1. **Što je strojno učenje? Koja je razlika između eksplicitnog programiranja računala i strojnog učenja? Navedite kakvi problemi se rješavaju pojedinim pristupom (primjeri).**

Strojno učenje je grana umjetne inteligencije koja se bavi razvojem algoritama i tehnika koje omogućuju računalima da nauče iz podataka i obavljaju zadatke bez eksplicitnog programiranja. Strojno učenje omogućuje računalu da “uči” iz primjera i iskustava te samo donese odluke na temelju tih podataka.

Razlika između eksplicitnog programiranja računala i strojnog učenja leži u pristupu rješavanju problema. U eksplicitnom programiranju, programer mora detaljno definirati pravila i korake za rješavanje problema. Ovo je prikladno za probleme za koje se mogu unaprijed predvidjeti svi mogući scenariji (ishodi). Strojno učenje koristi algoritme koji prilagođavaju svoje modele i parametre kako bi minimizirali greške na temelju primjera. Ovo je prikladno za probleme kod kojih ne možemo definirati sve moguće ishode.

Primjeri problema koji se rješavaju **eksplicitnim programiranjem računala**:

* Algoritam sortiranja
* Definiranje pravila za upravljanje prometnim signalima na raskrižjima

Primjeri problema koji se rješavaju **strojnim učenjem**:

* Prepoznavanje lica (face recognition)
* Predviđanje cijena nekretnina na tržištu na osnovu povijesnih podataka
* For you page na društvenim mrežama

1. **Navedite tipove strojnog učenja te za svaki tip minimalno dva primjera.**

Nadzirano učenje (*supervised learning*):

* Procjena cijena nekretnina (regresija)
* E-mail spam filtar (binarna klasifikacija - izlaz je 1 ili 0)
* Prepoznavanje rukom pisanih brojeva (višeklasna klasifikacija)

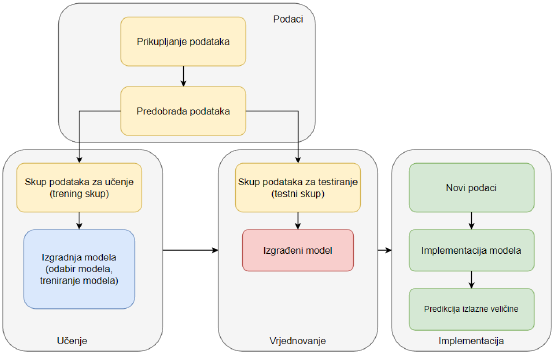
Nenadzirano učenje (*unsupervised learning*):

* Segmentacija kupaca
* Analiza društvenih mreža
* Kompresija podataka

Podržano učenje (*reinforcement leaning*):

* Igra snake
* Autonomna vozila
* Marketing i reklamiranje

1. **Nacrtajte tipičan redoslijed radnji kod primjene algoritma strojnog učenja u prediktivnom modeliranju. Objasnite pojedini korak.**



Prikupljanje podataka: ako podatkovni skup nije na raspolaganju

Predobrada podataka: dobro se upoznati s podacima

Učenje modela: primjenom linearne ili logističke regresije, neuronske mreže, ...

Izgradnja modela → odabir modela: odabir arhitekture

Izgradnja modela → treniranje modela: procjena parametara odabranog modela

**BITNO: Model se gradi isključivo na temelju skupa za učenje!**

Vrjednovanje modela: evaluiranje na skupu za testiranje. Modelu se predaju vrijednosti ulaznih veličina, a zatim se predikcije modela uspoređuju sa stvarnim vrijednostima.

Implementiranje modela: može biti zahtjevan zadatak ako model radi s velikim brojem podataka.

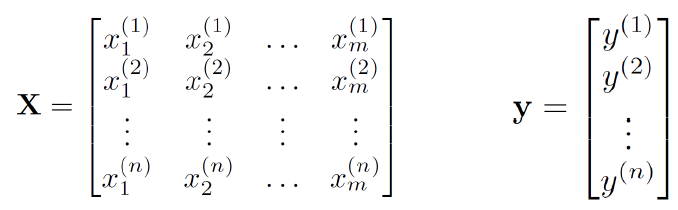
1. **Objasnite tabličnu organizaciju podatkovnog skupa i način zapisivanja podatkovnog skupa u matričnoj notaciji.**

Podaci se najčešće prikazuju u tabličnoj formi (eng. *dataset*).Stupci su veličine, a svaki redak predstavlja jedno mjerenje veličine. Česti formati su CSV, JSON, XML, ...

Npr. ulazne veličine bi bile: marka auta, tip goriva, gradska potrošnja, ...

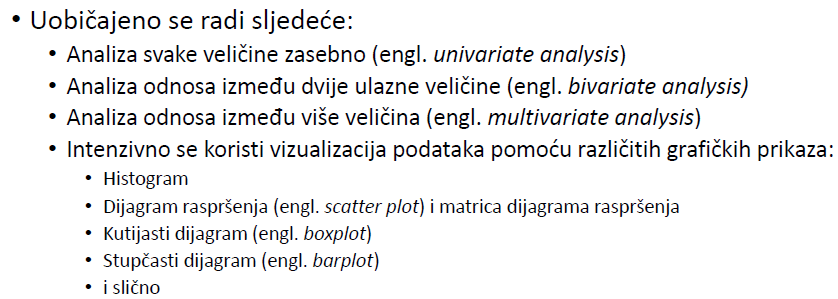
Skup od *n* podataka može se zapisati u obliku matrice X koja sadrži vrijednosti *m* ulaznih veličina i vektor y koji sadrži vrijednosti izlazne veličine za svaki podatak.

Način zapisivanja u matričnoj notaciji:



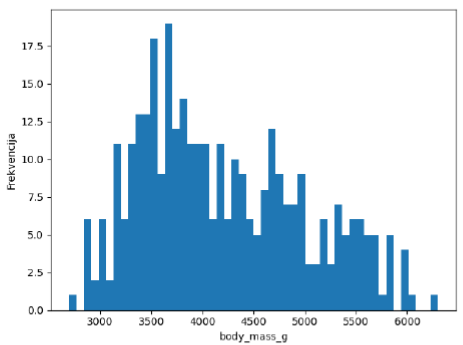
1. **Što je eksplorativna analiza podataka (EDA)? Koji grafički prikazi se koriste prilikom EDA? Objasnite ih.**

Prije predobrade podataka dobro je izvršiti EDA. To nam omogućuje uvid u podatke kao *missing values*, stršeće vrijednosti (eng. *outliers*) te odnos između ulaznih veličina.

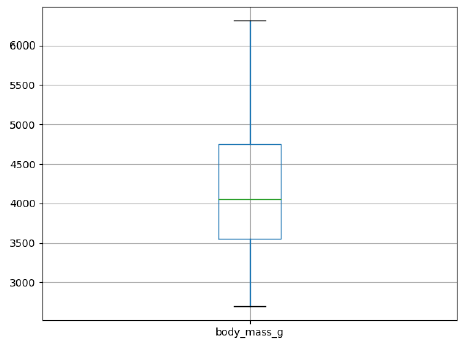


**Grafički prikazi**

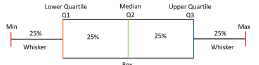
* **Histogram**:
  + Graf koji predstavlja distribuciju skupa kontinuiranih ili diskretnih podataka
  + Na x osi su vrijednosti, a na y osi je učestalost pojavljivanja vrijednosti (frekvencija)
  + Graf:

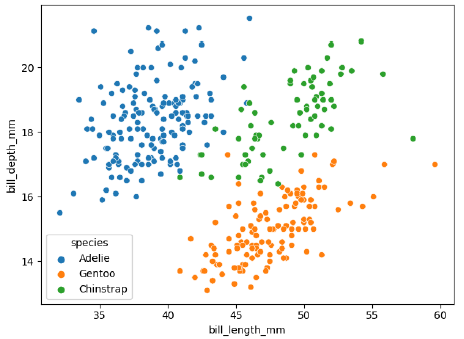


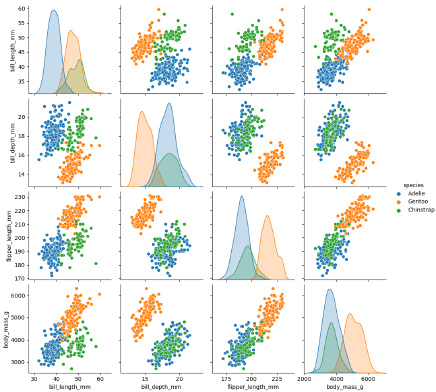
* **Kutijasti**
  + Graf koji predstavlja distribuciju skupa kontinuiranih ili diskretnih podataka
  + Koriste se za uspoređivanje dvije distribucije
  + Nacrtan je od prvog (donjeg) kvartila i trećeg (gornjeg) kvartila, a medijan je označen linijom unutar “kutije”. Ako ima stršećih vrijednosti, prikazuju se pojedinačnim točkama izvan kutije.
  + Graf:



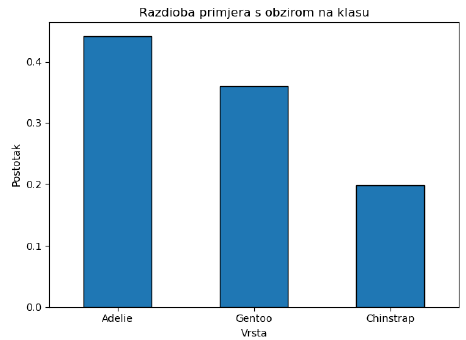
* + Tumač grafa:



* **Dijagram raspršenja**
  + Graf koji predstavlja distribuciju dvije kontinuirane ili diskretne varijable prikazane u dvodimenzionalnoj ravnini
  + Olakšava uvid u odnos između dvije veličine
  + Graf:
* **Matrica dijagrama raspršenja**
  + Prikazuje dijagrame raspršenja svih veličina (kao 1 figure)
  + Prikaz:



* **Stupčasti dijagram**
  + Daje uvid u distribuciju kategoričkih veličina, uključujući učestalost ili broj svake kategorije
  + Stupac = kategorija (x os)
  + Veličina stupca = učestalost ili broj primjera unutar kategorije (y os)
  + Graf:



1. **Objasnite osnovne postupke u predobradi podataka (podatkovnog skupa).**

* Odabrati relevantne ulazne veličine od svih dostupnih
* Ukloniti stršeće i izostale vrijednosti
* Transformirati kategoričke varijable u oblik koji zahtijevaju algoritmi strojnog učenja
* Skalirati numeričke varijable

1. **Objasnite kakve su to kategoričke veličine i načine njihovog kodiranja**

Kategoričke veličine mogu biti:

* **Nominalne** – ne postoji odnos između vrijednosti (npr. boja očiju, spol, ...)
  + Kodiranje: 1-od-K kodiranje (ako veličina ima K mogućih vrijednosti, tada se veličina kodira s K lažnih binarnih veličina (jedna lažna varijabla je 1, ostale su 0)).
* **Ordinalne** – postoji odnos između vrijednosti te je ih je **moguće sortirati** (ocjena, dobna skupina)
  + Kodiranje: Ručno se izvrši mapiranje znakovnih nizova u cjelobrojne vrijednosti
  + Primjer: Low → 0, Medium → 1, High → 2

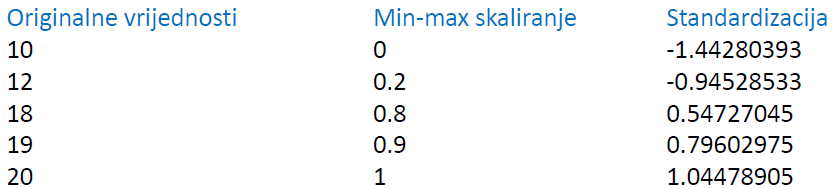
1. **Objasnite načine skaliranja numeričkih varijabli**

Prije procesa učenja skaliramo numeričke veličine tako da sve budu na istoj skali.

Načini skaliranja numeričkih varijabli:

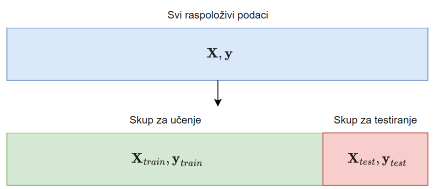
* **Min-max**: transformira ulazne vrijednosti na određeni interval (većinom [0, 1])
* **Standardizacija**: skaliranje gdje podaci za svaku ulaznu veličinu imaju srednju vrijednost 0 i varijancu 1

Primjer:



1. **Objasnite podjelu podatkovnog skupa na dva skupa. Koja je uloga svakog skupa podataka?**

Podatke dijelimo na dva skupa:

* Skup podataka za učenje – odvija se učenje modela i prilagođavanje parametara
* Skup podataka za testiranje – za evaluaciju nakon učenja. Cilj je procijeniti sposobnost modela da primjenjuje naučeno znanje na nove primjere.

**Bitno**:  
Skupovi podataka za učenje i testiranje moraju biti  
međusobno neovisni kako bi se dobila objektivna procjena  
performansi modela.

1. **Objasnite nadzirano učenje i princip označavanja dostupnog skupa podataka. Koja dva glavna tipa problema postoje u okviru nadziranog učenja?**

**Nadzirano učenje (eng. *supervised learning*)** je grana strojnog učenja koja se bavi razvojem algoritama i tehnika za obuku modela na temelju dostupnih podataka. Osnovna ideja je da model uči kako povezati ulazne podatke s odgovarajućim izlaznim podacima.

**Princip označavanja dostupnog skupa podataka** je postupak dodjeljivanja očekivanih izlaznih vrijednosti ulaznim vrijednostima. Npr. ako želimo trenirati model za prepoznavanje slika mačaka i pasa, označavanje podataka uključuje dodjeljivanje oznaka “mačka” ili “pas” svakoj slici u skupu podataka.

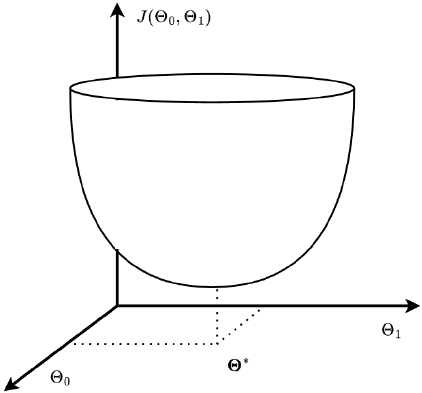
**Dva glavna tipa problema**:

* Klasifikacija
  + Uključuje dodjeljivanje ulaznih podataka u određene diskretne kategorije.
  + Primjer: spam pošta (binarna klasifikacija), prepoznavanje vrsta cvijeća, ...
* Regresija
  + Uključuje predviđanje kontinuiranih vrijednosti na temelju ulaznih podataka.
  + Primjer: predviđanje cijena nekretnina temeljeno na njezinim karakteristikama,  
    vremenske prognoze temeljeno na povijesnim podacima, ...

1. **Objasnite jednostavnu linearnu regresiju. Objasnite kriterijsku funkciju.**

**Jednostavna linearna regresija** modelira odnos između jedne ulazne veličine i kontinuirane izlazne veličine. Cilj je pronaći najbolju linearnu funkciju koja najbolje približava stvarne vrijednosti zavisne varijable na temelju nezavisnih (ulaznih) varijabli.

**Kriterijska funkcija** brojčano iskazuje koliko je dobar model s određenim parametrima na danom skupu podataka za učenje. Koristi se za procjenu koliko dobro linearna regresija približava stvarne vrijednosti zavisne varijable.



Izgled kriterijske funkcije

Minimum kriterijske funkcije – optimalni parametri modela

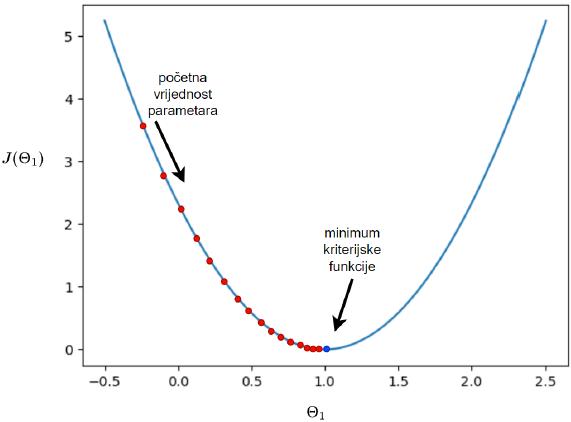
1. **Objasnite načine procjene parametara linearnog regresijskog modela. Navedite prednosti i nedostatke pojedinog pristupa.**

Rješenje u zatvorenoj formi (eng. *closed form solution*)

* Rješenje se dobija izravno, uvrštavanjem brojeva u matematički izraz
* Prednosti:
  + Brz i efikasan pristup
* Nedostaci:
  + Nepraktično za velike skupove podataka

Numerički iterativni postupak minimizacije

* Gradijentni spust
  + Ideja je krenuti od nekih početnih vrijednosti te ih podešavati dok ne dobijemo optimalne paremtre
  + Prednosti:
    - Fleksibilan – može se primijeniti na širok raspon problema optimizacije
    - Može se primijeniti na velike skupove podataka
  + Nedostaci:
    - Može zahtijevati velik broj iteracija
  + Graf:

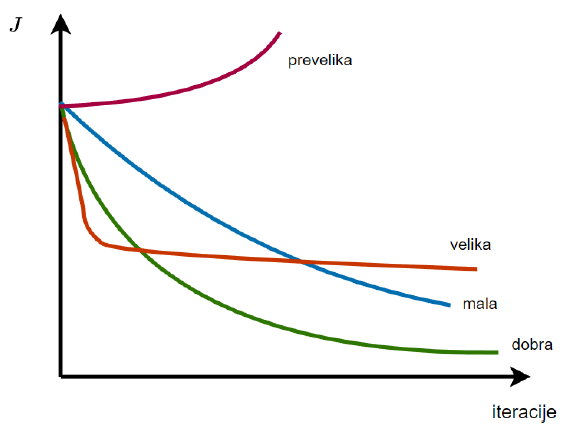


1. **Što je duljina koraka? Kako utječe na minimizaciju kriterijske funkcije? Objasnite na primjeru modela s jednim parametrom.**

**Duljina koraka** (eng. *learning rate*) utječe na proces optimizacije u algoritmima poput gradijentnog spusta. Određuje koliko “velik” korak algoritam treba napraviti u svakom koraku optimizacije pri ažuriranju parametara modela.

**Utječe na minimizaciju kriterijske funkcije** na način:

* Ako je duljina koraka prevelika, algoritam može preskočiti optimalno rješenje
* Ako je duljina koraka premala, algoritam može imati sporo konvergiranje i trebat će više iteracija



**Primjer modela s jednim parametrom**:

Model koji prevdiđa prodaju sladoleda na temelju temperature.

Pretpostavimo da imamo skup podataka o temperaturi (nezavisna (ulazna) vrijabla x) i broju sladoleda (zavisna varijabla y) za nekoliko dana i cilj nam je izraditi linearni regresijski model koji predviđa broj prodanih sladoleda na temelju temperature.

U ovom slučaju, temperatura je naš jedini parametar u modelu. Linearni regresijski model bi izgledao ovako:

y = ϴ0 + ϴ1 \* x

gdje je ϴ0 očekivani broj pdoranih sladoleda kada je temperatura 0, a ϴ1 mjeri koliko se broj prodanih sladoleda mijenja za jedinicu promjene temperature.

Ako model procjeni da je b1 = 3, to znači da očekujemo porast prodanih sladoleda za 3 komada po jednom stupnju temperature.

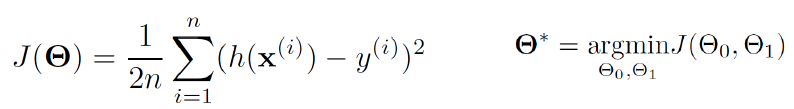
1. **Objasnite višedimenzionalnu linearnu regresiju.**

Ima više ulaznih varijabli. Primjer je određivanje cijene nekretnine gdje bi ulazne varijable bile godina izgradnje, površina, lokacija, broj soba itd...

Postupak procjene može se provesti primjenom metode gradijentnog spusta.

y = ϴ0 + ϴ1x1 + ϴ2x2 + ... + ϴnxn

1. **Objasnite načine procjene parametara višedimenzionalnog regresijskog modela.**

Provodi se na isti način kao i kod linearne regresije s jednom ulaznom veličinom – **minimizacijom kriterijske funkcije**.

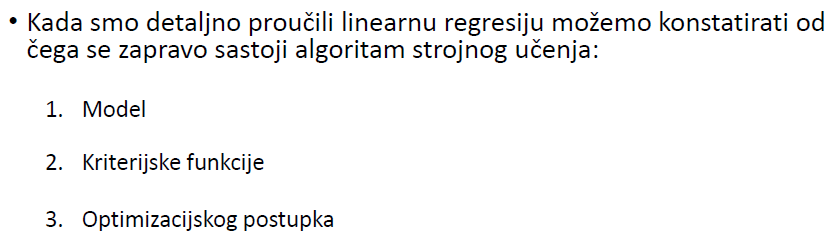
Formula:

1. **Objasnite razliku između batch metode gradijentnog spusta i stohastičke metode gradijentnog spusta. Napišite algoritam podešavanja parametara linearnog regresijskog modela stohastičkom gradijentnom metodom. Što je mini-batch stohastička metoda gradijentnog spusta?**

Razlika između batch metode gradijentnog spusta i stohastičke metode gradijentnog spusta leži u načinu kako se ažuriraju parametri modela i koriste podaci za izračunavanje gradijenta.

U batch metodi koristimo cijeli skup podataka (batch) kako bismo izračunali gradijent. Nakon izračuna gradijenta, parametri se ažuriraju koristeći formulu ažuriranja s dobivenim gradijentom.

U stohastičkoj metodi koristimo samo jedan primjer iz skupa podataka kako bismo izračunali gradijent. To se ponavlja za svaki primjer u skupu podataka ili za određeni broj iteracija.



**Mini-batch stohastička metoda gradijentnog spusta** – hibrid između batch i stohastičke metode. Umjesto da koristimo jedan ili sve podatke, koristimo podskup (mini-batch) određene veličine za izračunavanje gradijenta i ažuriranje parametara.

**Epoha** – jedan prolazak kroz sve podatke za učenje

1. **Navedite metrike za vrednovanje regresijskih modela**

Srednja kvadratna pogreška – MSE

Srednja apsolutna pogreška – MAE

Srednja apsolutna postotna pogreška – MAPE

Maksimalna pogreška – MaxError

R2 ili koeficijent determinacije

1. **Objasnite polinomsku regresiju. Kako se procjenjuju parametri ovakvog modela? Objasnite podusklađivanje i pretjerano usklađivanje na podatke na primjeru polinomske regresije.**

**Polinomska regresija**:

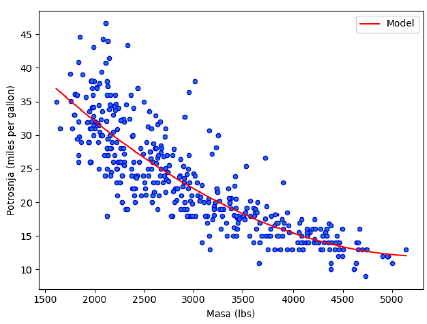
To je regresijska analiza u kojoj je veza između nezavisne varijable *Y* i zavisne varijable *X* opisana polinomom n-tog stupnja.

Promatramo jednu zavisnu i nezavisnu varijablu, ali njihov odnos ne mora biti određen pravcem nego i krivuljama višeg reda.

Primjer:

Ulazna veličina: masa vozila

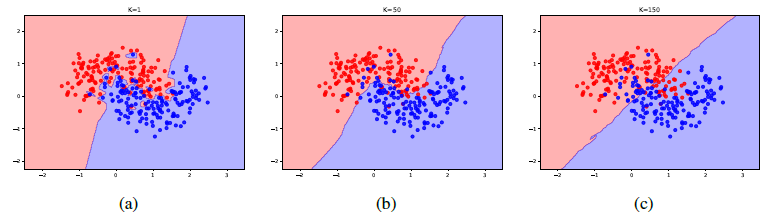
Izlazna veličina: potrošnja

¸

**Procjena parametara**:

Pitanje 12!

**Podusklađivanje**:

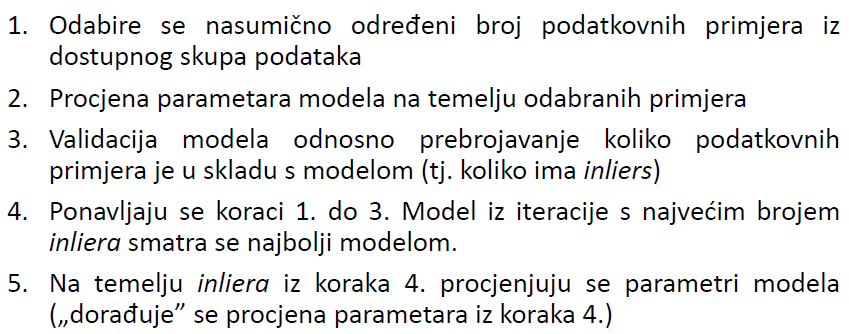


a) Overfitting

b) Optimalno

c) Underfitting

1. **Na primjeru određivanja parametara pravca objasnite princip RANSAC algoritma. Koje su prednosti i nedostaci RANSAC algoritma?**



Što je više iteracija, to će algoritam dati bolji rezultat.

**Prednosti**:

Robustan za prisustvo *outliera*, jednostavan i efikasan

**Nedostaci**:

Potrebno je odrediti parametra (broj iteracija), može biti dugotrajan kod većih skupova podataka

1. **Navedite tri primjera binarne klasifikacije i tri primjera višeklasne klasifikacije. Što su moguće ulazne veličine u model u svakom navedenom primjeru?**

**Primjeri binarne klasifikacije**:

Prepoznavanje spam poruka – Ulazne veličine: sadržaj poruke, duljina poruke, ...

Detekcija raka – Ulazne veličine: rezultati, ultrazvuk, ...

Prepoznavanje prijevare u kreditnim karticama – Ulazne veličine: iznos transakcije, vrsta trgovine, ...

**Primjeri višeklasne klasifikacije**:

Klasifikacija vrste cvijeta – Ulazne veličine: karakteristike cvijeta, širina latica, ...

Prepoznavanje rukom pisanih brojeva – Ulazne veličine: slika brojeva

Klasifikacija prometa na cestama – Ulazne veličine: kamere i senzori na cestama

1. **Objasnite logističku regresiju: model, kriterijska funkcija, optimizacija parametara**

**Logistička regresija** je model koji se koristi za binarnu klasifikaciju, tj. predviđanje dvije moguće klase ili kategorije na temelju ulaznih podataka. Koristi sigmoidnu funkciju koja stavlja parametre u intervalu [0, 1] pa se to interpretira kao vjerojatnost pripadnosti određenoj klasi. Prag odluke je 0.5

Parametri **modela** se procjenjuju optimizacijom kriterijske funkcije. Kriterijska funkcija je konveksna pa ne postoje lokalni minimumi. Cilj je minimizirati kriterijsku funkciju kako bi pronašli optimalne parametre modela.

**Optimizacija parametara** se vrši primjenom algoritama poput gradijentnog spusta. – pitanje 12

1. Objasnite OvR i OvO pristup na jednostavnom (2D) problemu s tri klase i dvije ulazne veličine. Koje su prednosti i nedostatci pojedinog pristupa?

OvR – One-vs-Rest

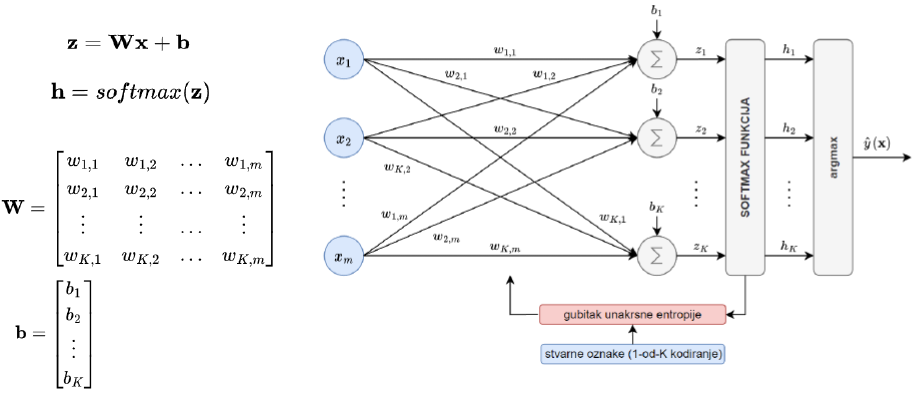
OvO – One-vs-One

Ako u binarnoj klasifikaciji imamo više klasa, možemo ih uspoređivati i dalje 1 na 1

1. **Kako izgleda model multinomijalne logističke regresije? Kako se kodira izlazna veličina za potrebe učenja ovakvog modela (objasnite na primjeru klasifikacije rukom pisanih znamenki). Koliko parametara ima model u ovom slučaju?**

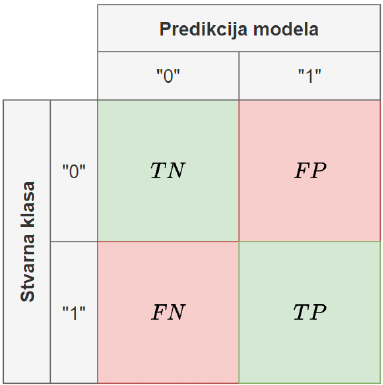
**-**

1. **Skicirajte softmax regresiju u formi potpuno povezanog sloja sa softmax aktivacijskom funkcijom. Napišite izraze za izračunavanje izlaznih vrijednosti.**



1. **Objasnite matricu zabune. Što predstavlja svaki njen element?**

**Matrica zabune** – koristi se za evaluaciju performansi klasifikacijskog modela. Redci su stvarne klase, a stupci predviđene klase



TN – broj primjera koji su TOČNO predviđeni kao negativni

TP – broj primjera koji su TOČNO predviđeni kao pozitivni

FN – broj primjera koji su NETOČNO predviđeni kao negativni

FP – broj primjera koji su NETOČNO predviđeni kao pozitivni

1. **Objasnite točnost, preciznost, odziv i F1 mjeru. Što je preciznost-odziv krivulja i ROC krivulja? Kako ih dobivamo?**

**Točnost** – udio točno klasificiranih primjera u skupu svih primjera. Izražava se u postotcima.

**Preciznost** – udio TP u skupu svih primjera koji su predviđeni kao POZITIVNI (TP + FP)

**Odziv** – udio TP u skupu svih pozitivnih primjera (TP + FN)

**F1 mjera** – objedinjuje preciznost i odziv u jedan broj. Uvijek će biti bliža manjoj vrijednosti.

Preciznost i odziv su obrnuto proporcionalne vrijednosti.

1. **Kako se izračunavaju metrike za evaluaciju u skučaju višeklasne klasifikacije? Pokažite na vlastitom primjeru s tri klase.**

U slučaju višeklasne klasifikacije, metrike za evaluaciju se računaju na temelju matrice zabune. Ako imamo problem klasifikacije cvjetova Iris na 3 klase: Iris-setosa, Iris-versicolor i Iris-virginica.

![Text

Description automatically generated](data:image/jpeg;base64,/9j/4AAQSkZJRgABAQEAeAB4AAD/4RDcRXhpZgAATU0AKgAAAAgABAE7AAIAAAAGAAAISodpAAQAAAABAAAIUJydAAEAAAAMAAAQyOocAAcAAAgMAAAAPgAAAAAc6gAAAAgAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAEpvc2lwAAAFkAMAAgAAABQAABCekAQAAgAAABQAABCykpEAAgAAAAM0OAAAkpIAAgAAAAM0OAAA6hwABwAACAwAAAiSAAAAABzqAAAACAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAMjAyMzowNTowMiAwOToxOTo0MwAyMDIzOjA1OjAyIDA5OjE5OjQzAAAASgBvAHMAaQBwAAAA/+ELGGh0dHA6Ly9ucy5hZG9iZS5jb20veGFwLzEuMC8APD94cGFja2V0IGJlZ2luPSfvu78nIGlkPSdXNU0wTXBDZWhpSHpyZVN6TlRjemtjOWQnPz4NCjx4OnhtcG1ldGEgeG1sbnM6eD0iYWRvYmU6bnM6bWV0YS8iPjxyZGY6UkRGIHhtbG5zOnJkZj0iaHR0cDovL3d3dy53My5vcmcvMTk5OS8wMi8yMi1yZGYtc3ludGF4LW5zIyI+PHJkZjpEZXNjcmlwdGlvbiByZGY6YWJvdXQ9InV1aWQ6ZmFmNWJkZDUtYmEzZC0xMWRhLWFkMzEtZDMzZDc1MTgyZjFiIiB4bWxuczpkYz0iaHR0cDovL3B1cmwub3JnL2RjL2VsZW1lbnRzLzEuMS8iLz48cmRmOkRlc2NyaXB0aW9uIHJkZjphYm91dD0idXVpZDpmYWY1YmRkNS1iYTNkLTExZGEtYWQzMS1kMzNkNzUxODJmMWIiIHhtbG5zOnhtcD0iaHR0cDovL25zLmFkb2JlLmNvbS94YXAvMS4wLyI+PHhtcDpDcmVhdGVEYXRlPjIwMjMtMDUtMDJUMDk6MTk6NDMuNDgyPC94bXA6Q3JlYXRlRGF0ZT48L3JkZjpEZXNjcmlwdGlvbj48cmRmOkRlc2NyaXB0aW9uIHJkZjphYm91dD0idXVpZDpmYWY1YmRkNS1iYTNkLTExZGEtYWQzMS1kMzNkNzUxODJmMWIiIHhtbG5zOmRjPSJodHRwOi8vcHVybC5vcmcvZGMvZWxlbWVudHMvMS4xLyI+PGRjOmNyZWF0b3I+PHJkZjpTZXEgeG1sbnM6cmRmPSJodHRwOi8vd3d3LnczLm9yZy8xOTk5LzAyLzIyLXJkZi1zeW50YXgtbnMjIj48cmRmOmxpPkpvc2lwPC9yZGY6bGk+PC9yZGY6U2VxPg0KCQkJPC9kYzpjcmVhdG9yPjwvcmRmOkRlc2NyaXB0aW9uPjwvcmRmOlJERj48L3g6eG1wbWV0YT4NCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgPD94cGFja2V0IGVuZD0ndyc/Pv/bAEMABwUFBgUEBwYFBggHBwgKEQsKCQkKFQ8QDBEYFRoZGBUYFxseJyEbHSUdFxgiLiIlKCkrLCsaIC8zLyoyJyorKv/bAEMBBwgICgkKFAsLFCocGBwqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKv/AABEIAMQDLwMBIgACEQEDEQH/xAAfAAABBQEBAQEBAQAAAAAAAAAAAQIDBAUGBwgJCgv/xAC1EAACAQMDAgQDBQUEBAAAAX0BAgMABBEFEiExQQYTUWEHInEUMoGRoQgjQrHBFVLR8CQzYnKCCQoWFxgZGiUmJygpKjQ1Njc4OTpDREVGR0hJSlNUVVZXWFlaY2RlZmdoaWpzdHV2d3h5eoOEhYaHiImKkpOUlZaXmJmaoqOkpaanqKmqsrO0tba3uLm6wsPExcbHyMnK0tPU1dbX2Nna4eLj5OXm5+jp6vHy8/T19vf4+fr/xAAfAQADAQEBAQEBAQEBAAAAAAAAAQIDBAUGBwgJCgv/xAC1EQACAQIEBAMEBwUEBAABAncAAQIDEQQFITEGEkFRB2FxEyIygQgUQpGhscEJIzNS8BVictEKFiQ04SXxFxgZGiYnKCkqNTY3ODk6Q0RFRkdISUpTVFVWV1hZWmNkZWZnaGlqc3R1dnd4eXqCg4SFhoeIiYqSk5SVlpeYmZqio6Slpqeoqaqys7S1tre4ubrCw8TFxsfIycrS09TV1tfY2dri4+Tl5ufo6ery8/T19vf4+fr/2gAMAwEAAhEDEQA/APKqKKK3EFFFFABRRVmxsZNQuGhhZVZYZZiXJA2xxtI344U496AK1FFWUsZJNLnvwy+VBNHCwJO4s6uRj2/dtn6igCtRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRU1nB9qvoLfdt82RUz6ZOK9Gum8LaNJdaRY3q6Pc2d3JHNLe6cl+bkAAAglPl5B4GOp69gDzOivRNWg8M+IdKvX0zdJqGm6es0l9DCLeKdg+CDEFGDhuox264587oAKKv6TpMmrTzKk9vbRW8LTzz3LlUjQEDJwCSSWVQACSWHFMv9Oex8t1miuraXPlXMG7Y5GNw+YBgRkZBAOCD0YEgFOiiigAooooAKKKKACirOo2MmmapdWE7K0trM8LlCSpZWIOM9uKrUAFeh/BLU7mz+JdpaQSEQX0csc6Z4YLGzqceoK/qfWvPK7j4Of8AJWdG/wC2/wD6IkpPYDh6KKKYHSQ6tf6X4JsDpl3LaPJqF1ukgbY5Ajt+N45A55GcEgE9BjR1m7mnvvGGnyP/AKDYb/slooCw25W8iQFEHyqdrMCQMnc2epzxVFAHpur2F/c+HdeX7Lql3Y2NvGLS5kbFoVWWMK1vCQxAESt+8Dtlclj89UJdQa8+KWqQavPcPp9u2pr5UGB5cflS7/LX7oYgZJ7tycnJrgaKAO7ubma/bTbwiKHw20ckdhZhh9ntbpY38tJQSV3+YQxeTh1bc2F3Kov9pf2In/Cdfavs/wDbNl5v27d5/k7Z9/X95sxnHbO/bzurhKKAO4s77UdOMtx4yW4ey2qlojAbXVnVZVtsYHlmBpRlCEGY+QfLNbuo2kekaP8AZXjcXHglUcTLMHEs10hdWC7Rwk/lkAnlQcgng+VUUAdxpOoXcd34L02Od0s7pUS5hThbhGvZlKSY++uMja2QMtx8xzw9FFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQBc0j/kN2P/XxH/6EK0vG/wDyPetf9fkn/oVYkUrwTJLEdrxsGU+hHIrqp/HMF7cPc3/hfRri5lO6WUpIC7dyfmoAj8If8gfxR/2Cj/6MSuWrprvxlv0y6stN0TTdMF2gjmlt0beyZB25LHA49K5mgDe0S1m1fRNS0mwTzL1pIryOLIBlWJZQ6oOrPiUMFHUK3cAE0/TP7M1iBNbWKCd45iltdDBjk8s+SZVYbVBk28PxgZYbCCcGigD0HRv7V+0n/hI/N/tjy7j7F/an3/K+yXG/d5v/ACz3eXjf8ud+P46zXv8AUtMjt7zxVHdTajDeRNDDfMwmmttr+dG27J8tvkX5gV5faD89czY30mn3DTQqrM0MsJDgkbZI2jb8cMce9Qwusc6O8SzKrAmNyQrgHocEHB9iD70Ab3iG2j0bT7PSra4W4WVmvmmTCmRH+WEOuSQfLXzAG5AnIwOSdHQpdUh0+0bwyFMRtZ11PehaLzT5gHnjGNvllNpf92DuPGJK5nVdTm1fU5b25SKN5MARwxhEjVQFVFUdAFAA9hVOgDqbD+1f7Hsv+EW83y/Lb+0/K+55nmPj7Rnjy/L2ff8A3eN/fzKuXH2nd/xQPm/2b5k32rys7MedJs+0b/8Aln5Pl/6z5Mb+/mVxVFAHpX2lrTxFrUujWepXOrJrdy0w0ycJKIQ4KblMblo92/IA252785SuB1eRJtbvpY4IrZHuJGWGF1dIwWPyqy/KQOgI4I6VTooAK7j4Of8AJWdG/wC2/wD6Ikrh67j4Of8AJWdG/wC2/wD6IkpPYDzj+3NO/wCfj/xxv8KP7c07/n4/8cb/AArkaKjmYzrv7c07/n4/8cb/AAo/tzTv+fj/AMcb/CuRoo5mB139uad/z8f+ON/hR/bmnf8APx/443+FcjRRzMDrv7c07/n4/wDHG/wo/tzTv+fj/wAcb/CuRoo5mB139uad/wA/H/jjf4Uf25p3/Px/443+FcjRRzMDrv7c07/n4/8AHG/wo/tzTv8An4/8cb/CuRoo5mB139uad/z8f+ON/hR/bmnf8/H/AI43+FcjRRzMDrv7c07/AJ+P/HG/wo/tzTv+fj/xxv8ACuRoo5mB139uad/z8f8Ajjf4Uf25p3/Px/443+FcjRRzMDrv7c07/n4/8cb/AAo/tzTv+fj/AMcb/CuRoo5mB139uad/z8f+ON/hR/bmnf8APx/443+FcjRRzMDrv7c07/n4/wDHG/wo/tzTv+fj/wAcb/CuRoo5mB139uad/wA/H/jjf4Uf25p3/Px/443+FcjRRzMDrv7c07/n4/8AHG/wo/tzTv8An4/8cb/CuRoo5mB139uad/z8f+ON/hR/bmnf8/H/AI43+FcjRRzMDrv7c07/AJ+P/HG/wo/tzTv+fj/xxv8ACuRoo5mB139uad/z8f8Ajjf4Uf25p3/Px/443+FcjRRzMDrv7c07/n4/8cb/AAo/tzTv+fj/AMcb/CuRoo5mB139uad/z8f+ON/hR/bmnf8APx/443+FcjRRzMDrv7c07/n4/wDHG/wo/tzTv+fj/wAcb/CuRoo5mB139uad/wA/H/jjf4Uf25p3/Px/443+FcjRRzMDrv7c07/n4/8AHG/wo/tzTv8An4/8cb/CuRoo5mB139uad/z8f+ON/hR/bmnf8/H/AI43+FcjRRzMDrv7c07/AJ+P/HG/wo/tzTv+fj/xxv8ACuRoo5mB139uad/z8f8Ajjf4Uf25p3/Px/443+FcjRRzMDrv7c07/n4/8cb/AAo/tzTv+fj/AMcb/CuRoo5mB139uad/z8f+ON/hR/bmnf8APx/443+FcjRRzMDrv7c07/n4/wDHG/wo/tzTv+fj/wAcb/CuRoo5mB139uad/wA/H/jjf4Uf25p3/Px/443+FcjRRzMDrv7c07/n4/8AHG/wo/tzTv8An4/8cb/CuRoo5mB139uad/z8f+ON/hR/bmnf8/H/AI43+FcjRRzMDrv7c07/AJ+P/HG/wo/tzTv+fj/xxv8ACuRoo5mB139uad/z8f8Ajjf4Uf25p3/Px/443+FcjRRzMDrv7c07/n4/8cb/AAo/tzTv+fj/AMcb/CuRoo5mB139uad/z8f+ON/hR/bmnf8APx/443+FcjRRzMDrv7c07/n4/wDHG/wo/tzTv+fj/wAcb/CuRoo5mB139uad/wA/H/jjf4Uf25p3/Px/443+FcjRRzMDrv7c07/n4/8AHG/wo/tzTv8An4/8cb/CuRoo5mB139uad/z8f+ON/hR/bmnf8/H/AI43+FcjRRzMDrv7c07/AJ+P/HG/wo/tzTv+fj/xxv8ACuRoo5mB139uad/z8f8Ajjf4Uf25p3/Px/443+FcjRRzMDrv7c07/n4/8cb/AAo/tzTv+fj/AMcb/CuRoo5mB139uad/z8f+ON/hR/bmnf8APx/443+FcjRRzMDrv7c07/n4/wDHG/wo/tzTv+fj/wAcb/CuRoo5mB139uad/wA/H/jjf4Uf25p3/Px/443+FcjRRzMDrv7c07/n4/8AHG/wo/tzTv8An4/8cb/CuRoo5mB139uad/z8f+ON/hR/bmnf8/H/AI43+FcjRRzMDrv7c07/AJ+P/HG/wo/tzTv+fj/xxv8ACuRoo5mB139uad/z8f8Ajjf4Uf25p3/Px/443+FcjRRzMDrv7c07/n4/8cb/AAo/tzTv+fj/AMcb/CuRoo5mB139uad/z8f+ON/hXd/BXVbO5+MGiRQzbnbz8DaR/wAu8h9K8Wr0b4Af8ly8P/8Abz/6TS0uZgec0UUVIBRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAV6N8AP+S5eH/+3n/0mlrzmvRvgB/yXLw//wBvP/pNLQB5zRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAV6l8LfCnw68b3+l+HtVl8UQ+IbzzvMe2a3W0GwO4wWBf7ijt972ry2vbf2fPh94n/4Tvw94y/sz/iQf6T/AKZ9oi/55Sxfc3b/AL/HT36c0Aef69pvhXV7/T7H4W2fii+u5fM8+31GKKSR8AFfKWEEnADls9gPeqS+AfGLXUlqvhPXDcRosjxDTZt6qxYKxG3IBKMAe+0+hrfT4X/Enwv4s0e0ttPuNN1vUfP/ALOa21GJJG8tMy4kST5cI3cjIOOa9X+KWlfF6HQNAutJvNUt7HR/DkUmrS2+qJE4uUUmdnIkDykBFOfmHXHJOQDxLw94TnutG1DV9X0DxJNpi6fPJZXum2RaETp0aV2G3yhtfcQcjHsa5/T9NvtXv47HSrO4vruXPl29tE0kj4BJwqgk4AJ+gr3L4MeGPHmo+CdfYfbJ/DupeHL+00qB79TC1yz7QFiL/ISwk+YgDk8880fhnpnjXwRq+t+HV0jT7u7u5reG80ZfECWOoSbInmV4JI5QQoDqWIzkHaOj7QDx/VdF1XQbpbXXNMvNNuHQSLFeW7QuykkBgGAOMgjPsapV7l8bpZ18F6bba8mqaFq/9oGSLw9da8dURoPLIN2XO5lJbEaqzYG1yq5ZzXnV/wCP/t/g1PD3/CJ+F7bbDFD/AGnb6bsvT5ZU7zLu+8235jjnc3rQBiaR4c1zxB539g6NqGp+Rt837FavN5e7ON20HGcHGfQ1W/s2+/tX+zPsdx/aHnfZ/snlN5vm7tvl7MZ3buMYznivf/hneeL5fAmm6Tp/h238S2VjCZoRoni86fPaedLI5S6EcvzMcAqCBtGR97cBe0rVNKi/aEaObWZG1qbw4bK3TUb1bj+zNTYgC0juDEy5RcrvKuWZnDb3ZlIB8/33hPxHpd1aWup6BqlncXz+XaxXFlJG9w2QNqAgFjllGBnqPWup1T4OeKrDwnoOrW2j6xeXep/aPtOnx6VL5lj5bhV34yfnB3DIXgd+tegeN7vxb4Y+F9/ZeJLDULaaa9t7nTrnW/FVte3ttLG4LS2oSNXOBtVsNlfMBAwXNSxp8SvHPwc8GXXgXWdUvb62e/j1aWDWRFKGMwMSylpFLEJyAc4UjoCMgHhGlaLquvXTWuh6ZealcIhkaKzt2mdVBALEKCcZIGfcVHqGm32kX8ljqtncWN3FjzLe5iaORMgEZVgCMgg/Q16/8MY9b1f4M61onw4v47Xxb/a0d1dRQzLb3VxYhFVQkpxwspJOGGAWB++A3Qa9qeu6Br3gL7Xqnh+fx7oGmXsmpyateJGoR1xbwyzFk3ybHIC7idzFydrFyAeG6r4T8R6Darda5oGqabbu4jWW8spIUZiCQoLADOATj2NGleE/EevWrXWh6BqmpW6OY2ls7KSZFYAEqSoIzgg49xX0T4vk1W18BeID4u0jXPBdrJp8kcFzJ4xbUvt05x5dp5TlyA55ZlAO1GBYKWzL4JsfFdrpHw+bSbu3vfDLQ2813fyagtlb2rvL81skNu8Zlk3AITN5weRuiZcEA+W6K6n4nW09r8WPFUd1DJC7atcyBZEKkq8rMrYPYqwIPcEHvXLUAFet+CPCnws8VaVeefL4wj1DSdFfU9R2NaiJvKVfNEWQSfmb5Q2OOpFeSV7t8Ivhb4xttK8R6lPo+2013wldQadJ9qhPnvMsbRDAfK5HdgAO+KAPMtV0XSte19bX4V6Z4k1K3S1Ek0V5brNcKwYhmAhBGzBQZPcn2qjJ4K8VRQ3k0vhrWEisc/a3awlC2+EDnedvy4RlbnHBB6Gup0r4b/FDQfGjaLodreabrz6ebtks9TihdrYyBCTIsgGN4Hy5zwDjvXon7QFj8TY9e1zWILvUIPBkUMMAEGoJHEyOqI4aJXDPmR2BLKSQf7oGADyCHwnPa+CdT1fW9A8SQuyW8mmXqWRWxKu/ztK7DoVZdhU8k89aw9K0XVdeumtdD0y81K4RDI0VnbtM6qCAWIUE4yQM+4r3fwn4I+I2s/AfxNp1/FeXqara6Y+gwXGoo6eSsu9tgaTEQ2bDg7egGOMVW+FqeMtD0a88L2uh2etBNQuZ7jTdP8TjT9Rtpo/LhZ5GjlB8rIKhcHJG7ONhYA8N1DTb7SL+Sx1WzuLG7ix5lvcxNHImQCMqwBGQQfoarV638d5t03h+C+n1C11iGGb7ZoV3qv8AaK6YpcGPE2Ms0gy53MzAFBwoQVyPivx//wAJVpUdj/wifhfRvLmE32jR9N+zythWGwtuOV+bOPUD0oA5u902+077P/aFncWn2mFbiDz4mTzYm+7IuR8ynBwRwcVZ/wCEc1z+3v7E/sbUP7W/58Psr+f93f8A6vG77vzdOnPSvVvh5q+jax4R8Pza/LJAfh5qE2ossE0UZmtHDTKQJGzI/wBqjiiwu0YmQdW3C18HddfxBqHxAeB44PFuvoLixjtLhbSaQGSSS4it55EkEZIK4B5+UHI2l1APH73w5rmnarb6ZqGjahaahc7fItJ7V0ll3NtXahGWywIGByRirzeAfGK3Udq3hPXBcSI0iRHTZt7KpUMwG3JALqCe24eor1b4h/8ACWr4b8M+GtQ024g8QjWlk0y41DxHbXmpxb+EAZERljaTo5Y4aLBIASup8QeGfjPN8K/COl6TPqg1eF7t9XdNWjWYnzf3G6YyAsNjNwGI6ZGQMAHztY+E/EeqXV3a6ZoGqXlxYv5d1Fb2Ukj27ZI2uACVOVYYOOh9KrQaLqt1rJ0i10y8m1NXeM2UduzTBkzuXYBuyNpyMcYPpX0B8PNV8baj4PsrKLRI/FkUCNdCXSfGLWN1A1xLJIwu9kuXdj8wyBtGQfmLAcj48t9b1r406bZ+CNU1S48S/wBnpHLAdVV5NNnVJDLaJc/KCI0BBbcSzF9zFi1AHAeLtFg0G6061i0zXNNuH0+KS8i1q3ELtOSwdohgHysjCk88HNVYPCfiO60Y6va6Bqk2mKjyG9jspGhCpnc28DbgbTk54wfSvXPjr4G8XNpnh/xHqdlJLb6X4csrTVLuW6jd1ud7K4b5izndIuWGQc5z1rpPGNh8Rta8UWfiP4Y+II7bwOtrbvYXFtqKWtlYwxqA/nQkrwjK5YFGwvykfLsAB87aVouq69dNa6Hpl5qVwiGRorO3aZ1UEAsQoJxkgZ9xUo03+yPEkFj4ss9QsYopo/t1uIvLuUiOC21XAwxQ5G7jkdq9p8K/8JJrngLxhpHgnXbObxw/iNrvUJ9PuEgl1C04XfDNhQU84luCoCkjjeFbI+LVlql/beAPC+r3NvqfjyOF7bUAkitKPNkT7LFLLwCwUnqx5JYkh9zAHNfEPwp4V0jwn4W8Q+C5dYa0137XuTVmiMieS6J0jAAyS3c8Y6Vy2leE/EevWrXWh6BqmpW6OY2ls7KSZFYAEqSoIzgg49xXsHjr4W+MYvgl4Ogk0fEvh2HU59UX7VD/AKOjSiRT9/5sopOFyeMda14dP8eeJvC/gGf4O6xHBpFjp8Ntex2l0tulrfKwM7XEZAMgJKtjD5wx24fLgHgmkeHNc8Qed/YOjahqfkbfN+xWrzeXuzjdtBxnBxn0NSXPhPxHZ6ZLqV5oGqQWMLmOW6lspFijYP5ZVnIwCHG0gn73HWvbrGHW/E37SWrar8L7mR/DsOrWEmryaddrbwzKMeYzLuXzgWWc5Abdljzuyef+N9j8TbLVdZ1DxLd6gnhi+1NoLSL+0EMDoGZoR5CPx8kYbJUHIyfmNAHjdFFFAHSeEf8AhCP9M/4Tz/hIP4Psn9i+R/tb9/m/8Bxj3z2rrvil4U+HXgi/1Tw9pUviibxDZ+T5b3LW7Wh3hHOSoD/cY9vve1cj4R+H3ifx39s/4RTTPt/2LZ9o/wBIii2b92377LnOxumelek/tB/D7xP/AMJ34h8Zf2Z/xIP9G/0z7RF/zyii+5u3/f46e/TmgDzKXwD4xgkhjn8J65G9w5jhV9NmBkbaW2qCvJ2qxwOyk9qk0fwVrlz4kFjfeGvEE0VnND/advY2DtcwRPz90r8rFMld3B+le9xeGfjPD8E5tLSfVD4km10Ozvq0bTCy8kfdmMh2jzVHCsD14wTnz/4X6V8StR+LEskF5qkr6bq1rH4jY6oAWWOUqVkJk/fALHIMDdwMDqMgHlut20Fnr+oWtnDeQW8N1JHFFfoFuI1DEBZQOA4AwwHfNWdV8J+I9BtVutc0DVNNt3cRrLeWUkKMxBIUFgBnAJx7GvV9V8O+MfAfx5vPFC2+j2DXV7qd7psus6hDFBcJ8wx/rAQxEyBVJBywJwqsR2Xi+TVbXwF4gPi7SNc8F2smnyRwXMnjFtS+3TnHl2nlOXIDnlmUA7UYFgpbIB8wVZg02+ubC6vrazuJrSz2fabiOJmjg3nC72AwuTwM9TXSeFPH/wDwiulSWP8AwifhfWfMmM32jWNN+0SrlVGwNuGF+XOPUn1qz8J9esdG8ZSWOtSeTpOv2U2jX1wGVWginAHmBmIVdrBSWbIC7uCcUAcl/Zt9/ZX9p/Y7j+z/ADvs/wBr8pvK83bu8vfjG7bzjOcc10Gq+E59B8Frda5oHiTTdXfUBGst5ZGGyaAxkhQWAbzdwJx02g961vidfR2Fh4a8D24z/wAIxZMt4zOkjC9nIluIw6MVKo2EHAIKuCTgGvTtV+G/xQ174GrouuWt5qWvJ4jF2qXmpxTOtsLYoCJGkIxvJ+XOeScd6APCIPCfiO60Y6va6Bqk2mKjyG9jspGhCpnc28DbgbTk54wfSi+8J+I9LurS11PQNUs7i+fy7WK4spI3uGyBtQEAscsowM9R617d8B9J+I0Ph3VNU06S8OkTaFeJoaPeo0IvfMGzbCzEKd6ycsoHXJweaPgy48V+E/ismm/FLUtQt9TuNFuRpN9fTLqS6W8n/Lzy7BFCwybjkDA+bCZYAHjeoeHNc0iGSbVdG1CxiimFvI9zavGqSlA4jJYDDFCGx1wc9K14fCc9r4J1PV9b0DxJC7JbyaZepZFbEq7/ADtK7DoVZdhU8k89a9N+IrXy/s52Vrqvi638WXdr4mEcl7bTNPHFm2kcRCZhmXAcNu7F9n8FXvCfgj4jaz8B/E2nX8V5epqtrpj6DBcaijp5Ky722BpMRDZsODt6AY4xQB4bp/hzXNXhjm0rRtQvopZjbxvbWryK8oQuYwVBywQFsdcDPSop9F1W11kaRdaZeQ6mzpGLKS3ZZiz42rsI3ZO4YGOcj1r1e21LxD4F/Zz1O0sby40jUU8ZvYXbW0oEi7bZSyCRTx88Y5U8gYyQTmt8UjrGr+KvhwdPuLibW7zwzpZgn8/bK9w7vtbzCRhi5B3E9ec0AcR4l8Of8I/oOi/bdG8QaZq0/n/bf7UtfJgk2sPL8jIDHCn589yMcVR0rwn4j161a60PQNU1K3RzG0tnZSTIrAAlSVBGcEHHuK9p+Jfw/wDH+o/CHwlNrNpcXmoaFDqU+sTXN/HLJEhkDoSxcl/3adFLEAY9qlvNM8eeKvAXgyX4O6lJ/ZFnpKWt7baVfrZPDfLgzmUFkLFiVI5P8TcB8sAeCafpt9q9/HY6VZ3F9dy58u3tomkkfAJOFUEnABP0FSarouq6DdLa65pl5ptw6CRYry3aF2UkgMAwBxkEZ9jXuXg2LxbqPjLx5fWfiTT/ABD4oSytUVNInto11NWCqStwVVkjjUKHEXlsW2/OrBTVL47Wusn4deBJ9bjja6he/hvHtrqW7igmMiYhaaR3YuNjqQXOGjcDheADw2iiigAooooAKKKKACiiigAooooAKKKKACvRvgB/yXLw/wD9vP8A6TS15zXo3wA/5Ll4f/7ef/SaWgDzmiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACtvwh4v1jwP4kg1vw/ceTcxfK6MMxzocbo3X+JTge4IBBBAIxKKAOt8Y/EjWPGdhY6bPbafpWk2G5oNL0m38i2VySTJsyct8x74GTgAsxPJUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFejfAD/kuXh//ALef/SaWvOa9G+AH/JcvD/8A28/+k0tAHnNFFFAHsnwN8QWOr+LNC8F6r4P8L31pL9o8y+udMWS7fCSSjMjEg4IC/d+6Me9cjfaz/wALN17S9M/s3wf4Q2eb/pdvB9ggbK7v3z5b+5heOrY710vwdm8CeFde0bxdr3jj7JqFt5/m6R/ZM77dyyRD98uQflYNwO+PeuavvC3gSDXtLtbL4i/a9PufN+2339hzp9i2rmP92TmTe3y8fdxk0Abl78BdY037PJqnizwfY2l3Cs1neXOreXFdg9fLJTLYGwk4xiRcE84reCvBF9ceDdZ8Q6ffeD9Q3aLdtPpl/M0t7aRIcNMkQX5JBtGxycfOPWtv4iD4deJNK0ufT/iHuu9C8PxabBa/2JcD7Y8KuVO84Ee8sBzkL1yarfCybwJoeg6tda944+w6hrmi3elS2P8AZM8n2TzGAEnmLkP8qBtox97GeKAOI8I+BdU8Y/bJrK40+w0+w2fbdR1K7WCC237hHuJ5+Zl2jAPJGcZzUvizwFe+E7W3vG1fQ9YsbhzEl3pGopOnmAZZNvD5UFSTt2jenOSBXU+EdU8PeDfE15BoPxM1jTd0yeVqtvoxeymiELE+dbs+9m3uVXKELt3fxZST4r+ONK8QaBZ6Yutx+MNXW6+0P4hOkLYPDAFZVtAu0Mw3M0hPAGR1OdoB5/P4T8R2ujDV7rQNUh0xkSQXsllIsJV8bW3kbcHcMHPOR610Hh74Waprmg2+s3uteH/D+n3m77FNrWpLB9r2MUk2KAx+Vhg5A+8MZzXPz+LPEd1ow0i61/VJtMVEjFlJeyNCFTG1dhO3A2jAxxgelenfD3xpY6HoMdnp/wAVtY8L7IU8+1vtCW+gMpZ2b7OAz+Wo3AHIUsfm77VAONtvhZ4kvfHsXhOwSzvbyVBOtxa3STW627crOzpnahUqw3AMQyYXLKDd1b4PaxplhZ6lBrfh/VdJuL1bKfVNNv8AzrawdioBnfaNi/OMtggcZwSoPXWXxT8K2/xcuLue23aTfaK2iajq0FjFG13K77pb9rcoVO5uqMGO3kgn93WR4v8AEWhWXgnUNH8P+No9YfVHhM0Gm+E7fTIWWJ92JmCq5OSGUrnGxgR8+QAa+t/BPS4/Anheay8V+D7PUJftf23UbjWWEF/iUCPySQQdi/K2AuCec9a8/wDB3w31jxnYX2pQXOn6TpNhtWfVNWuPItlckAR78HLfMO2BkZILKD1Npd+B/F3wr8LaJ4h8YSeG77w+96ro+ky3QnE0okDKYzgAAY55znjGCaXg7WvCmrfDK+8D+MNVuNAxqa6tZ6pHbNcxs/liJonjUbvu5IIIGTyRtw4Bman8I/Fen+LNO0C2trfU5tWhM+m3VlcKba9QJvYxyvtBwvUHB6HkMpMus/Cm90vRrnUrLxT4T1tLNPNuYtL1hHkij6byHC5G4quASxZ1ABzXQa34v8IWn/CL+H9B17xBHp/hiyu/K13TYRHPcXVzhjtRmjKRq2c5O4jKf9ND0Hiv4l6VL4R1S11fx5H8QnurV7ey0+Tw2tkLOdxgXfmsvBRS4AHJLjpyQAcR4f8Agtr+uaNpuo3Wq6HoQ1d9um2+sXhgmvB8oDRoFJYEsAO5yDjDKTwmpafdaRqt3puoReTd2czwTx7g2x0YqwyCQcEHkHFe2+FPFfgDSLbwlqNn4w1Dw6thDbrq2i6fp8iyX9wsh3yzXCnEkeTu2NuxGGVArNtrynx7d2Wo/EPX7/S7yO9s73UJrqCeNHUMsjlwMOqsCN2Dx1BxkYJAOfooooAKKKKACiiigAooooAKKKKACiiigAooooA0vD39h/29b/8ACV/2h/ZPzfaP7N2ef907du/5fvbc57Z712/xp03Q9O1Xwr/wjFn9k0+58M2dwm+JEll3NJiSXYMNIVC7j3IrkfCml6Fq+qyQeJ/Ef/CPWiwl0uvsL3W99ygJsQgjILHPT5cd69A+Kc3gTXNB0m60Hxx9u1DQ9FtNKisf7Jnj+1+WxBk8xsBPlcttOfu4zzQBt6T4k8FXfxB0nwH4R8D6Pr/hu48uyk1O4s3OoT+YuZZ/NKqybNxJ+XgRkqUGNvjfiOzsdO8VarY6Rc/a9Ptr2aG1uPMV/NiVyEfcvDZUA5HBzxXsCp8LtN8LyaN4Y+I8elPe2qwapqJ0C7lurwbmLoGyFiibKgoq5IQBmfmvFtSgtbbVbuDT7v7daRTOkF15Rj89AxCvsPK5GDg8jOKAK1FFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFaXhzSP+Eg8VaVo3n/Z/wC0b2G087Zv8vzHCbtuRnGc4yKzau6Jqs+g6/p+r2aRvcafdR3USyglGZGDAMAQcZHOCKAPYP7R8If8LH/4Vf8A8Ilp/wDYH23+x/7V+yj+1vPzs87z9+P+Pj5vu42cbcfLXkniPSP+Ef8AFWq6N5/2j+zr2a087Zs8zy3KbtuTjOM4ya9S/t34bf8ACZf8LM/tK4/tDzv7Q/4RH7JLu+2Y/wCfvdt2+d+9zj7vGz+CvKNb1WfXtf1DV7xI0uNQupLqVYgQis7FiFBJOMnjJNAFKiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACtLw5pH/AAkHirStG8/7P/aN7Daeds3+X5jhN23IzjOcZFZtXdE1WfQdf0/V7NI3uNPuo7qJZQSjMjBgGAIOMjnBFAHsH9o+EP8AhY//AAq//hEtP/sD7b/Y/wDav2Uf2t5+dnnefvx/x8fN93Gzjbj5a8k8R6R/wj/irVdG8/7R/Z17Naeds2eZ5blN23JxnGcZNepf278Nv+Ey/wCFmf2lcf2h539of8Ij9kl3fbMf8/e7bt8797nH3eNn8FeUa3qs+va/qGr3iRpcahdSXUqxAhFZ2LEKCScZPGSaAKVFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABXo3wA/5Ll4f/AO3n/wBJpa85r0b4Af8AJcvD/wD28/8ApNLQB5zRRRQAUVt6f4K8VavYR32leGtYvrSXPl3FtYSyRvgkHDKpBwQR9RVbV/DmueH/ACf7e0bUNM8/d5X221eHzNuM7dwGcZGceooAzaKKs2Wm32o/aP7Ps7i7+zQtcT+REz+VEv3pGwPlUZGSeBmgCtRRRQAUUUUAFFFFABRRRQAUUVZn02+trC1vrmzuIbS83/ZriSJljn2HDbGIw2DwcdDQBWooooAKKK1tV8J+I9BtVutc0DVNNt3cRrLeWUkKMxBIUFgBnAJx7GgDJooooAKKKKACiiigAooooAKKKKACiiigAooooAKK0v8AhHNc/sH+2/7G1D+yf+f/AOyv5H3tn+sxt+98vXrx1rNoAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKs3um32nfZ/7Qs7i0+0wrcQefEyebE33ZFyPmU4OCODigCtRWl/wjmuf2D/bf9jah/ZP/AD//AGV/I+9s/wBZjb975evXjrWbQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFa0HhPxHdaMdXtdA1SbTFR5Dex2UjQhUzubeBtwNpyc8YPpWTQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFejfAD/kuXh//t5/9Jpa85r0b4Af8ly8P/8Abz/6TS0Aec0UUUAet/s+eI9c/wCFseHtE/tnUP7J/wBJ/wBA+1P5H+olf/V52/e+bp1561xFr4j/AOEg16x/4WTrPiDU9Jg8zd5V1508e5ePL84lRlgmfYeuK6X4ffEnwx4E/s+//wCED+36/Zeb/wATT+2JYt+/cP8AVbWQYR9vfpnrWZfeKfAk+vaXdWXw6+yafbeb9tsf7cnf7buXEf7wjMexvm4+9nBoA67xN4O+EXhP+yJ9S1HxhPFrWmQ6hbWtslsZYEfJDSOwCndwAq5KlGJJDLWb8PtG8K6z4N8Q/YtS8UWHiay8P313efZ54orKeJDxEcAuysDHuU4Bw3PSjxX8UPB3inSo4J/hx5N3Z6YNN066/t2ZvsiIrCI7AgD7S2fm5PQmq3gX4k+GPB2gz2s/gP8AtHUL2ymsL++/tiWL7TBK2SnlhSE+UKuVwflzkZNAFX4U+AtJ8c396mq6hcCW18vyNJsXgS71DcH3eU0zqo8vaGbg/LnocVe+K/w0sPA9rZ3enWniSzFw/lvBq9vDIi8Md32mBzHlsYEWC3yOxwCuciDxl4Sh1e63fDnT5dEuJkmFnJqNz9ph2xbAiXIYYUuWcgoc7sH7qFY/Fnjqy1jQLfw74X8Px+HNBjujfSWYu3unmuSuzzDI/IAQABRgcknPGACK/wDAH2DwaniH/hLPC9zuhim/sy31LfejzCo2GLb95d3zDPG1vSpbbwXBrHwti8QaELy51e31Yade2EWJ8rKuYZlVVDRgt+6AO7e3Qg/LXG11PgXx1d+BtQuZ4LWO/t7hEL2k8siRmaKRZYJSEIyUlRTg9VLrxuyADsrX4R6PqXxkvfCGn6tcSadoVkLjVb2MefLcFAnnLBHGh2tvkCBDvKlW5Y/JR4x+FWj2vw+vPFHhm38UaV/ZcyJeWXiiy8mSZJGVEeFlUKcMcEZJxyduF38b4N8e6j4S8XTa3JFHqwvkkg1O2vv3gvoZCDIjswJyxAO7nkchgSDZ8TeLfCuo6M2n+F/ANnoRldXlupdQnu5gV+75bMQEGC4YEMGypwCqkAHZf8Id8IpfAP8AwmMeo+MLfTodT/s97aZLZp7h/K37Y9o2rjcG3McYRlxkqaPBXwx8E+N7nUb3RbzxBfWkU3k22iQy2UepBBHGWuZDJIE8vezJhVzyvPUVW/4Wv4E/4RX/AIRv/hVn/Ep+2/b/ALP/AMJDP/r9nl792zd93jGce2a5rTfGXhKCaWDVfhzp99pwmnmtkj1G5huY/McFUecMfMVEUKBsH97gltwBe+IXgTRvA3i7S4LiHxJBpV04a6hvbWJZoowV3rFOjGKZ9pLfL8q70UktuAl+JOjaHZeBPBmp+FNS8QXGk6j9u+z2mtTo/wBl8uVFbYiDam5ixOCc8HrWR4i8dWWu6hoVrH4fjs/C2huTbaGl27lleQSThrg/OS5BGf4RjAyCT0Gv/FDwdrfg228Px/Dj7NFp0Nwmlyf27M/2N5juZ8bB5nz4bDEjjHAoAG8HfD7wl4b0Ob4g6j4gudW1qyXUo7fQkiEdvbyf6sOZR8zHDZweCCMYAZq3h7wB4Uvb/wAS6rfeKLi98IeHIYXlvLKzaG5vHmGI0jjkBCfvAVJbjgdAxZS0+J+hX/hvStN8e+BrfxJc6RD9ks76PUHspBbjGyN/LX59uDgk9D0yWZqui/FT7Br2tzah4X0e40PX4Uh1DQrSL7LAfLXEbxEbjGwbLZGeWJ4O1lADxj4O8NR+B7Hxn4C1HUJtJlvW026tNWRRcwXAUyA5QbGUoAeOmRySSF19V1vVde/ZjW61zU7zUrhPGAjWW8uGmdVFmSFBYk4ySce5rn/GPxBtde8N2Phrwz4dt/DegWkzXbWkc5uJJ7g5HmPKwDHCnaB6dSQFC9L/AMLX8Cf8Ir/wjf8Awqz/AIlP237f9n/4SGf/AF+zy9+7Zu+7xjOPbNAGbovg7wZpPw+03xR8RNR1hv7bmlTTbLREj8xUhYpI8jSjb97AABzgg/Nk7DXfB3gzwZ44SHxFqOsaj4Z1PTE1LSbjSUjW5ZJW/diUSgAfKr5wAT8pwuSoq+HviRpdv4Nt/DHjXwlb+J9P0+ZptNb7W1nLa7yTInmIpLKzHdg4565wu01D4kaX4j8cSaz4v8JW+paXFZCx07R7a7azjsolYGMB41y2BvHIAJfgAAKADc+Ifg74YeA9X1fQW1HxRe6zDDvthEkAggdogUSZmAL/ADEMSgA2uB95Sa8kr0n4g/Enwx47/tC//wCED+wa/e+V/wATT+2JZdmzaP8AVbVQ5RNvbrnrXm1ABRRRQAUUUUAFFFFABRRRQAVt+CtPtdX8feH9N1CLzrS81O2gnj3Fd6PKqsMggjIJ5BzWJUltcz2d1FdWc0kFxC4kilicq8bA5DKRyCCMgigD2n/hPNc/4aO/sTz/APiQf21/Yn9h73+wfZd32Xb5G7Z9z5sY27ucY4ry3xrp9rpHj7xBpunxeTaWep3MEEe4tsRJWVRkkk4AHJOa7f8A4XND9p/4SL/hFbf/AITzyfK/4SP7bJt3eX5PmfZceXu8rjH3d3zbcfLXmNzcz3l1LdXk0k9xM5klllcs8jE5LMTySSckmgCOiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKANLw9rP/CP69b6n/Zun6n5G7/RNSg86CTcpX5kyM4zkc9QDXrfxBjtfFHxT+FseoWNvFaaroukLPZ2qmKJUlnbdGgByqgMQMHIGOa8t8KapoWkarJP4n8Of8JDaNCUS1+3Pa7H3KQ+9AScAMMdPmz2rsvGPxS8P+JdMsxpngqTRtX0u1t7TS9Ui1qV3so4XDIFXaoYgbgGJ3DOc5AoA6D/AITzXP8Aho7+xPP/AOJB/bX9if2Hvf7B9l3fZdvkbtn3Pmxjbu5xjivLfGun2ukePvEGm6fF5NpZ6ncwQR7i2xElZVGSSTgAck5rt/8Ahc0P2n/hIv8AhFbf/hPPJ8r/AISP7bJt3eX5PmfZceXu8rjH3d3zbcfLXmNzcz3l1LdXk0k9xM5klllcs8jE5LMTySSckmgCOiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigD3Lw98S/FHin4mQ+INNu5PDPg3QUga9s/tH+g2dooCmMqqBXeT5lRdhfJULwgK+P+I72x1HxVqt9pFt9l0+5vZprW38tU8qJnJRNq8LhSBgcDHFem6l8YPBeraNY6Re/DORtM09FS2so/ElxHCmN3z7EQKXO9suQWO45JrynUp7W51W7n0+0+w2kszvBa+aZPIQsSqbzy2BgZPJxmgCtRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAV6N8AP+S5eH/8At5/9Jpa85r0b4Af8ly8P/wDbz/6TS0Aec0UUUAFFd/4O+EV943sLObSvFHheG7vN/l6Zc6gy3Y2FgcxKhPRS3+7zWJ4u8G/8Il9j/wCKj8P639q3/wDIFvvtPk7dv3/lG3O7j1wfSgDm6KK0tG0b+2ft/wDxMtPsPsVlJd/6dP5Xn7Mfuo+DukbPyrxnB5oAzaKKKACiiigAoorS1nRv7G+wf8TLT7/7bZR3f+gz+b5G/P7qTgbZFx8y84yOaAM2iiigAoorSvtG+xaDpep/2lp9x/aPm/6Jbz757Xy22/vkx8m7OV5ORzQBm0UUUAFFaXiHRv8AhH9euNM/tLT9T8jb/pemz+dBJuUN8r4GcZweOoIrNoAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiu//wCFN+If7K8z7bo/9s+T5/8Awjf20f2pt278fZ8Z3eX8+zO7Hbd8tcBQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFdTD8OvEb+CdT8VXVjJY6Zp6W8ga8hkjN2sz7VaEldrgZBJyOGHXNctQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUVZ0/Tb7V7+Ox0qzuL67lz5dvbRNJI+AScKoJOACfoK1/F3gvVfBN1p1rrgjS41DT4r9Yl3B4VcsAkgZQVcFSCOcetAHP0V6bafAXxRc2tqs2oaHZaveWrXVvoN5feVfyqAxAERXAJCHqRjndtIbHm1zbT2d1La3kMkFxC5jlilQq8bA4KsDyCCMEGgCOiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAr0b4Af8ly8P/8Abz/6TS15zXo3wA/5Ll4f/wC3n/0mloA85ooooA9J/Z8/5Lt4e/7ef/SaWuI8Paz/AMI/r1vqf9m6fqfkbv8ARNSg86CTcpX5kyM4zkc9QDXSeHvjF478K6Db6NoOu/ZNPtt3lQ/Y4H27mLn5mQk/MxPJ70X3xi8d6jr2l6ze675moaT5v2Kb7HAPK81dknyhMNlRjkHHbFAHoHj/AMaaP4U/4RmbS/h34Pku9Z8P2uo3jXOm74gZNxCxxAgJg78tklgVB+4KxPhjqljrfg3xR4e1Dwx4fl/svwzqF9BqZ09TemUHKkykn7vmHGACNq88Vh6l8cviLq+lXem6h4h860vIXgnj+xW670dSrDIjBGQTyDmq3h74xeO/Cug2+jaDrv2TT7bd5UP2OB9u5i5+ZkJPzMTye9AF74UXvhHTZNRu/GOnxs4eGOy1K906S+sbNispdZoUdd5dVwgwcFS3G01t/Fex0PVPBdn4o8Kw+G57NNQ/s6a+0jTrjTXMnltIIvszsUwBljLks25FAAQ7uSi+LnjyDxFNrcHia8jvLhxJME2iGRvLEW5oAPLJ2KoyV/hB6jNZvizx14k8c3VvceKtUkv3tkKQgokaRgnJwiALk8ZOMnABPAwAXb//AIV1/wAIan9l/wDCUf8ACTeTFu+0fZ/sXm5Xzcbfn2437e/3c963PDOg2Pjr4X/2TYx6fb+JtM1q3CXEyrE0tndOsIXKAtLtndCWcHYrAKedtebVpaF4i1jwzfve+H9RuNPuZIXgeW3faxRxhh/Ig9QQGGCAQAdb4v8AE2h6T8Xp73wlouj3eiaV/odnZ3VokttOqRmNnfaf32XLOsjMWPyknjAtfGmKx/tXwrfafpWn6V/aXhmzvp7fTrZYIvNkaQsQo/Ac5OAOeK4jw94h1Twrr1vrOg3P2TULbd5U3lq+3cpQ/KwIPysRyO9dJ4h+MXjvxVoNxo2va79r0+52+bD9jgTdtYOPmVAR8yg8HtQB6B/wmmjy/Bf/AITG5+Hfg9tUi8Qf2dbxw6b5cCj7PvLSIDmTguNpYAEq3VRmPwkYNZ8F+I/HOjeAtD13xIdQhszokNmHtbC28tP3yWgJYl2UgkHOdzDAVweW/wCGg/id/wBDN/5IW3/xujwR8Q9Hsptfk8YR6xDqGsTfajr3h64+z3yOX3PH95U8tiSSAMZxkHCGMA0vjTYWPg/x94VfT/DOn6Z5Gi2dzPpLqs0RlWWTdHKwx533QrMTlgOtVvihPY6v8NfAXiG20HR9Gu9T/tH7Smk2a28b+XMiLwMk4A7k8k+tYfjb4hf25440zWPDNn/ZFpoEMFpo0bHzJI4oGLRM+4sC2TnHIxgHdgsbOpfHL4i6vpV3puoeIfOtLyF4J4/sVuu9HUqwyIwRkE8g5oA9FOpaBoGv/DTQLfwN4bu38QaTpR1G8vrESu4lbZlF4VX++S5yWyufuCvMbrVbX4ffFPxLHp+gaPqlpb3t1ZQWesWxuookWf5SAWzuAQDcSTgn1rEn8Y67c6roupT32670KGCDTpPJQeQkLbohgLhsHuwJPfNWdG+IPifw/wCKr/xHpGp/Z9W1HzPtVx9nifzPMcSP8rKVGWAPAHtxQB2XxG8MaVP+023hq1tY9O0y71CwtTFYxrEI1ljhDFABtB+Ynp1Oea1/EXiOwtPiDf8AgaD4QaPfafZzNaxWltayx6lcJGuVkFwnz/MFEmdpJQ4JOSx8/wDFfxS8Y+N9Kj03xPrH260imE6R/ZYY8OFZQcogPRm4zjmr0/xs+I1zow0uTxXeC3CIgeNUSbC4x++VRJngZO7J5yTk0AdT4HfR/Cvw+0q+1Xwnp99e+Ir2SKymu7H+157oRts2xWuYliXcwXPml2b+BlKlOW+NejadoHxj17TtGtI7OzjeJ0giGEQvCjttHYbmJAHA6AAACqOhfFPxr4a8NvoOh+ILi0059+IlRGaPf97Y5UtH3PykYJJGCSayPE3ijVvGGstq3iG4jur50VHmS3jhLheBu8tVDEDjJ5wAM4AoAyaKKKACiiigAooooAKKKKACiiigAroPANzBZ/Enw1dXk0cFvDq1rJLLK4VI1EyksxPAAAySa5+igD2T7B4h/wCGwPI33H9of8JB5+ftI3fY8+Zjdu+79m42Z+78uP4a878fXMF58SfEt1ZzRz282rXUkUsThkkUzMQykcEEHIIq7/wtPxr/AMIb/wAIr/wkFx/Y3k/Z/I2Ju8rP+r8zbv24+XG7G35fu8VyVABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQB6t4X1vVdU/Z/+Ilrqep3l5b2KaTHaxXFw0iW6/aCNqAkhRhVGBjoPSvKa9J/4aD+J3/Qzf8Akhbf/G68/wBS1C61fVbvUtQl867vJnnnk2hd7uxZjgAAZJPAGKAK1FFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAWdP1K+0i/jvtKvLixu4s+XcW0rRyJkEHDKQRkEj6GvRPjbcz3l14IuryaSe4m8H2EkssrlnkYmQlmJ5JJOSTXG+FPGOu+CNVk1LwxffYbuWEwPJ5KSZQsrEYdSOqrzjPFbfiH4xeO/FWg3Gja9rv2vT7nb5sP2OBN21g4+ZUBHzKDwe1AHb+DruTwf8AFbS9M+Jvgy41TxfeanbmHWNQ1R5GiSTZFGwX50l2EEhs5BGMgpx5b41gurbx94gg1C7+3XcWp3KT3XlCPz3ErBn2DhcnJwOBnFben/GPx/pXhuPQtP8AElxDp8UJgjURRmSNDkYWUrvXAOAQ2VAGMYGOJoAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACvRvgB/yXLw//wBvP/pNLXnNejfAD/kuXh//ALef/SaWgDzmiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAK9G+AH/JcvD/8A28/+k0tec16P8AB/xfLQP+3n/wBJpaAPOXG2RlHQEikoooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACvef2WNFs7vxPrGrzoWurCBI7c54TzC2449cLj6E0UUAf//Z)

Iz ovoga možemo izračunati točnost, preciznost, odziv i F1 mjeru.

1. **Objasnite regularizaciju. Na koji način se modificira kriterijska funkcija? Skicirajte efekt regularizacije na jednostavnim primjerima (regresija i klasifikacija)**

**Regularizacija** se koristi kako bi se smanjila složenost modela i izbjegao overfitting na skupu podataka za učenje. Koristi se kada ima puno podataka. Kriterijska funkcija se proširuje članom koji kontrolira složenost modela.

A picture containing text, diagram, line, font

Description automatically generated

1. **Objasnite algoritam K najbližih susjeda (KNN). Kako se najčešće definira mjera udaljenosti? Kako broj susjeda utječe na rezultate KNN algoritma? Koje su prednosti i nedostatci algoritma KNN?**

KNN se može koristiti i za klasifikaciju i regresiju. Temelji se na tome da se novi primjer klasificira na temelju *k* najbližih susjeda na skupu za učenje.

Postupak algoritma:

1. Definiraj broj susjeda *k*
2. Odredi udaljenost novog primjera od svih primjera na skupu za učenje
3. Odaberi *k* primjera iz skupa za učenje koji su najbliži novom primjeru
4. Za klasifikaciju, novi primjer se klasificira na temelju većinske klase K najbližih primjera. Za regresiju, vrijednost ciljne varijable se procjenjuje kao srednja vrijednost ciljne varijable K najbližih primjera.

**Mjere udaljenosti**:

1. Euklidska – najzasupljenija
2. Manhattan
3. ...

Mal broj susjeda može dovesti do overfittinga, a veći do underfittinga.

**Prednosti**:

1. Jednostavnost
2. Lako se implementira
3. Može modelirati složenije funkcije

**Nedostatci**:

1. Spor za velike skupove podataka
2. Mjera udaljenosti i broj susjeda moraju biti zadani
3. Memorijski i računalni hatjevi
4. **Skicirajte princip određivanja optimalne vrijednosti nekog hiperparametra pomoću jednostavne provjere (validacije). Koji su nedostatci ovakvog pristupa?**

-

1. **Objasnite k-struku unakrsnu validaciju. Kako se određuje optimalna vrijednost jednog ili više hiperparametara pomoću unakrsne validacije?**

**K-struka unakrsna validacija** se koristi za procjenu performansi modela i umanjuje rizik od overfittinga. Skup podataka za učenje se dijeli u *k* podskupa, gdje se *k-1* koristi za učenje modela, a 1 za validaciju. Procedura se ponavlja *k* puta te se dobiva *k* modela s pripadnom procjenom pogreške na temelju koje se čuna prosječna pogreška. *K* se obično uzima između 5 i 10.

1. **Od čega se sastoji stablo odlučivanja, koje su prednosti i nedostatci stabla odlučivanja? Što predstavlja mjera nečistoće (npr. entropija) i kako se ona koristi u procesu učenja? Kako je moguće spriječiti pretjerano usklađivanje na podatke stabla odlučivanja?**

**Stablo odlučivanja** se sastoji od čvorova, grana i listova. Čvorovi predstavljaju test određene ulazne veličine. Grane predstavljaju moguće ishode testa, tj. odluke. Listovi predstavljaju konačnu odluku ili ishod

Primjer:

A picture containing text, screenshot, diagram, font

Description automatically generated

**Prednosti**:

* Lako razumljivi
* Može se koristiti i za klasifikaciju i za regresiju
* Ne zathevaju predobradu podataka u smislu skaliranja

**Nedostatci**:

* Sklona su pretjeranom usklađivanju (overfitting)
* Ograničena u svojoj aproksimaciji

**Mjera nečistoće** se koristi za određivanje toga koliko dobro neka značajka podijeli skup primjera u različite klase. Najčešća mjera nečistoće je entropija. Entropija mjeri koliko je neodređeno neko stanje.

Kako bi se **spriječio overfitting** može se ograničiti dubina stabla, broj primjera, ...

1. **Objasnite temeljnu pretpostavku strojeva s potpornim vektorima. Kako se dobiva procjena izlazne veličine pomoću ovog algoritma? Kako se dobiva nelinearna (složena) granica odluke? Objasnite što predstavljaju hiperparametri C i gama.**

-

A picture containing text, screenshot, font

Description automatically generated

1. **Što je cilj nenadziranog učenja? Kakve podatke koristimo i zašto su baš takvi?**

**Nenadzirano učenje** je vrsta strojnog učenja gdje je cilj istražiti strukturu dostupnih podataka kako bi izvukli važne informacije. Cilj je otkriti uzorke i strukture u **neoznačenim** podacima, tj. podacima koji nemaju oznake o tome kojoj klasi pripadaju.

Koristimo **neoznačene** podatke. To znači da nećemo imati vrijednosti odgovarajuće izlazne veličine. Razlog tome je što ne znamo unaprijed koje klase postoje u podacima. Primjer: Grupiranje korisnika – ne znamo uopće unaprijed kakvi tipovi korisnika postoje.

1. **Objasnite algoritam *K* srednjih vrijednosti na jednostavnom problemu s dvije ulazne veličine. Koji su načini inicijalizacije centara? Objasnite princip određivanja optimalnog broja grupa K.**

**Algoritam *K* srednjih vrijednosti** je algoritam grupiranja koji kao rezultat daje podijelu podatkovnog skupa u *K* grupa gdje svaki podatkovni primjer pripada samo jednoj grupi. Svaka grupa se predstavlja *m* dimenzionalnim vektorom koji se naziva centar.

Cilj algoritma je odrediti vrijednosti centara; broj *K* potrebno je unaprijed definirati.

Primjer s dvije ulazne veličine:

A picture containing text, screenshot, pattern

Description automatically generated

Nakon 4 iteracije algoritma dobijamo:

A picture containing diagram, text, line, screenshot

Description automatically generated

Algoritam počinje sa slučajnim odabirom K centara grupa, a zatim iterativno radi sljedeće korake:

1. Svaki podatak se dodjeljuje najbližem centru grupa
2. Centri grupa se ažuriraju tako da predstavljaju srednju vrijednost svih podataka koji su im dodijeljeni

Postupak se ponavlja sve dok se centri grupa ne stabiliziraju ili dok se ne postigne unaprijed definirani broj iteracija. Kako bi se odredio optimalan broj grupa K, koristi se metoda tzv. lakta (engl. elbow method). Ideja ove metode je da se za svaki K izračuna suma kvadrata udaljenosti svakog podatka do njemu najbližeg centra grupe. Dobivene vrijednosti se zatim prikažu na grafu, te se traži vrijednost K nakon koje promjena vrijednosti sume kvadrata postaje manje izražena.

**Načini inicijalizacije centara**:

* Nasumično odabrani podaci kao centri grupa
* Korištenje nekog algoritma za grupiranje
* Ručno određivanje početnih centara

1. **Objasnite kvantizaciju boje digitalne slike pomoću algoritma *K* srednjih vrijednosti i kako se može postići kompresija slike.**

**Kvantizacija boje digitalne slike** je postupak smanjivanja broja boja koje se koriste za prikazivanje slike. Jedan od načina je primjena algoritma *K* srednjih vrijednosti (*K*-means).

Postupak:

1. Odabiremo *K* boja u slici
2. Konvertiranje slike iz RGB u 2D matricu gdje je svaki redak jedan piksel, a svaki stupac jedan kanal (R, G, B)
3. Pokrećemo algoritam *K* srednjih vrijednosti koji će izračunati pozicije centara
4. Vrijednosti svakog piksela (R, G, B) zamijeni s vrijednostima najbližeg centra

Cilj je pronaći manji broj boja od originalne slike s tim da nova slika bude što sličnija originalnoj.

1. **Objasnite hijerarhijsko aglomerartivno grupiranje na jednostavnom primjeru s dvije ulazne veličine (značajke). Na koje se sve načine može definirati udaljenost (sličnost) dva primjera ili grupe?**

-

1. **Objasnite princip rada DBSCAN algoritma.**

-

1. **Što je smanjivanje dimenzionalnosti? Zašto se ono radi? Koja su dva pristupa smanjivanju dimenzionalnosti?**

To je smanjivanje broja dimenzija u skupu podataka. Radi se kako bi se smanjila složenost algoritma te kako bi se olakšalo učenje.

**Dva pristupa**:

* Odabir značajki
* Izvlačenje značajki

1. **Objasnite analizu glavnih komponenti. Kako se određuje potrebni broj komponenti? Na konkretnom primjeru (npr. skup podataka koji sadrže lica) objasnite kako se postiže kompresija podataka.**

**Analiza glavnih komponenti** je metoda za smanjivanje dimenzionalnosti podataka.

-

1. **Skicirajte umjetni neuron. Koje aktivacijske funkcije se najčešće koriste i kako izgledaju? Kako se izračunava izlaz iz neurona na temelju ulaznih veličina?**

Skica:

A diagram of a mathematical equation

Description automatically generated with low confidence

Lijevi dio neurona je linearna funkcija, a *g(z)* je aktivacijska funkcija.

Umjetni neuron prima signale od drugih neurona isto kao biološki neuron.

**Aktivacijske funkcije**:

A picture containing text, diagram, line, plot

Description automatically generated

1. **Izvedite izraze za podešavanje parametara umjetnog neurona s dvije ulazne veličine.**

---

1. **Što je umjetna neuronska mreža? Kakva je to unaprijedna potpuno povezana neuronska mreža? Skicirajte potpuno povezanu mrežu s 3 ulazne veličine, 4 neurona u skrivenom sloju i 3 neurona u izlaznom sloju. Koliko ovakva mreža ima paremetara?**

**Umjetna neuronska mreža** je spoj više neurona u strukturu.

**Unaprijedne potpuno povezane mreže** su mreže u kojima se propagacija signala odvija samo u jednom smjeru, od ulaza prema izlazu. Ne sadrži povratne veze.

Prikaz mreže:

A diagram of a network

Description automatically generated with low confidence

Mreža ima 31 parametar:

3 (ulazne vrijednosti) \* 4 (elementi u skrivenom sloju) = 12

4 (elementi u skrivenom sloju) \* 3 (izlazne vrijednosti) = 12

4 (elementi u skrivenom sloju) + 3 (izlazne vrijednosti) = 7

12 + 12 + 7 = 31

Prvo zbrajamo sve kombinacije do skrivenog sloja i onda od njega isto te dodamo skriveni i izlazni sloj.

1. **Kako se strukturira izlazni sloj neuronske mreže prilikom rješavanja regresijskih problema, a kako prilikom rješavanja klasifikacijskih problema?**

Kada se rješavaju **regresijski problemi**, tada se izlazni sloj mreže sastoji od **jednog neurona s linearnom aktivacijskom funkcijom**.

Kada se rješavaju **klasifikacijski problemi**, tada se izlazni sloj mreže sastoji od:

* **jednog neurona sa sigmoidnom aktivacijskom funkcijom**
* ***K* neurona sa softmax aktivacijskom funkcijom**

1. **Navedite osnovne korake prilikom izgradnje neuronske mreže. Objasnite načelno princip rada algoritma unazadne propagacije (engl. *backpropagation algorithm*). Što nam omogućava ovaj algoritam?**

**Osnovni koraci prilikom izgradnje neuronske mreže**:

1. Model

2. Odabir kriterijske funkcije

3. Optimizacijski postupak

**Algoritam unazadne propagacije** omogućava računanje gradijenta kriterijske funkcije. Sastoji se od 2 faze:

* **Unaprijedna propagacija**
  + Svaki neuron izračunava svoj izlaz na temelju ulaza
* **Unazadna propagacija pogreške**
  + Greška se propagira od izlaza prema ulazima

1. **Napišite u općem obliku izraze koji opisuju unaprijednu propagaciju kroz mrežu s jednim skrivenim slojem.**

---

1. **Što je epoha i veličina *batcha*? Što je stopa učenja? Što je inicijalizacija parametara mreže?**

**Epoha** je jedan prolazak kroz čitav skup za učenje prilikom treniranja mreže

**Veličina batch-a** – koliko se primjera koristi u svakoj iteraciji prilikom treniranja mreže

**Inicijalizacija parametara mreže** – određivanje početnih vrijednosti neuronske mreže

1. **Objasnite princip sprječavanja pretjeranog usklađivanja na podatke pomoću ranog zaustavljanja (engl. *earyl stopping*).**

Tijekom treniranja modela pratimo gubitak i određenu metriku na validacijskom skupu te po potrebi zaustavimo traniranje modela kada performanse na validacijskom skupu počnu opadati.

1. **Objasnite tehniku nasumičnog izbacivanja neurona tijekom učenja (engl. *dropout*).**

Koristi se za sprječavanje pretjeranog usklađivanja na podatke. Ideja je nasumično izbacivanje neurona u određenom sloju s nekom, unaprijed definiranom, vjerojatnosti *p*. To znači da se mreža u svakoj iteraciji mora prilagoditi uvjetima (ako se izbaci neuron).

Primjer:

A picture containing diagram, circle, line, origami

Description automatically generated

1. **Što je 2D konvolucija? Objasnite/skicirajte na jednostavnom primjeru filtriranja digitalne slike.**

**2D konvolucija** je matematička operacija koja se koristi u obradi slika, a temelji se na konceptu konvolucije. Konvolucija je specijalan slučaj filtriranja slike koji koristi operator (“maska”) koji se primjenjuje na piksel slike uzimajući u obzir njegovo susjedstvo. Maska je matrica unaprijed definirane veličine koja se stavlja na ulaznu matricu (početnu sliku) i zatim se primjenjuje **operacija zbrajanja i množenja** na piksele koji se preklapaju. Masku mičemo po cijeloj ulaznoj matrici i stvaramo novu matricu (filtriranu sliku).

Primjer:

A picture containing text, rectangle, diagram

Description automatically generated

Izoštravanje slike postiže se maskom:

A picture containing diagram, line, design

Description automatically generated

1. **Što je konvolucijski sloj i zašto se koristi? Skicirajte primjer primjene konvolucijskog sloja na neki ulazni volumen, naznačite ulazne i izlazne dimenzije. Što je *stride*? Što je *padding*? O čemu ovisi broj parametara konvolucijskog sloja?**

**Konvolucijski sloj** je glavni gradbeni element u CNN. Svaki konvolucijski sloj sastoji se od većeg broja filtera.

A picture containing text, screenshot, diagram, line

Description automatically generatedA picture containing screenshot, diagram, text, design

Description automatically generated

A picture containing text, diagram, screenshot, line

Description automatically generated

**Stride** – broj koji kaže za koliko se elemenata pomiče filtar po prostornoj dimenziji ulaznog volumena

**Padding**:

A picture containing diagram, text, square, rectangle

Description automatically generated

1. **Što je sloj sažimanja po maksimalnoj vrijednosti (eng. *max pooling*) i zašto se koristi? Skicirajte primjer primjene ovog sloja na neki volumen, naznačite ulazne i izlazne dimenzije.**

**Sloj sažimanja** radi smanjivanje svake pojedine mape značajki u ulaznom volumenu. Time se smanjuje količina računskim operacija i memorijski zahtjevi pa time i broj parametara mreže.

**Sažimanje po maksimalnoj vrijednosti**:

A picture containing text, screenshot, diagram, line

Description automatically generated

1. **Za danu konvolucijsku mrežu napišite dimenzije pojedinih volumena/izlaza iz slojeva te ukupan broj parametara mreže.**

-

1. **Što je augmentacija skupa podataka za učenje?**

To je postupak umjetnog povećavanja skupa podataka za učenje na način da se novi primjeri kreiraju od postojećih primjera. Odnosi se na modifikaciju postojećih slika. Glavni cilj je sprječavanje overfittinga. Moguće je primijeniti i kod obrade teksta i zvuka

1. **Navedite najpopularnije “gotove” strukture konvolucijskih neuronskih mreža koje se mogu koristiti kao ekstraktor značajki (što je karakteristično za svaku pojedinu mrežu?). Na kojem skupu su najčešće već istrenirane ove mreže i koje su karakteristike ovog skupa?**

-

1. **Što je učenje prijenosom (engl. *transfer learning*) i kako se odvija?**

**Učenje prijenosom** je tehnika u kojoj se za izradu modela koristi već istrenirani model kao polazna točka. Omogućuje treniranje i kad nemamo velik skup podataka. Smanjuje vrijeme treniranja. Ideja je preuzeti naučene značajke na jednom problemu i primijeniti ih na novi, sličan problem.