

Real World Smartphones's

Samostalni projekat iz Istrazivanja podataka 1

Autor: Pavle Medic Asistent: Stefan Kapunac Profesor: Nenad Mitic

September 2023

Contents

1	Uvod	2
1.1	Skup podataka	2
1.1.1	Smartphones datoteka	2
1.2	Predprocesiranje	8
1.2.1	Ciscenje podataka - Smartphones datoteka	8
2	Klasifikacija	9
2.1	KNN Algoritam	9
2.2	Stabla odlucivanja	16
3	Klasterovanje	23
3.1	Algoritam K sredina	23
3.2	Sakupjajuce klasterovanje	27
4	Pravila pridruzivanja - SPSS	31
4.1	Apriori algoritam	31
4.2	Pretprocesiranje - Python	31
4.3	SPSS	32
4.3.1	Pokretanje algoritma	32
4.3.2	Rezultati	33
5	Zakljucak	34

1 Uvod

Svrha ovog projekta je demonstracija algoritama

- Klasifikacije (KNN, Algoritam stabla odlucivanja)
- Klasterovanja (Algoritam K-sredina, Algoritam Sakupljajucih Klasterovanja)
- Pravila prifruzivanja u SPSS-u (Apriori algoritam)

Skup podataka na kojem je radjeno moze se naci na linku:

Algoritmi klasifikacije, klasterovanja i predprocesiranja su radjeni u okruzenju Jupyter Notebook, dok su pravila pridruzivanja radjena u IBM SPSS Modeleru

Kod ovog projekta mozete naci na linku:

1.1 Skup podataka

Skup podataka se sastoji od jedne datoteke: smartphones.csv

U nastavku će biti opisana datoteka, zajedno sa promenama koje su učinjene kako bi se prilagodila za upotrebu s algoritmima.

1.1.1 Smartphones datoteka

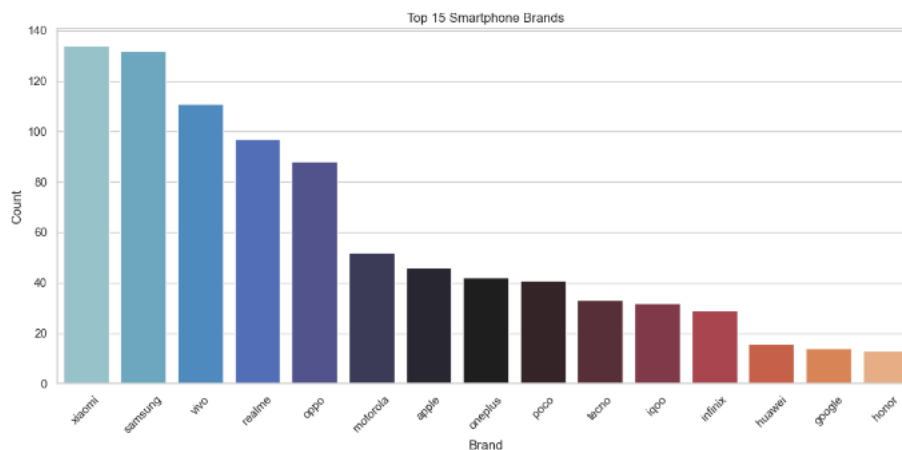
Ovaj skup podataka obuhvata devetsto osamdeset različitih modela mobilnih telefona, zajedno sa dvadeset i dve kolone koje detaljnije opisuju specifikacije.

- *brand_name*: Ime brenda telefona
- *model*: Model telefona
- *price*: Cena telefona
- *avg_rating*: Prosecna ocena
- *5G_or_not*: Da li poseduje mogucnost 5G
- *processor_brand*: Proizvodjac procesora
- *num_cores*: Broj jezgara procesora
- *processor_speed*: Brzina procesora
- *battery_capacity*: Kapacitet baterije
- *fast_charging_available*: Da li poseduje brzo punjenje
- *fast_charging*: Jacinu brzog punjenja

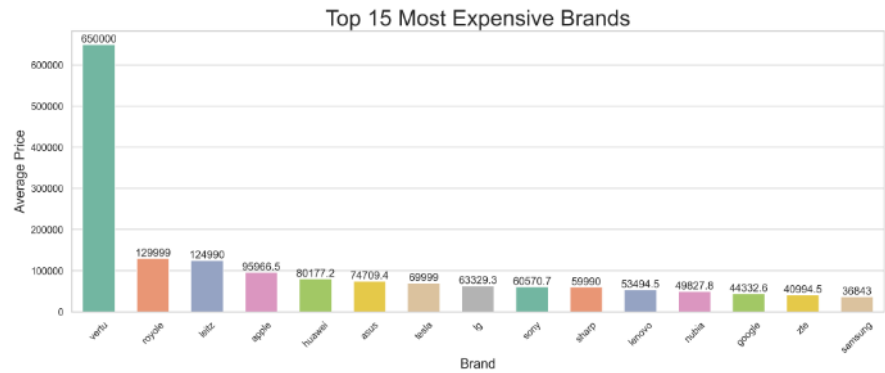
- *ram_capacity*: Kapacitet ram memorija
- *interna_memory*: Kapacitet interne memorija
- *screen_size*: Velicina ekrana
- *refresh_rate*: Brzina osvezavanja ekrana
- *num_rear_cameras*: Koliko poseduje zadnjih kamera
- *os*: operativni sistem
- *primary_camera_rear*: Jacina zadnje kamere
- *primary_camera_front*: Jacina prednje kamere
- *extended_memory_available*: Da li dozvoljava eksternu memoriju
- *resolution_height*: Visina slike
- *resolution_width*: Sirina sirina

Tokom analize ovog skupa podataka saznali smo sledece:

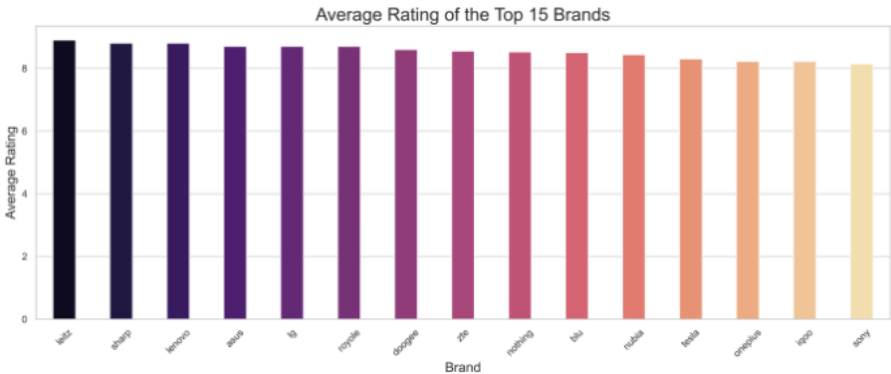
- U skupu podataka prisutno je 46 razlicitih proizvajaca mobilnih telefona.
- Brendovi sa najvećim brojem modela.



- Brendovi sa najvišim prosečnim cenama.

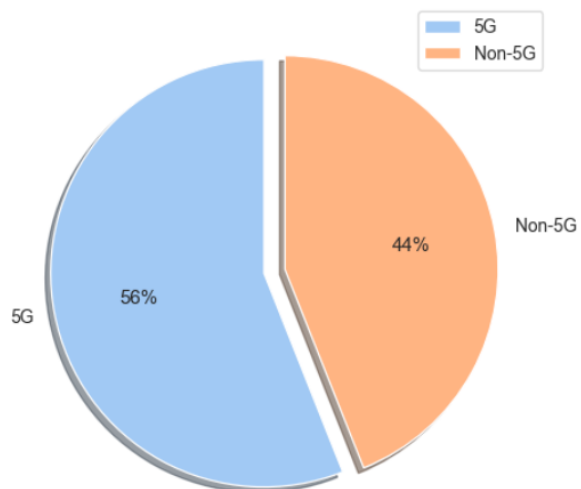


- Brendovi sa najvišim prosečnim ocenama.

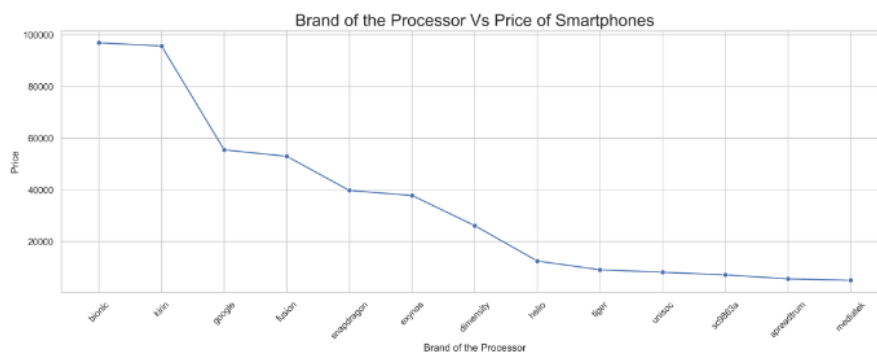


- Procenat kompatibilnosti sa 5G mrežom.

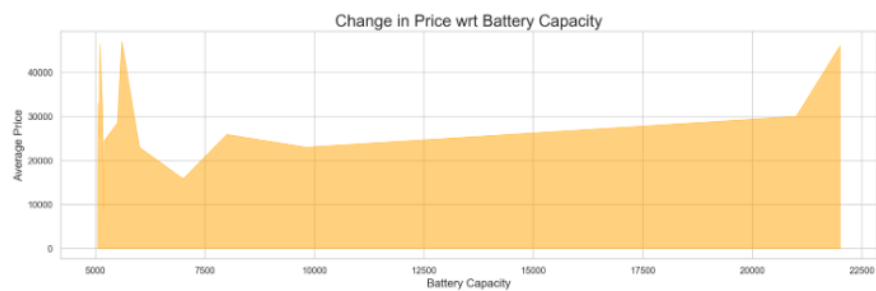
5G vs. Non-5G Smartphones



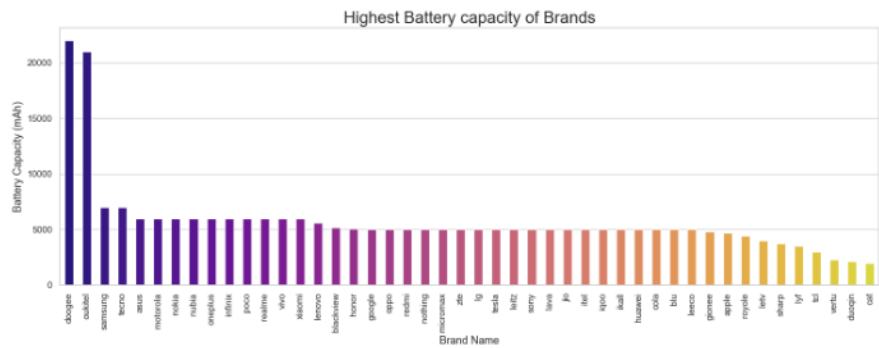
- Veza izmedju procesora i cene.



- Odnos izmedju kapaciteta baterije i cene.

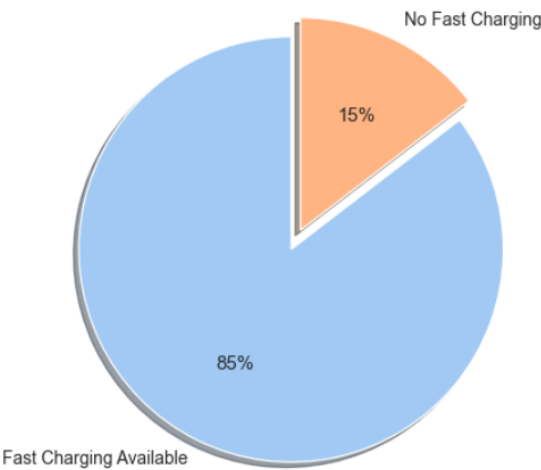


- Brendovi sa najvećim kapacitetom baterije.

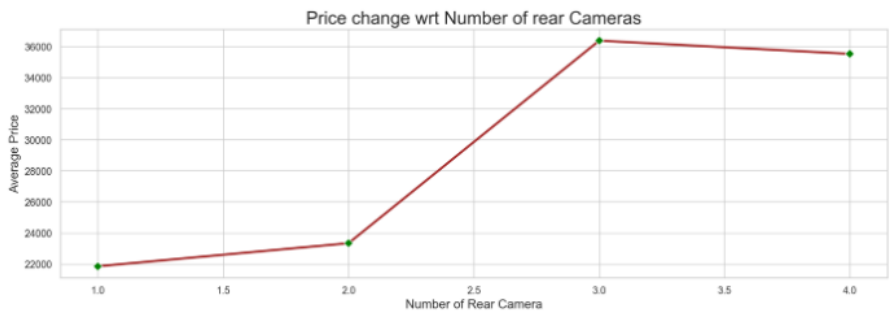


- Procenat kompatibilnosti sa brzim punjenjem.

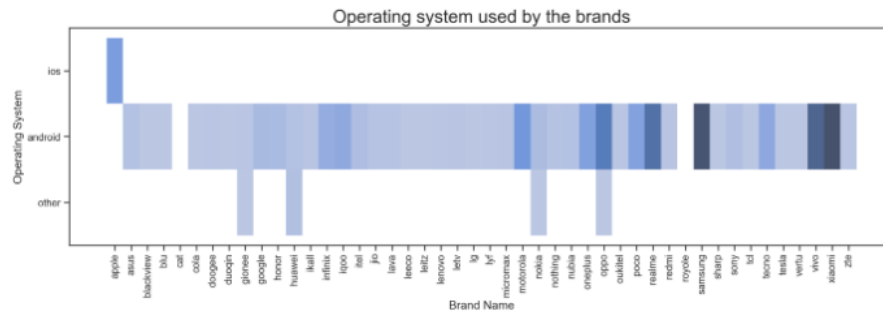
Fast Charging Availability in Smartphones



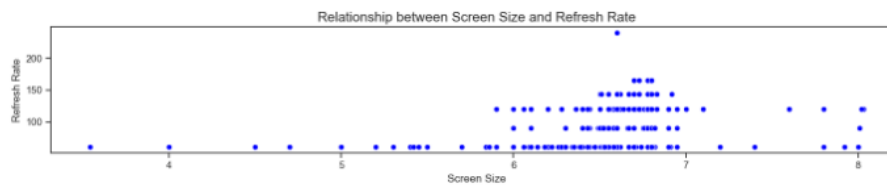
- Fluktuacija cena u zavisnosti od broja zadnjih kamera.



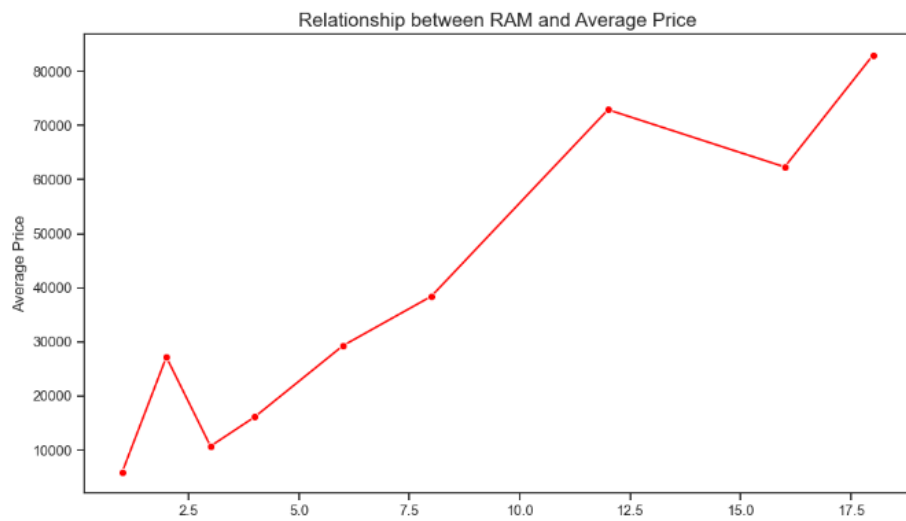
- Zapažamo da je Android najrašireniji operativni sistem.



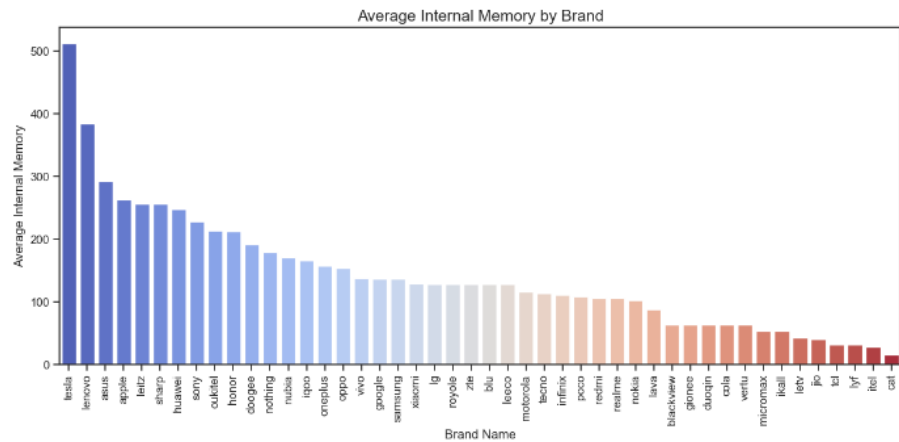
- Povezanost izmedju dimenzija ekrana i brzine osvežavanja.



- Odnos izmedju RAM memorije i cene.



- Brendovi sa najvećim unutrašnjim skladištem.



1.2 Predprocesiranje

1.2.1 Ciscenje podataka - Smartphones datoteka

U skupu podataka naišli smo na određeni broj nedostajućih vrednosti

```

brand_name      0
model           0
price           0
avg_rating      101
5G_or_not       0
processor_brand  20
num_cores       6
processor_speed  42
battery_capacity 11
fast_charging_available 0
fast_charging   211
ram_capacity     0
internal_memory  0
screen_size     0
refresh_rate    0
num_rear_cameras 0
os              14
primary_camera_rear 0
primary_camera_front 5
extended_memory_available 0
resolution_height 0
resolution_width 0
dtype: int64

```


Umesto da se brišu podaci koji imaju nedostajuće vrednosti, većina ovih polja je zamenjena sa srednjom vrednošću, i to polja za atribut *avg_rating*, *processor_speed*, *num_cores*, *battery_capacity*, i *primary_camera_front*.

Dok su atributi *processor_brand* i *os* zamenjeni modom. Moda je vrednost koja se najčešće ponavlja u skupu podataka za taj atribut. Atribut *fast_charging* je zamenjen random vrednošću. Ovakva odluka je doneta zbog ograničenog broja instanci u skupu podataka.

2 Klasifikacija

Klasifikacija predstavlja izazov koji podrazumeva pronalaženje funkcije f koja mapira skup X na unapred definisane klasne oznake Y .

Obično se ta ciljna funkcija naziva klasifikacionim modelom.

U procesu klasifikacije, podaci se često dele na tri skupa: trening skup, skup za validaciju i testni skup. U nekim slučajevima, skup za validaciju može biti izostavljen.

Trening skup se koristi za obuku modela koji ćemo kasnije primeniti za klasifikaciju.

Test skup se koristi kako bismo procenili koliko je naš model dobar u obradi podataka koje prethodno nije video.

Skup za validaciju igra ključnu ulogu u izboru optimalnih parametara za model ili čak u izboru samog modela.

2.1 KNN Algoritam

KNN (K najbližih suseda) algoritam je jedan od najpoznatijih algoritama za klasifikaciju.

Za primenu ovog algoritma potrebni su nam sledeći elementi:

- Trening skup podataka sa čuvanim instancama.
- Metrika koja meri rastojanje izmedju instanci (obično se koristi Euklidsko rastojanje).
- Parametar "k" koji određuje broj suseda koje želimo da uzmemo u obzir.

Opis rada algoritma:

- Izračunavamo rastojanje izmedju test instance i svih instanci u trening skupu.

- Odabiremo "k" najbližih suseda.
- Klasifikujemo test instancu na osnovu glasanja medju "k" najbližih suseda.

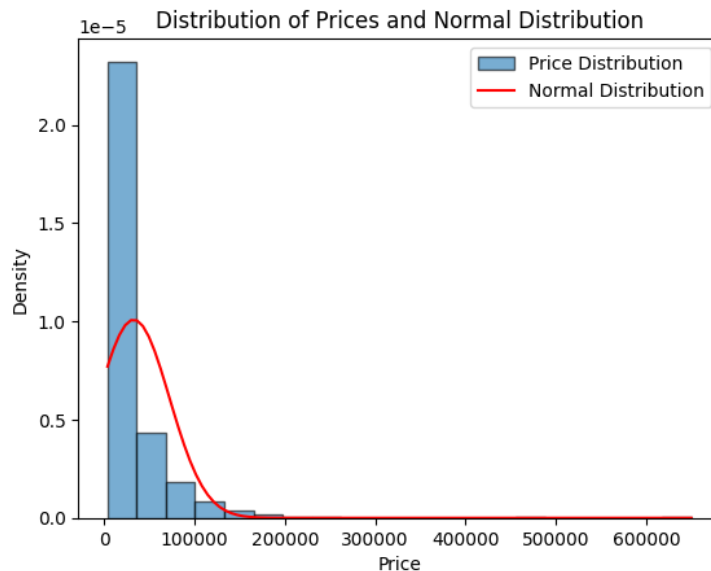
Ovaj algoritam se koristi za klasifikaciju na osnovu sličnosti izmedju instanci. Kao što smo ranije istakli, iako smo sproveli opšti proces pretprocesiranja, potrebno je da svaki element podlegne specifičnom pretprocesiranju.

Prvo vršimo analizu cena ("price") podataka. Atribut 'price' predstavlja ciljni atribut i karakteriše se neprekidnim vrednostima. S obzirom na našu nameru da sprovedemo klasifikaciju, a ne regresiju, postoji potreba da prethodno izvršimo diskretizaciju ovog atributa. Da bismo doneli ispravnu odluku o tome kako ga najbolje diskretizovati, prvi korak koji ćemo preduzeti jeste analiza raspodele vrednosti atributa 'price'.

Izračunavamo srednju vrednost cena (mean_price) i standardnu devijaciju cena (std_price) iz kolone "price".

Sledeći korak je kreiranje niza vrednosti ('x') koji predstavljaju opseg cena iz skupa podataka, a zatim se računaju vrednosti normalne distribucije ('normal_dist') za taj opseg. Ovo se radi kako bi se prikazala očekivana normalna raspodela za cene sa istim parametrima srednje vrednosti i standardne devijacije kao i stvarni podaci.

Ovaj grafik omogućava vizualno poredjenje stvarne raspodele cena sa očekivanom normalnom raspodelom, što može biti korisno u statističkoj analizi i proceni podataka.



Izračunavamo merenja asimetrije (eng. skewness) za kolonu "price". Vrednost merenja asimetrije daje informaciju o tome kako je raspodela podataka asimetrična ili simetrična.

Vrednost merenja asimetrije koja iznosi 6.591790999665569 ukazuje na to da raspodela cena u koloni "price" ima pozitivnu asimetriju. Pozitivna asimetrija znači da je rep raspodele sa višim vrednostima cena (ekstremima) produžen na desnu stranu distribucije, dok je glavni deo raspodele sa nižim vrednostima cena koncentrisan na levu stranu.

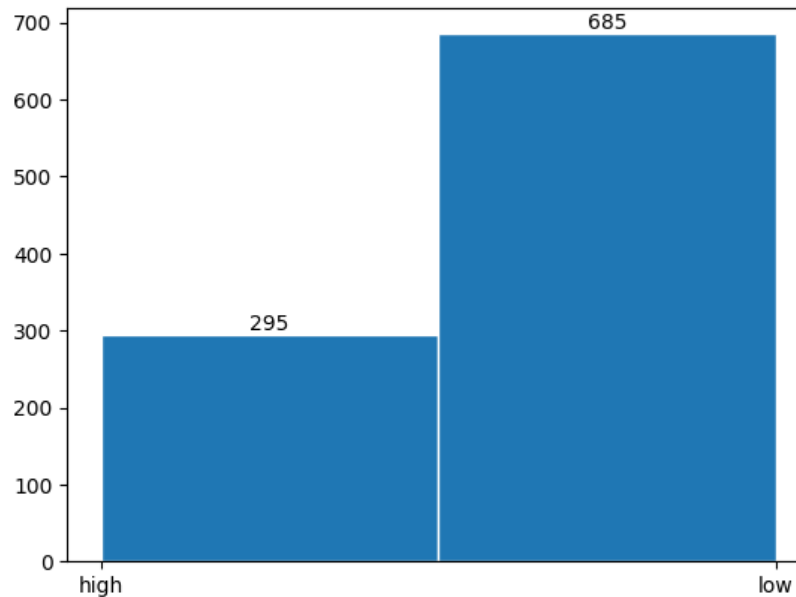
U ovom kontekstu, to znači da postoji mali broj telefona sa veoma visokim cenama koji "izvlače" srednju vrednost cena na višu vrednost u odnosu na srednju vrednost idealno simetrične raspodele. Ovo je korisna informacija jer ukazuje na to da skup podataka ima izražene ekstreme u cenama telefona. Naglasimo da ovo nisu outliers u klasičnom smislu, ove cene telefona, za date marke su cene koje su validne.

Atribut *price* je diskretizovan u 2 kategorije, odnosno binarizovan. Binarizacija je proces pretvaranja kontinuiranih atributa u binarne (dve vrednosti) kako bi se olakšala analiza ili klasifikacija podataka. Dobijene kategorije su *low* i *high price*. Zbog gore opisane raspodele ovog atributa, podaci su podeljeni u odnosu 7 na 3, u korist *low*. Dobijena je nova kolona *price_category*.

Razdvajamo kolonu 'os' na više kolona na osnovu prisutnih operativnih sistema u toj koloni (vrednosti u ovim kolonama mogu biti 0 ili 1). Na koloni 'os' je izvršena binarizacija.

Zatim pripremamo podatke za treniranje modela tako da odvajamo ulazne karakteristike od ciljane varijable i prati nazive kolona koje se koriste kao ulazne karakteristike.

Rezultirajući histogramom koji prikazuje distribuciju vrednosti u ciljnoj varijabli 'price_category' sa brojem pojavljivanja iznad svake korpe na histogramu



Zatim razdvajamo podatke na test i trening skupove
Definišmo funkciju IQR (Interquartile Range) koja se koristi za analizu distribucije podataka i identifikaciju ekstremnih vrednosti (outliers) u okviru različitih karakteristika (features) u vašem skupu podataka
Ova funkcija prima dva argumenta:

- data: DataFrame koji sadrži podatke koje želite analizirati.
- feature_names: Lista naziva karakteristika (features) koje želite analizirati.

Funkcija se zatim izračunava i vraća DataFrame iqr sa sledećim informacijama za svaku karakteristiku:

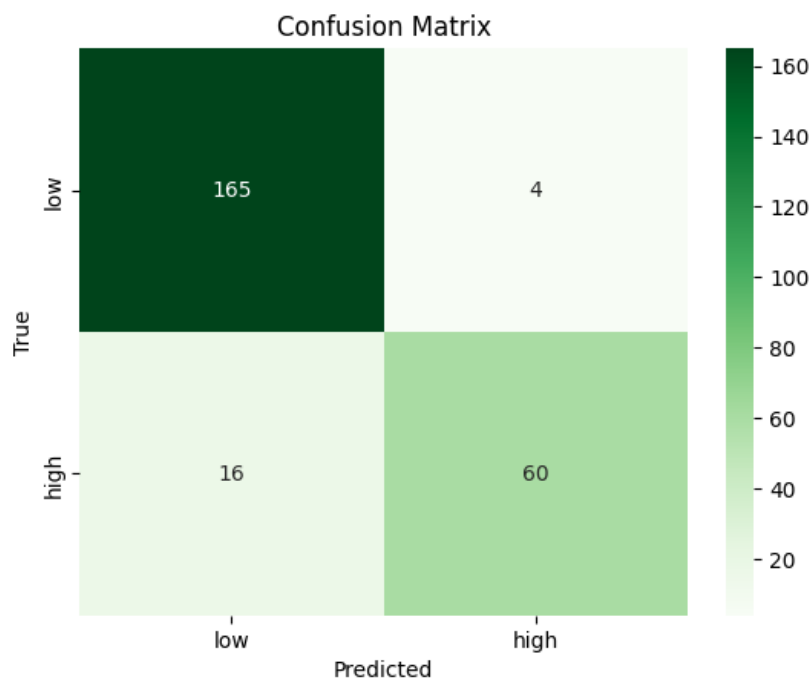
- lower: Donja granica za detekciju ekstremnih vrednosti (outliers).
- min: Minimum vrednosti u karakteristici.
- num_lower: Broj ekstremnih vrednosti ispod donje granice.
- upper: Gornja granica za detekciju ekstremnih vrednosti.
- max: Maksimum vrednosti u karakteristici.
- num_upper: Broj ekstremnih vrednosti iznad gornje granice.
- percentage: Procenat ekstremnih vrednosti u odnosu na ukupan broj podataka.

Zatim vršimo skaliranje koje može biti korisno za pripremu podataka kako bismo dobili tačnije i pouzdanije rezultate.

Treniramo algoritam nad trening podacima

Kada se trening izvrši, KNN model će "naučiti" kako da klasifikuje ili predviđa ciljnu varijablu na osnovu trening podataka. Nakon treniranja, model će biti sposoban da primeni svoje znanje na nove, nevidjene podatke kako bi donosio predikcije.

Na test i trening skupu dobijena tačnost je 92%. Matrica konfuzije, možemo videti na narednom grafiku.



GridSearch

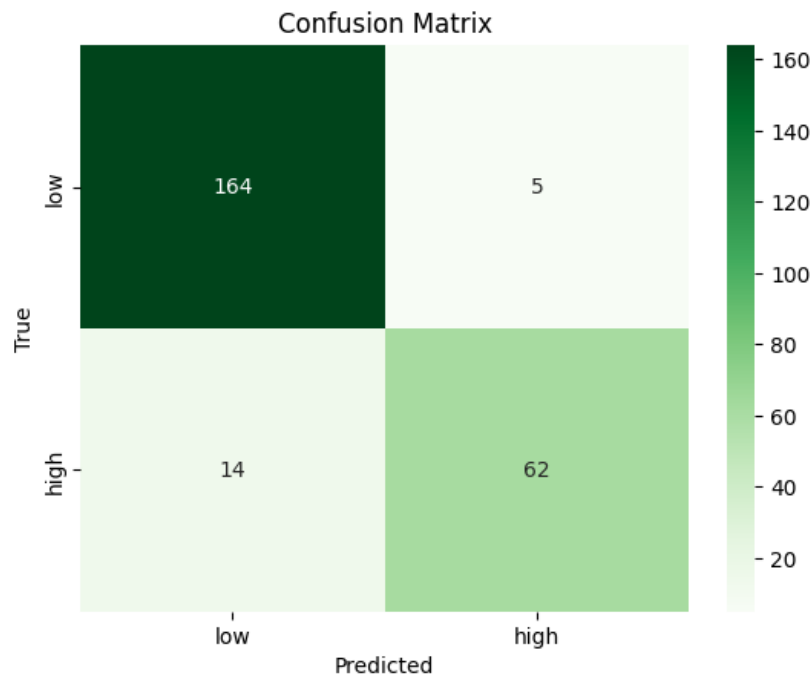
Postavljamo parametre za K-nearest neighbors (KNN) model. Ovi parametri će se koristiti u procesu pretrage parametara (grid search) kako bi se pronašla najbolja kombinacija za ovaj model. Grid search će ispitati različite kombinacije parametara i odabrati one koje daju najbolje performanse za dati skup podataka i problem.

GridSearch je dao najbolje rezultate, za sledeću kombinaciju parametra:

```
[72]: estimator.best_params_
```

```
[72]: {'algorithm': 'auto',  
      'leaf_size': 10,  
      'metric': 'manhattan',  
      'n_neighbors': 10,  
      'p': 1,  
      'weights': 'distance'}
```

Dobijena je sledeca matrica konfuzije

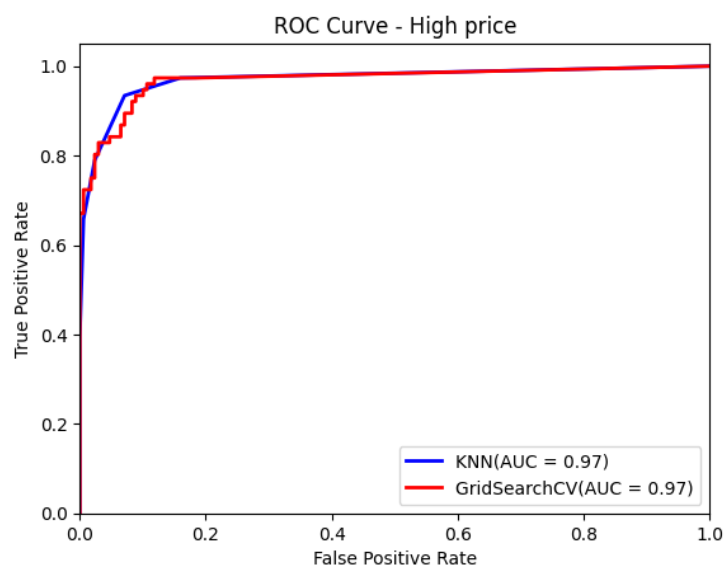
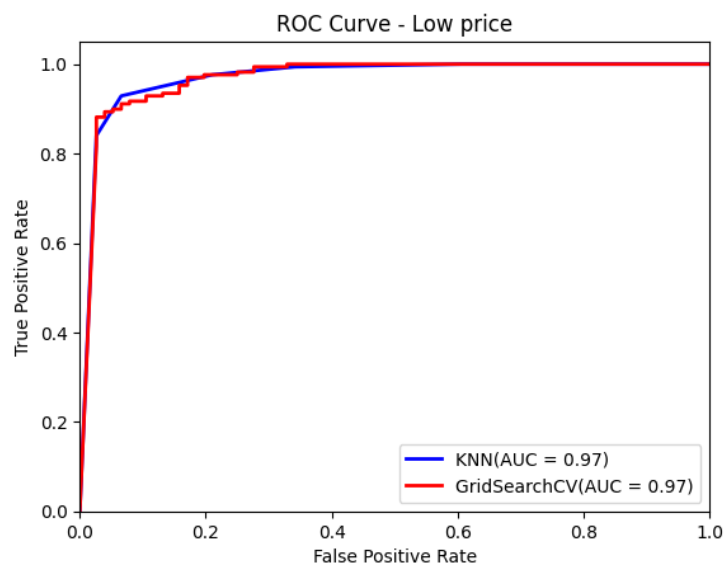


Ovaj model, takodje daje tacnost 92% na test skupu, nema velike razlike u odnosu na KNN model sa podrazumevanim podacima - u kontekstu tacnosti. Da bismo uporedili KNN model sa podrazumevanim parametrima i model dobijen putem GridSearch-a, koristicemo ROC krivu kako bismo vizuelno prikazali njihovo ponašanje i razlike.

ROC (Receiver Operating Characteristic) kriva je grafički prikaz performansi klasifikacionog modela, koji se koristi za analizu sposobnosti modela da razlikuje izmedju dve klase (obično pozitivne i negativne). ROC kriva ima dve osnovne ose:

- X-osa: False Positive Rate (FPR) - Odnos lažno pozitivnih rezultata u odnosu na ukupan broj stvarno negativnih primera.

- Y-osa: True Positive Rate (TPR), takodje poznat kao Recall ili Sensitivity - Odnos tačno pozitivnih rezultata u odnosu na ukupan broj stvarno pozitivnih primera.



2.2 Stabla odlucivanja

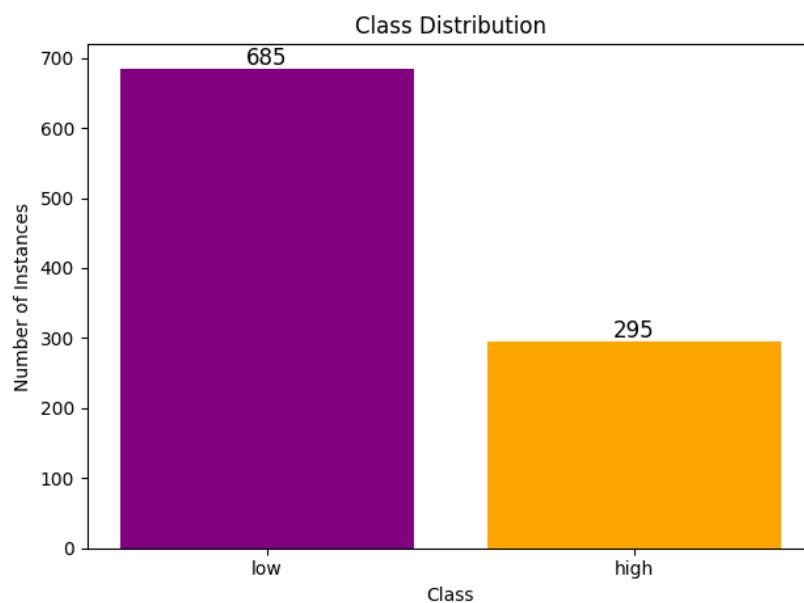
Stabla odlučivanja su često korišćena metoda za rešavanje problema klasifikacije. Ovaj metod pripada kategoriji nadgledanog učenja i funkcioniše tako što gradi model odluka u obliku drveta, gde svaki čvor predstavlja odluku, a svaka grana označava mogući ishod te odluke.

Proces konstrukcije stabla odlučivanja je rekurzivan i uključuje particionisanje podataka na osnovu vrednosti ulaznih atributa. Pri svakom čvoru u stablu, algoritam bira karakteristiku koja najefikasnije razdvaja podatke u različite klase i deli podatke na odgovarajući način, smanjujuci necistocu, odnosno gresku pri svakoj podeli.

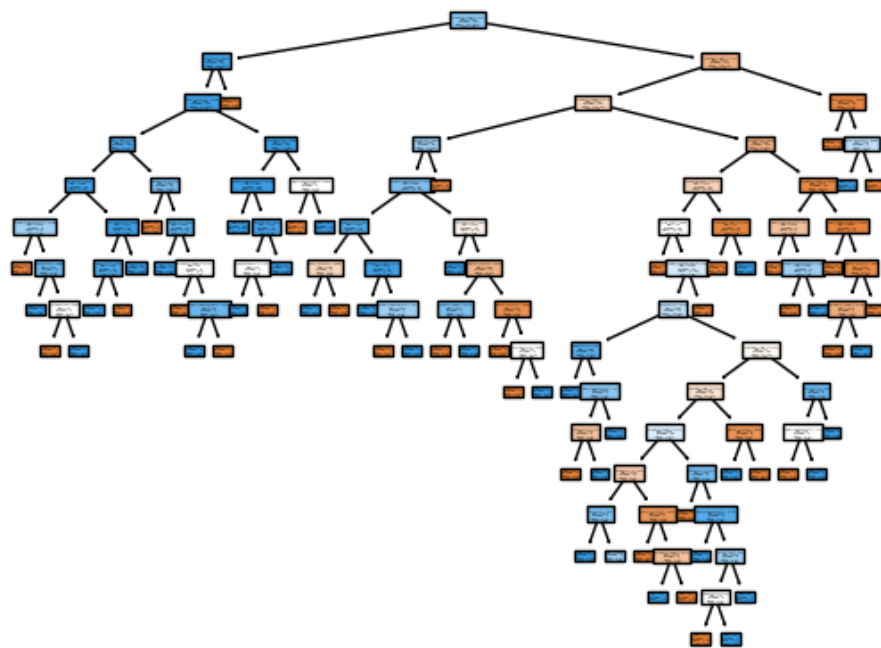
Predprocesiranje podataka za ovaj model obavlja se na isti način kao i za KNN model.

Ciljna promenljiva je, kao i u KNN algoritmu 'price_category'.

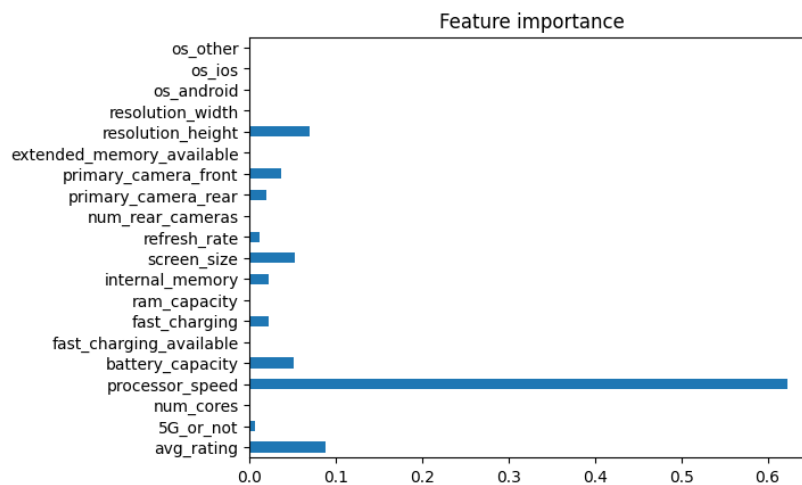
Na narednom grafiku, mozemo videti kako su podaci rasporedjeni za atribut price_category.



Postavljamo i treniramo drvo odlucivanja za dati trening skup podataka i generisemo predvidjanja na testnom skupu podataka koristeći naše naučeno drvo odlucivanja i zatim vizualizujemo stablo.



Mozemo videti koja svojstva imaju najveći uticaj na donošenje odluka naseg drveća odlučivanja, na narednom grafiku.



Dobijeni su sledeci rezultati:

Classification report for model DecisionTreeClassifier on training data

	precision	recall	f1-score	support
high	1.00	1.00	1.00	219
low	1.00	1.00	1.00	516
accuracy			1.00	735
macro avg	1.00	1.00	1.00	735
weighted avg	1.00	1.00	1.00	735

Confusion matrix for model DecisionTreeClassifier on training data

	high	low
high	218	1
low	0	516

Mozemo primetiti da je doslo do preprilagodjavanja na trening skupu. Iako se na slici ispod moze videti da je velika tacnost modela na test skupu, ovaj model nije dobar.

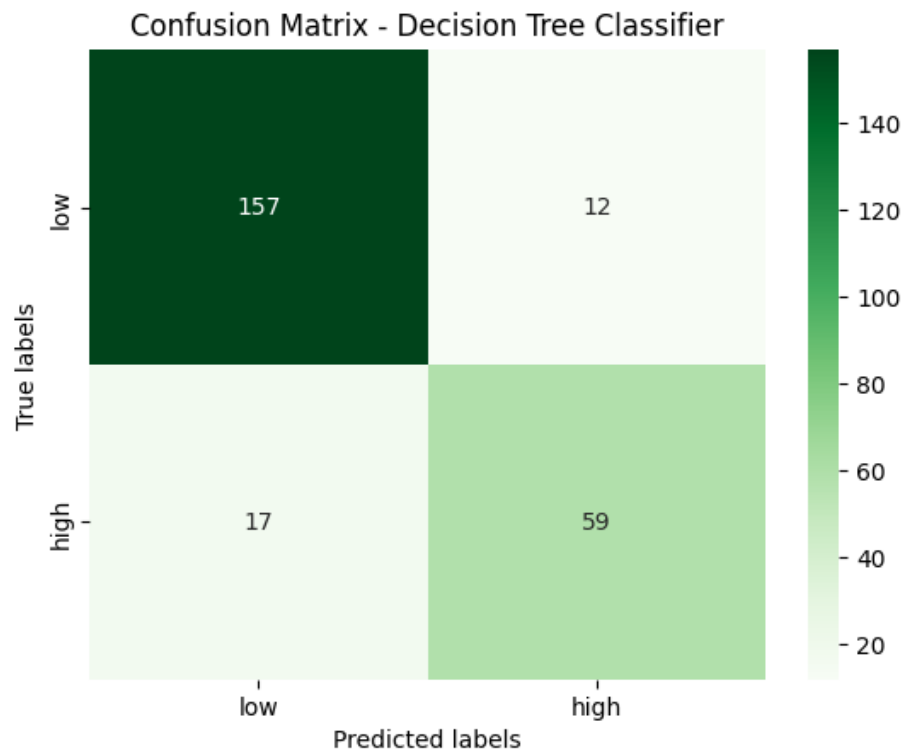
Classification report for model DecisionTreeClassifier on test data

	precision	recall	f1-score	support
high	0.83	0.78	0.80	76
low	0.90	0.93	0.92	169
accuracy			0.88	245
macro avg	0.87	0.85	0.86	245
weighted avg	0.88	0.88	0.88	245

Confusion matrix for model DecisionTreeClassifier on test data

	high	low
high	59	17
low	12	157

Generisanje i vizualizacija matrice konfuzije za analizu preformansi naseg klasifikacionog modela, koja nam omogucava da vidimo koliko tacno i netacno model klasifikuje instance u svakoj klasi



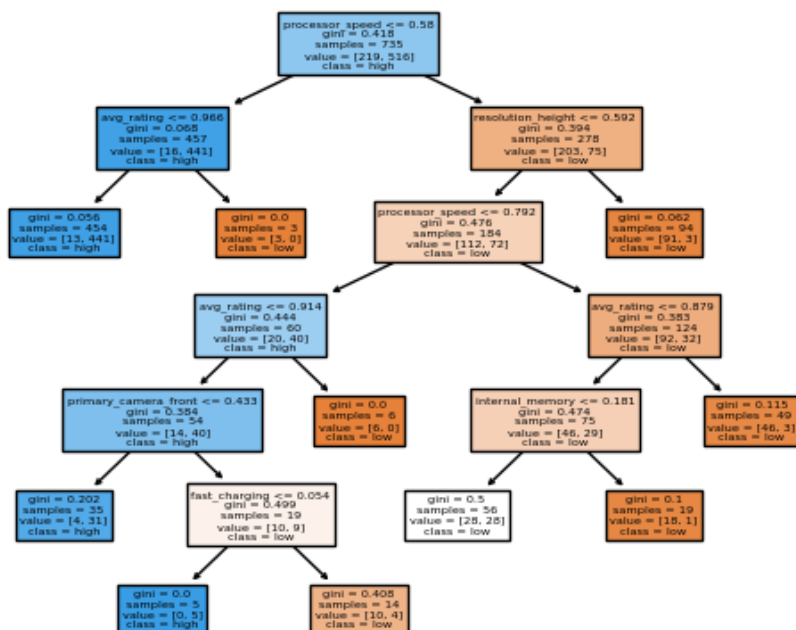
Posto je doslo do preprilagodjavanja, pokrenucemo GridSearch da bi nasli parametre, za koje cemo dobiti bolji model. Na slici ispod mozemo videti koji su to najbolji parametri.

```
estimator.best_params_
```

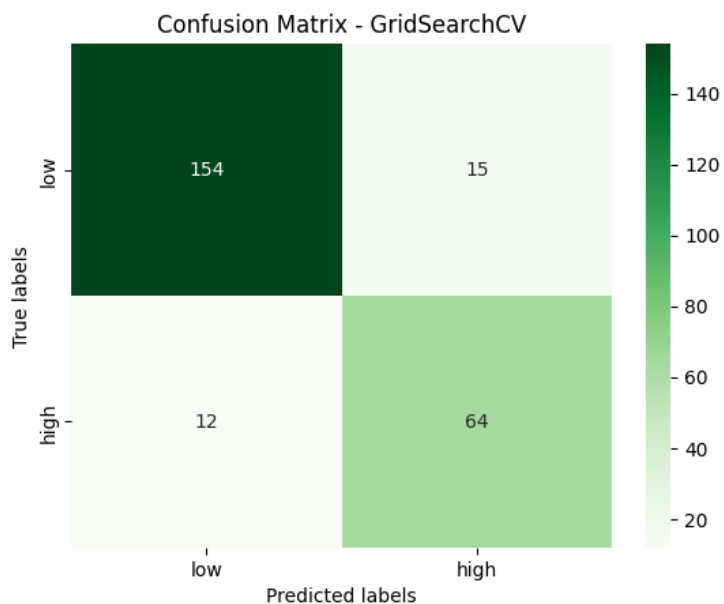
```
{'ccp_alpha': 0.0,  
 'criterion': 'gini',  
 'max_depth': 8,  
 'max_leaf_nodes': 10,  
 'min_impurity_decrease': 0.0,  
 'min_samples_leaf': 1,  
 'min_samples_split': 2}
```

Na trening skupu dobijamo tacnost 92%, a na test skupu 89%, zakljucujemo da jeGridSearch je dao bolji model.

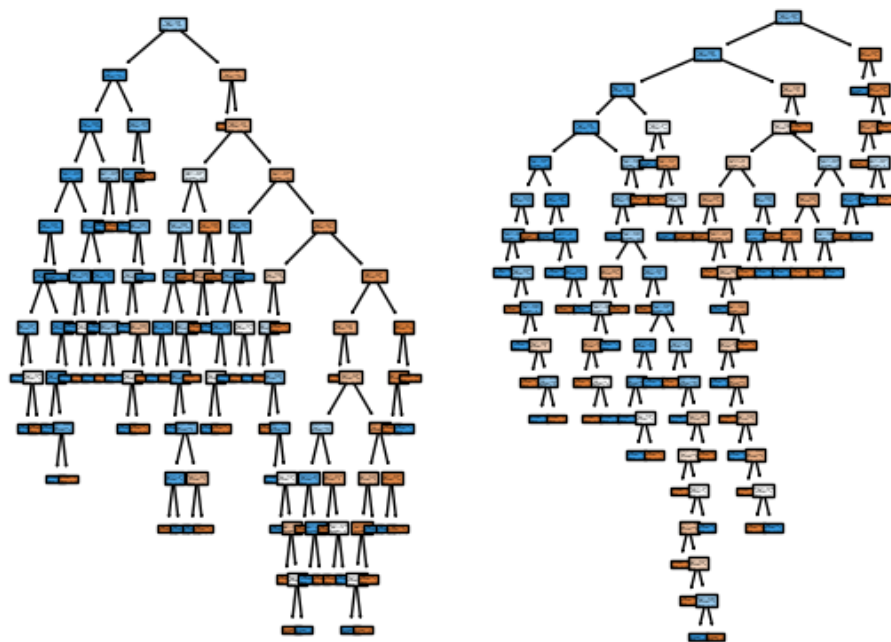
Na narednoj slici mozemo videti, kako izgleda drvo odlucivanja za dobijeni model.



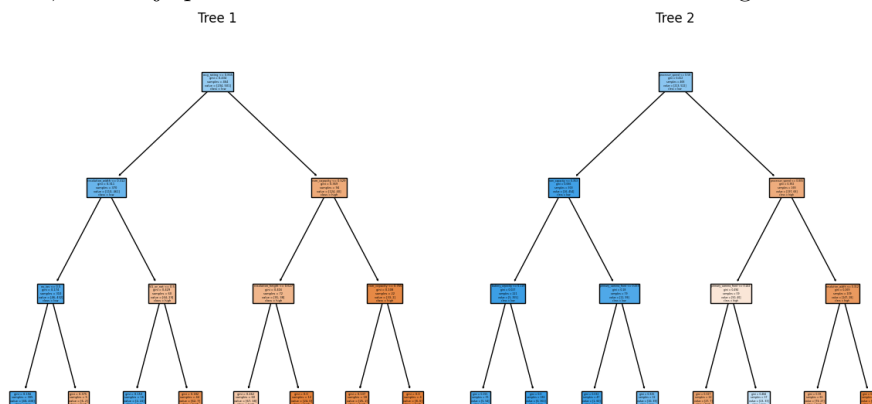
Dok matrica konfuzije izgleda, ovako:



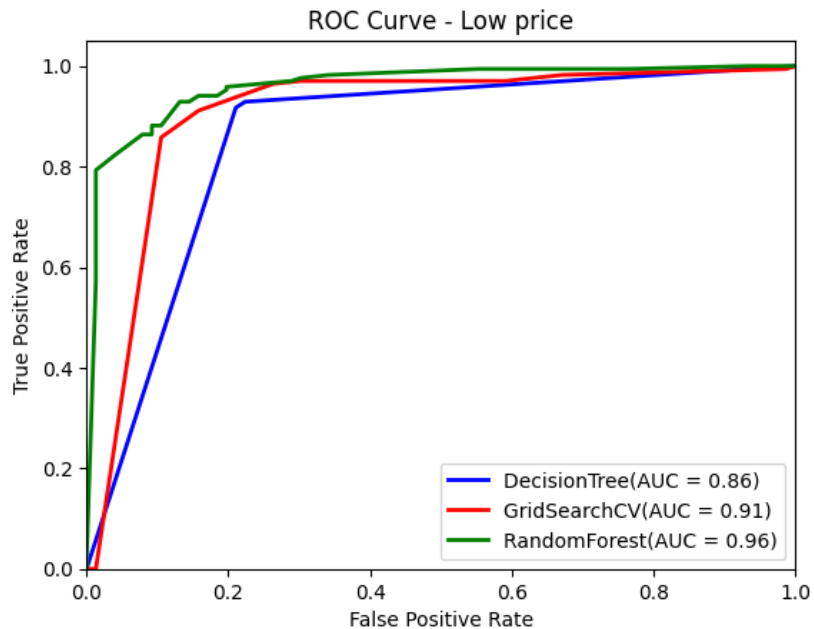
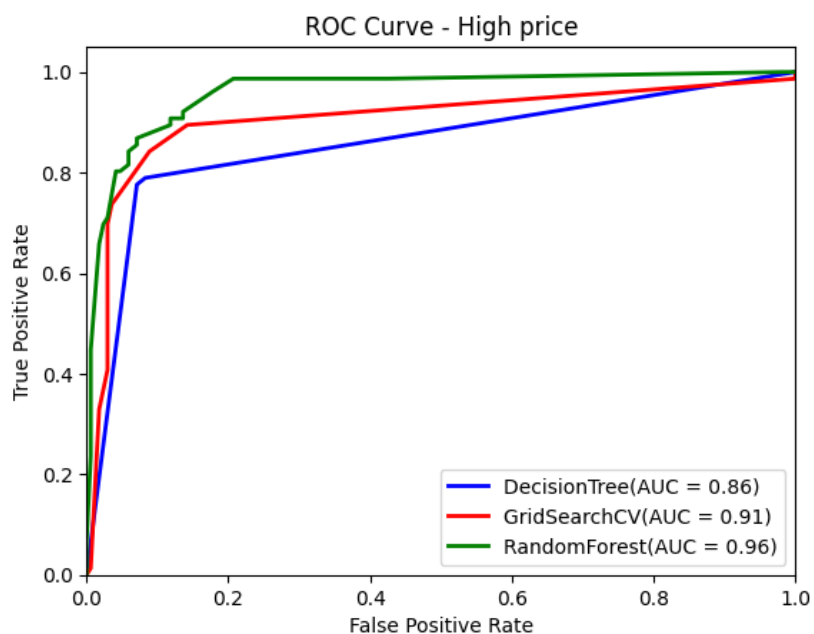
Takodje mozemo videti kakve rezultate daje RandomForest klkasifikator, koji na test skupu ima tacnost 90%.



Rezultat RandomForest klasifikatora sa dodatim parametrom za maksimalnu dubinu 3, nam daje preciznosi 91%. Mozemo videti vizuelno kako izgleda



Ova tri modela, najbolje mozemo uporedi na ROC krivi. Posmatramo koji model ima najveću površinu ispod njegove krive.



RandomForest klasifikator ima najveću površinu ispod ROC krive, takodje i najveću tačnost - najbolji je od ova tri modela.

3 Klasterovanje

Klasterovanje je tehnika koja se koristi kako bi se grupisali slični objekti ili podaci na osnovu njihovih karakteristika. Cilj klasterovanja je identifikovanje prirodnih grupa ili obrazaca unutar skupa podataka, pri čemu su objekti unutar iste grupe sličniji jedni drugima nego objektima u drugim grupama.

U klasterovanju, algoritam analizira podatke i dodeljuje objekte klasterima na osnovu njihove sličnosti. Sličnost između objekata određuje se uzimajući u obzir karakteristike atributa. Često korišćeni algoritmi klasterovanja uključuju K-sredina, hijerarhijsko klasterovanje i DBSCAN (klasterovanje na osnovu gustine prostornih podataka sa šumom)

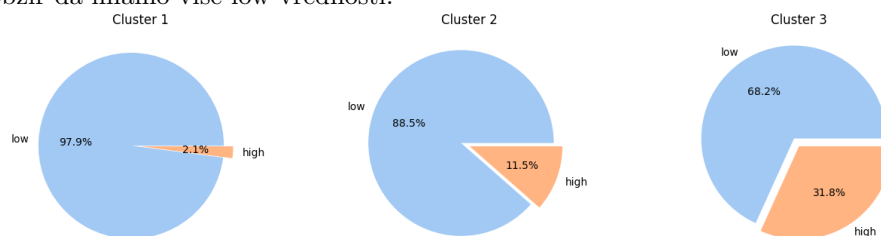
3.1 Algoritam K sredina

K-sredina (K-means) je jedan od najčešće korišćenih algoritama klasterovanja u mašinskom učenju i analizi podataka. Cilj K-sredina algoritma je da minimizuje zbir kvadratnih udaljenosti između tačaka podataka i njihovih odgovarajućih klaster centara.

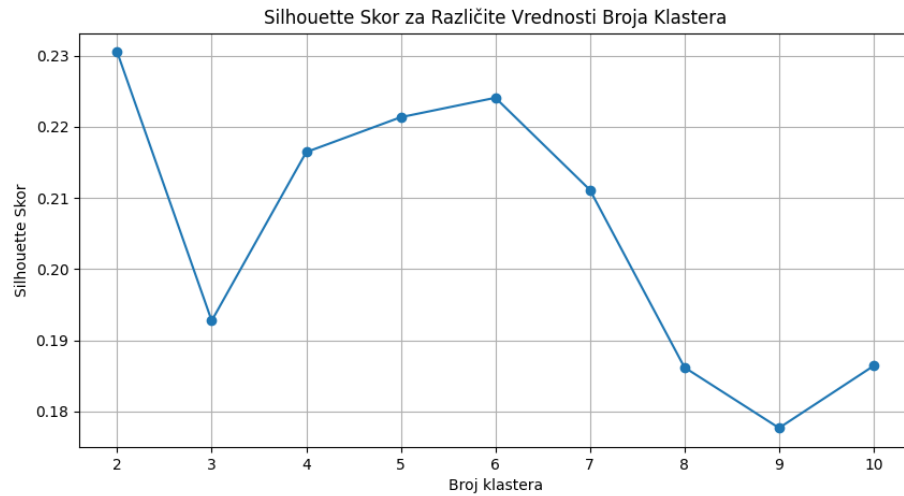
Cena telefona je podeljena na dve kategorije, low i high, kao i ranije.

KMeans algoritam, čemo za početak pokrenuti, za 3 klastera.

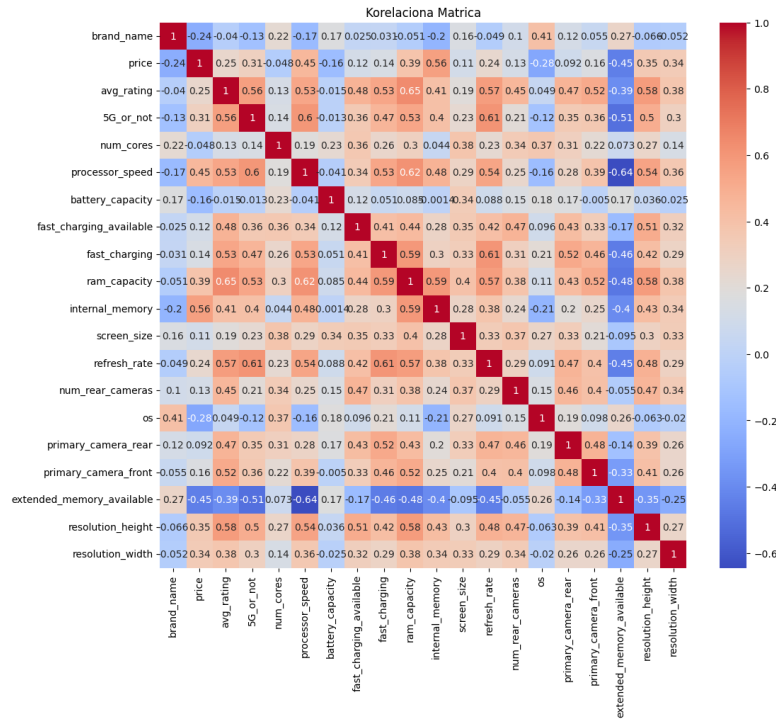
Na narednom grafiku, kakav je odnos high i low cena u svakom klasteru. Moramo u obzir da imamo više low vrednosti.



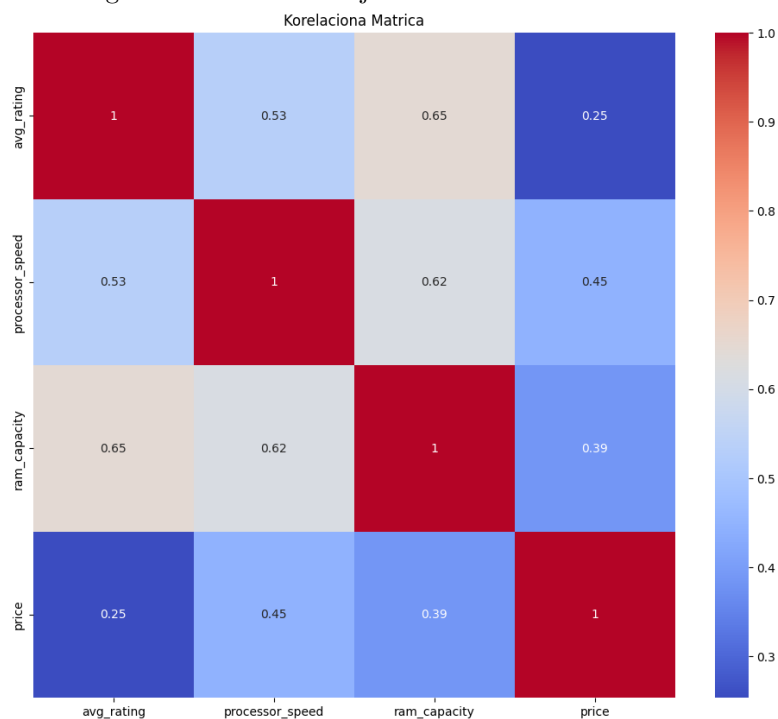
Na narednom grafikonu, možemo videti odnos silhouette skora i broja klastera. Silhouette score je statistička mera koja se koristi za procenu kvaliteta klasterovanja podataka. Ova mera omogućava procenu koliko su instance unutar istog klastera slične jedna drugoj u poređenju sa instancama u drugim klasterima. Možemo primetiti da je silhouette skor nizak. U ovakvom skupu podataka, najbolje je podatke podeliti u dva klastera.



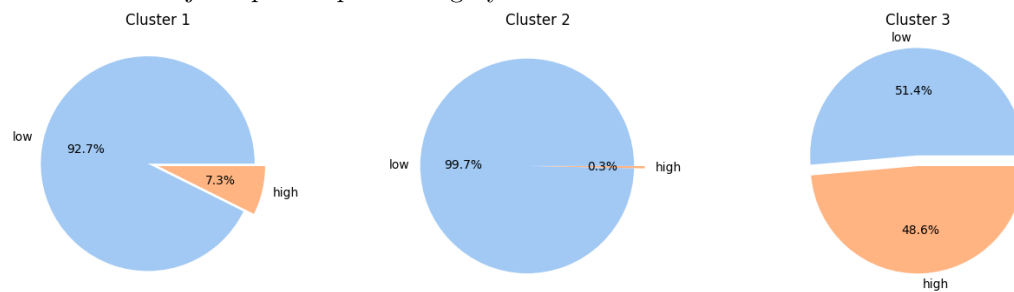
Korelaciona matrica je koristan alat za analizu kako su različite osobine u podacima međusobno povezane. Visoke vrednosti korelacionih koeficijenata (bliske 1) ukazuju na jaku pozitivnu korelaciju izmedju osobina, dok niske vrednosti (bliske -1) ukazuju na jaku negativnu korelaciju. Vrednosti bliske 0 ukazuju na slabo ili nema korelacije izmedju osobina. Prikazivanje ovih veza pomoću heatmap-a može pomoći u identifikaciji uzoraka u podacima i donošenju informisanih odluka u analizi podataka.



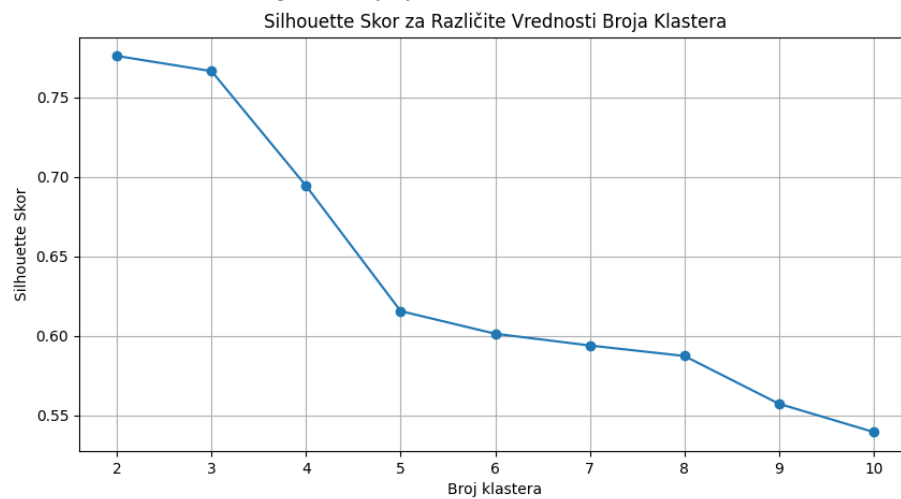
Uvidjamo da naredni atributi imaju veliku korelaciju, iz skupa podataka biramo da radimo samo nad njima. To su: avg_rating, processor_speed, ram_capacity, price. Očekujemo da će podaci biti podeljeni u bolje klastere. Možemo videti, kako sada izgleda matrica korelacije:



Ako sada podelimo ovakav skup podataka u tri klastera, možemo videti da je unutar klastera bolja raspodela price_category.



Na narenom grafiku, mozemo videti kako se silhouete skor povecao, kada smo uzeli korelisane atribute. Opet dobijamo da je najbolje da podelimo podatke u 2 klastera, ali sada sa mnogo znacajnijim silhouete skorom.



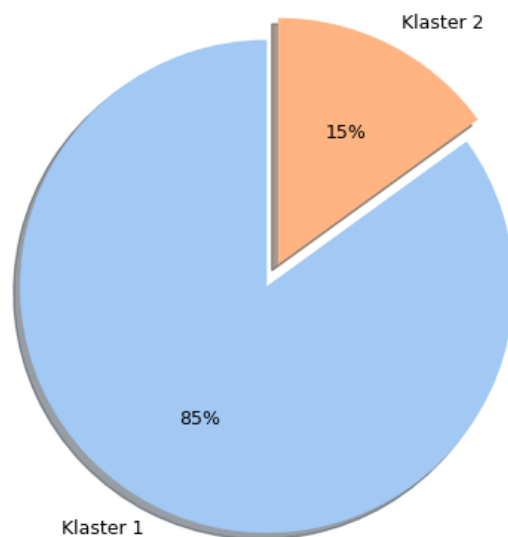
3.2 Sakupljajuće klasterovanje

Sakupljajuće klasterovanje (Agglomerative Clustering) je algoritam hijerarhijskog klasterovanja koji se koristi za grupisanje sličnih tačaka podataka u klasterima. Ovo je pristup odozdo prema gore (bottom-up), gde svaka tačka podataka počinje kao sopstveni klaster, a zatim se iterativno spajaju klasteri na osnovu njihove sličnosti sve dok se ne dostigne željeni broj klastera. Pretprocesiranje skupa podataka i priprema za ovaj algoritam je identična kao za algoritam K sredina

Radimo Min-Max skaliranje, da bismo osigurali da karakteristike neće imati neprikladan uticaj na konačan rezultat analize podataka

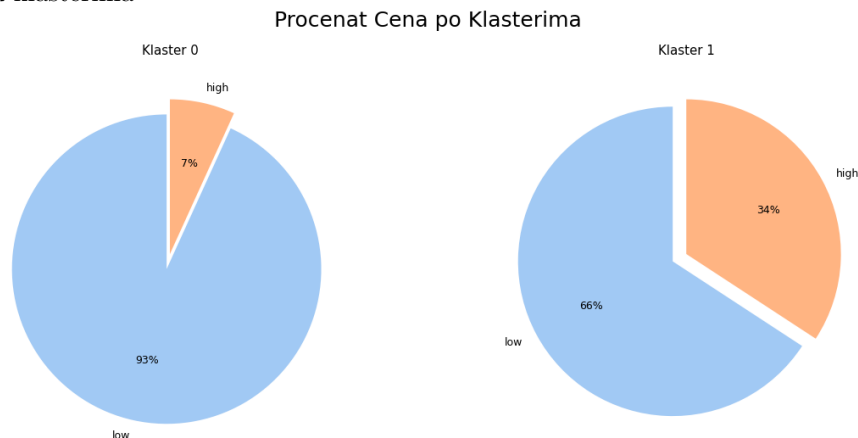
Za pocetak, podelicemo podatke u dva klastera. Klasteri će se formirati hijerarhijski, počevši od pojedinačnih tačaka i spajajući ih postepeno u veće grupe.

Broj Instanci po Klasterima



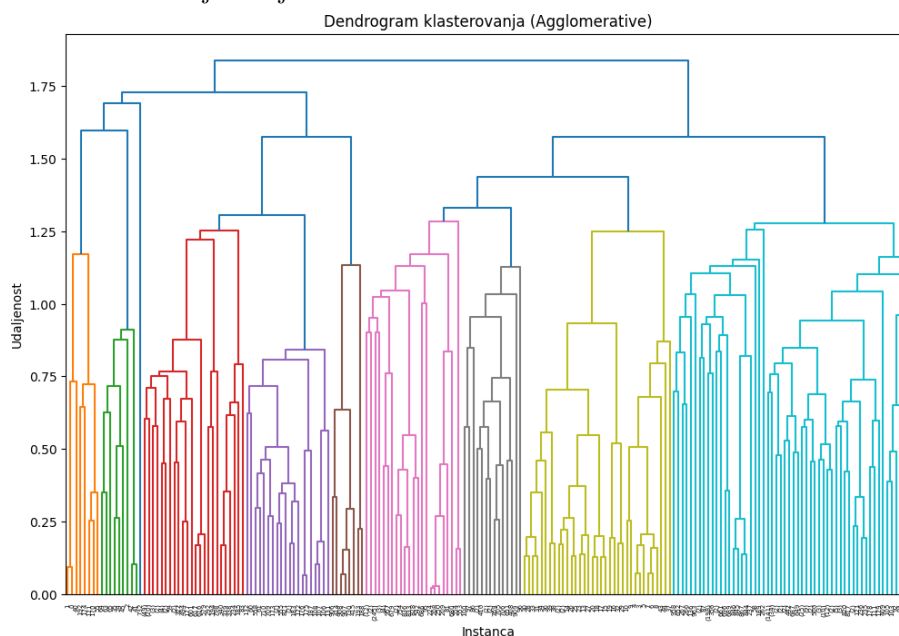
Mozemo primetiti da je jedan klaster dosta veci u odnosu na drugi.

Bavimo se i analizom cena proizvoda, mozemo videti procentualnu raspodelu cena u klasterima

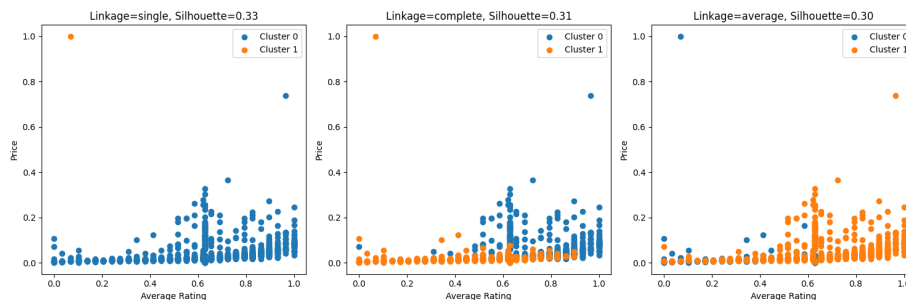


Posmatramo silhouette skor za ocenu kvaliteta klasteriranja, rezultat će biti vrednost izmedju -1 i 1, gde veće vrednosti ukazuju na bolje klasteriranje. Ova vrednost nam pomaže da procenimo kvalitet klasteriranja koje je izvršeno modelom. Nas rezultat je 0.30

Mozemo videti dendrogram, koji prikazuje veze koje su generisane aglomerativnim algoritmom klasteriranja. Dendrogram je grafički prikaz hijerarhijskog klasteriranja i omogućava vizualizaciju kako su tačke grupisane u klasterima na različitim nivoima hijerarhije.

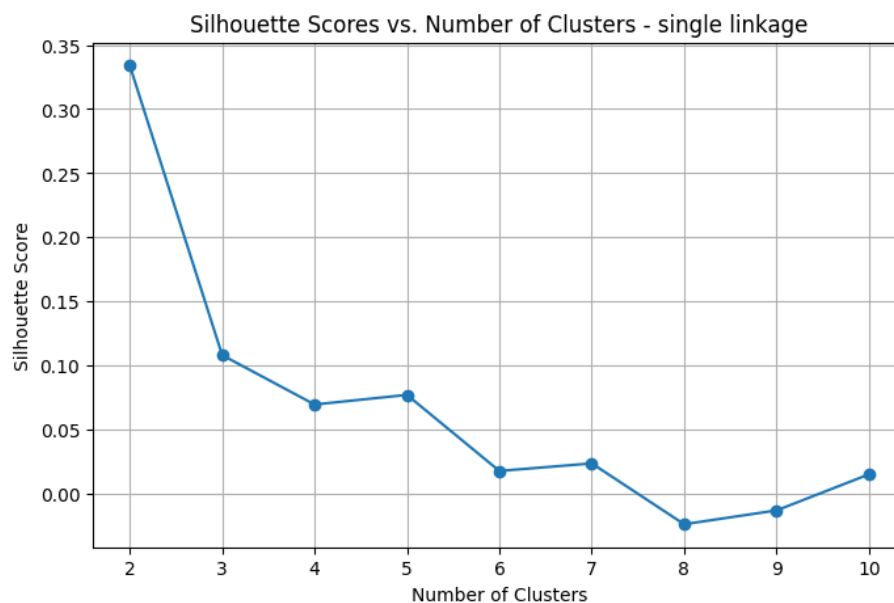


Zatim prikazujemo rezultate klasterovanja za različite tipove veza (linkage). Konačni rezultat su tri grafička prikaza (subplota), svaki prikazujući kako su podaci raspoređeni u klasterima za različite tipove veza u aglomerativnom klasterovanju. Ovi grafikoni omogućavaju vizualnu analizu kako različite veze utiču na strukturu klastera i kako se podaci grupišu unutar tih klastera. Takođe, silhouette skor je uključen kao metrika za ocenu kvaliteta klasteriranja za svaki tip veze.



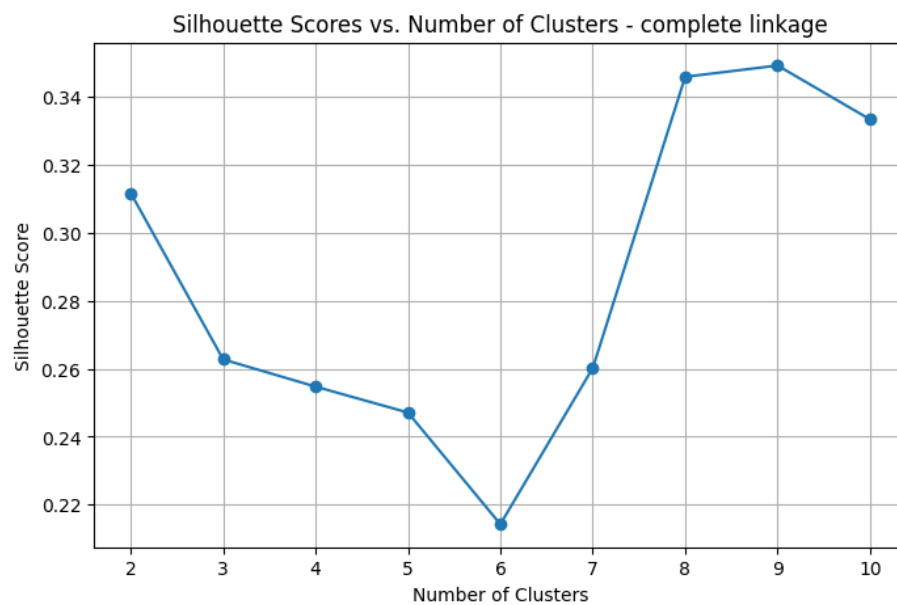
Kada je broj klastera dva, dobijamo najbolji silhouette skor.

Na narednom grafiku možemo videti kako se silhouette skor menja u zavisnosti od broja klastera, kada je tip veze single.



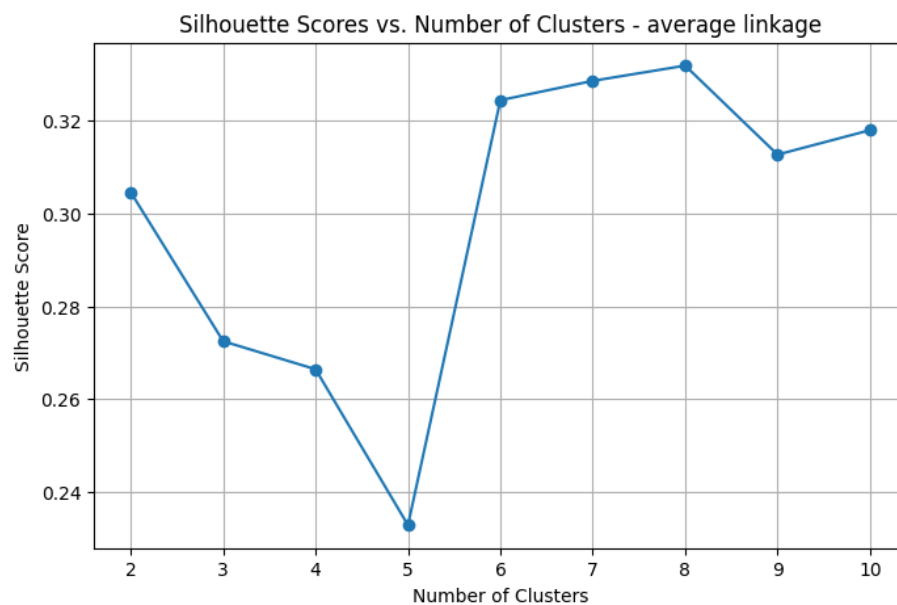
Najbolje je da se izabere 2 klastera za single linkage

Na narednom grafiku možemo videti kako se silhouette skor menja u zavisnosti od broja klastera, kada je tip veze complete.



Najbolje je da se izabere 9 klastera za complete linkage

Na narednom grafiku možemo videti kako se silhouette skor menja u zavisnosti od broja klastera, kada je tip veze average.



Najbolje je da se izabere 8 klastera za average linkage. S obzirom da imamo mali broj instanci u skupu podataka, a silhoutte skor je slican i kad se podaci dele u 6 klastera, bolje je izabrati da delimo podatke u manje klastera - izbegavamo premale klastere.

4 Pravila pridruzivanja - SPSS

Ovu metodu koristimo, kada se suočavamo sa velikim brojem atributa i želimo da identifikujemo medjusobne veze izmedju tih atributa. Pravila pridruživanja nam omogućavaju da otkrijemo te veze medju podacima. Ove veze su značajne jer nam pomažu da dublje razumemo podatke i bolje ih interpretiramo.

4.1 Apriori algoritam

Apriori algoritam je jedan od najpoznatijih algoritama za izdvajanje pravila pridruživanja.

4.2 Pretprocesiranje - Python

Da bi se olakšao rad s podacima, deo pretprocesiranja je izvršena pomoću Pythona.

S obzirom na to da Apriori algoritam radi s diskretizovanim i kategoričkim podacima, a u skupu podataka prevladavaju neprekidni podaci, bilo je potrebno sprovesti preprocesiranje podataka. Konkretno, atribut "*price*" je podvrgnut procesu diskretizacije. Diskretizacija predstavlja postupak pretvaranja neprekidnih numeričkih atributa u diskretne kategorije ili intervale kako bi se olakšala analiza podataka. Podela podataka je izvršena u odnosu 7 naprema 3, kao što je uradjeno u algoritmima za klasifikaciju i klasterovanje.

Takodje, nad atributom "*os*" je sprovedena binarizacija primenom tehnike One-Hot Enkodiranja. Ova tehnika je specifična vrsta binarizacije cesto korišćena za rad s kategoričkim atributima. Svaka moguća vrednost kategoričkog atributa je pretvorena u zasebnu binarnu kolonu, gde svaka kolona sadrži vrednosti 0 ili 1 i označava prisustvo ili odsustvo određene kategorije za svaku instancu podataka.

Dodatno, atribut "*processor_brand*" je podvrgnut faktorizaciji, jer Apriori algoritam zahteva numeričke podatke za rad. Iz tabele su takodje obrisani nenumerički podaci: *brand_name* i *model*.

4.3 SPSS

Nakon predprocesiranja u Pythonu, ovi podaci su učitani u SPSS Modeler

Field	Measurement	Values	Missing	Check	Role
5G_or_not	Flag	1/0		Coerce	Both
processor_b...	Nominal	0,1,2,3,4,5...		Coerce	Both
num_cores	Continuous	[4.0,8.0]		Coerce	Both
processor_s...	Continuous	[1.2,3.22]		Coerce	Both
battery_capa...	Continuous	[1821.0,22...		Coerce	Both
fast_chargin...	Flag	1/0		Coerce	Both
fast_charging	Continuous	[0.0,240.0]		Coerce	Both
ram_capacity	Continuous	[1,18]		Coerce	Both
internal_me...	Continuous	[8,1024]		Coerce	Both
screen_size	Continuous	[3.54,8.03]		Coerce	Both
refresh_rate	Continuous	[60,240]		Coerce	Both
num_rear_c...	Nominal	1,2,3,4		Coerce	Both
primary_cam...	Continuous	[2.0,200.0]		Coerce	Both
primary_cam...	Continuous	[0.0,60.0]		Coerce	Both
extended_m...	Flag	1/0		Coerce	Both
resolution_h...	Continuous	[480,3840]		Coerce	Both
resolution_w...	Continuous	[480,2460]		Coerce	Both
price_binary	Flag	1/0		Coerce	Both
avg_rating_b...	Flag	1/0		Coerce	Both
os_android	Flag	1/0		Coerce	Both
os_ios	Flag	1/0		Coerce	Both
os_other	Flag	1/0		Coerce	Both

Nakon citanja svih vrednosti i postavljanja njihovih uloga na "Both"

Naredni korak je da diskretizujemo vrednosti internal_memory i ram_capacity jer su neprekidni atributi

internal_memory, battery_capacity i ram_capacity smo podelili na 5 kategorija.





4.3.1 Pokretanje algoritma

Nakon predprocesiranja pokrenuli smo apriori algoritam

U listu Consequents se dodaju stavke koje mogu da se pojave u glavi pravila, a u listu Antecedents se dodaju stavke koje mogu da se pojave u telu pravila (Oblik pravila pridruživanja je telo - glava, odnosno antecedents - consequents).

U listi Consequent, se nalaze price_binary i avg_rating_binary, dok su u listi Antecedents svi ostali atributi.

price_binary & avg_rating_binary

Fields

Model

Expert

Annotations

Model name:

☒ Auto
 ☐ Custom

☒ Use partitioned data

Minimum antecedent support (%):

2.0

Minimum rule confidence (%):

75.0

Maximum number of antecedents:

5

☒ Only true values for flags

Optimize:

☒ Speed
 ☐ Memory

OK

▶ Run

Cancel

Apply

Reset

Activate Windows
 Go to Settings to activate Windows

4.3.2 Rezultati

Support meri učestalost pojave određenog pravila ili uzorka u skupu podataka. Konkretno, support za pravilo (A -> B) meri koliko često se skup atributa A i atributa B zajedno pojavljuju u skupu podataka. Confidence meri koliko je često pravilo (A -> B) tačno, odnosno koliko često se atribut B pojavljuje kada je atribut A prisutan.

Consequent	Antecedent	Support %	Confidence %
	5G_or_not internal_memory_BIN = ...		
avg_rating_binary	processor_brand = 2 num_rear_cameras = 3 5G_or_not internal_memory_BIN = ... os_android	15.816	98.065
avg_rating_binary	num_rear_cameras = 3 5G_or_not internal_memory_BIN = ... fast_charging_available os_android	25.918	98.031
avg_rating_binary	processor_brand = 2 num_rear_cameras = 3 5G_or_not internal_memory_BIN = ... fast_charging_available	15.51	98.026
avg_rating_binary	processor_brand = 2 5G_or_not fast_charging_available	29.898	97.952
avg_rating_binary	processor_brand = 2 5G_or_not fast_charging_available os_android	29.694	97.938
avg_rating_binary	processor_brand = 2 5G_or_not battery_capacity_BIN = 1	29.184	97.902
avg_rating_binary	processor_brand = 2		

5 Zaključak

Kad je u pitanju klasifikacije, KNN klasifikator sa podrazumevanim parametrima i KNN klasifikator dobijen korišćenjem Grid Search-a imaju isti nivo tačnosti 92%.

KMeans algoritam je postigao najbolju tačnost, sa smanjenim brojem atributa. Međutim, kada je u pitanju ceo skup podataka, AgglomerativeClustering je dao najbolju tačnost, a statisticka mera pomocu koje je merena tacnost je silhouette skor.