

# Klasifikacija Zvezda i Galaksija Korišćenjem Algoritama Mašinskog Učenja

Aleksa Toroman 84/2018

Milica Sudar 79/2017

Univerzitet u Beogradu, Matematički fakultet

29. maj 2024.

# Sadržaj

<b>1</b>	<b>Uvod</b>	<b>3</b>
<b>2</b>	<b>Metodologija</b>	<b>3</b>
2.1	Softverski alati i okruženje . . . . .	3
2.2	Korišćene biblioteke . . . . .	3
<b>3</b>	<b>Podaci i pretprocesiranje</b>	<b>3</b>
3.1	SuperCOSMOS Sky Survey . . . . .	3
3.2	Sloan Digital Sky Survey . . . . .	4
3.3	Učitavanje i čišćenje podataka . . . . .	5
3.3.1	Učitavanje podataka . . . . .	5
3.3.2	Identifikacija i uklanjanje anomalija . . . . .	5
3.4	Odabir atributa za treniranje modela . . . . .	7
3.4.1	Odstranivanje nepotrebnih atributa . . . . .	7
3.4.2	Izračunati atributi . . . . .	7
3.4.3	RA & DEC . . . . .	7
3.4.4	Magnituda . . . . .	8
3.4.5	Redshift . . . . .	9
3.4.6	Class (klasa) . . . . .	10
3.4.7	Zaključak . . . . .	10
3.5	Priprema podataka za treniranje modela . . . . .	11
3.5.1	Unakrsna validacija . . . . .	12
3.5.2	Skaliranje podataka . . . . .	12
<b>4</b>	<b>Klasifikacioni algoritmi</b>	<b>13</b>
4.1	Metrike za evaluaciju modela . . . . .	13
4.2	Stabla odlučivanja . . . . .	14
4.2.1	Trening . . . . .	14
4.2.2	Rezultati . . . . .	15
4.3	Logistička regresija . . . . .	16
4.3.1	Treniranje i evaluacija . . . . .	16
4.3.2	Rezultati unakrsne validacije . . . . .	16
4.3.3	Rezultati . . . . .	16
4.4	K-najbližih suseda . . . . .	17
4.4.1	Normalizacija podataka . . . . .	17
4.4.2	Prilagođavanje i evaluacija modela . . . . .	17
4.4.3	Rezultati . . . . .	19
4.5	Slučajna šuma . . . . .	20
4.5.1	Treniranje modela . . . . .	20
4.5.2	Optimizacija modela . . . . .	20
4.5.3	Rezultati . . . . .	21
4.6	Naive Bayes (Gaus) . . . . .	22
4.6.1	Treniranje i evaluacija . . . . .	23
4.6.2	Rezultati . . . . .	23
<b>5</b>	<b>Zaključak</b>	<b>24</b>

# 1 Uvod

Klasifikacija nebeskih objekata na zvezde i galaksije predstavlja jedan od ključnih izazova u astronomiji i astrofizici. Tačna klasifikacija ovih objekata omogućava bolje razumevanje strukture i evolucije svemira. Ovo istraživanje ima za cilj da primenom različitih klasifikacionih algoritama postigne visoku tačnost u klasifikaciji, kao i da identifikuje ključne attribute koji najviše doprinose tačnosti predikcija.

## 2 Metodologija

### 2.1 Softverski alati i okruženje

Za potrebe ovog istraživanja korišćeni su sledeći softverski alati i okruženja:

- **PyCharm:** Integrisano razvojno okruženje (IDE) za Python, koje omogućava efikasno kodiranje, debugovanje i testiranje.
- **Python:** Glavni programski jezik korišćen za implementaciju algoritama, analizu podataka i vizualizaciju rezultata.
- **Jupyter Notebook:** Interaktivno okruženje za analizu podataka i vizualizaciju, koje omogućava lako eksperimentisanje sa kodom i pregled rezultata.

### 2.2 Korišćene biblioteke

Za analizu podataka i implementaciju klasifikacionih algoritama korišćene su sledeće Python biblioteke:

- **pandas:** Za manipulaciju i analizu podataka.
- **numpy:** Za numeričke operacije i rad sa nizovima.
- **scikit-learn:** Za implementaciju i evaluaciju klasifikacionih algoritama.
- **matplotlib** i **seaborn:** Za vizualizaciju podataka i rezultata.

## 3 Podaci i preprocesiranje

U ovom istraživanju razmatrana su dva različita skupa podataka: jedan sa SuperCOSMOS Sky Survey sajta i drugi sa Sloan Digital Sky Survey sajta. U nastavku ćemo opisati način prikupljanja, učitavanja i preprocesiranja podataka za oba skupa, dok za ključke do kojih smo došli u ovom postupku ćemo iskoristiti da izaberemo jedan skup i da nad njim pravimo modele za klasifikaciju nebeskih tela.

### 3.1 SuperCOSMOS Sky Survey

Podaci sa SuperCOSMOS Sky Survey sajta prikupljeni su korišćenjem dostupnih pretraga i alata za ekstrakciju podataka dostupnih na njihovom zvaničnom sajtu. Originalno, skup podataka je sadržao 901,322 redova i 43 atributa. Tokom preprocesiranja ovih podataka identifikovano je nekoliko značajnih problema u samom skupu:

- **Magnitude:** Mnoge vrednosti za magnitude (u, g, r, i, z) su bile prazne ili su imale placeholder vrednosti koje su bile velike cifre, što je indiciralo nedostatak stvarnih podataka ili prisustvo grešaka. Ove vrednosti su morale biti odstranjene pre dalje analize.
- **RA i DEC:** Vrednosti za Celestial Right Ascension (RA) i Celestial Declination (DEC) kao dva glavna lokacijska atributa su bila vrlo slična za sve redove, što je ukazivalo na to da su podaci dobijeni sa ograničenog područja neba. Ovo je bilo posledica limitacije upita koji je vraćao podatke samo iz određenih regiona sa definisanim radijusom pretrage. Ovaj nedostatak bi možda uticao da se ne vide određeni šabloni u raspodeli zvezda i galaksija na osnovu njihovih lokacija.

Nakon uklanjanja nedostajućih vrednosti i outlier-a, preostalo je svega 82,880 redova za dalju analizu, što je znatno manje od originalnih 901,322 redova. Ovo smanjenje broja redova je posledica visokog procenta nedostajućih vrednosti i pre svega velikog prisustva ranije pomenutih placeholder vrednosti koji su značajno umanjili kvalitet podataka.

## 3.2 Sloan Digital Sky Survey

Podaci sa Sloan Digital Sky Survey (SDSS) sajta prikupljeni su korišćenjem odgovarajućeg SQL upita u njihovom data centru. Upit je preuzet sa SDSS sajta i modifikovan tako da obuhvati podatke koji uključuju samo zvezde i galaksije, isključujući quasare kao tip nebeskih objekata.

```
-- https://skyserver.sdss.org/dr18/SearchTools/sql
SELECT TOP 500000
  p.objid, p.ra, p.dec, p.u, p.g, p.r, p.i, p.z,
  p.run, p.rerun, p.camcol, p.field,
  s.specobjid, s.class, s.z as redshift,
  s.plate, s.mjd, s.fiberid
FROM PhotoObj AS p
  JOIN SpecObj AS s ON s.bestobjid = p.objid
WHERE s.class IN ('STAR', 'GALAXY')
```

Slika 1: SQL upit korišćen za prikupljanje podataka sa SDSS sajta

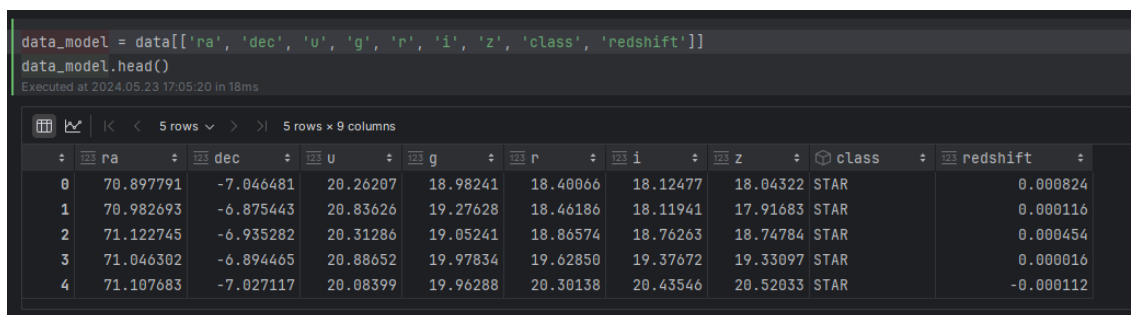
Upitom je prikupljeno 500.000 redova. Spektar atributa dostupnih putem SDSS baze podataka mnogo je širi u poređenju sa SuperCOSMOS bazom, što omogućava detaljniju analizu i bolje treniranje modela. Na osnovu preporuka i primera sa njihovog sajta, izabrali smo određen broj ključnih atributa iz dostupnih tabela za dohvaćanje podataka.

Pretprocesiranje podataka sa SDSS sajta pokazalo je znatno manji broj nedostajućih vrednosti i outlier-a u poređenju sa SuperCOSMOS Sky Survey skupom podataka. Nakon uklanjanja redova sa takvim anomalijama, preostalo je oko 492,000 redova koji su bili kvalitetni za dalju analizu. S obzirom na viši kvalitet podataka kao i veći broj slogova, dalji rad i analiza u ovom istraživanju fokusirani su na ovaj skup podataka.

## 3.3 Učitavanje i čišćenje podataka

### 3.3.1 Učitavanje podataka

Podaci su učitani korišćenjem biblioteke pandas, koja je moćan alat za manipulaciju i analizu podataka u Pythonu. Nakon učitavanja, proverili smo da li je skup podataka uspešno dovučen i uradili smo osnovne operacije za pregled podataka i njegove statistike. Takođe u ovom procesu su i isključene kolone koje nisu potrebne za dalju analizu i treniranje modela za klasifikaciju (više o izboru atributa u narednoj sekciji):



```
data_model = data[['ra', 'dec', 'u', 'g', 'r', 'i', 'z', 'class', 'redshift']]
data_model.head()
```

Executed at 2024.05.23 17:05:20 in 18ms

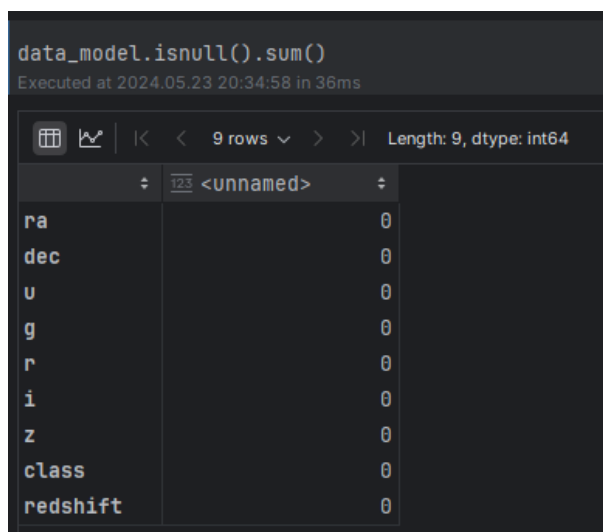
	ra	dec	u	g	r	i	z	class	redshift
0	70.897791	-7.046481	20.26207	18.98241	18.40066	18.12477	18.04322	STAR	0.000824
1	70.982693	-6.875443	20.83626	19.27628	18.46186	18.11941	17.91683	STAR	0.000116
2	71.122745	-6.935282	20.31286	19.05241	18.86574	18.76263	18.74784	STAR	0.000454
3	71.046302	-6.894465	20.88652	19.97834	19.62850	19.37672	19.33097	STAR	0.000016
4	71.107683	-7.027117	20.08399	19.96288	20.30138	20.43546	20.52033	STAR	-0.000112

Slika 2: Izlistavanje nekolicine redova učitanoog skupa podataka

### 3.3.2 Identifikacija i uklanjanje anomalija

Proces čišćenja podataka je ključan korak u pripremi podataka za treniranje modela. Ovaj proces uključuje identifikaciju i razrešavanje anomalija, outlier-a i praznih vrednosti. S obzirom na količinu dostupnih podataka, a imajući i u vidu osetljivost u vrednostima atributa, strategija za razrešavanje u našem scenariju je bila jednostavno odstranjivanje redova sa takvim anomalijama.

Prvo smo proverili da li skup podataka ima nedostajuće vrednosti u nekim kolonama:



```
data_model.isnull().sum()
```

Executed at 2024.05.23 20:34:58 in 36ms

	<unnamed>
ra	0
dec	0
u	0
g	0
r	0
i	0
z	0
class	0
redshift	0

Slika 3: Provera nedostajućih vrednosti u kolonama

Međutim, naš skup podataka nije imao takav slučaj pa smo obradu ovakvih redova preskočili.

Za uklanjanje outlier-a koristili smo algoritam Local Outlier Factor (LOF) za identifikaciju i uklanjanje outlier-a. LOF algoritam procenjuje udaljenost svakog podatka od njegovih suseda i identifikuje one podatke koji se značajno razlikuju od ostatka skupa.

Takođe, i bez algoritma jednostavnim pregledom osnovnih statistika smo uvideli da neke kolone u sebi imaju određene anomalije:

	ra	dec	u	g	r	i	z	redshift
count	500000.000000	500000.000000	500000.000000	500000.000000	500000.000000	500000.000000	500000.000000	500000.000000
mean	157.041133	21.733104	21.836999	20.346590	19.253018	18.470464	18.260552	0.308527
std	101.681940	19.596665	56.737171	42.566563	37.536372	53.038040	37.522237	0.301834
min	0.000464	-11.241591	-9999.000000	-9999.000000	-9999.000000	-9999.000000	-9999.000000	-0.011219
25%	49.481773	3.391814	20.201615	18.560318	17.694418	17.285407	17.015497	0.000203
50%	162.432338	22.664959	22.450030	21.040390	19.697010	18.996600	18.600295	0.226605
75%	211.180766	34.459235	23.908010	22.278320	20.910872	19.986250	19.529592	0.544987
max	359.998704	84.221075	31.475460	32.149970	31.990100	32.101780	30.017040	2.002617

Slika 4: Prikaz osnovne statistike skupa podataka

Na slici se vidi da određeni broj kolona koji se odnosi na magnitude ima vrednost od -9999.00 što je predstavljalo placeholder za redove koje imaju grešku ili su nepoznate.

Koraci u procesu čišćenja podataka uključivali su:

- **Provera atributa:** Prvo je provereno da li skup podataka sadrži kategoričke attribute. Pošto su svi atributi numerički, svi atributi su uključeni u primeni algoritma za detekciju outlier-a.
- **Filtriranje outlier-a:** Na osnovu rezultata LOF algoritma, podaci identifikovani kao outlier-i su uklonjeni iz skupa podataka.

Korišćenjem ovog pristupa, uspeali smo da identifikujemo i uklonimo outlier-e iz skupa podataka, čime smo poboljšali kvalitet podataka za treniranje modela. Nakon čišćenja podataka, preostalo je 492,153 redova koji su bili kvalitetni za dalju analizu.

	ra	dec	u	g	r	i	z	redshift
count	492153.000000	492153.000000	492153.000000	492153.000000	492153.000000	492153.000000	492153.000000	492153.000000
mean	156.884300	21.752059	22.156484	20.520144	19.384453	18.745129	18.392884	0.310627
std	101.687495	19.587729	2.363537	2.194855	1.938050	1.757989	1.724364	0.300718
min	0.000464	-11.241591	12.055210	10.487280	9.432361	8.809973	9.701558	-0.011219
25%	49.159440	3.437514	20.202210	18.558580	17.693200	17.284070	17.014670	0.000238
50%	162.372542	22.714609	22.446720	21.039360	19.689000	18.991330	18.606210	0.235375
75%	211.034336	34.462908	23.893790	22.271070	20.898850	19.976680	19.519890	0.546242
max	359.998704	84.221075	28.405650	27.467450	25.346950	24.726670	24.333020	2.002617

Slika 5: Skup podataka nakon uklanjanja outlier-a

## 3.4 Odabir atributa za treniranje modela

Proces odabira atributa započeo je razumevanjem i analizom svakog atributa. Nakon toga, koristili smo različite grafičke prikaze kako bismo identifikovali koji atributi mogu biti najznačajniji za treniranje modela.

### 3.4.1 Odstranivanje nepotrebnih atributa

Za početak bez ikakve dodatne analize mogli smo da odstranimo neke attribute samo na osnovu njihovog značenja.

Atributi kao što su 'objid', 'run' i 'rerun' odnose se na administraciju podataka i nisu direktno bitni za izradu samog klasifikacionog modela.

Atributi 'camcol', 'field', 'plate' i 'fiberid' predstavljaju tehničke podatke koji se odnose na uslove pod kojima je snimljen nebeski objekat.

Atribut 'mjd' predstavlja modifikovani Julianov datum posmatranja i nije relevantan za određivanje da li je objekat zvezda ili galaksija.

### 3.4.2 Izračunati atributi

U razmatranje smo takođe uzeli i attribute koji nisu direktno dostupni u samom skupu podataka, ali se mogu izračunati i imaju određeno značenje u konkretnom domenu. Takvi atributi u našem slučaju su takozvani 'colour-indexes'.

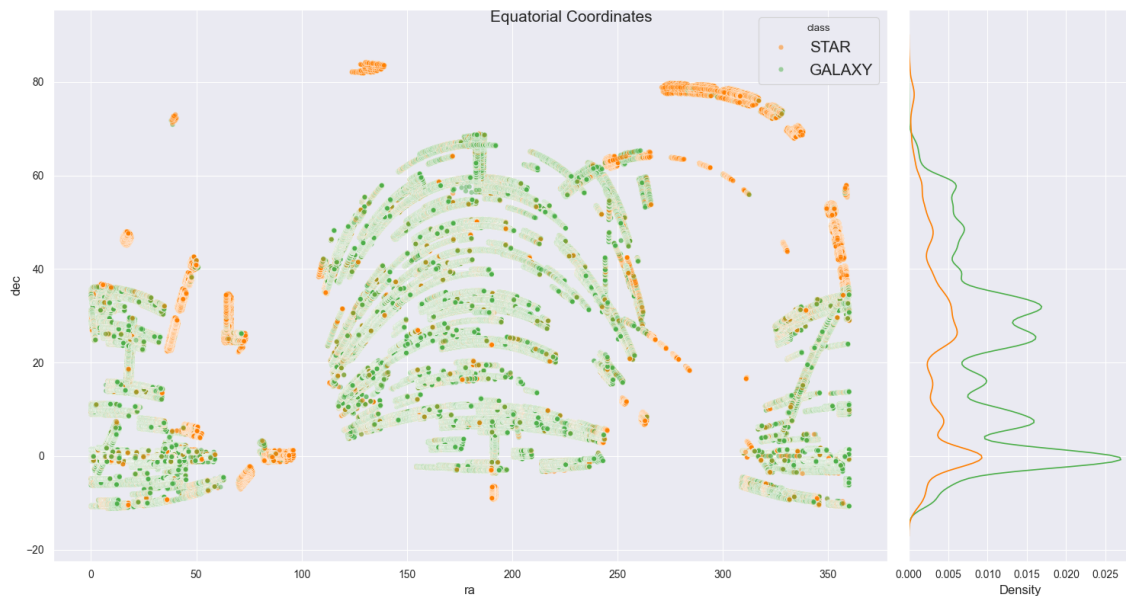
Ovi atributi su izračunati na osnovu magnituda i predstavljaju razliku između sjaja u različitim delovima spektra. Oni pružaju dodatne informacije o karakteristikama nebeskih objekata.

- **color\_u\_g**: Razlika između magnituda u ultraljubičastom (u) i zelenom (g) delu spektra:  $color\_u\_g = u - g$ .
- **color\_g\_r**: Razlika između magnituda u zelenom (g) i crvenom (r) delu spektra:  $color\_g\_r = g - r$ .
- **color\_r\_i**: Razlika između magnituda u crvenom (r) i bliskom infracrvenom (i) delu spektra:  $color\_r\_i = r - i$ .
- **color\_i\_z**: Razlika između magnituda u bliskom infracrvenom delu spektra (i) i (z):  $color\_i\_z = i - z$ .

### 3.4.3 RA & DEC

- **RA (Right Ascension)**: Ovo je jedna od dve osnovne nebeske koordinate koje se koriste za određivanje položaja nebeskih objekata na nebeskoj sferi. RA se meri u satima, minutima i sekundama, i predstavlja ugaonu udaljenost objekta istočno od Prolećne tačke. RA je ekvivalent geografskoj dužini na Zemlji.
- **DEC (Declination)**: Druga osnovna nebeska koordinata koja se koristi za određivanje položaja objekata na nebeskoj sferi. DEC se meri u stepenima, minutima i sekundama, i predstavlja ugaonu udaljenost objekta severno ili južno od nebeskog ekvatora. DEC je ekvivalent geografskoj širini na Zemlji.

Sledeći grafikon prikazuje prostorni raspored zvezda i galaksija u ekvatorijalnim koordinatama.



Slika 6: Prostorni raspored zvezda i galaksija u ekvatorijalnim koordinatama

Iz grafikona se može videti da:

- Zvezde (narandžaste tačke) su raspoređene u određenim regijama, često u skupovima, što ukazuje na oblasti sa visokom koncentracijom zvezda.
- Galaksije (zelene tačke) su takođe raspoređene u skupovima, ali pokazuju drugačiji obrazac rasporeda u odnosu na zvezde.
- Gustinski graf na desnoj strani pokazuje da postoji razlika u raspodeli zvezda i galaksija duž deklinacije, što može pomoći u daljem istraživanju i klasifikaciji objekata.

#### 3.4.4 Magnituda

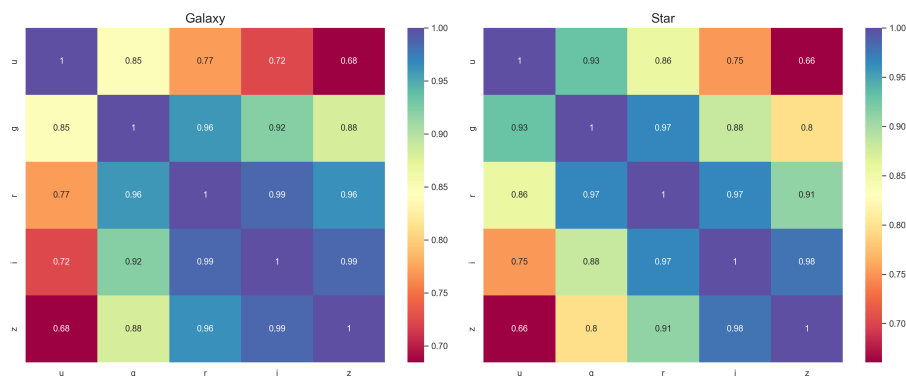
Magnituda je mera sjaja nebeskog objekta viđenog sa Zemlje. Postoji pet filtera (u, g, r, i, z) koji mere sjaj u različitim delovima elektromagnetnog spektra:

- **u (ultraljubičasti filter):** Mera sjaja u ultraljubičastom delu spektra (354 nm).
- **g (zeleni filter):** Mera sjaja u zelenom delu spektra (476 nm).
- **r (crveni filter):** Mera sjaja u crvenom delu spektra (628 nm).
- **i (bliski infracrveni filter):** Mera sjaja u bliskom infracrvenom delu spektra (769 nm).
- **z (bliski infracrveni filter):** Mera sjaja u bliskom infracrvenom delu spektra (925 nm).



Značaj magnituda može se uočiti kroz prikaz odgovarajuće korelacione matrice za ove atribute.

Korelaciona matrica je vizualni prikaz koji pokazuje međusobnu povezanost različitih atributa korišćenih u skupu podataka.



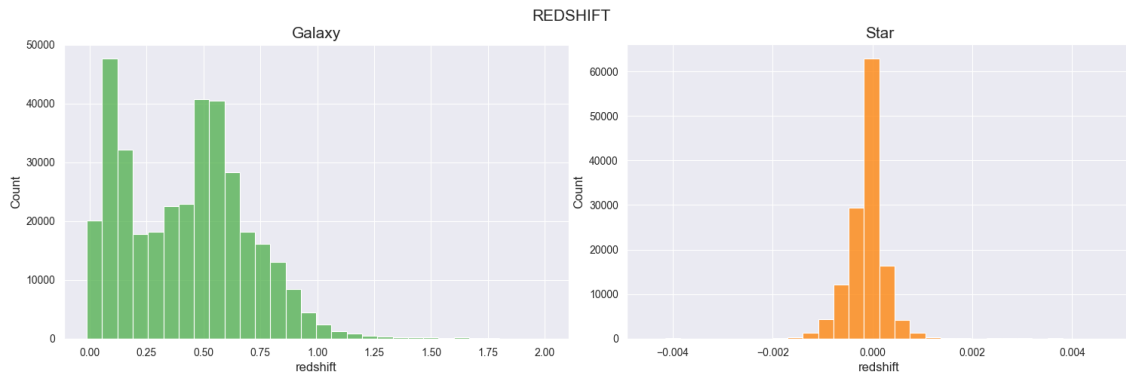
Slika 7: Korelaciona matrica za atribute zvezda i galaksija

Na grafikonu su prikazane korelacije između magnituda za zvezde (desno) i galaksije (levo).

Na prvi pogled možemo primetiti da su korelacione matrice veoma slične za obe klase (zvezde i galaksije). Možemo zaključiti da postoje visoke korelacije između različitih opsega. Ovo nije iznenađujuće; intuitivno, očekivali bismo da ako jedan opseg zabeleži svetlost nekog objekta, i ostali opsezi bi trebali zabeležiti svetlost. Međutim, interesantno je primetiti da je opseg 'u' manje povezan sa ostalim opsezima. Ovo je u skladu sa činjenicom da opsezi 'u', 'g', 'r', 'i', 'z' zabeležavaju svetlost na talasnim dužinama od 354, 476, 628, 769 i 925 nm, redom. Ovo može ukazivati da galaksije i zvezde sjaje jače na talasnim dužinama od 476 do 925 nm. Ipak, treba biti oprezan sa ovakvim interpretacijama.

### 3.4.5 Redshift

Crveni pomak je ključan fenomen u astronomiji jer omogućava određivanje udaljenosti i brzine udaljavanja nebeskih objekata, što je posebno važno za klasifikaciju između zvezda i galaksija. Razumevanje crvenog pomaka pomaže u identifikaciji galaksija koje se udaljavaju od nas zbog širenja svemira, dok zvezde, sa svojim karakterističnim relativnim brzinama unutar naše galaksije, pružaju dodatne informacije o njihovim kretanjima i položajima u odnosu na Sunčev sistem.



Slika 8: Histogram crvenog pomaka za zvezde i galaksije

Na slici 8 prikazan je histogram crvenog pomaka za zvezde (desno) i galaksije (levo). Iz ovog histograma možemo zaključiti nekoliko važnih činjenica:

- **Galaksije:**

- Crveni pomak galaksija varira u širokom rasponu, od 0 do preko 2.0.
- Većina galaksija ima crveni pomak između 0.0 i 1.0, sa vrhom distribucije oko 0.5.
- Ovaj široki raspon crvenog pomaka potvrđuje da se galaksije udaljavaju različitim brzinama, što je u skladu sa Hablovim zakonom koji kaže da je brzina udaljavanja galaksija proporcionalna njihovoj udaljenosti od nas.

- **Zvezde:**

- Crveni pomak zvezda je veoma koncentrisan oko nule, sa većinom vrednosti između -0.004 i 0.004.
- Ova koncentracija oko nule ukazuje na to da se zvezde unutar naše galaksije kreću relativno malim brzinama u poređenju sa galaksijama, te se njihov crveni pomak uglavnom javlja zbog njihovih orbitalnih kretanja unutar naše galaksije.

Crveni pomak može se koristiti kao procena udaljenosti objekta od Zemlje. Na osnovu histograma, većina posmatranih zvezda je bliža Zemlji nego galaksije. Galaksije su generalno dalje, što se može objasniti time da galaksije, zbog svoje veličine i fizičke strukture, emituju jače zračenje i mogu se posmatrati sa većih udaljenosti nego 'male' zvezde.

Kako možemo razlikovati klase objekata na osnovu kolone 'crveni pomak', ovaj atribut će verovatno biti od velike pomoći u klasifikaciji novih objekata.

### 3.4.6 Class (klasa)

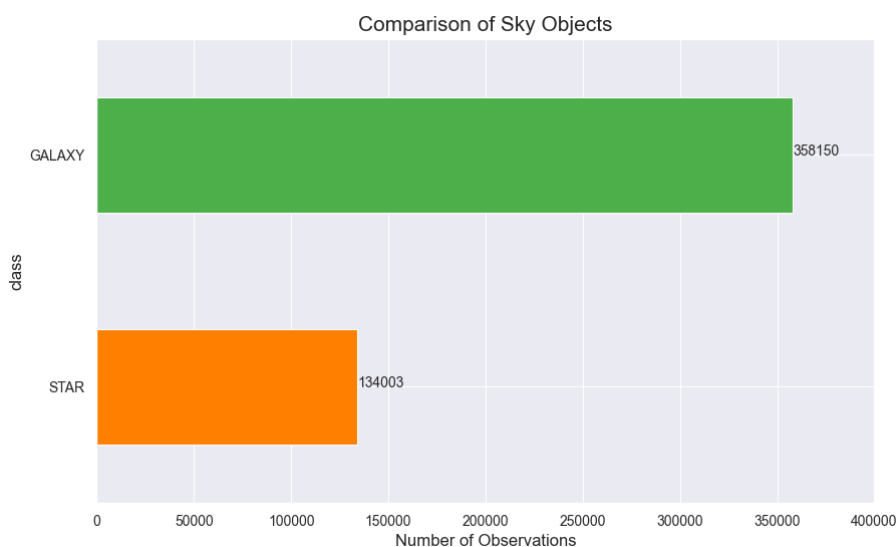
Ovaj atribut označava klasifikaciju objekta kao zvezda ili galaksija. Ovo je ciljna promenljiva u klasifikacionom modelu.

### 3.4.7 Zaključak

Za treniranje klasifikacionih modela korišćeni su sledeći atributi: 'ra', 'dec', 'u', 'g', 'r', 'i', 'z', 'class', 'redshift', kao i izračunati atributi 'color\_u\_g', 'color\_g\_r', 'color\_r\_i', 'color\_i\_z'.

### 3.5 Priprema podataka za treniranje modela

Prvo što želimo da uradimo, želimo da proverimo koji je odnos klasa koje imamo u našem skupu podataka.



Slika 9: Odnos klasa u skupu podataka

Vidimo da u skupu podataka imamo mnogo veći broj galaksija u odnosu na broj zvezda, svakako model ćemo trenirati na svim dostupnim instancama.

Za uspešno treniranje i evaluaciju klasifikacionih modela, skup podataka je podeljen na dva dela: trening skup i test skup. Ovaj pristup omogućava pravilnu procenu performansi modela i njegovu generalizaciju na nove podatke.

Skup podataka je podeljen na sledeći način:

- **Trening Skup (Training Set):** Ovaj skup podataka se koristi za treniranje modela. Model uči obrasce i veze između atributa na osnovu ovog skupa.
- **Test Skup (Test Set):** Ovaj skup se koristi za konačnu evaluaciju modela nakon što je treniran. Test skup pruža nezavisnu procenu performansi modela na novim, nepoznatim podacima.

Podaci su podeljeni tako da trening skup čini 75% ukupnih podataka, i test skup preostalih 25%. Ova podela je odabrana kako bi se osiguralo da ima dovoljno podataka za treniranje modela, ali i dovoljno za testiranje.

Takođe, prilikom podele podataka vodili smo računa da zadržimo originalni odnos zvezda i galaksija i u ovim podskupovima, kako ne bismo napravili trening i test skup koji su potencijalno još više disbalansirani.

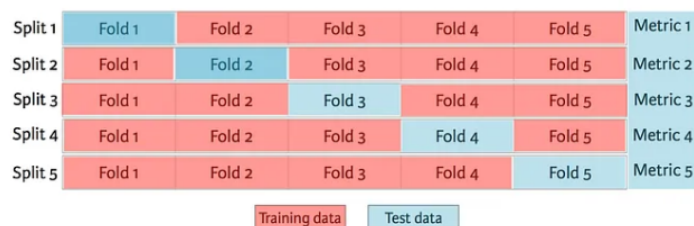
Treba napomenuti da nismo unapred definisali i odredili validacioni skup, koji bismo koristili za korigovanje različitih hiperparametara kod različitih klasifikacionih algoritama. Umesto toga, koristili smo unakrsnu validaciju (cross-validation) kako bismo evaluirali performanse modela i optimizovali hiperparametre.

### 3.5.1 Unakrsna validacija

Unakrsna validacija je tehnika koja se koristi za procenu performansi modela i osiguranje njegove generalizacije na nezavisnim skupovima podataka. U ovoj tehnici, prilikom treniranja modela, trening podaci se dele na  $k$  podskupova (ili foldova). Model se trenira na  $k - 1$  podskupovima, a validira se na preostalom podskupu. Ovaj proces se ponavlja  $k$  puta, tako da svaki podskup bude korišćen kao validacioni barem jednom. Na kraju, rezultati svih iteracija se agregiraju kako bi se dobila pouzdana procena performansi modela.

Korišćenjem unakrsne validacije, postigli smo nekoliko stvari:

- **Optimizaciju hiperparametara:** Unakrsna validacija omogućava efikasno podešavanje hiperparametara modela bez potrebe za unapred definisanim validacionim skupom, što može povećati tačnost modela.
- **Bolju ocenu generalizacije:** Unakrsna validacija pomaže da se model generalizuje bolje na nezavisnim podacima jer koristi različite podskupove za treniranje i validaciju i samim tim možemo biti sigurniji da će se model ponašati slično i na testnim podacima kasnije.



Slika 10: Unakrsna validacija

### 3.5.2 Skaliranje podataka

Skaliranje podataka je ključan korak u pripremi podataka za treniranje modela. Podaci su standardizovani tako da imaju srednju vrednost 0 i standardnu devijaciju 1. Ovo je važno jer određeni algoritmi klasifikacije zahtevaju skalirane podatke kako bi se izbeglo da atributi sa većim opsegom vrednosti postanu automatski važniji od ostalih atributa. Skaliranje osigurava da svi atributi imaju jednaku težinu prilikom treniranja modela.

Skaliranje podataka je takođe značajno za mnoge optimizacione algoritme, kao što je gradijentni spust, jer doprinosi bržoj konvergenciji i stabilnijem učenju modela. Standardizacija podataka omogućava algoritmima da efikasnije pretražuju prostor rešenja, što rezultira boljim performansama i bržim treniranjem. Još jedna bitna napomena jeste da podatke treba normalizovati nakon što ih podelimo na trening i test skupove kako bismo sprečili 'curenje podataka' koje bi modelu dalo dodatne informacije o test skupu ako bismo normalizovali sve podatke odjednom, a podelu podataka izvršili naknadno.

## 4 Klasifikacioni algoritmi

### 4.1 Metrike za evaluaciju modela

Za evaluaciju performansi klasifikacionih modela korišćene su sledeće metrike: tačnost (accuracy), preciznost (precision), odziv (recall), F1-score i podrška (support). Ove metrike omogućavaju sveobuhvatnu procenu kvaliteta modela i njegove sposobnosti da tačno klasifikuje nebeske objekte kao zvezde ili galaksije.

Važno je napomenuti da je skup podataka neuravnotežen, sa većim brojem galaksija u odnosu na zvezde. U ovakvim situacijama, određene metrike, kao što su F1-score i odziv, postaju posebno važne za procenu performansi modela.

- **Tačnost (Accuracy):** Tačnost je mera koja pokazuje procenat tačno klasifikovanih instanci od ukupnog broja instanci. Visoka tačnost ukazuje na to da model dobro prepoznaje i zvezde i galaksije.

$$\text{Tačnost} = \frac{\text{Broj tačno klasifikovanih instanci}}{\text{Ukupan broj instanci}}$$

- **Preciznost (Precision):** Preciznost pokazuje koliko od svih instanci koje su modelom klasifikovane kao pozitivne (npr. zvezde) zaista pripada pozitivnoj klasi. Visoka preciznost znači da je mali broj lažno pozitivnih klasifikacija.

$$\text{Preciznost} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

gde su TP pravi pozitivni, a FP lažno pozitivni primeri.

- **Odziv (Recall):** Odziv pokazuje koliko od svih stvarno pozitivnih instanci model tačno prepoznaje kao pozitivne. Visok odziv znači da je model sposoban da prepozna većinu pozitivnih instanci, što je posebno važno u neuravnoteženim datasetima gde je potrebno identifikovati što više pozitivnih instanci.

$$\text{Odziv} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

gde su TP pravi pozitivni, a FN lažno negativni primeri.

- **F1-score:** F1-score je harmonijska sredina preciznosti i odziva i koristi se kao jedinstvena mera performansi modela, naročito kada postoji neuravnoteženost klasa. F1-score uzima u obzir i preciznost i odziv, pružajući balansiranu meru performansi modela.

$$\text{F1-score} = 2 \cdot \frac{\text{Preciznost} \cdot \text{Odziv}}{\text{Preciznost} + \text{Odziv}}$$

- **Podrška (Support):** Podrška označava broj instanci u svakoj klasi u skupu podataka. Koristi se za procenu uravnoteženosti klasa i može uticati na izbor metrika za evaluaciju modela.

Uz ove metrike, koristili smo i matricu konfuzije za dodatnu evaluaciju modela. Matrica konfuzije je tabela koja prikazuje stvarne klase naspram predviđenih klasa, omogućavajući identifikaciju tačno i pogrešno klasifikovanih instanci. Redovi matrice predstavljaju

stvarne klase, dok kolone predstavljaju predviđene klase. Matrica konfuzije pomaže u vizualizaciji performansi modela i identifikaciji specifičnih vrsta grešaka, kao što su lažno pozitivni i lažno negativni primerci.

## 4.2 Stabla odlučivanja

Decision tree (stablo odluke) je jedan od najjednostavnijih i najintuitivnijih algoritama za klasifikaciju. Radi tako što deli podatke na osnovu vrednosti atributa i pravi hijerarhijsku strukturu odluka, gde svaka grana predstavlja ishod odluke, a svaki čvor predstavlja atribut po kojem se podaci dele.

### 4.2.1 Trening

Za treniranje Decision Tree modela korišćen je skup podataka koji je prethodno podeljen na trening i test skup. Za treniranje modela nije bilo potrebno koristiti skalirane podatke, jer Decision Tree algoritmi ne zavise od udaljenosti između tačaka.

Decision Tree algoritmi imaju različite hiperparametre koji mogu uticati na kvalitet modela. Neki od tih hiperparametara uključuju maksimalnu dubinu stabla, minimalan broj uzoraka potrebnih za podelu čvora, minimalan broj uzoraka u listu i broj karakteristika koje se koriste za traženje najbolje podele.

Da bismo pronašli najbolju kombinaciju hiperparametara, koristili smo GridSearchCV funkciju iz paketa scikit-learn. Ova funkcija automatski pretražuje različite kombinacije hiperparametara i procenjuje performanse modela za svaku od njih. Na ovaj način možemo identifikovati konfiguraciju hiperparametara koja daje najbolje rezultate za dati skup podataka.

Jedan od parametara koji se prosleđuje GridSearchCV funkciji je metrika koja će se koristiti za procenu kvaliteta modela tokom pretrage. U našem slučaju, koristili smo tačnost (accuracy).

Važno je napomenuti da GridSearchCV koristi unakrsnu validaciju (cross-validation) kako bi procenio performanse modela u svakoj iteraciji pretrage. Za naše modele smo koristili 10-fold unakrsnu validaciju, što znači da je skup podataka podeljen na 10 delova, pri čemu se model trenira 10 puta, svaki put koristeći drugačiji deo kao test skup, a preostalih 9 delova kao trening skup.

Mana ovog pristupa je što može biti vremenski zahtevan, jer se broj treniranja modela (fitova) računa kao broj mogućih kombinacija hiperparametara pomnožen sa brojem foldova.

Nakon što smo identifikovali najbolji skup hiperparametara, iskoristili smo ih da ponovo istreniramo model, ali ovog puta koristeći ceo trening skup podataka.

```
best_dtree = DecisionTreeClassifier(max_depth=15,
                                    min_samples_leaf=5,
                                    min_samples_split = 8,
                                    max_features = None)
```

Slika 11: Najbolji parametri za model stabla odlučivanja

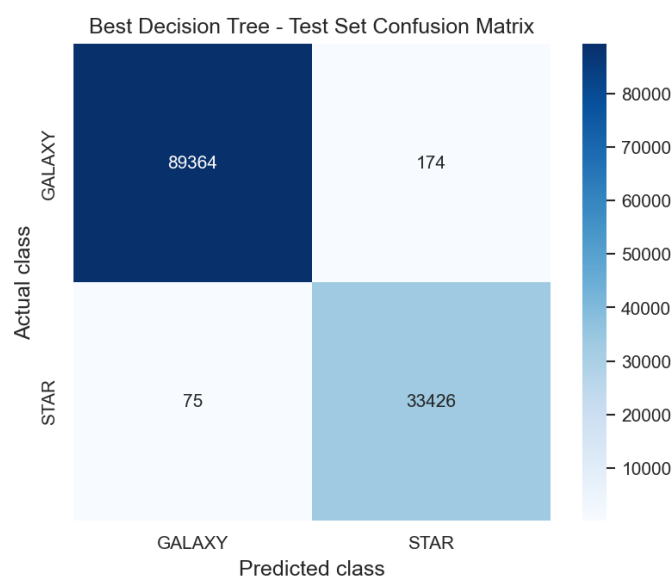
#### 4.2.2 Rezultati

Nakon optimizacije hiperparametara, model je evaluiran na test skupu podataka. Rezultati pokazuju visoku tačnost i odlične metrike za klasifikaciju.

Decision Tree - Test Set				
Decision Tree accuracy best params: 0.9979762514324726				
	precision	recall	f1-score	support
GALAXY	0.99916	0.99806	0.99861	89538
STAR	0.99482	0.99776	0.99629	33501
accuracy			0.99798	123039
macro avg	0.99699	0.99791	0.99745	123039
weighted avg	0.99798	0.99798	0.99798	123039

Slika 12: Rezultati klasifikacije stabla odlučivanja na test skupu

Rezultati pokazuju da je model stabla odlučivanja vrlo efikasan u klasifikaciji nebeskih objekata, sa visokim vrednostima preciznosti, odziva i F1-score-a za obe klase.



Slika 13: Matrica konfuzije za model stabla odlučivanja na test skupu

Matrica konfuzije pruža dodatni uvid u rezultate evaluacije modela, prikazujući tačne

i netačne klasifikacije za svaku klasu.

Ovi rezultati potvrđuju da algoritam stabla odlučivanja može biti izuzetno efikasan za klasifikaciju nebeskih objekata, pružajući visok nivo tačnosti i pouzdanosti.

### 4.3 Logistička regresija

Logistička regresija (logistic regression) je popularan algoritam za binarnu klasifikaciju koji modelira verovatnoću da primerak pripada određenoj klasi. Algoritam koristi logističku funkciju za predikciju verovatnosti, a zatim klasifikuje primerke na osnovu praga verovatnoće (obično 0.5).

#### 4.3.1 Treniranje i evaluacija

Za treniranje modela korišćen je skup podataka koji je prethodno podeljen na trening i test skup. Podaci su skalirani kako bi se obezbedila bolja konvergencija i kvalitet modela.

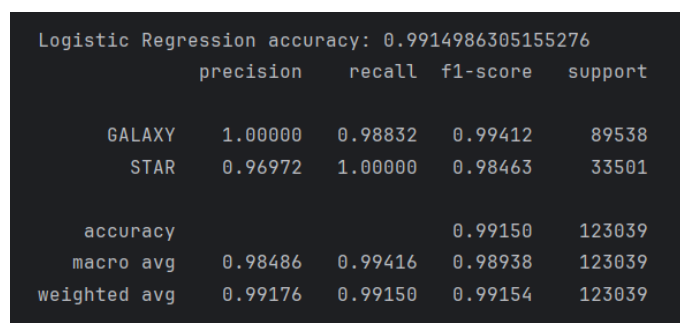
Kako bismo dodatno poboljšali kvalitet modela i smanjili varijansu, korišćena je unakrsna validacija.

#### 4.3.2 Rezultati unakrsne validacije

Prosečna tačnost modela procenjena unakrsnom validacijom je pokazala visok nivo tačnosti, sa malim standardnim odstupanjem, što ukazuje na konzistentne performanse modela.

Mean Accuracy: 0.99019 (0.001)

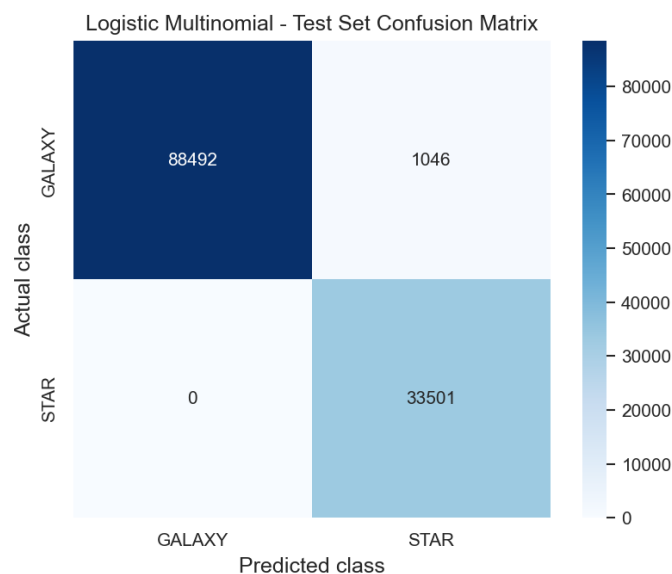
#### 4.3.3 Rezultati



Logistic Regression accuracy: 0.9914986305155276				
	precision	recall	f1-score	support
GALAXY	1.00000	0.98832	0.99412	89538
STAR	0.96972	1.00000	0.98463	33501
accuracy			0.99150	123039
macro avg	0.98486	0.99416	0.98938	123039
weighted avg	0.99176	0.99150	0.99154	123039

Slika 14: Rezultati klasifikacije modela na test skupu





Slika 15: Matrica konfuzije za Logistic Regression model na test skupu

Iako ovaj model postiže dobre sveobuhvatne metrike, kao što su visoka tačnost, preciznost i odziv, primećeno je da ima tendenciju grešaka u klasifikaciji galaksija. Konkretno, model često pogrešno klasifikuje galaksije kao zvezde, što je prikazano u matrici konfuzije na slici 15.

Ova greška može biti značajna u određenim aplikacijama, čime stabla odlučivanja postaju bolji izbor za ovaj zadatak zbog veće preciznosti u razlikovanju klasa.

## 4.4 K-najbližih suseda

K-najbližih suseda (KNN) je jednostavan, ali moćan algoritam mašinskog učenja koji se koristi za zadatke klasifikacije i regresije. U ovom radu koristićemo ga za klasifikaciju. KNN ne gradi globalni model, već koristi primere iz trening skupa za pravljenje predikcija za test instance. Takvi algoritmi zahtevaju meru blizine kako bi odredili sličnost ili udaljenost između instanci i funkciju klasifikacije koja vraća predviđenu klasu test instance na osnovu njene blizine drugim instancama. Neke od mera udaljenosti koje se koriste za pronalaženje najbližeg suseda su: Euklidsko rastojanje, rastojanje Minkovskog, Hamingovo rastojanje i Menhetn rastojanje.

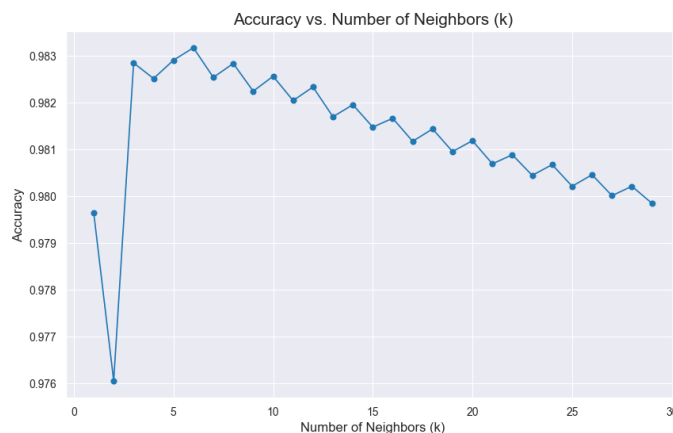
### 4.4.1 Normalizacija podataka

Kod KNN algoritma neophodno je da su podaci normalizovani, zato što algoritam koristi meru za udaljenost između tačaka.

### 4.4.2 Prilagođavanje i evaluacija modela

Za prilagođavanje KNN modela, koristili smo grid pretragu u kombinaciji sa unakrsnom validacijom kako bismo optimizovali vrednosti hiperparametara. Izbor vrednosti 'k' igra ključnu ulogu u performansama KNN modela. Klasifikator sa malim brojem suseda može dovesti do preprilagođavanja, dok model sa mnogo suseda može dovesti do nedovoljno prilagođenog modela.

Ukoliko vizualizujemo sad performanse modela za različite vrednosti  $k$  na trening podacima, najbolje performanse su koristeći oko 6 suseda. Što je broj suseda veći to model postaje jednostavniji i tačnost na trening skupu opada. Međutim, ako pogledamo y-osu, razlika između tačnosti nije velika i bilo koja vrednost ' $k$ ' bi dobro obavila posao na našem problemu jer je najgora performansa oko 0.976.



Slika 16: Odnos performansi modela i broja suseda

Slično kao i kod prethodnih modela, za traženje najboljeg skupa hiperparametara koristili smo grid pretragu u kombinaciji sa unakrsnom validacijom.

Kako bismo poboljšali model optimizujemo tri glavna hiperparametra koja utiču na prilagođavanje modela: `n_neighbours`, `weights` i `metric`. Svi ostali hiperparametri ostali su na svojim podrazumevanim vrednostima.

```
param_grid = {
    "n_neighbors": np.arange(1,12),
    "weights": ['uniform', 'distance'],
    "metric": ["euclidean", "manhattan"],
    "n_jobs": [4]
}
```

- **n\_neighbors**: Ovo predstavlja broj najbližih suseda koje K-NN algoritam treba da uzme u obzir.
- **weights**: Ovo određuje način na koji se računa doprinos svakog suseda. Postoje dve opcije:
  - **uniform**: Svi susedi imaju isti doprinos bez obzira na udaljenost
  - **distance**: Susedi bliži tački imaju veći doprinos nego udaljeni susedi.
- **metric**: Ovo određuje način na koji se meri udaljenost između tačaka. Postoje dve opcije:
  - **euclidean**
  - **manhattan**

- `n_jobs`: Ovo određuje broj paralelnih zadataka koje treba pokrenuti prilikom obrade. Vrednost 4 znači da će se koristiti četiri procesorska jezgra za paralelnu obradu.

Rezultati su pokazali da je najbolja kombinacija parametara

```
Best params: {'metric': 'manhattan',
              'n_neighbors': 6,
              'weights': 'uniform'}
Best score: 0.9835416993884458
```

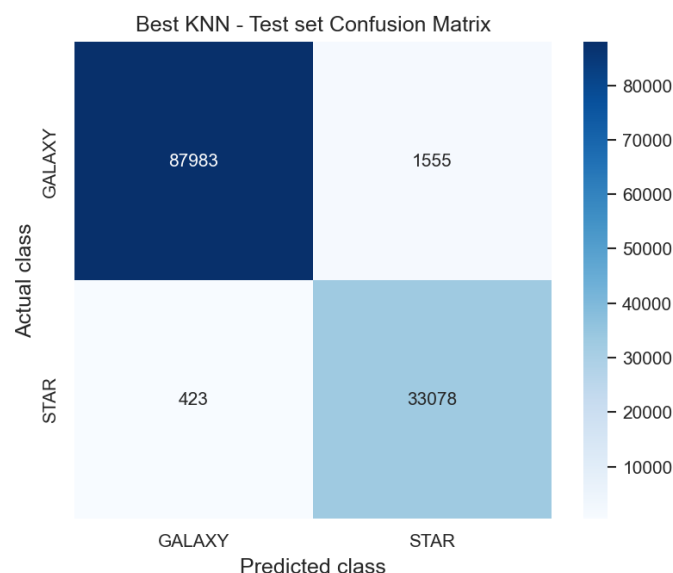
Nakon što smo odradili optimizaciju modela vreme je da pokrenemo najbolje dobijen model na test podacima i da vidimo kolika će biti uspešnost modela.

#### 4.4.3 Rezultati

```
KNN - Test Set
KNN accuracy with best params: 0.9839237965198027
```

	precision	recall	f1-score	support
GALAXY	0.99522	0.98263	0.98888	89538
STAR	0.95510	0.98737	0.97097	33501
accuracy			0.98392	123039
macro avg	0.97516	0.98500	0.97993	123039
weighted avg	0.98429	0.98392	0.98401	123039

Slika 17: Rezultati KNN modela na test skupu



Slika 18: Matrica konfuzije za KNN model na test skupu

Vidimo da ovaj model postiže lošije rezultate u odnosu na prvobitna dva algoritma.

## 4.5 Slučajna šuma

Metod slučajnih šuma se zasniva na prosto agregaciji stabala odlučivanja. Ansambl (skupovi većeg broja modela koji zajednički donose odluke) se sastoji od  $m$  stabala treniranih na različitim podskupovima skupa za obučavanje. Jedno stablo se obučava tako što se izabere podskup skupa za obučavanje određene veličine, pri čemu je moguće koristiti i samo podskup ukupnog skupa atributa. Stabla se obučavaju na različitim podskupovima kako bi njihove greške bile što slabije korelisane, što ostavlja prostor za popravku agregacijom.

Slučajne šume su jedan od najprimenjenijih algoritama mašinskog učenja. Njihovo obučavanje je relativno efikasno, a preciznost predviđanja obično među najboljim za vektorski predstavljene podatke. Njegova jednostavnost korišćenja i fleksibilnost doprineli su njegovoj popularnosti, jer može da rešava i klasifikacione i regresione probleme. Neki od nedostataka upotrebe algoritma slučajne šume su: zahteva više resursa za obradu, troši više vremena u poređenju sa algoritmom stabla odluke, manje je intuitivan kada imamo veliki broj stabala odluke, izuzetno je složen i zahteva više računске snage.

### 4.5.1 Treniranje modela

Kao i kod prethodnih algoritama, koristili smo prethodno podeljene skupove za trening i test. Korišćenje slučajne šume obično ne zahteva skaliranje podataka. Algoritmi za slučajne šume, kao i drugi algoritmi bazirani na stablima odluke, nisu osetljivi na skalu ili distribuciju podataka.

### 4.5.2 Optimizacija modela

Hiperparametri se koriste u slučajnim šumama da bi se poboljšale performanse i prediktivna moć modela ili da bi model bio brži. Hiperparametri na koje smo se odlučili da se fokusiramo su:

- **n\_estimators** - broj stabala u šumi
- **max\_features** - maksimalan broj atributa korišćenih pri podeli čvora, obično manji od broja atributa u skupu podataka
- **max\_depth** - maksimalan broj nivoa u svakom stablu odluke
- **min\_samples\_split** - minimalan broj podataka u čvoru pre nego što se čvor podeli
- **min\_samples\_leaf** - minimalan broj podataka dozvoljen u listu
- **criterion** - funkcija za merenje kvaliteta podele

Postoji nekoliko tehnika optimizacije hiperparametara za algoritam slučajne šume, kao što su grid pretraga, Bajesova optimizacija i nasumična pretraga. Za razliku od grid pretrage, za podešavanje hiperparametara zasnovano na nasumičnoj pretrazi se ne koriste sve moguće kombinacije hiperparametara. Grid pretraga iscrpno pretražuje svaku kombinaciju unapred definisanih vrednosti hiperparametara, što značajno povećava vreme obrade. Međutim, nasumična pretraga uzima uzorak konstantne veličine konfiguracija hiperparametara određenih raspodelom njihovih vrednosti. Korišćenje nasumične pretrage

omogućava pronalaženje raznovrsnije regije hiperparametara. Iako ne garantuje pronalaženje hiperparametara koji čine najbolju kombinaciju, dovoljno je dobra da brzo postigne vrlo dobru kombinaciju, što je posebno važno kada se radi sa skupovima podataka u astronomiji, gde obično postoji veliki broj opservacija, što generiše značajno vreme obrade.

Izvršili smo nasumičnu pretragu hiperparametara koristeći unakrsnu validaciju ( $cv=10$ ), kroz 100 različitih kombinacija ( $n\_iter=100$ ), i sa svim dostupnim jezgrima istovremeno ( $n\_jobs=-1$ ). Nasumična pretraga nasumično bira kombinaciju hiperparametara umesto iteriranja kroz svaku moguću kombinaciju. Veći  $n\_iter$  i  $cv$  rezultiraju većim brojem kombinacija i manjom mogućnošću prekomernog prilagođavanja. U svakoj iteraciji, algoritam će odabrati različitu kombinaciju hiperparametara.

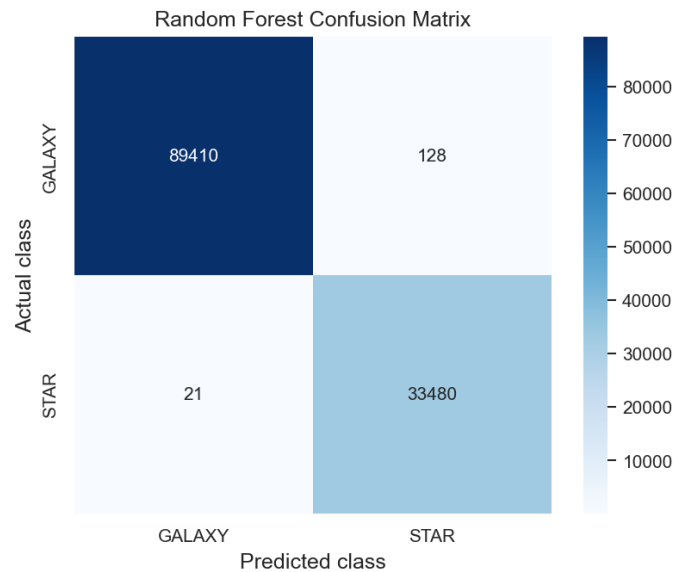
Najbolji hiperparametara koji su dobijeni nasumičnom pretragom

```
{
  'n_estimators': 100,
  'min_samples_split': 5,
  'min_samples_leaf': 2,
  'max_features': None,
  'max_depth': 15,
  'criterion': 'entropy'
}
```

### 4.5.3 Rezultati

Random Forest - Test Set				
Random Forest accuracy with best params: 0.9987646193483367				
	precision	recall	f1-score	support
GALAXY	0.99974	0.99856	0.99915	89538
STAR	0.99616	0.99931	0.99773	33501
accuracy			0.99876	123039
macro avg	0.99795	0.99894	0.99844	123039
weighted avg	0.99877	0.99876	0.99877	123039

Slika 19: Rezultati Slučajne sume na test skupu



Slika 20: Matrica konfuzije za Slučajnu sumu na test skupu

Još jedna od dobrih karakteristika slučajne šume jeste što možemo da odredimo važnost svakog pojedinačnog atributa.

feature	importance
redshift	0.984701
color_u_g	0.002335
z	0.001965
g	0.001499
color_i_z	0.001452
color_g_r	0.001378
color_r_i	0.001370
ra	0.001270
dec	0.001212
u	0.001181

Slika 21: Važnost atributa

Možemo primetiti da je redshift najvažniji atribut od svih kao što smo i pretpostavili na početku na osnovu histograma.

Takođe vidimo da do sada, ovaj klasifikacioni algoritam daje najbolje rezultate.

## 4.6 Naive Bayes (Gaus)

Gaussian Naive Bayes (GNB) je algoritam za klasifikaciju zasnovan na Bayesovoj teoremi sa pretpostavkom da su karakteristike podjednako i nezavisno normalno distribuirane. Ovaj algoritam je jednostavan za implementaciju i efikasan, posebno za visokodimenzionalne podatke.

### 4.6.1 Treniranje i evaluacija

Treniranje i evaluacija Gaussian Naive Bayes modela izvedeni su korišćenjem pretrage za optimizaciju hiperparametara. Hiperparametar *var\_smoothing*, koji dodaje varijansu na procene verovatnoće kako bi se stabilizovale procene, optimizovan je korišćenjem unakrsne validacije (cross-validation) sa deset preklopnih skupova (foldova).

### 4.6.2 Rezultati

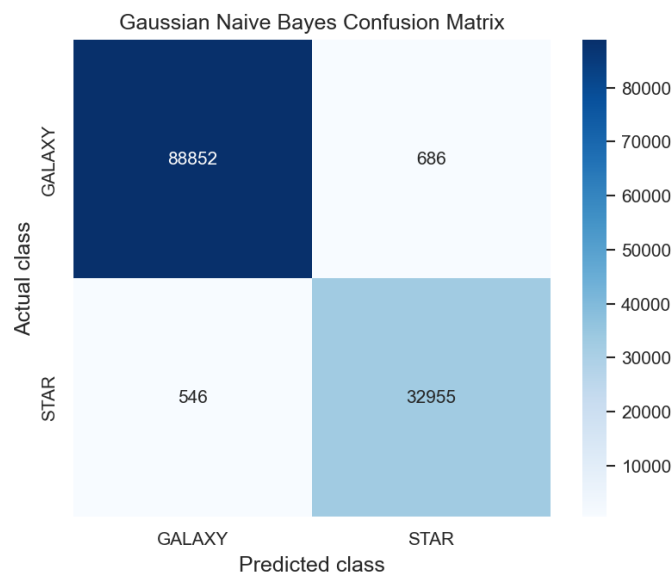
Nakon optimizacije hiperparametara, Gaussian Naive Bayes model je evaluiran na test skupu podataka. Rezultati su prikazani u sledećim slikama.

```
Gaussian NB - Test Set
Gaussian NB accuracy best params: 0.9899869147180975
      precision    recall  f1-score   support

   GALAXY      0.99389    0.99234    0.99311     89538
    STAR      0.97961    0.98370    0.98165     33501

 accuracy                   0.98999     123039
 macro avg      0.98675    0.98802    0.98738     123039
weighted avg      0.99000    0.98999    0.98999     123039
```

Slika 22: Rezultati klasifikacije Gaussian Naive Bayes modela na test skupu



Slika 23: Matrica konfuzije za Gaussian Naive Bayes model na test skupu

## 5 Zaključak

÷	Classifier	÷ <small>123</small> Accuracy	÷ <small>123</small> F1 Score	÷ <small>123</small> Precision	÷ <small>123</small> Recall	÷
0	Logistic Regression	0.99150	0.99154	0.99176	0.99150	
1	Decision Tree	0.99798	0.99799	0.99799	0.99798	
2	KNN	0.98392	0.98401	0.98429	0.98392	
3	Gaussian Naive Bayes	0.98999	0.98999	0.99000	0.98999	
4	Random forest	0.99877	0.99877	0.99878	0.99877	

Slika 24: Rezultati korišćenih modela

Na osnovu ovih rezultata, možemo zaključiti da je Random Forest model pokazao najbolje performanse u svim evaluacionim metrikama. Decision Tree se takođe može smatrati dobrom alternativom zbog svojih visokih performansi i jednostavnosti interpretacije.

Ipak, moramo napomenuti da ne postoji univerzalno najbolji model u realnom svetu. Kao što je George E.P. Box jednom rekao: „Svi modeli su pogrešni, ali neki su korisni.“ Važno je samo pronaći model koji je najkorisniji za konkretan problem i podatke.



# Literatura

- [1] Pang-Ning Tan - Introduction to Data Mining (2019, Pearson Education Limited)
- [2] The SuperCOSMOS Sky Survey - I. Introduction and description (2001) by N. Hambly, H. MacGillivray, M. Read, S. Tritton, E. Thomson, D. Kelly, D. Morgan, R. Smith, S. Driver, J. Williamson, Q. Parker, M. Hawkins, P. Williams and A. Lawrence
- [3] The SuperCOSMOS Sky Survey. Paper II: Image detection, parameterisation, classification and photometry (2001) by N. Hambly, M. Irwin and H. MacGillivray
- [4] Identifying galaxies, quasars, and stars with machine learning: A new catalogue of classifications for 111 million SDSS sources without spectra A. O. Clarke, A. M. M. Scaife, R. Greenhalgh and V. Griguta
- [5] Probst, P.; Wright, M.N.; Boulesteix, A.L. Hyperparameters and tuning strategies for random forest. Wiley Interdiscip. Rev. Data Min. Knowl. Discov. 2019, 9, 1301.
- [6] Xu, B.; Huang, J.Z.; Williams, G.; Wang, Q.; Ye, Y. Classifying very high-dimensional data with random forests built from small subspaces. Int. J. Data Warehous. Min. \*\*2012\*\*, 8, 44–63.
- [7] Javeed, A.; Zhou, S.; Yongjian, L.; Qasim, I.; Noor, A.; Nour, R. An intelligent learning system based on random search algorithm and optimized random forest model for improved heart disease detection.