modelu zarad predikcije odgovarajuće vrednosti. Kao što je već napomenuto same kolone nazivaju se još i "*features*" ili osobine.

Same kolone mogu se predstaviti i kao n-dimenzionalni prostor osobina, dok redovi podataka mogu predstavljati odgovarajuće tačke u definisanom prostoru. Ovo može biti jako korisna geometrijska interpretacija samog skupa podataka. Naime, u koliko prostor ima veliki broj dimenzija, sama zapremina navedenog prostora će biti ogromna. Kao posledica ovoga, tačke koje odgovaraju podacima, veoma često predstavljaju male uzorke koji se ne mogu prikazati u navadenom prostoru. Ova nebalansiranost može negativno uticati na algoritme mašinskog učenja i njihove performanse. Ovaj uslov je još poznat pod nazivom "kletva" dimenzionalnosti.

Sam pojam "kletve" dimenzionalnosti odnosi se na fenomen koji se javlja pri analizi podataka visoke dimenzionalnosti koji se ne mogu prikazati u prostorima sa manjom dimenzijom, poput 3D prostora. Ovaj fenomen može se pojaviti u različitim aspektima: numeričkim analizama, kombinatorici, mašinskom učenju, bazama podataka, pretraga podataka itd. Uobičajeno kod ovog problema je da se sa povećanjem dimenzionalnosti, brzo povećava i zapremina prostora pa samim tim podaci postaju razređeniji. Kako bi se održali dobri rezultati, sa povećanjem dimenzionalnosti eksponencijalno se povećava i broj potrebnih podataka. Takođe, procesi organizacije i pretrage podataka oslanjaju se na detekciju oblasti podataka sa sličnim osobinama. Kod visoko dimenzionalnih prostora, objekti su razređeni i teško se može pronaći bilo koji vid sličnosti. Ovo povlači i efikasnot procesa organizacije. [2]

Sa stanovišta mašinskog učenja, svaka od osobina poseduje konačan skup mogućih vrednosti. Da bi model mogao da nauči "prirodu" podataka konačnog skupa predstavljenog u okviru prostora visoke dimenzionalnosti, za svaku od mogućih kombinacija određene osobine potrebno je posedovati adekvatan broj primeraka. Sa abstraktne strane, potrebno je sa porastom dimenzionalnosti uvećati broj primeraka eksponencijalno. U samom mašinskom učenju, problem "kletve" poklapa se sa "*Hjuzovim fenomenom*". Ovaj fenomen ukazuje na to da sa fiksiranom količinom primeraka za treniranje, prosečna moć predikcije jednog klasifikatora ili regresora se prvo poboljšava sa povećanjem broja ulaza, ali po dostizanju određene dimenzionalnosti pošinje da se pogoršava umesto da nastavi polako da raste.

[1] https://machinelearningmastery.com/dimensionality-reduction-for-machine-learning/
[2] https://en.wikipedia.org/wiki/Curse_of_dimensionality

Kako se otklanja “kletva” dimenzionalnosti? U koliko postoji ovaj problem sa određenim skup podataka, jedan od načina za njegovo rešavanje je smanjenje dimanzionalnosti. Prosto rečeno, potrebno je ukloniti određeni broj kolona iz fajla i smanjiti dimenzionalnost prostora. Sve ovo odgovara upravo redukciji dimenzionalnosti.

3. Redukcija dimenzionalnosti [3]

Kao što je u prethodnom poglavlju već navedeno, sam problem redukcije dimenzionalnosti podrazimeva smanjenje broja ulaznih promenljivih koje se prosleđuju modelu za predikciju. Ovo podrazumeva projekciju podataka na podprostor niže dimenzionalnosti kako bi uhvatila njihova suština.

U kontekstu modela mašinskog učenja, manja dimenzionalnost ulaznih vrednosti odgovara manjem broju parametara, to jest jednostavnijoj strukturi. Ovo se takođe može nazvati is stepenom slobode modela. Sa povećanjem stepena slobode, povećava se i mogućnost “overfitinga” samog modela. Ovo sa sobom povlači i performanse modela pri radu sa novim podacima. Najpoželjnije je posedovati jednostavniji model sa odgovarajućim performansama i sa manjim brojem ulaza. Navedena osobina najbolje odgovara linearnim modelima kod kojih su broj ulaza i stepen slobode usko povezani. Međutim, u kontekstu jednostavnijih modela analitički i empirijski pokazano je da u koliko je relativna kumulativna efikasnost navedenog dodatnog skupa osobina veća (ili manja) od veličine navedenog skupa, očekuje se da greška modela konstruisanog upotrebom dodatnog skupa osobina bude veća (ili manja) od grečke modela konstruisanog bez njega. Drugim rečima, pored veličine ulaznih promenljivih, potrebno je sagledati i uticaj relativne kumulativne efikasnosti u procesu nadgledanja moći predikcije samog modela. Pri učenju metrika, povećan broj osobina može poboljšati performanse modela. Zbog toga, potrebno je razgraničiti situacije u kojima je neophodno odraditi redukciju dimenzionalnosti od onih u kojima osobine ne prave problem.

Da bi se ovo zaobišlo, redukcija dimenzionalnosti uglavnom se primenjuje nakon detaljne analize podataka i njihovog čišćenja i skaliranja. Po završetku procesa redukcije, pokreće se treniranje modela za predikciju. Kako se podaci za treniranje modela dele na podatke za treniranje, podatke za validaciju i podatke za testiranje, sam proces redukcije neophodno je primeniti na svaki od njih. Ujedno potrebno je primeniti ga i na podacima koji se koriste nad finalnim modelom

Prednosti koje redukcija dimenzionalnosti donosi su:

- .Kompleksnost se smanjuje sa brojem osobina

[3] https://www.techtarget.com/whatis/definition/dimensionality-reduction

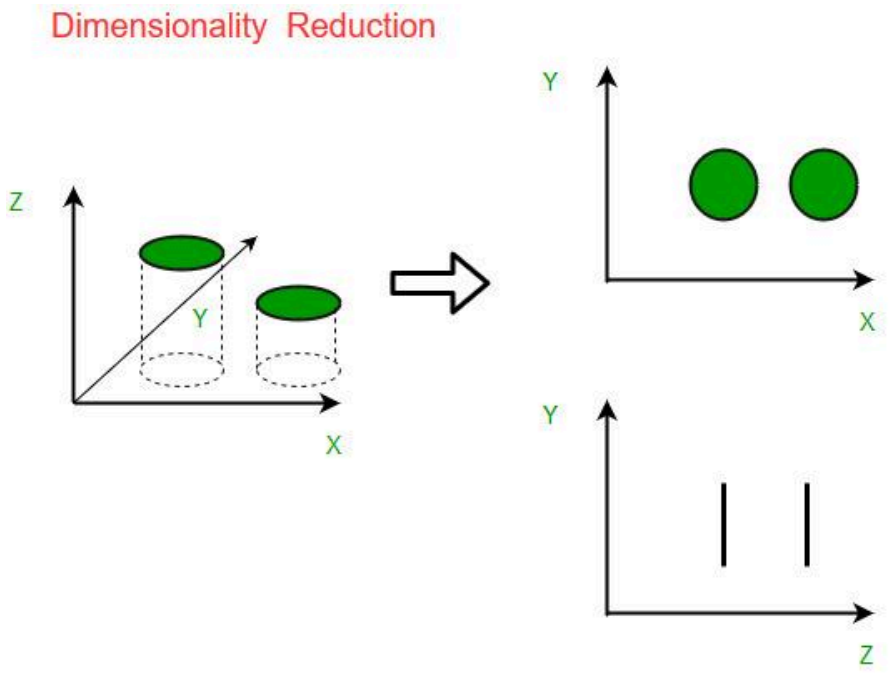
- Manja količina podataka zahteva manje prostora
- Manji broj osobina zahteva manje vremena za računanje
- Tačnost modela se povećava, zbog manje količine podataka koji mogu dovesti do pogrešnog navođenja modela
- Treniranje algoritama je brže
- Visualizacija podataka je brža
- Uklanja nebitne osobine

Nedostaci samog procesa redukcije dimenzionalnosti su:

- Samo redukcija podrazumeva izbacivanje određenog skupa podataka. Ovaj postupak može uticati na tačnost algoritma i njegov način rada.
- Može zahtevati veliku procesorsku snagu
- Interpretiranje dobijenih rezultata može biti komplikovano
- Nezavisne promenljive je teško obuhvatiti u okviru rezultata

Intuitivni primer za redakciju dimenzionalnosti može se prikazati putem jednostavne klasifikacije email-ova (spam ili ne). Ovaj problem može uključiti veliki broj osobina poput: da li email ima generički naslov, sadržaj samog email-a, da li koristi templejt itd. Međutim, potrebno je uzeti u obzir da neke od navedenih osobina mogu da se preklapaju. Drugim rečima, ako posmatramo set podataka koji se oslanja na vlažnost i padavine, može se lako zaključiti da se problem klasifikacije zasniva na jednoj osobini. Samim tim u takvoj situaciji može se primeniti redukcija. 3D klasifikacione probleme je malo teže prikazati. Sam 3D prostor osobina se deli na dva 2D prostora i u kolko se ustanovi velika korelisanost između osobina, osobine se mogu redukovati dalje. [Slika 1] [4]

[4] https://www.geeksforgeeks.org/dimensionality-reduction/



Slika 1

Sam proces redukcije dimenzionalnosti sastoji se od dve glavne komponente:

- **Selekcija osobina:** biraju se određene osobine iz originalnog skupa kako bi se dobio manji podskup koji se može iskoristiti za modelovanje problema. Ugalvnom uključuje tri načina:
    - Filter metoda - poredi osobine po njihovoj povezanosti sa ciljanom promenljivom
    - Vraper metoda - koristi performanse modela za odabir osobina
    - Embeded metoda - kombinuje odabir osobina sa procesom treniranja modela

- **Izvlačenje osobina:** predstavlja proces ubacivanja novih osobina kombinovanjem i transformisanjem postojećih. Cilj ovog postupka je kreiranje skupa osobina koji obuhvata srž originalnih podataka u nižem dimenzionalnom prostoru. Postoje par metoda za izvlačenje osobina: PCA, LDA, GDA

4. Metode za redukciju dimenzionalnosti

4.1 Metode Linearne algebra [5]

Pri odabiru metoda za redukciju podataka potrebno je uzeti određene mogućnosti koje nudi sam skup podataka. Potrebno je sagledati da li su dostupne labele podataka. U zavisnoti od cilja obrade podataka, potrebno je proveriti koja od metoda je najadekvatnija. Takođe, veliki broj skupova podataka nije u potpunosti popunjen, već ima vrednosti koje nedostaju. Pored toga može se desiti da podaci nisu lepo distribuirani i da se ne mogu linearno razdvojiti. U zavisnosti od osobina samog skupa podataka neće baš sve metode dati validne rezultate.

4.1.1 Non-negative Matrix factorization

Jedan od načina za redukciju dimenzionalnosti osobina predstavljenih u vidu matrice ne-negativnih vrednosti je upravo primena algoritma NMF. NMF predstavlja nenadgledanu tehniku za linearnu redukciju dimenzionalnosti. Sama tehnika bazira se na faktorizaciji originalne matrice, to jest njenoj podeli na više manjih matrica pri čemu njihov proizvod odgovara originalnoj. Početna matrica se razbija na veći broj manjih matrica koje predstavljaju latentni vezu između opažanja i njihovih osobina. Ovaj algoritam se može koristiti za redukciju dimenzionalnosti upravo zbog samog procesa množenja matrica. Množenje matrica uključuje dva faktora (same matrice koje se množe), koji mogu biti značajono manje dimenzionalnosti od rezultujuće matrice. Formalno konstruisano, u koliko je željeni broj rezultujućih osobina *r,* NMF tehnika će odraditi faktorizaciju originalne matrice tako da:

$$V \approx W * H$$

gde V predstavlja originalnu matricu veličine (*d,_n)*, gde su *d* osobine, *_n* opažanja., W matrica dimanzije *(d,r)* i H matrica dimenzije *(r,n)*. Podešavanjem vrednosti *r* možemo dobiti željenu redukciju dimenzionalnosti.

Jedan od glavnih zahteva ovog algoritma, što se može i zaključiti na osnovu samog naziva, je da matrica ne sme sadržati negativne vrednosti. Takođe, jedan od problema sa ovim algoritmom je što se ne može prikazati kako se dobijaju redukovane osobine. Tako da je najbolji način za dobijanje optimalnog broja komponenata je grubom silom, to jest testiranjem određenog skupa vrednosti.

[5] Machine Learning with Python Cookbook: Practical Solutions from Preprocessing to Deep Learning

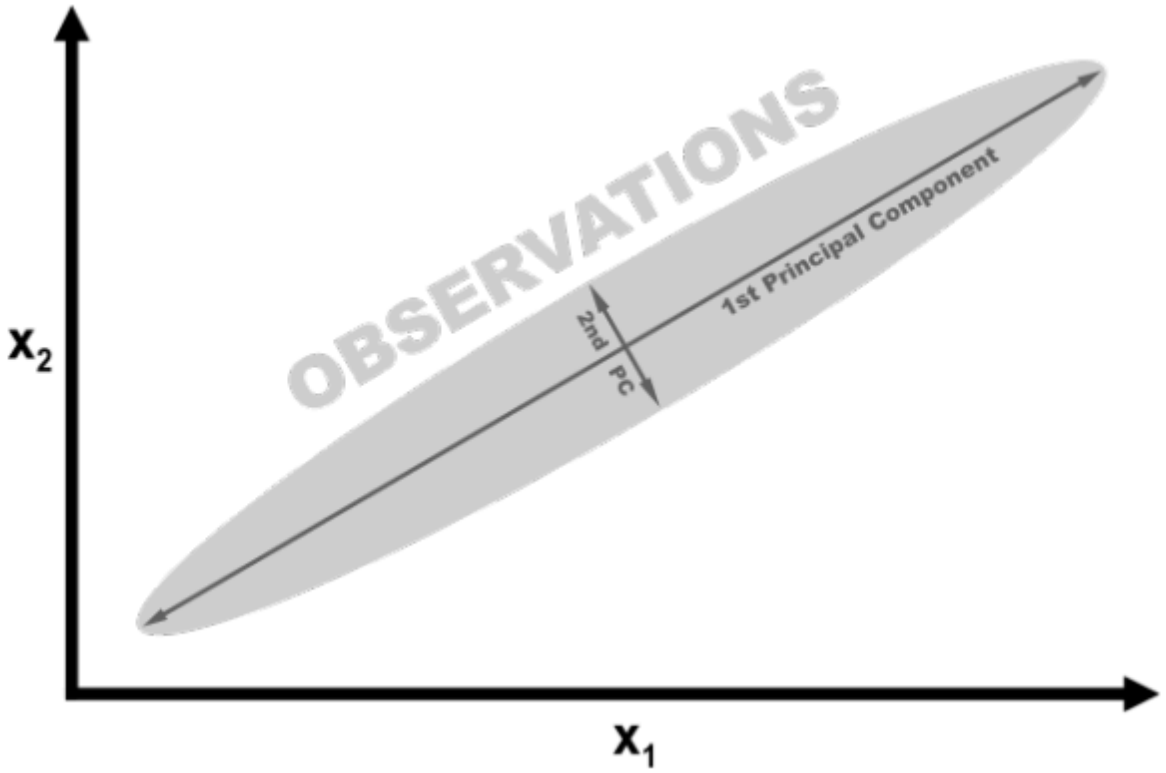
4.1.2 PCA

PCA (*Principal Component Analysis - "Analiza glavne komponente"*) - je jedna od popularnijih tehnika za linearnu redukciju dimenzionalnosti. Ova tehnika bazira se na projekciji opažanja na (poželjnije manji) određeni broj glavnih komponenata matrice osobina, koje zadržavaju varijansu, to jest održavaju povezanost podataka. PCA, poput prethodne tehnike, predstavlja nenadgledanu tehniku, to jest ne preuzima informacije iz ciljanog vektora, već se bazira samo na matrici osobina.

Sam PCA primenjuje se u situacija kada je porebno zadržati varijansu podataka, a otkloniti, smanjiti pri tome dimenzionalnost. Sama tehnika podrazumeva da se u okviru varijanse osobina prenose informacije. Zbog toga vrši potragu za najvećom varijansom jer podrazumeva da ona nosi najviše informacija. Sam PCA bazira se na par koraka:

1. *Standardizacija:* primenjuje se u slučaju da se podaci nalaze na različitim skalama (oduzima se srednja vrednost i vrši se deljenje standardnom devijacijom)
2. *Izračunavanje matrice kovarijanse*
3. *Izračunavanje eigen vrednosti i vektora zarad određivanja komponenti:* eigen vektora predstavljaju smer maksimalne varijanse, dok vrednosti ukazuju na magnitutu same varijanse duž definisanog pravca
4. *Sortiranje eigen vektora po vredsnotima:* sortiranje se vrši u opadajućem redosledu
5. *Odabir komonenata:* brira se k eigen vektora, to jest glavnih komponenti, pri čemu k ukazuje na željenu dimenzionalnost podataka
6. *Transformacija podataka:* množenje originalnih standardizovanih podataka dobijenim komponentama zarad dobijanja novog skupa

Zarad razumevanja procesa na kome se bazira PCA u nastavku biće prikazan primer koji se sastoji od dve osobine: *x1* i *x2*. [Slika 2]



Slika 2

Na osnovu same ilustracije, može se zaključiti da dužina ili ti smer sa najvećom varijansom predstavlja prvu glavnu komponentu, dok visina ili ti smer sa sledećom najboljom varijansom koja je ortogonalna prvoj komponenti predstavlja drugu glavnu komponentu.

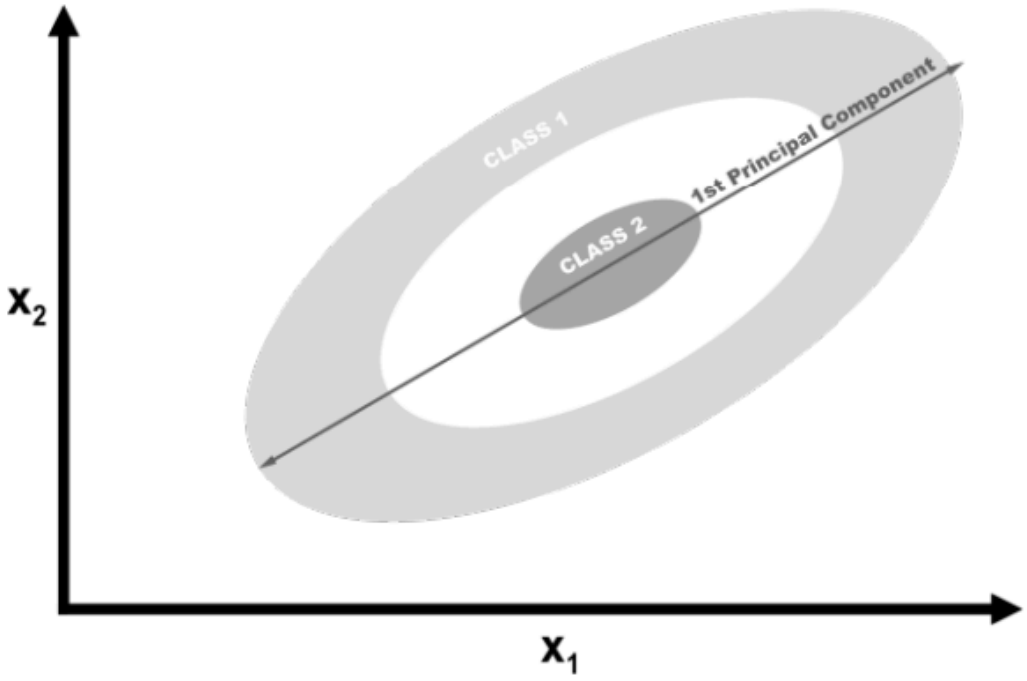
Kako bi se uspešno redukovala dimenzionalnost, jedna od tehnika je preslikavanje 2D opažanja na 1D glavnu komponentu. U slučaju da se primeni navedena tehnika, gube se podaci obuhvaćeni u drugoj glavnoj komponenti. Međutim, u većini slučajeva, ovakav vid razmene smatra se prihvatljivim. Na ovom principu se bazira PCA.

Kod standardne PCA tehnike potrebno je navesti odgovarajući broj komponenata. Navedeni broj ima dve operacije u zavisnosti od vrednosti argumenta. U slučaju da je navedena vrednost veća od 1, algoritam će vratiti navedeni broj osobina. Ovo samim tim zahteva odabir optimalnog broja osobina. Da bi se izbegla potraga za optimalnim brojem osobina, u koliko se argument postavi na vrednost između 0 i 1, PCA će vratiti minimalan broj osobina koji zadržava navedenu varijansu podataka. Najčešće korišćene vrednosti su između 0.95 i 0.99, to jest zadržava se nekih 95% varijanse originalnih podataka. Još jedan od parametara koji se može postaviti je *whiten.* U koliko je parametar postavljen na potvrdnu vrednost, vrednosti svake glavne komponente transformišu se tako da njihova njihova srednja vrednost bude nula a

varijansa jedinična. Još jedan parametar je *svd_solver,* koji u koliko je postavljen na *randomize* pokreće stahostički algoritam za određivanje prve glavne komponente.

Pored toga, PCA tehnika se može primeniti i u drugačijim situacijama. U koliko je potrebno smanjiti dimenzionalnost podataka koji se linearno ne mogu razdvojiti, PCA je najadekvatnije rešenje. Standardni PCA koristi linearnu projekciju zarad smanjenja dimenzionalnosti. Ako se podaci mogu linearno razdvojiti (to jest moguće je povući pravu liniju između podataka koji pripadaju različitim klasama), onda PCA radi kako treba. Međutim, kod podataka koji se ne mogu linearno razdvojiti (podaci koji pripadaju različitim klasama mogu se samo razdvojiti krivudavom linijom), linearna transformacija neće raditi baš najbolje.

Zarad bolje ilustracije, iskorišćen je primer u okviru koga podaci koji pripadaju jednoj klasi u potpunosti okružuju podatke druge. [Slika 3]



Slika 3

U koliko se na navedenom primeru primeni linearni PCA za redukciju dimenzionalnosti, navedene klase bi se linearno projektovale na prvu glavnu komponentu, tako da bi postale isprepletane. U idealnoj situaciji, potrebno je rešenje koje bi redukovalo dimenzionalnost i učinilo podatke linearno separabilnim. Poseban tip PCA tehnike, Kernel PCA rešava ovaj problem.

Kernel omogućava projekciju podataka na veću dimenzionalnost, gde su podaci linearno razdvojivi. Najčešće korišćeni kernel je *Gaussian radial basis* funkcija, ali postoje i polinomialni kerneli kao i sigmoid kernel. Jedna od mana kod kernel PCA tehnike je broj parametara koje je potrebno specificirati. Kod ove tehnike navodi se broj parametara. Pored toga, kerneli dolaze sa sopstvenim hiperparametrima koji se moraju postaviti. Koje parametre je potrebno postaviti? Određivanje parametara vrši se grubom silom i tesiranjem. Sam model može se trenirati više puta sa različitim parametrima kernels. Kada se pronađe skup parametara koji generiše najbolje rezultate, proces potrage je završen.

Kada se PCA primenju na prediktivne modele potrebno je uzeti u obzir raspodelu podataka za treniranje samog modela. Naime, pre početka procesa treniranja set podataka je potrebno podeliti na deo za treniranje i deo za testiranje. Samim tim, u koliko primenimo PCA tehniku nad seton za treniranje potrebno je primeniti je i nad podacima za testiranje. Međutim potrebno je obratiti pažnju na sledeće:

- Podaci za testiranje i treniranje se ne smeju kombinovati zarad određivanja komponenata PCA tehnike. Ovo bi narušilo proces generalizacije, jer bi podaci za testiranje na neki način bili ubačeni u set za treniranje. Na ovaj način ugrozila bi se mogućnost modela za generalizaciju.
- Tehinika PCA se ne sme odvojeno sprovoditi nad skupovima podataka. Dobijeni vektori se mogu razlikovati za zasebne skupove podataka, pa će samim tim jedinstvene varijanse biti različite. U ovom slučaju vršilo bi se poređenje podataka na različitim osama.

Zbog navedenih razloga neophodno je nad odvojenim skupovima podataka primeniiti istu transformaciju.

Prednosti PCA tehnike:

- Efikasno vrši redukciju dimenzionalnosti, što doprinosi modelima
- Komponente koje se definišu u okviru PCA tehnike su ortogonalne (nezavisne), što obuhvata informacije koje su nezavisne, pri čemu se uprošćava interpretacija redukovanih osobina
- Uklanja nepotrebne informacije jer se oslanja na najveću varijansu
- Olakšava samu vizualizaciju podataka i njihovo razumevanje

Nedostaci PCA tehnike:

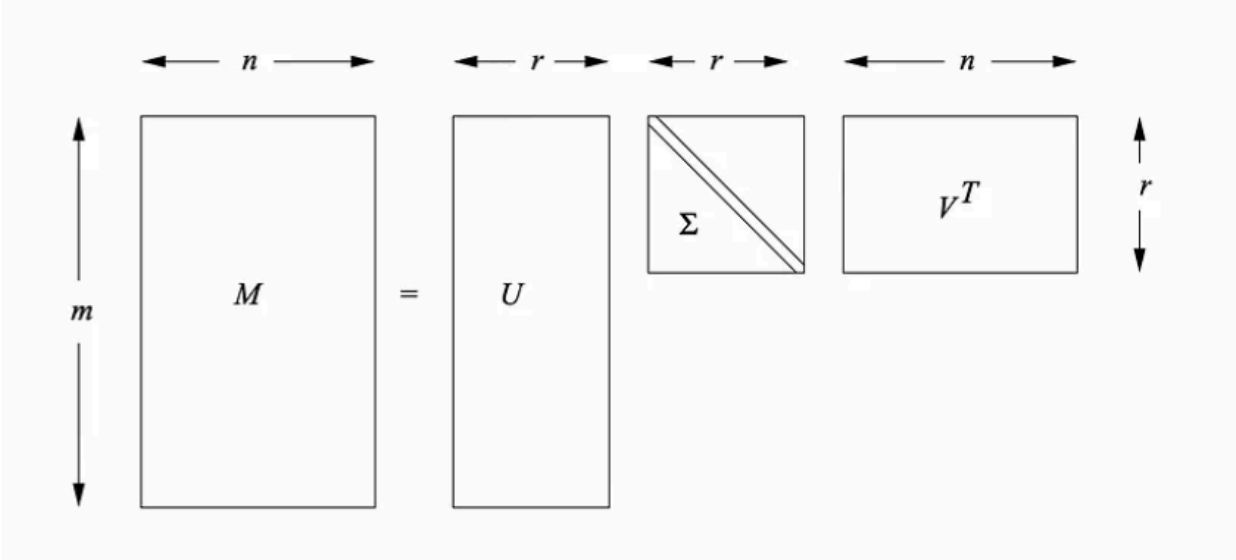
- Kako dobijene komponente predstavljaju linearne kombinacije originalnih podataka, interpretabilnost originalnih osobina se može izgubiti
- Potrebno je voditi računa o lineatnosti podataka
- Osetljiva tehnika na skaliranj
- Outlajeri mogu značajno uticati na rezultate same PCA tehnike

Iz navedenih razloga potrebno je voditi računa kada se sama tehnika primenjuje. Najbolje je primenjivati kod visoko dimenzionalnih skupova podataka koje je teško prikazati, kod

podataka koji su linearno povezani i kada u skupu postoji veliki broj osobina koje su visoko korelisane.

4.1.3 SVD

SVD (*Singular Value Decomposition*) predstavlja tehniku koja se bazira na faktorizacija matrica. Ova tehnika vrši dekompoziciju originalne matrice na tri nove i na taj način omogućava prikaz početne matrice u redukovanoj formi. [Slika 4]



Slika 4

Od početne matrice M dimenzija *(m,n)*, to jest originalnog skupa podataka sa n osobina i m podataka, SVD vrši dekompoziciju na tri matrice. Matrica U predstavlja ortogonalnu matricu, Σ je diagonalna matrica dimenzija *(m,r)*, dok V predstavlja ortogonalnu matricu dimenzija *(r,n)*. Sam dimenzija *r* odgovara ranku matrice M.
Σ - dijagonalni elementi ove matrice predstavljaju singularne vrednosti matrice M organizovane u opadajućem redosledu.
U - kolone ove matrice predstavljaju leve singularne vektore matrice M.
V - kolone ove matrice predstavljaju desne singularne vektore matrice M.
Navedeni vektori kreiraju ortogonalnu basu prostora reda matrice M.

Za samo redukovanje dimenzionalnosti podataka koristi se posebna verzija SVD tehnike - TSVD.

TSVD (*Truncated Singular Value Decomposition*) je tehnika slična PCA tehnici. Potrebno je naglasiti da veoma često PCA upravo koristi gore navedenu SVD tehniku. Sama SVD tehnika će kreirati faktor matrice, dok će TSVD vratiti odgovarajuće faktore prethodno definisanih dimenzija. TSVD tehnika će vratiti k najvećih singularnirnih vrednosti u matrici Σ. Ove kolone se mogu selektovati iz ove matrice, ali pored toga se mogu uzeti i iz redova matrice V. Nova matrica B se može rekonstruisati iz matrice M putem formule:

$$B = U * \Sigma$$
$$B = V^{t} * A$$

Pri čemu će matrica Σ posedovati k najboljih kolona iz originalne matrice baziranih na singularnim vrednostima, a V sadržati k najboljih redova iz originalne matrice koji takođe odgovaraju singularnim vrednostima. Sam TSVD se , za razliku od PCA, može primenjivati na razređene skupove podataka.

Prednosti SVD tehnike:

- Omogućava redukciju dimenzionalnosti zadržavajući samo najbitnijr singularne vrednosti i vektore.
- Vrši kompresiju podataka, smanjujući potreban prostor za čuvanje matrice.
- Uklanja suvišne i nepotrebne podatke.
- Numerički stabilna metoda korinsna za rešavanje linearni jednačina smeštenih u lošem okruženju.
- Dekompozicija je ortogonalna, pa se čuva vezu između redova i kolona originalne matrice.
- Pogodno za sisteme preporuke.

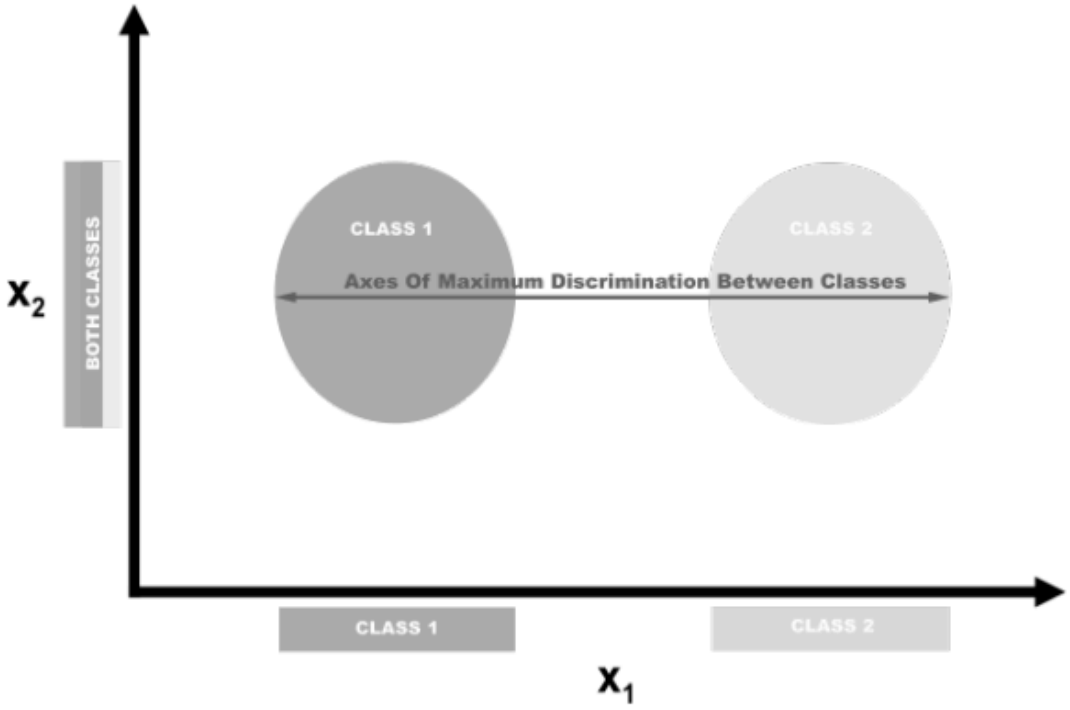
Mane SVD tehnike:

- Izračunavanje za velike matrice može biti skupo.
- Zahteva dosta memorije za velike matrice.
- Osetljiv na nedostajuće vrednosti u skupu podataka, pa je potrebno primenjivati posebne tehnike za njihovo popunjavanje.

SVD metodu je najbolje primenjivati u situacijama gde se radi sa sistemima preporuke, kada je potrebno redukovati dimenzionalnost a zadržati strukturu, kada je potrebno kompresovati velike skupove podataka, za procesiranje signala i za rešavanje linearnih jednačina postavljenih u lošem okruženju.

4.1.4 LDA

LDA (*Linear Discriminant Analysis*) predstavlja metodu koja se pre svega koristi zarad klasifikacije podataka. Pored toga jednu od većih primena ova metoda pronašla je na području redukcije dimenzionalnosti. Sama metoda funkcioniše na sličan način kao i PCA. Vrši se projekcija iz više dimenzionalnog prostora na niži. Međutim, kod PCA tehnike najvažnije je pronaći ose koje zadržavaju maksimalnu varijansu, tj predstavljaju pravac najbitnijih informacija. Kod LDA metode postoji još jedan dodatan cilj. To je maksimizacija razlike između klasa. [Slika 5]. Sam LDA algoritam spada u grupu nadgledanih tehnika jer svaka od tačaka iz skupa podataka poseduje unapred definisanu klasu kojoj pripada.



Slika 5

Sam primer ukazuje na postojanje dveju različitih klasa i dve osobine. U koliko se podaci projektuju po y-osi, dve klase podataka se ne mogu lako razdvojiti. Dok kada se projekcija izvrši po x-osi ostaje vektor osobine i dimenzionalnost se redukuje za jedan. Ovaj postupak idalje zadržava odvojenost klasa.

Kada se primenjuje sama metoda neophodno je navesti broj komponenata koji ukazuje na broj očekivanih osobina. Za određivanje broja komponenata može se iskoristiti dodatna mogućnost za prikaz varijanse za skup izlaznih osobina. Koraci sprovođe LDA algoritma su:

- Izračunavanje srednje vrednosti vektora različitih klasa
- Izračunavanje rasute matrice koja prikazuje resutost podataka za svaku od klasa
- Izračunavanje razlike između različitih klasa na osnovu izračunatih matrica

- Izračunavanje eigen vektora i vrednosti
- Sortiranje eigen vektora u opadajućem redosledu po vrednostima, kako bi se odabrao k najboljih. Nakon sortiranja parova, potrebno je konstruisati eigen matricu na osnovu dva najbolja para.
- Na osnovu dobijenje matrice i originalne matrice vrše se izračunavanje redukovanih vrednosti.

Prednosti LDA metode:

- LDA je dizajniran da povećava distancu između različitih klasa, što poboljšava efikasnost procesa klasifikacije.
- LDA redukuje dimenzionalnost i pri tome uzima u obzir informacije klasa.

Mane LDA metode:

- Navedena metoda je osetljiva na postojanje outlajera.
- Sam algoritam podrazumeva da su podaci u okviru klase normalno raspoređeni
- Ne može dati dobre rezultati u slučaju malog broja primera po klasi. Sa većim brojem primeraka povećava se estimacija klasnih parametara.

Sam LDA algoritam najbolje je primenjivati za klasifikaciju podataka u unapred predefinisane klase, za redukciju dimenzionalnosti kada je potrebno zadržati informacije bitne za razdvajanje klasa, kada su podaci normalno distribuirani među klasama ili kada zadatak zahteva redukciju dimenzionalnosti uz korišćenje labela klasa.

4.2 Manifold Learning Metode (Non-linear) [6]

"*Manifold učenje*" (*Manifold Learning*) predstavlja posebno polje mašinskog učenja i analize podataka. Samo polje pokazalo se dobro u potrazi za skrivenim strukturama u okviru kompleksnih skupova podataka. Jednostavnije rečeno, "*manifold učenje*" razmatra svaki sloj kompleksnosti posebno, dimenziju po dimenziju.

Metodama "*manifold učenja*" može se pristupiti i pod drugačijim nazivom - nelinearno redukcija dimenzionalnosti. Tehnike nelinearne redukcije dimenzionalnosti se primenjuju kod složenih skupova podataka visoke dimenzionalnosti. Suština ovih metoda ogleda se u otkrivanju unutrašnje strukture podataka njihovim mapiranjem na prostor niže dimenzionalnosti bez uklanjanja suštinskih karakteristika. Sam termin "*manifold*" odnosi se na matematički koncept - topološki prostor koji lokalno odgovara Euklidovom prostoru ali koji globalno može izložiti kompleksnu strukturu.

[6] Manifold - skup tačaka koji čine određenu vrstu skupa, poput topološki zatvorenih oblasti ili analogija ovoga u tri ili više dimenzije.

Osnovni koncepti nelinearnih tehnika redukcije dimezionalnosti su: *lokalna linearnost i unutršnja dimenzionalnost.* Pod pojmom *lokalne linearnosti* ove tehnike podrazumevaju da se tačke koje leže na ili blizu osnove manifolda mogu lokalno aproksimirati kao linearna veza. Ova lokalna linearno omogućava proces mapiranja visoko dimenzionalnih podataka na niži nivo. *Unutrašnja dimenzionalnost* samog skupa podataka predstavlja minimalan broj dimenzija potrebnih da se obuvati njegova struktura. Tehnike koje spadaju u ovo polje pokušavaju da otkriju upravo navedenu vrednost samim smanjivanjem dimenzionalnosti skupa podataka.

Same nelinerane tehnike redukcije dimenzionalnosti pronašle su primenu u različitim poljima, poput:

1. **Procesiranje slika i računarski vid** - omogućavaju izvlačenje osobina iz podataka slika i na taj način omogućavaju zadatke poput prepoznavanja objekata, uklanjanja nepravilnosti sa slika i prepoznavanja lica.
2. **“Genomik” analiza podataka -** mogu biti korisne u slučaju analize i prikaza velike količine podataka za identifikaciju genetski šablona i otkrivanja bolesti.
3. **Detekcija anomalija -** smanjenjem dimenzionalnosti podataka bez remećenja njihove strukture, mogu se poboljšati performanse algoritama za detekciju anomalija.

U okviru ovog polja razvijeno je par algoritama koji se bave redukcijom dimenzionalnosti, poput: Isomap, t-SNE, LLE.

4.2.1 Isomap [7]

Isomap zapravo predstavlja izometrijsko mapiranje. Ova tehnika spada u skup nelinearnih metoda za redukciju dimenzionalnosti i bazira se na spektralnoj teoriji. Isomap metoda pokušava da očuva geodetsku distancu u prostoru manjih dimenzija. Pre svega, ova metoda kreira mrežu suseda i koristi distancu grafa kako bi aproksimirala geodetsku distancu između svih mogućih parova tačaka. Potom upotrebom dekompozicije eigen vrednosti nad definisanom matricom distanci, pronalazi odgovarajuću nižu dimenzionalnost. Kod ovih metoda, Euklidijanova distanca je dobra ako i samo ako struktura suseda može biti aproksimirana kao linearna. U koliko struktura suseda ima dosta rupa, onda Euklidova distanca može biti pogrešna. U poređenju sa ovim, u koliko se proprati manifold zarad izračunavanja distance dobiće se bolja aproksimacija distance dve tačke.

Kada Isomap kreira graf, izvuče distance iz njega i primeni eigen dekompoziciju, kao rezultat ne dobija se samo globalna struktura skupa podataka u okviru niže dimenzionalnosti, već se pored toga dobija i lokalna struktura. Jedna od mana ove metode je da neće pokazati dobro u koliko manifold ima dosta rupa. Pored toga u koliko se ne konstruiše lepo graf suseda čak i malo pomeranian parametri mogu dati jako loše rezultate.

[7] https://www.geeksforgeeks.org/isomap-a-non-linear-dimensionality-reduction-technique/

4.2.2 t-SNE [8]

Ova metoda ne obuhvata samo lokalnu strukturu podataka visoke dimenzionalnosti već pored toga čuva i globalnu strukturu podataka poput klastera. Ima jako dobru sposobnost da dobro odvoji klastere podataka. Sama metoda bazira se na stohastičkom ugrađivanju suseda (*Stochastic neighbor embedding - SNE*).

SNE -koristi pristup verovatnoće kako bi ugradio skup podataka visoke dimenzionalnosti u nešto nižu dimenzionalnost održavajući pri tome strukturu suseda. SNE pokušava da smanji razliku u verovatnoći distribucije kako u visokoj dimenzionalnosti tako i u niskoj, kako bi dobio savršenu reprezentaciju tačaka u prostoru niže dimenzionalnosti.

Koraci za sprovođenje t-SNE algoritma:

1. Potrebno je slične tačke okupiti na jednom mestu, dok će se različite tačke udaljiti. Samo premeštanje se vrši na osnovu sličnosti.
2. Sam koncept sličnosti ovde se bazira na verovatnoći distibucije. Na osnovu ove vrednosti određuju se susedi određene tačke.
3. t-SNE komplikovanost - ovo ukazije na zbijenost određenih tačaka u osnosu na određenu, posmatranu tačku. U koliko tačke koje se nalaze u okviru istog klastera imaju veliku gustinu, onda će imati visoke vrednosti u odnosu na one koje se ne nalaze u okviru istig klastera. Tačke sa manjom gustinom imaju ravnije krive.
4. Normalizacija kompleksnosti - u koliko posmatramo dva skupa tačaka i prosečna distanca između u grupi koja je gušće raspoređena je polovina prosečne distance grupe koja nije toliko gusto raspoređena, da bi bile tretirane isto potrebno je primeniti normalizaciju. Potrebno je za svaku grupu sumu sličnosti svesti na istu vrednost.
5. Kreiranje matrice sličnosti
6. Na osnvovu matrice sličnosti vrši se kreiranje nižeg dimenzionalnog prostora

Prednosti algoritma:

- Može raditi sa nelinearnim podacima
- Održava lokalnu strukturu

Mane algoritma:

- Složenost izračunavanja je velika
- Nije stabilan - za iste hiperparametre mogu se dobiti različite vrednosti jer je uključen proces randomizacije.

[8] https://www.geeksforgeeks.org/ml-t-distributed-stochastic-neighbor-embedding-t-sne-algorithm/

4.2.3 LLE [9]

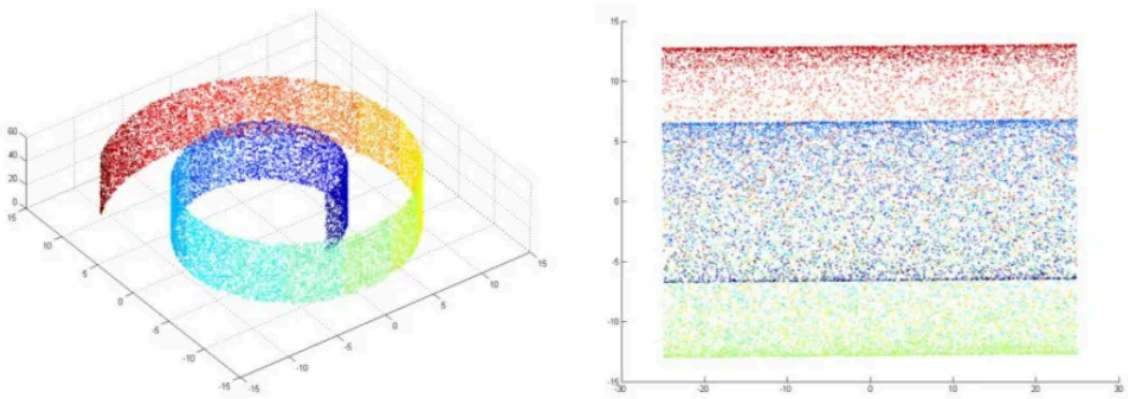
LLE (*Locally Linear Embedding*) predstavlja nenadgledanu metodu za redukciju dimenzionalnosti. Ova tehnika pokušava da smanji dimenzionalnost a da pri tome održi geometrijske osobine originalne nelinearne strukture.

LLE metoda pre svega pronalazi k najbližih suseda za svaku od tačaka. Potom aproksimira za svaki podatak vektor koji predstavlja linearnu kombinaciju k suseda. Na samom kraju vrši proračunavanje težina koje najbolje mogu rekonstruisati definisane vektore iz svojih suseda i zatim kreira vektore niže dimenzionalnosti na osnovu dobijenih težina.

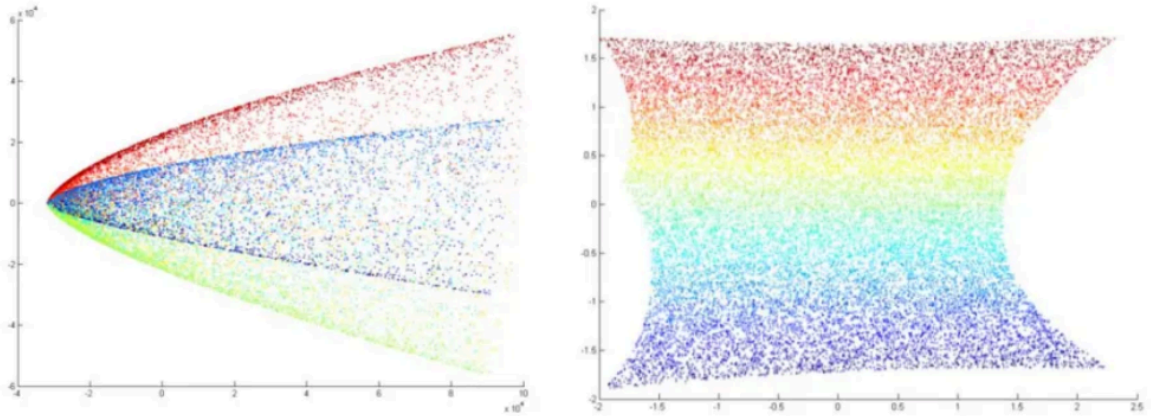
Jedna od prednosti ove tehnike ogleda se u tome da je potrebno prilagoditi samo jedan parametar, a to je broj najbližih suseda koji će činiti jedan klaster. U koliko ovaj broj bude prevelik ili isuviše mali neće biti lako održati geometriju originalnih podataka. Po određivanju suseda vrši se sakupljanje njihovih težina za svaku od tačaka kako bi se konstruisala nova. U ovom slučaju pokušava se minimizacija kost funkcije. Iz ovoga se kreira novi prostor vektora.

Prednosti ovog algoritma ogledaju se u tome što on zapravo razmatra opciju nelinearnostu u okviru strukture. LLE ima mogućnost detekcije nepravilnosti na nižim nivoima dimenzionalnih projekcija originalnih podataka. Pored toga ova metoda je bolja i od Kernel PCA i od Isomap metode jer može detektovati pojedine osobine koje ove dve metode ne mogu videti [Slika 6]. Pored toga LLE ima bolje vreme izračunavanja i zahteva manje prostora.

[9] https://medium.com/analytics-vidhya/locally-linear-embedding-lle-data-mining-b956616d24e9



From left : Swiss Roll features in 3 dimensions, Swiss Roll example using PCA Source



From left : Swiss Roll features using Kernel PCA, Swiss Roll features using LLE

Slika 6

4.3 Odabir osobina (Feature selection) [10]

Do sada diskutovalo se o tome kako smanjiti dimenzionalnost matrice osobina kreiranje novih osobina sa sličnim mogućnostima za treniranje modela ali značajno manjih dimenzija. Ovaj proces predstavlja proces izvlačenja osobina. U ovom delu biće pokrivena druga komponenta redukcije dimenzionalnosti poznata pod nazivom odabir osobina. Ovo predstavlja alternativni pristup koji se bazira na selekciji osobina visokog kvaliteta, onih koje pružaju više informacija a odbacivanjem onih koje su manje bitne.

Postoje tri tipa ovih metoda: filter metode, vraper metode i embedding metode. Filter metode biraju osobine na osnovu statističkih vrednosti. Vraper metode koriste grubu silu za

[10] Machine Learning with Python Cookbook: Practical Solutions from Preprocessing to Deep Learning

određivanje najboljeg podskupa osobina koji kreira model sa najboljom mogućnošću predikcije. Na kraju embeded metode biraju podskup kao nastavak na proces treniranja modela. Same embeded metode su ugrađene u okviru odrešenih algoritama za učenje pa ih je teže objasniti i pokriti njihov način rada. Neke od osnovnih metoda za odabir osobina su: selekcija na osnovu varijanse, na osnovu bitnosti osobina i korelacione matrice. Pored toga može se vršiti i rekurzivno uklanjanje osobina.

4.3.1 Selekcija na osnovu varijanse

Uklanjanje osobina na osnovu njihove varijanse spada u jedan od osnovnih pristupa selekcije. Ovaj pristup motivisan je idiom da osobine sa niskom varijansom ne poseduju dovoljnu količinu bitnih informacija, pa se samim tim smaraju manje zanimljivim i korisnim. Ovaj pristup podrazumeva izračunavanje varijanse za svaku od osobina:

$$operatornameVar(x) = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^2$$

gde x predstavlja vektor osobine, xi je individualna vrednost osobine i μ je srednja vrednost osobine. Nakon određivanja vrednosti varijanse svake od osobina potrebno je odbaciti one koji se nalaze ispod unapred definisane granične vrednosti.

Prilikom primene ove metode potrebno je uzeti u obzir da svi podaci moraju biti definisani u istim jedinicama, jer varijanse nisu centrirane. U koliko nisu u jednakim jedinicama, ovaj pristup nije validan. Pored toga, granična vrednost se bira ručno, samim tim potrebno je voditi računa.

Pored toga potrebno je uzeti u obzir da gore navedena formula važi samo za numeričke podatke. Kod binarnih primenjuje se sledeće:

$$\text{Var}\ (x) = p(1-p)$$

gde p odgovara proporcionalnosti primeraka u klasi. Samim tim, postavljanjem vrednosti p mogu se otkloniti sve osobine kod kojih većina primerakapripada jednoj klasi.

4.3.2 Selekcija na osnovu važnosti osobina

Koliko je neka osobina značajna za određeni model može se videti naosnovu njegovog propertija *feature_importance.* Ovaj properti vraća skor za svaku od osobina u okviru navedenog

skupa podataka. Što je skor to je osobina bitnija za sam model i određivanje izlazne vrednosti. Ova metoda uglavnom dolazi uz klasifikatore.

Kod kategoričkih vektora moguće je primeniti i *chi-squere* metode koja statistički proverava povezanost dva kategorička vektora. Ova statistika je jedan broj koji ukazije na to koliko razlike postoji između posmatranog broja primeraka i broja u slučaju da nema nikakve povezanosti. Izračunavanjem *chi-square* statistike između osobine o ciljanog vektora, dobija se mera nezavisnosti između navedenih. U koliko je cilj nezavistan od navedene osobine, ona se može ukloniti iz skupa jer ne poseduje bitne informacije za proces klasifikacije. Sa drudge strane, u koliko je velika povezanost onda je navedena osobina jako bitna i pruža dosta informacija u procesu treniranja

U koliko se *chi-square* koristi u procesu odabira osobina, onda je potrebno za svaku osobinu proračunati povezanost sa ciljem. Nakon izračunavanja neophodno je odabrati k najboljih osobina a ostale ukloniti. Potrebno je naglasiti da se ova vrednost može izračunati samo između kategoričkih vektora.

4.3.3 Selekcija na osnovu korelacione matrice

Jedan od klavnih problema pri radu sa velikim skupovima podataka je postojanje osobina koje su visoko korelisane. U koliko su dve osobine visoko korelisane to ukazuje da obe osobine poseduju iste informacije i u većini slučajeva je redundantno posedovati obe. Rešenje za visoko korelisane osobine je jednostavno - ukloniti jednu od njih iz skupa podataka.

4.3.4 Rekurzivno uklanjanje osobina

Ideja iza rekurzivnog uklanjanja osobina (*Recursivelz Eliminating Features - RFE*) je treniranje modela sa određenim parametrima (težinama ili koeficijentima) neprekidno. Prvi put kada se model trenira, uključuju se sve osobine. Nakon toga, pronalaze se osobine sa najmanji parametrima (podrazumeva se da su osobine skalirane ili standardizovane), što označava da su manje bitni pa se mogu ukloniti iz skupa.

Problem nastaje u određivanju koliko puta treba ponoviti treniranje samog modela i na koliko osobina se treba zadržati. Sam proces može se ponavljati dok ne preostane samo jedna osobina u okviru skupa, ali jedan od boljih pristupa je uljučivanje “kros validacije”.

Sami podaci poseduju cilj koji je potrebno predvideti i matricu osobina na osnovu kojih treba izvesti predikciju. Kada se trenira model, podatke je potrebno podeliti u dve grupe: set za treniranje i set za testiranje. Sam model se trenira upotrebom skupa za testiranje. Nakon treniranja, model se primenjuje nad test skupom zarad provere ispravnosti njegovih predikcija.

To je i osnovna ideja “kros validacije”. Sam model pušta se nad određenim brojem raspodela i proverava se njegova ispravnost.

“Kros validacija” može se iskoristiti u procesu određivanja optimalnog broja osobina koje je potrebno zadržati u procesu RFE-a. Bolje rečeno, nakon svake iteracije, pušta se “kros validacija” nad modelom kako bi se proverila ispravnost rada samog modela. U koliko rezultati ukažu da je model napredovao sa uklanjanjem osobine, proces se nastavlja. Međutim, u koliko je rad modela postao gori, osobina se vraća nazad u skup i selektuje se kao najbolja osobina za rad modela.

4.3.5 Lasso Regresija [11]

Lasso regresija odnosi se na model koji podrazumeva linearnu povezanost između ulaznih vrednosti i ciljane vrednosti. Kada govorimo o jednom ulazi, ova veza predstavlja liniju, dok u više dimenzionalnim prostorima ova veza je predstavljena kao hiper-ravan koja povezuje ulazne vrednosti sa izlaznim. Koeficijenti ovog modela pronalaze se u postupku optimizacije koji pokušava da minimizuje sumu kvadratne greške između predikcija i očekivanih vrednosti.

Problem koji može nastati sa linearnom regresijom je da koeficijent modela može postati jako veliki, čineći model osetljivijim na ulaze i nestabilnim. Ovo je naročito tačno za premiere gde postoji nekoliko primeraka nego li ulaznih promenljivih.

Jedan od pristupa za postizanje stabilnosti modela je promena *loss funkcije* da uključuje veći kost za modele sa velikim koeficijentima. Modeli linearne regresije koji se baziraju na ovim funkcijama i koriste ih u toku treniranja se nazivaju još modelima sa kaznom.

Navedeni modeli mogu se iskoristiti za odabir osobina. Naime nakon treniranja, iz modela je moguće izvići značaj za svaku od osobina i odabrati samo one sa većim vrednostima.

[11] https://machinelearningmastery.com/lasso-regression-with-python/

[1] Jason Brownlee, 30 June, 2020, Introduction to Dimensionality Reduction for Machine Learning, Introduction to Dimensionality Reduction for Machine Learning - MachineLearningMastery.com

[2] From Wikipedia, the free encyclopedia, 20 December, 2023, Curse of dimensionality, Curse of dimensionality - Wikipedia

[3] Gillis, S., Alexandar, September 2023, Dimensionality Reduction, What is Dimensionality Reduction? | Definition from TechTarget

[4] May 06 2023, Introduction to dimensionality reduction, Introduction to Dimensionality Reduction - GeeksforGeeks

[5] Albon, Chris, April 2018, Machine Learning with Python Cookbook: Practical Solutions from Preprocessing to Deep Learning, Chapter 9

[7] 02 January,2024, Isomap: Non-linear Dimensionality Reduction Technique, Isomap | A Non-linear Dimensionality Reduction Technique - GeeksforGeeks

[8] 03 August, 2023, T-distributed Stochastic Neighbor Embedding (t-SNE) Algorithm, ML | T-distributed Stochastic Neighbor Embedding (t-SNE) Algorithm - GeeksforGeeks

[9] Mihir, 10 October, 2019, Locally Linear Embedding (LLE), Data Mining and Machine Learning, Locally Linear Embedding (LLE) | Data Mining and Machine Learning | by Mihir | Analytics Vidhya | Medium

[10] Albon, Chris, April 2018, Machine Learning with Python Cookbook: Practical Solutions from Preprocessing to Deep Learning, Chapter 10

[11] Brownlee, Jason, 06 October, 2021, How to develop LASSO Regression Models with Python , How to Develop LASSO Regression Models in Python - MachineLearningMastery.com