



# PRIRODNO-MATEMATIČKI FAKULTET

## INFORMATIKA

PREDMET: Uvod u nauku o podacima

SEMINARSKI RAD NA TEMU: All Computer Prices

Članovi tima:

Bojan Kovarbašić 63/2022

Mateja Lapatanović 67/2022

Predmetni profesor

Branko Arsić

## Sadržaj

Uvod .....	5
Opis problema i motivacija.....	6
Opis podataka .....	7
Opis ciljne promenljive.....	9
Vizuelizacija podataka .....	12
Uvod .....	12
Univariantna analiza.....	12
Bivariantna analiza .....	17
Multivariantna analiza.....	34
Čišćenje i obrada podataka .....	41
Uvod .....	41
Nedostajuće vrednosti .....	42
Pogrešno unete vrednosti .....	42
Nelogične vrednosti .....	42
Analiza i potencijalno izbacivanje outlier i high leverage tačaka.....	45
EDA (Exploratory Data Analysis).....	47
Uvod .....	47
Matrica korelaciјe .....	48
Pretvaranje kategorijskih obeležja u factor obeležja .....	51
Najbitniji grafici EDA faze i ANOVA statistički test.....	52
Feature engineering.....	64
Feature Selection .....	64
Feature Engineering .....	67
Cpu power score .....	67
Cgt score .....	70

Cpu generation.....	73
Struktura skupa .....	77
Treniranje modela .....	77
Priprema skupa .....	77
Linearna regresija.....	79
Uvod .....	79
Model 1.....	80
Model 2.....	82
Model 3.....	84
Model 4.....	85
Model 5.....	86
Model 6.....	87
Model 8.....	90
Model 9.....	92
Model 10.....	93
Model 11.....	95
Model 12.....	97
Model samo laptopovi .....	99
Model samo računari .....	102
Random Forest.....	104
Uvod .....	104
Treniranje modela .....	104
Predikcija.....	106
Feature importance.....	107
XGBoost .....	109
Uvod .....	109

Treniranje modela .....	109
Predikcija.....	110
Feature importance.....	111
Lasso regresija.....	112
Uvod .....	112
Treniranje .....	113
Predikcija.....	114
Feature importance.....	115
Poređenje modela.....	116
Zaključak .....	117
Literatura .....	119

## Uvod

Predviđanje cena računara i laptopova danas ima sve veći značaj, jer omogućava bolje razumevanje tržišta i efikasnije donošenje poslovnih i potrošačkih odluka. U svetu, u kojem se računar i informacione tehnologije menjaju iz dana u dan, cene uređaja zavise od velikog broja faktora — od brenda i tehničkih karakteristika, do trenutnih trendova i ponude na tržištu. Tačne procene vrednosti uređaja pomažu potrošačima da ne plate više nego što uređaj zaista vredi, dok proizvođačima i prodavcima omogućavaju da postave realne i konkurentne cene svojih proizvoda.

Osim u komercijalne svrhe, predviđanje cena ima i širu analitičku primenu. Na osnovu istorijskih podataka može se razumeti na koji način pojedine komponente, poput procesora, količine RAM memorije ili grafičke kartice, utiču na krajnju cenu uređaja. Takve analize doprinose boljem planiranju proizvodnje, razvoju novih modela i praćenju trendova u IT industriji. Istovremeno, ovakav pristup pokazuje kako se principi nauke o podacima mogu primeniti u svakodnevnim i praktičnim kontekstima.

Tradicionalno, formiranje cena računara oslanjalo se na subjektivnu procenu stručnjaka ili prodajnih agenata, što je često dovodilo do nepreciznih i neujednačenih rezultata. Danas, zahvaljujući razvoju metoda analize podataka i mašinskog učenja, moguće je objektivno utvrditi odnose između karakteristika uređaja i njegove tržišne vrednosti. Analitički pristupi zasnovani na podacima omogućavaju da se uoče obrasci i zavisnosti koji nisu odmah vidljivi na prvi pogled, čime se povećava tačnost i pouzdanost procena.

Za potrebe ovog projekta korišćen je programski jezik R, koji se pokazao kao odličan alat za analizu i obradu podataka, vizuelizaciju i izradu prediktivnih modela. On pruža mogućnost da se rezultati prikažu jasno, pregledno i grafički, što dodatno olakšava njihovu interpretaciju. Na taj način, predviđanje cena računara ne predstavlja samo tehnički zadatak, već i istraživački proces koji spaja analitičko razmišljanje i praktičnu primenu podataka u realnom svetu.

## Opis problema i motivacija

Iz ugla nauke o podacima, problem predviđanja cena računara spada u oblast regresione analize, gde se na osnovu poznatih karakteristika uređaja pokušava proceniti njegova tržišna vrednost. U ovom slučaju, cilj je izgraditi model koji na osnovu tehničkih specifikacija, poput procesora, grafičke kartice, količine RAM memorije, vrste skladišta podataka ili operativnog sistema, može da predvidi kolika će biti cena uređaja.

Ovakva analiza ima praktičan značaj jer omogućava da se utvrdi koje karakteristike najviše utiču na cenu i u kolikoj meri. Na primer, može se istražiti da li procesori viših generacija uvek nose veću cenu, koliko dodatna RAM memorija povećava vrednost uređaja ili da li grafičke kartice određenih proizvođača imaju veći uticaj na konačnu cenu. Takve informacije mogu biti korisne ne samo krajnjim kupcima, već i prodavcima, proizvođačima i analitičarima tržišta.

Sa aspekta analize podataka, ovaj zadatak predstavlja zanimljiv izazov jer karakteristike uređaja obuhvataju različite tipove podataka — numeričke (poput cene i veličine memorije) i kategoriske (kao što su proizvođač, tip procesora ili operativni sistem). Potrebno je pravilno pripremiti podatke, izdvojiti korisne informacije iz tekstualnih zapisa i pretvoriti ih u oblik pogodan za modelovanje. Time se stvara osnova za izgradnju prediktivnog modela koji može da prepozna obrasce i relacije između obeležja i cene.

Motivacija za izbor ove teme potiče iz činjenice da je tržište računara i laptopova izuzetno dinamično i da se stalno pojavljuju novi modeli i tehnologije. Razumevanje faktora koji utiču na formiranje cena doprinosi boljem sagledavanju tržišnih trendova, racionalnijem planiranju kupovine ili prodaje i uopšteno većem razumevanju načina na koji se vrednuju tehničke performanse.

Cilj projekta je, dakle, izgraditi model koji može da predvidi cenu računara na osnovu raspoloživih karakteristika, ali i da pruži uvid u to koje osobine najviše određuju vrednost uređaja.

## Opis podataka

Za potrebe ovog projekta korišćen je skup podataka pod nazivom „All Computer Prices“, preuzet sa platforme „Kaggle“. Skup sadrži objedinjene informacije o velikom broju modela računara i laptopova i njihovim tehničkim karakteristikama, zajedno sa cenama izraženim u američkim dolarima.

U datasetu se nalazi 100.000 redova i 33 kolone, veoma je obiman što omogućava dublju analizu, pri čemu svaki red predstavlja jedan računar ili laptop, a svaka kolona opisuje određenu karakteristiku uređaja.

Uz pomoć funkcije `names` možemo videti spisak svih kolona koje skup podataka sadrži:

```
> names(data)
[1] "device_type"          "brand"           "model"          "release_year"
[5] "os"                  "form_factor"     "cpu_brand"      "cpu_model"
[9] "cpu_tier"             "cpu_cores"        "cpu_threads"    "cpu_base_ghz"
[13] "cpu_boost_ghz"        "gpu_brand"        "gpu_model"      "gpu_tier"
[17] "vram_gb"              "ram_gb"          "storage_type"   "storage_gb"
[21] "storage_drive_count" "display_type"    "display_size_in" "resolution"
[25] "refresh_hz"           "battery_wh"      "charger_watts" "psu_watts"
[29] "wifi"                 "bluetooth"       "weight_kg"      "warranty_months"
[33] "price"
```

Slika 1 Rezultat korišćenja funkcije `names()`

Ne postoji kolona koja predstavlja jedinstveni identifikator (`id`), a ciljna kolona je kolona koja predstavlja cenu uređaja - `price`.

Uz pomoć funkcije `str` možemo videti celu strukturu skupa podataka i tip svakog atributa. Pojavljuju se i numeričke i kategoriske promenljive. U skupu podataka postoji:

- 13 promenljivih tekstualnog tipa (`chr`)
- 14 promenljivih koje imaju celobrojne vrednosti (`int`)
- 6 promenljivih koje imaju vrednosti realnih brojeva (`num`)

Tekstualne kolone uglavnom opisuju specifikacije uređaja poput proizvođača, modela, tipa procesora, itd. Numeričke kolone uglavnom sadrže neke konkretnе vrednosti poput brzine procesora, količine memorije i slično. Ovakva kombinacija kategoriskih i numeričkih promenljivih pogodna je za primenu regresione analize i vizuelizaciju odnosa između različitih karakteristika i cene uređaja.

```

> str(data)
'data.frame': 100000 obs. of 33 variables:
 $ device_type      : chr "Desktop" "Laptop" "Desktop" "Desktop" ...
 $ brand            : chr "Samsung" "Samsung" "Lenovo" "Dell" ...
 $ model            : chr "Samsung Forge XDI" "Samsung Pro K8" "Lenovo Strix BIE" "Dell Cube AXR" ...
 $ release_year     : int 2022 2022 2024 2024 2024 2025 2024 2023 2024 2025 ...
 $ os               : chr "Windows" "Windows" "macOS" "Windows" ...
 $ form_factor      : chr "ATX" "Mainstream" "SFF" "ATX" ...
 $ cpu_brand        : chr "Intel" "Intel" "AMD" "AMD" ...
 $ cpu_model        : chr "Intel i5-11129" "Intel i7-11114" "AMD Ryzen 5 5168" "AMD Ryzen 5 7550" ...
 $ cpu_tier         : int 3 4 2 2 5 5 2 3 6 1 ...
 $ cpu_cores        : int 12 12 8 6 16 16 6 8 26 4 ...
 $ cpu_threads      : int 24 24 16 12 32 32 6 8 52 8 ...
 $ cpu_base_ghz    : num 2.8 2.6 2.6 2.6 2.8 3.2 2.6 2.8 3 2 ...
 $ cpu_boost_ghz   : num 3.8 3.6 3.6 3.6 3.9 4.3 3.5 3.7 4.1 2.9 ...
 $ gpu_brand        : chr "NVIDIA" "NVIDIA" "NVIDIA" "AMD" ...
 $ gpu_model        : chr "RTX 40 60" "RTX 40 80" "RTX 40 50" "RX 7000 60" ...
 $ gpu_tier         : int 2 4 1 2 5 6 1 4 6 1 ...
 $ vram_gb          : int 6 10 4 6 12 16 0 0 16 4 ...
 $ ram_gb           : int 16 64 8 16 96 96 8 32 128 8 ...
 $ storage_type     : chr "NVMe" "NVMe" "NVMe" "HDD" ...
 $ storage_gb       : int 1024 512 512 512 256 512 2048 1024 1024 512 ...
 $ storage_drive_count: int 1 1 2 2 1 2 2 1 1 1 ...
 $ display_type     : chr "LED" "OLED" "LED" "IPS" ...
 $ display_size_in  : num 27 16 32 27 15.6 24 32 27 14 15.6 ...
 $ resolution       : chr "2560x1440" "1920x1080" "3440x1440" "3440x1440" ...
 $ refresh_hz       : int 90 90 120 120 90 90 60 60 60 120 ...
 $ battery_wh       : int 0 56 0 0 80 0 0 0 80 60 ...
 $ charger_watts   : int 0 120 0 0 90 0 0 0 240 45 ...
 $ psu_watts        : int 750 0 850 650 0 1000 850 650 0 0 ...
 $ wifi             : chr "Wi-Fi 6" "Wi-Fi 6" "Wi-Fi 6" "Wi-Fi 6" ...
 $ bluetooth        : num 5.1 5.3 5 5.2 5.2 5 5.1 5 5 5.3 ...
 $ weight_kg        : num 11 2.03 7 6 1.5 9 9 8 1.17 1.5 ...
 $ warranty_months  : int 36 12 24 36 12 36 24 12 48 24 ...
 $ price            : num 1384 2275 1880 1332 2682 ...

```

*Slika 2 Rezultat korišćenja funkcije str()*

Sledi spisak svih kolona i kratak opis svake:

1. device\_type – tip tj. vrsta uređaja, desktop računar ili laptop
2. brand – proizvođač uređaja
3. model – naziv i oznaka konkretnog modela uređaja
4. release\_year – godina u kojoj je uređaj pušten na tržište
5. os – operativni sistem koji je instaliran na uređaju
6. form\_factor – dodatne informacije o uređaju (veličina uređaja, namena i slično)
7. cpu\_brand – proizvođač procesora
8. cpu\_model – naziv i oznaka konkretnog modela procesora
9. cpu\_tier – rang procesora
10. cpu\_cores – broj jezgara procesora
11. cpu\_threads – broj niti procesora

12. cpu\_baze\_ghz – frekvencija procesora (izražena u GHz)
13. cpu\_boost\_ghz – maksimalna frekvencija do koje procesor može da se nadograđi (takođe u GHz)
14. gpu\_brand – proizvođač grafičke kartice
15. gpu\_model – naziv i oznaka modela grafičke kartice
16. gpu\_tier – rang grafičke kartice
17. vram\_gb – količina video memorije grafičke kartice (izražena u GB)
18. ram\_gb – količina radne memorije (takođe u GB)
19. storage\_type – tip skladišta podataka
20. storage\_gb – ukupan kapacitet skladišta podataka (takođe u GB)
21. storage\_drive\_count – broj fizičkih diskova na uređaju koji služe za skladištenje
22. display\_type – tip ekrana uređaja
23. display\_size\_in – dijagonalna ekrana uređaja (izražena u inčima)
24. resolution – rezolucija ekrana uređaja
25. refresh\_hz – brzina osvežavanja ekrana uređaja (izražena u Hz)
26. battery\_wh – kapacitet baterije uređaja (izražen u vat-satima - Wh)
27. charger\_watts – snaga punjača uređaja (izražena u W)
28. psu\_watts – snaga napajanja za desktop računare (takođe u W)
29. wifi – verzija Wi-Fi standarda koji uređaj podržava
30. bluetooth – verzija bluetooth standarda koji uređaj podržava
31. weight\_kg – ukupna masa uređaja (izražena u kg)
32. warranty\_months – trajanje garancije (izraženo u mesecima)
33. price – tržišna cena računara (izražena u \$) - ciljna promenljiva

## Opis ciljne promenljive

```
> summary(data$price)
   Min. 1st Qu. Median      Mean 3rd Qu.      Max.
   373     1504    1864     1929    2288    10985
```

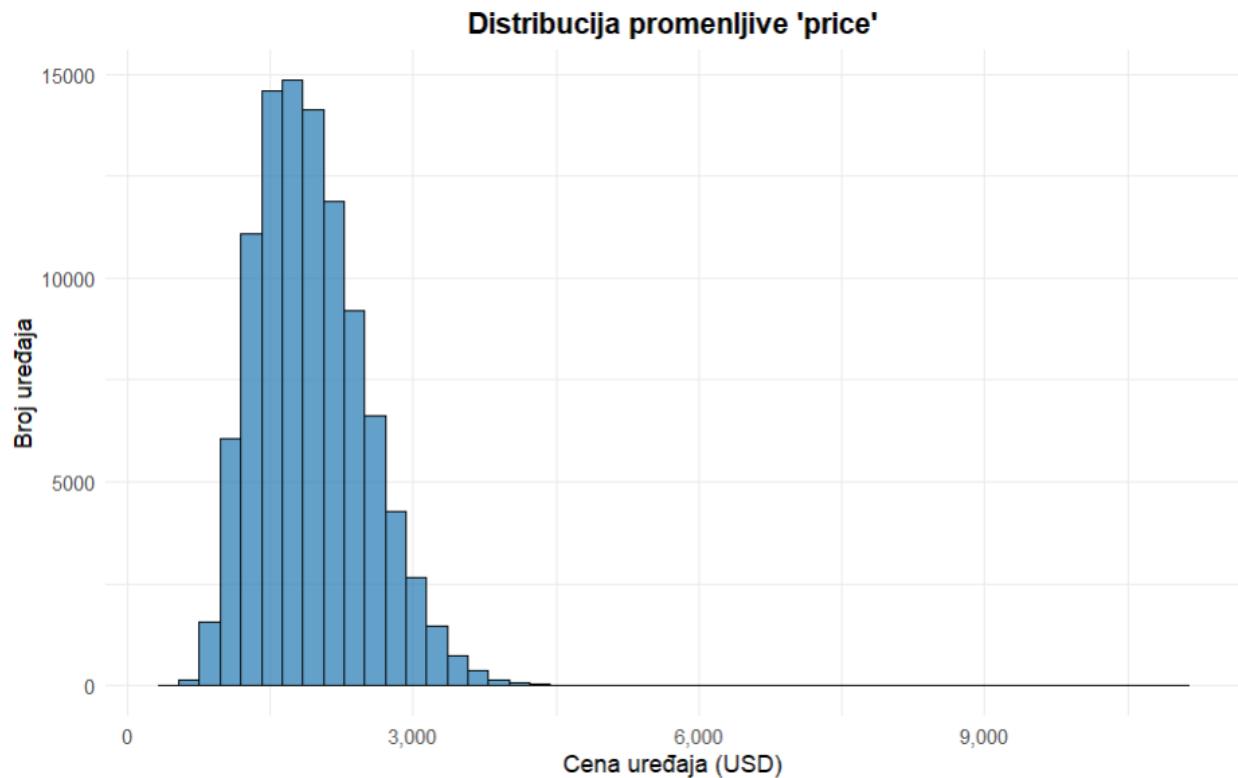
Slika 3 Rezultat korišćenja funkcije `summary()` nad ciljnom promenljivom

Uz pomoć funkcije `summary` možemo videti da se cene uređaja kreću od 373 do 10985 američkih dolara. Srednja vrednost (mean) je nešto viša od medijane

(median) što nam govori da je raspodela podataka blago asimetrična. Većina cena se nalazi u rasponu od 1500\$ do 2300\$, a manji broj uređaja ima ekstremno visoke cene koje i povlače srednju vrednost naviše, što je i pomenuto malopre. Uređaji uglavnom jeftiniji od 800\$ predstavljaju računare i laptopove slabijih performansi i konfiguracija, a uređaji sa cenama iznad 7000\$ predstavljaju računare i laptopove sa vrhunskim komponentama i performansama.

```
ggplot(data, aes(x = price)) +
  geom_histogram(bins = 50, fill = "#1f78b4", color = "black", alpha = 0.7) +
  scale_x_continuous(labels = comma) +
  labs(
    title = "Distribucija promenljive 'price'",
    x = "Cena uređaja (USD)",
    y = "Broj uređaja"
  ) +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5, face = "bold"))
```

*Slika 4 R kod za crtanje histograma distribucije cene uređaja*

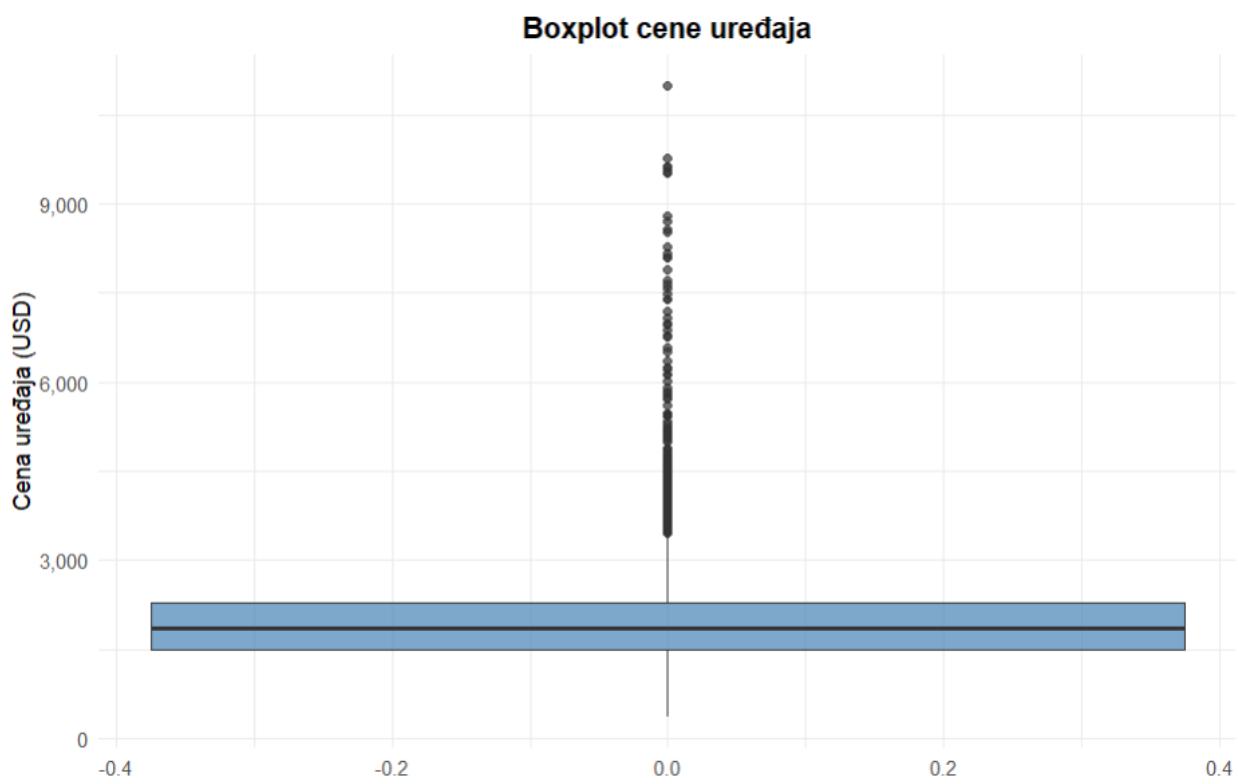


*Slika 5 Histogram distribucije cene uređaja*

Histogram prikazuje raspodelu ciljne promenljive price. Ovde takođe možemo videti da raspodela nije simetrična, već pozitivno asimetrična. Kao što je malopre spomenuto najveći broj uređaja se nalazi u srednjem cenovnom rangu, manji u nižem, a najmanji broj uređaja u skupljem cenovnom rangu. Ovo je i očekivano jer većina korisnika kupuje upravo uređaje srednjeg ranga.

```
ggplot(data, aes(y = price)) +  
  geom_boxplot(fill = "steelblue", alpha = 0.7) +  
  scale_y_continuous(labels = comma) +  
  labs(  
    title = "Boxplot cene uređaja",  
    y = "Cena uređaja (USD)"  
) +  
  theme_minimal() +  
  theme(plot.title = element_text(hjust = 0.5, face = "bold"))
```

Slika 6 R kod za crtanje boxplot-a cene uređaja



Slika 7 Boxplot cene uređaja

Boxplot i potvrđuje ono što smo videli na histogramu i ovde vidimo tačke koje na histogramu nisu bile prikazane, jer ih je jako malo sa visokom cenom i te tačke

potencijalno predstavljaju outlier-e. Oni mogu značajno da utiču na to koliko je naš model dobar što će i biti provereno kasnije.

Pošto linearna regresija prepostavlja da je raspodela ciljne promenljive približno normalna, a kod nas to nije slučaj i zbog toga će kasnije biti primenjena logaritamska transformacija. Ona približava raspodelu normalnoj tj. čini da histogram ima zvonastu strukturu. Samim tim se smanjuje uticaj skupljih uređaja i model postaje bolji i stabilniji.

## Vizuelizacija podataka

### Uvod

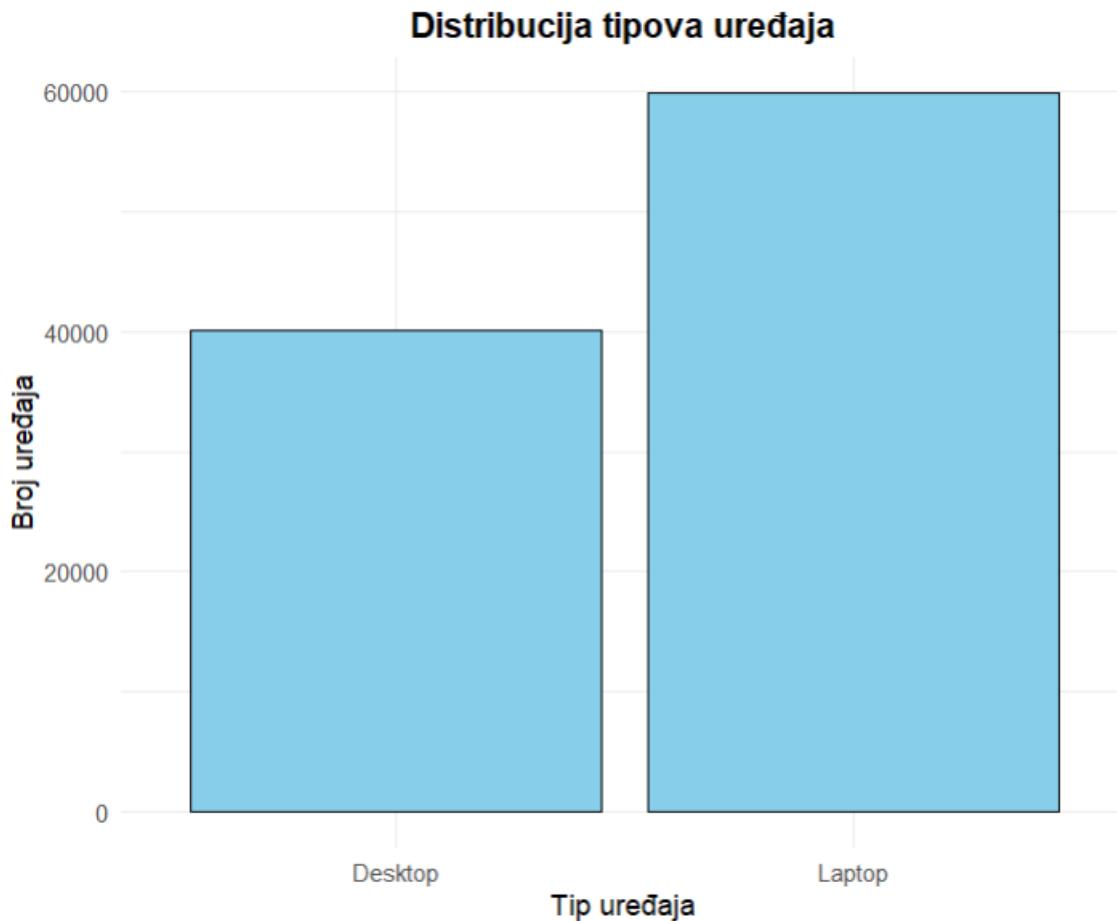
Pre nego što možemo bilo šta da radimo nad skupom podataka, moramo se prvo upoznati sa tim skupom. U ovom delu dokumenta će biti predstavljeni razni grafici, koji će nam pomoći da razumemo svaki podatak u našem skupu i kako on utiče na cenu računara.

Proći ćemo kroz univarijantnu analizu, gde ćemo pogledati raspodelu feature-a. Posle toga sledi bivarijanta analiza, gde ćemo uporediti većinu feature-a sa cenom računara i videti kako oni utiču na cenu i da li postoje ekstremne vrednosti. Nakon toga će biti odrađena multivarijantna analiza, gde će biti iscrtani grafici sa više feature-a i cenom, gde možemo analizirati kako više feature-a zajedno utiču jedan na drugi i kako zajedno utiču na cenu uređaja.

### Univarijantna analiza

```
ggplot(data, aes(device_type)) +  
  geom_bar(fill = "skyblue", color = "black") +  
  labs(  
    title = "Distribucija tipova uređaja",  
    x = "Tip uređaja",  
    y = "Broj uređaja"  
  ) +  
  theme_minimal() +  
  theme(plot.title = element_text(hjust = 0.5, face = "bold"))
```

Slika 8 R kod za prikaz distribucije tipova uređaja



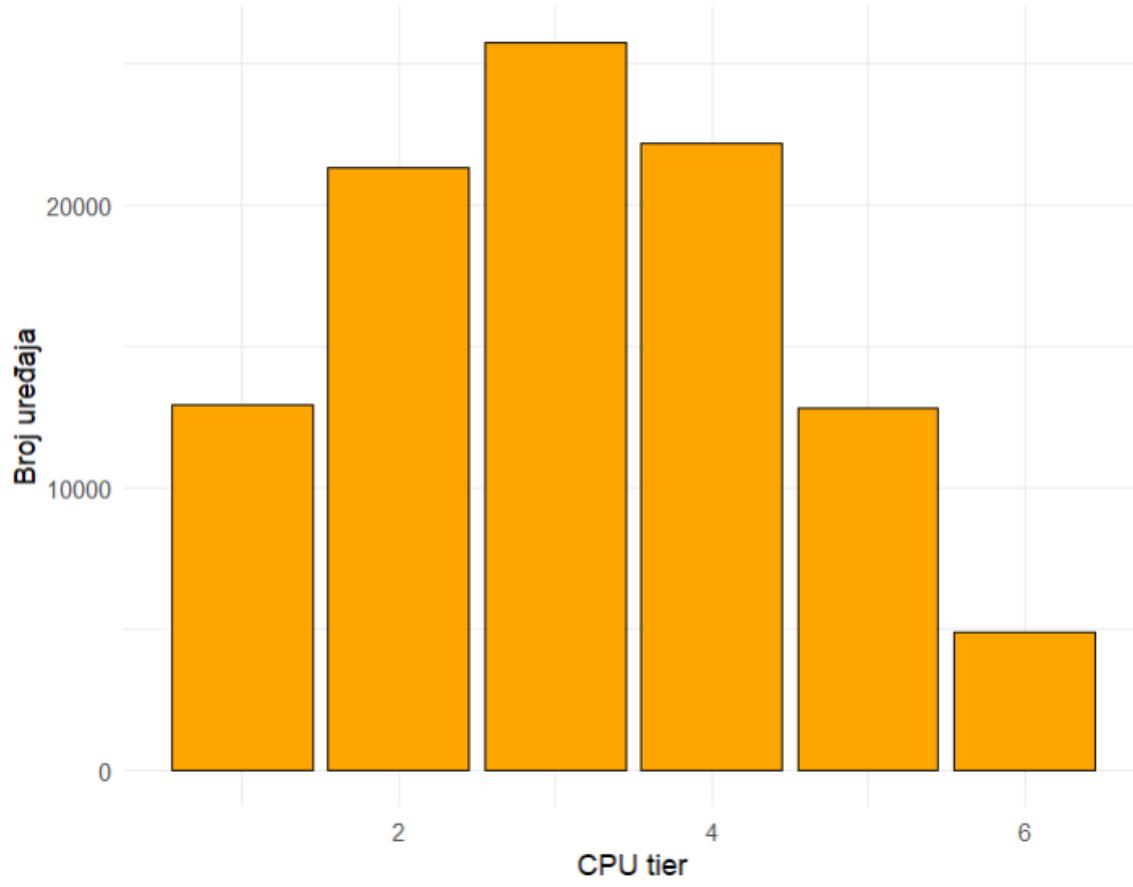
*Slika 9 Distribucija tipova uređaja*

Na grafiku iznad se vidi da u skupu ima više laptopova nego desktop računara, što je i očekivano jer su laptopovi danas mnogo zastupljeniji na tržištu. Ova razlika nije problematična, ali je važno da se zabeleži, već u početnoj analizi, jer nam govori kakav je sastav podataka pre daljeg čišćenja i obrade. U ovoj promenljivoj nema nelogičnosti ili vrednosti koje bi trebalo ukloniti.

```
ggplot(data, aes(cpu_tier)) +
  geom_bar(fill = "orange", color = "black") +
  labs(
    title = "Distribucija CPU tier kategorija",
    x = "CPU tier",
    y = "Broj uređaja"
  ) +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5, face = "bold"))
```

*Slika 10 R kod za prikaz distribucije CPU tier kategorija*

### Distribucija CPU tier kategorija



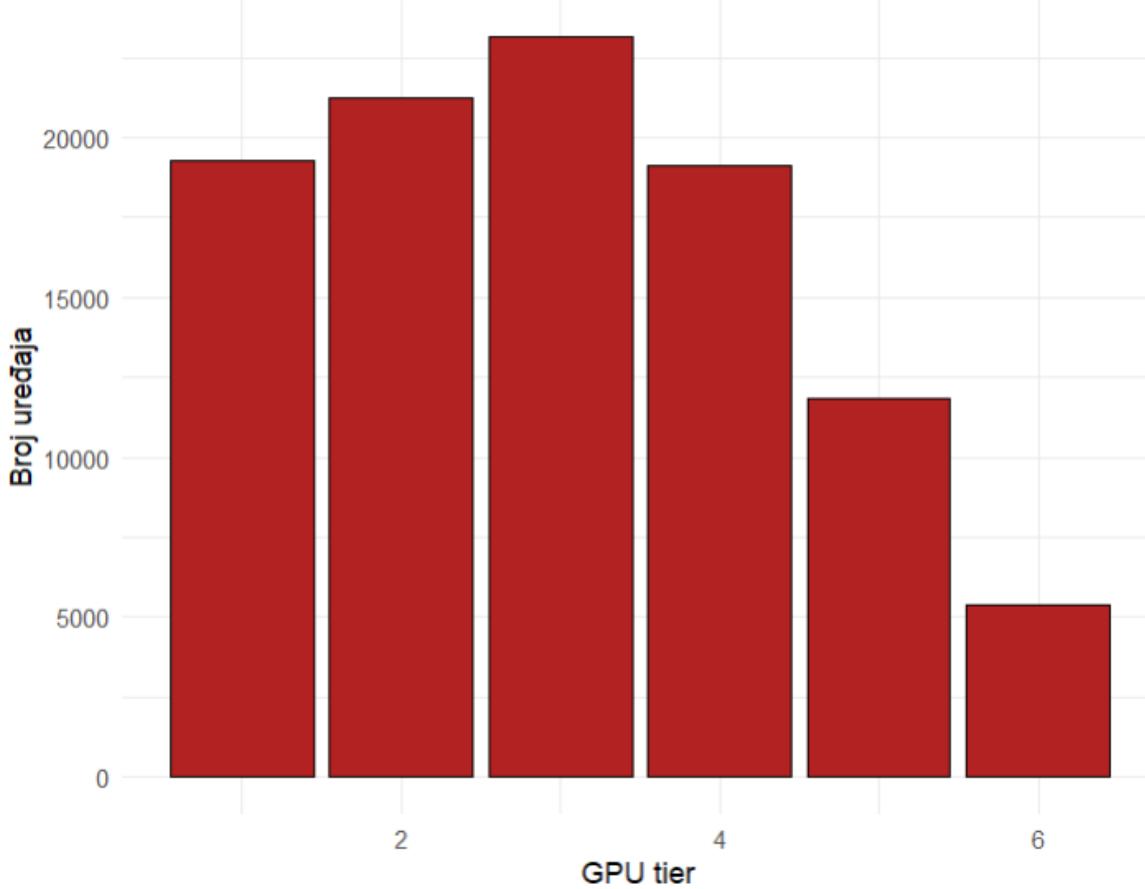
Slika 11 R Distribucija CPU tier kategorija

Raspodela CPU tier vrednosti izgleda prilično prirodno. Najviše uređaja pripada srednjim kategorijama (tier 2–4), dok su najjači i najslabiji procesori očekivano ređi. Ovo nam govori da dataset dobro pokriva različite nivoe procesorske snage i da u ovoj promenljivoj nema očiglednih grešaka ili nelogičnosti koje bi trebalo odmah uklanjati.

```
ggplot(data, aes(cpu_tier)) +  
  geom_bar(fill = "firebrick", color = "black") +  
  labs(  
    title = "Distribucija GPU tier kategorija",  
    x = "GPU tier",  
    y = "Broj uređaja"  
) +  
  theme_minimal() +  
  theme(plot.title = element_text(hjust = 0.5, face = "bold"))
```

Slika 12 R kod za prikaz distribucije GPU tier kategorija

### Distribucija GPU tier kategorija

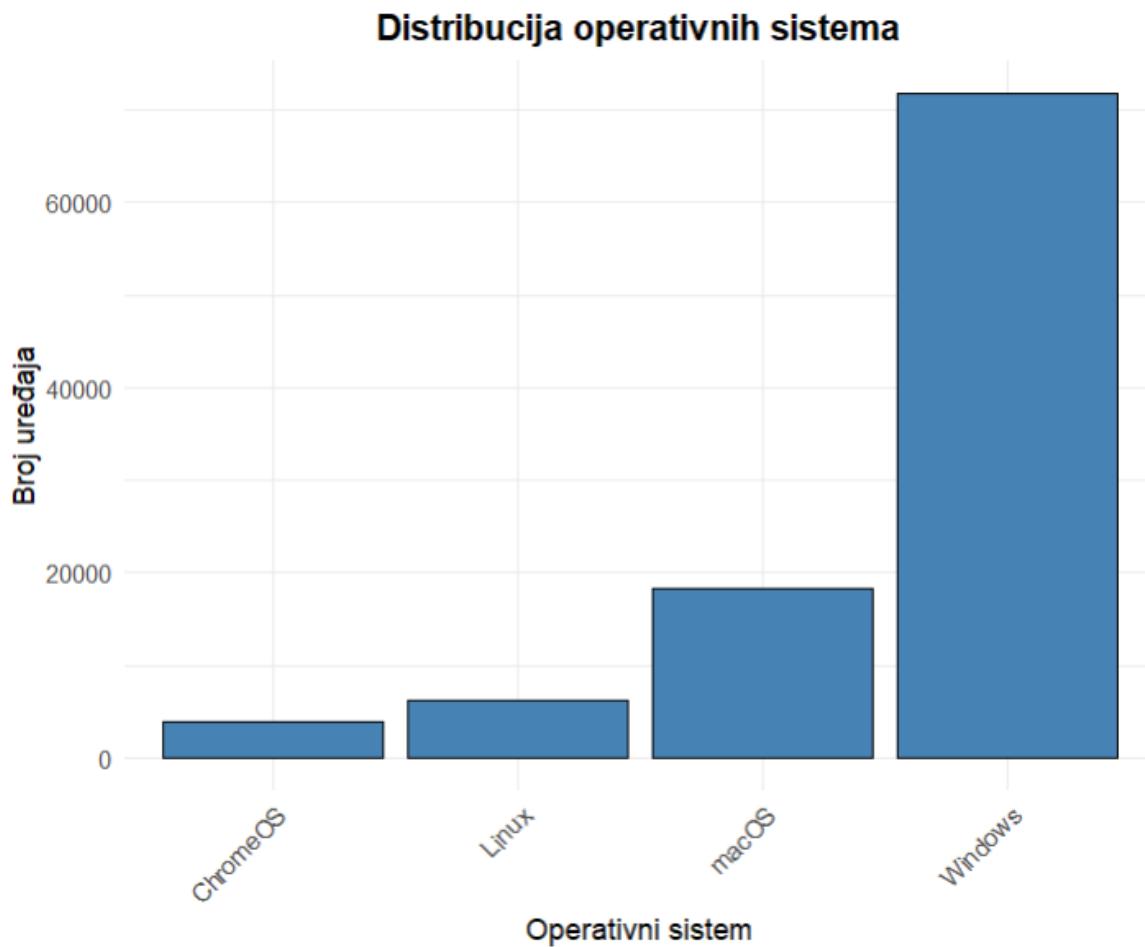


Slika 13 Distribucija CPU tier kategorija

Na grafiku se vidi da i raspodela GPU tier vrednosti izgleda očekivano tj. najzastupljenije su srednje kategorije, dok najjače grafičke kartice imaju znatno manje primeraka. Ovo je tipičan raspored za tržište uređaja i ne ukazuje na bilo kakve greške u podacima i nema nikakvih vrednosti koje bi trebalo odmah ukloniti.

```
ggplot(data, aes(os)) +  
  geom_bar(fill = "steelblue", color = "black") +  
  labs(  
    title = "Distribucija operativnih sistema",  
    x = "Operativni sistem",  
    y = "Broj uređaja"  
) +  
  theme_minimal() +  
  theme(axis.text.x = element_text(angle = 45, hjust = 1),  
    plot.title = element_text(hjust = 0.5, face = "bold"))
```

Slika 14 R kod za prikaz distribucije operativnih sistema



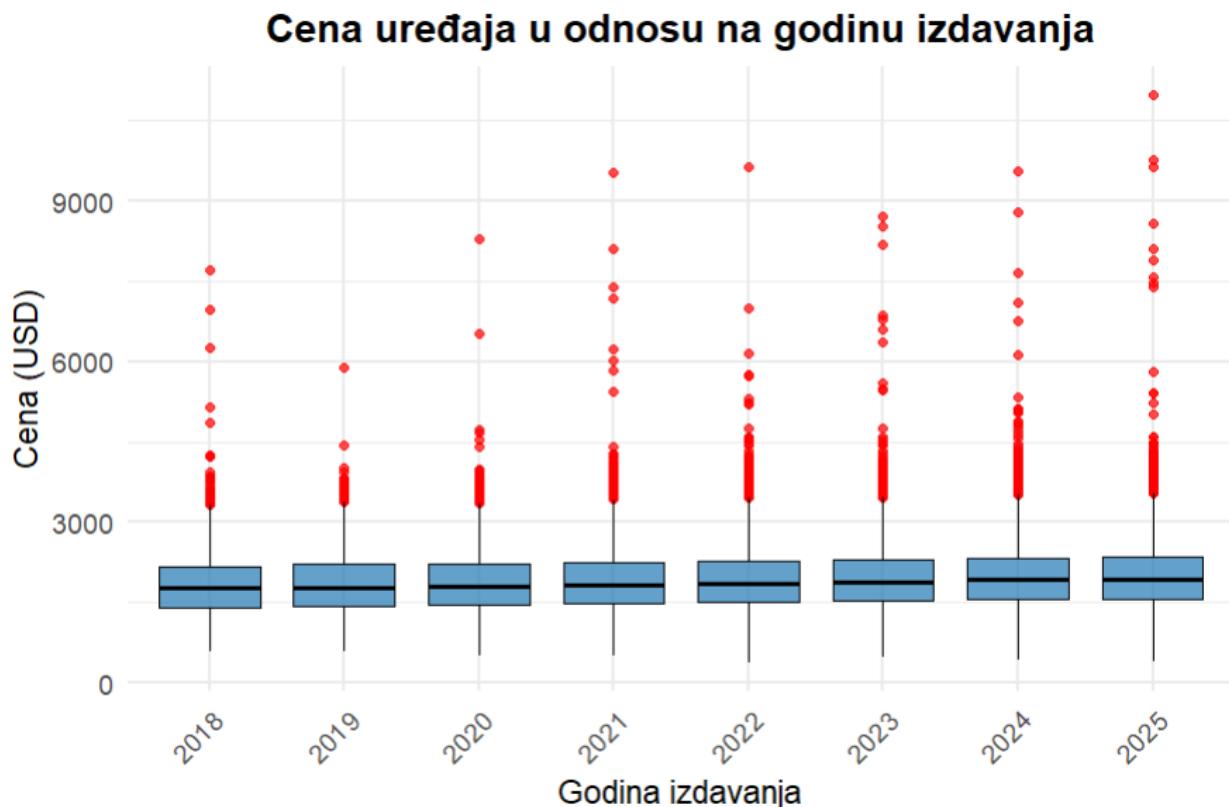
*Slika 15 Distribucija operativnih sistema*

Distribucija operativnih sistema pokazuje da veliki broj uređaja koristi Windows, što je u skladu sa realnim tržištem. ChromeOS i Linux su očekivano znatno manje zastupljeni, dok macOS ima nešto veći broj uređaja nego inače na tržištu, pa je potrebno kasnije to ispitati, trenutno nema nekih grešaka u podacima ili vrednosti koje bi trebalo odmah ukloniti.

## Bivarijančna analiza

```
ggplot(data, aes(x = factor(release_year), y = price)) +  
  geom_boxplot(fill = "#1f78b4", color = "black", alpha = 0.7, outlier.colour = "red")  
  labs(  
    title = "Cena uređaja u odnosu na godinu izdavanja",  
    x = "Godina izdavanja",  
    y = "Cena (USD)"  
) +  
  theme_minimal(base_size = 14) +  
  theme(  
    plot.title = element_text(hjust = 0.5, face = "bold"),  
    axis.text.x = element_text(angle = 45, hjust = 1)  
)
```

Slika 16 R kod za crtanje grafika cene u odnosu na godinu izdavanja



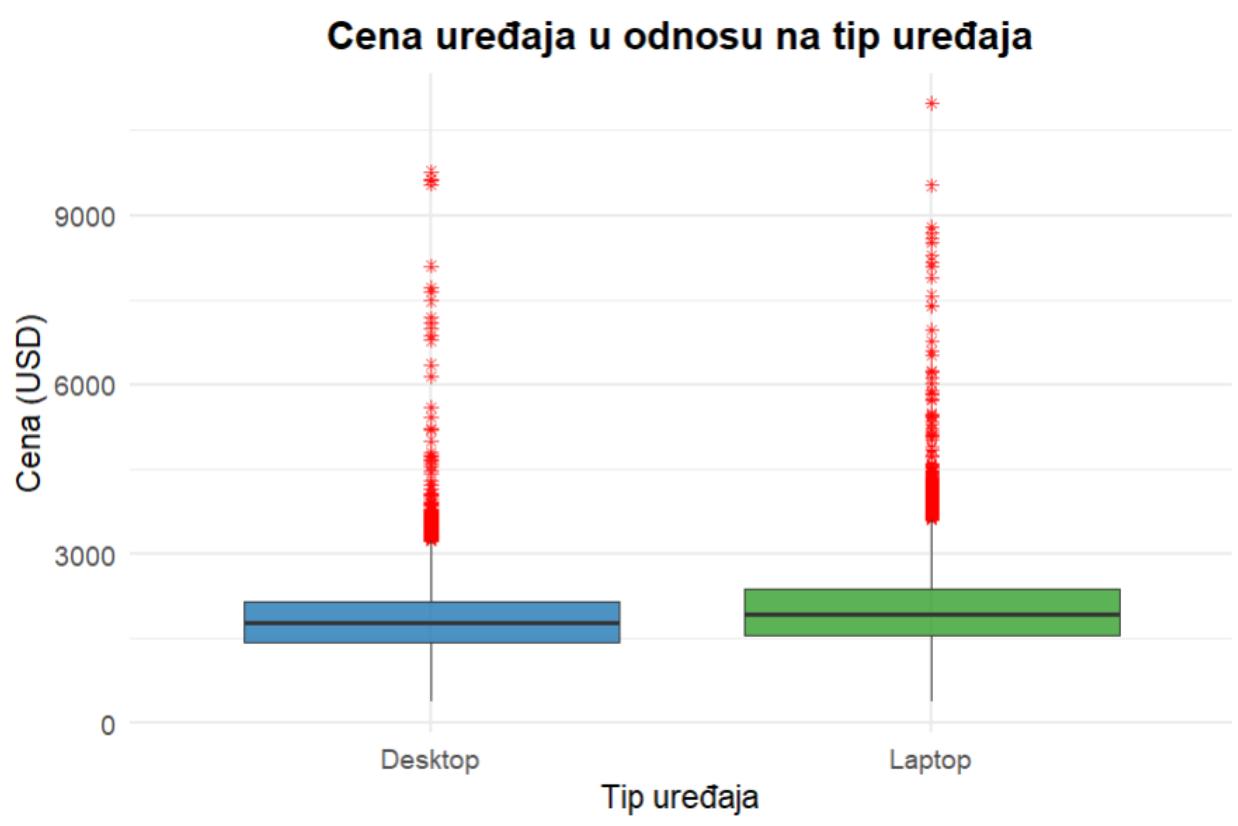
Slika 17 Grafik odnosa cene uređaja i godine izdavanja

Na osnovu boxplot-ova možemo videti da su medijane uglavnom slične kroz godine ili se minimalno povećavaju. Iako bi se trebalo pretpostaviti da će noviji uređaji biti malo skuplji ovo je sasvim u redu. Postoji dosta tačaka koje odskaču od većine, to su sve verovatno profesionalni uređaji sa veoma jakim performansama i skupim komponentama, ali će kasnije biti ispitani. Iako modeli srednjeg cenovnog ranga

blago rastu jeftini modeli uglavnom ostaju sličnih cena što znači da u tom delu ne dolazi do nekih većih promena.

```
ggplot(data, aes(x = device_type, y = price, fill = device_type)) +  
  geom_boxplot(alpha = 0.8, outlier.colour = "red", outlier.shape = 8) +  
  labs(  
    title = "Cena uređaja u odnosu na tip uređaja",  
    x = "Tip uređaja",  
    y = "Cena (USD)"  
) +  
  scale_fill_manual(values = c("#1f78b4", "#33a02c")) +  
  theme_minimal(base_size = 14) +  
  theme(  
    plot.title = element_text(hjust = 0.5, face = "bold"),  
    legend.position = "none"  
)
```

Slika 18 R kod za crtanje boxplot-a cene u odnosu na tip uređaja



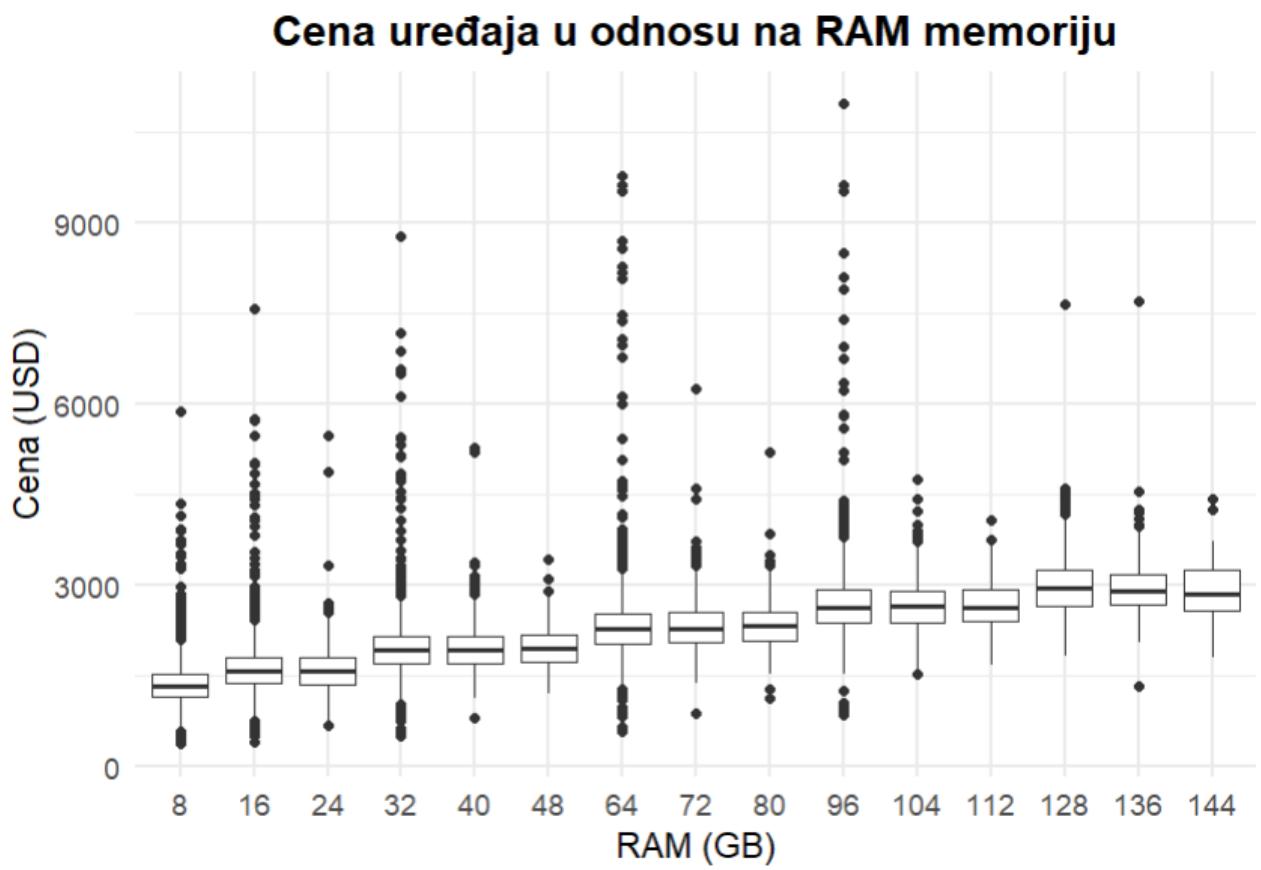
Slika 19 Boxplot-ovi cene uređaja u odnosu na tip (Desktop/Laptop)

Na osnovu boxplot dijagrama možemo videti da postoji razlika u ceni između desktop računara i laptopova, ali nije velika. Medijana cene laptopova je nešto veća

što je i očekivano, jer laptopovi imaju integrisane komponente i prenosivi su, pa su skuplji od računara. Postoji dosta cena koje odskaču od većine (srednje klase) i to su verovatno profesionalni računari i gaming laptopovi.

```
ggplot(data, aes(x = factor(ram_gb), y = price)) +  
  geom_boxplot() +  
  labs(  
    title = "Cena uređaja u odnosu na RAM memoriju",  
    x = "RAM (GB)",  
    y = "Cena (USD)"  
) +  
  theme_minimal(base_size = 14) +  
  theme(  
    plot.title = element_text(face = "bold", hjust = 0.5)  
)
```

Slika 20 R kod za crtanje boxplot-ova cene u odnosu na RAM memoriju



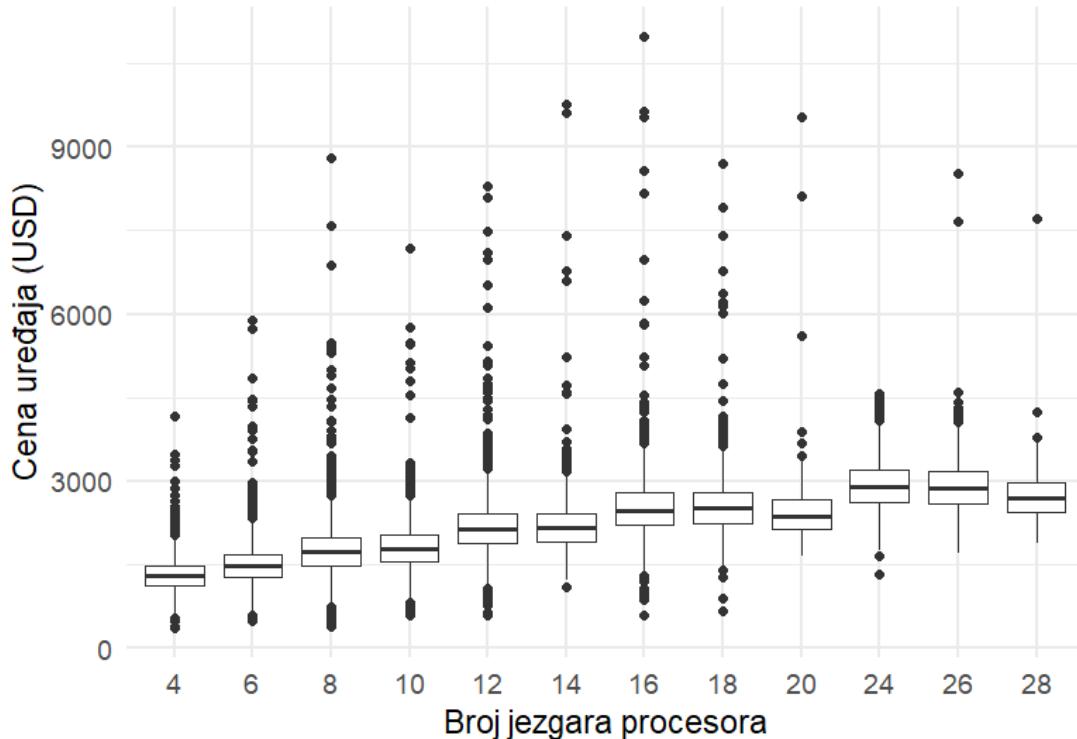
Slika 21 Boxplot-ovi cene u odnosu na količinu RAM memorije

Sa boxplot-ova možemo videti da postoje samo određene vrednosti RAM memorije. Medijane uglavnom rastu sa porastom cene i to tek na po 24 ili 32 GB ram-a, npr. medijana za 32, 40 i 48 GB ram-a je skoro identična tek se za 64 GB ram-a povećava. Postoji mnoštvo potencijalnih outlier-a, posebno u veličinama RAM memorije koje su i inače najčešće, 16, 32, 48 i 96 GB ram-a, ali su za sve vrednosti outlier-i u granicama normale i možemo zaključiti da cena dosta zavisi i od drugih prediktora, jer ovde imamo samo vrednosti količine RAM-a, nemamo podatke o tip-u, frekvenciji i slično. Npr. tokom godina se povećava brzina RAM-a u MHz-ima, pa bi bilo dobro to proveriti.

```
ggplot(data, aes(x = factor(cpu_cores), y = price)) +  
  geom_boxplot() +  
  labs(  
    title = "Cena uređaja u odnosu na broj jezgara procesora",  
    x = "Broj jezgara procesora",  
    y = "Cena uređaja (USD)"  
) +  
  theme_minimal(base_size = 14) +  
  theme(  
    plot.title = element_text(face = "bold", hjust = 0.5)  
)
```

*Slika 22 R kod za crtanje grafika cene u odnosu na broj jezgara procesora*

## Cena uređaja u odnosu na broj jezgara procesora



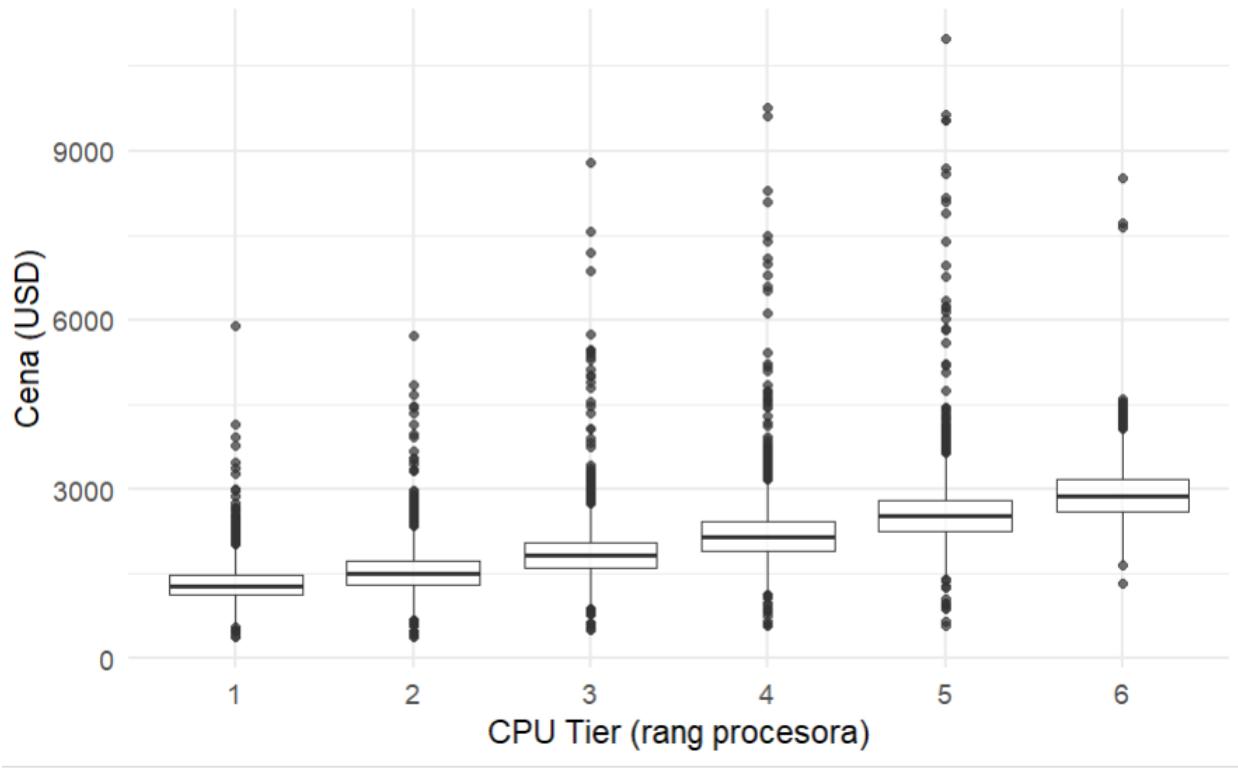
Slika 23 Grafik odnosa broja jezgara procesora i cene

Sa grafika se može videti da sa porastom broja jezgara blago raste i cena uređaja što je i logično jer je broj jezgara indikator procesorske moći. Takođe postoji nekoliko primera sa manje od 10 jezgara, a cenom preko 8000\$ i verovatno je isti razlog kao za ove prethodne, a takođe je moguće da je došlo do nekih grešaka pri unosu, što je potrebno dodatno ispitati.

```
ggplot(data, aes(x = as.factor(cpu_tier), y = price)) +  
  geom_boxplot(alpha = 0.7) +  
  labs(  
    title = "Odnos između CPU Tier-a i cene uređaja",  
    x = "CPU Tier (rang procesora)",  
    y = "Cena (USD)")  
  ) +  
  theme_minimal(base_size = 14) +  
  theme(plot.title = element_text(hjust = 0.5, face = "bold"))
```

Slika 24 R kod za crtanje grafika zavisnosti cene od broja jezgara

## Odnos između CPU Tier-a i cene uređaja



Slika 25 Boxplot-ovi cene u odnosu na rang procesora

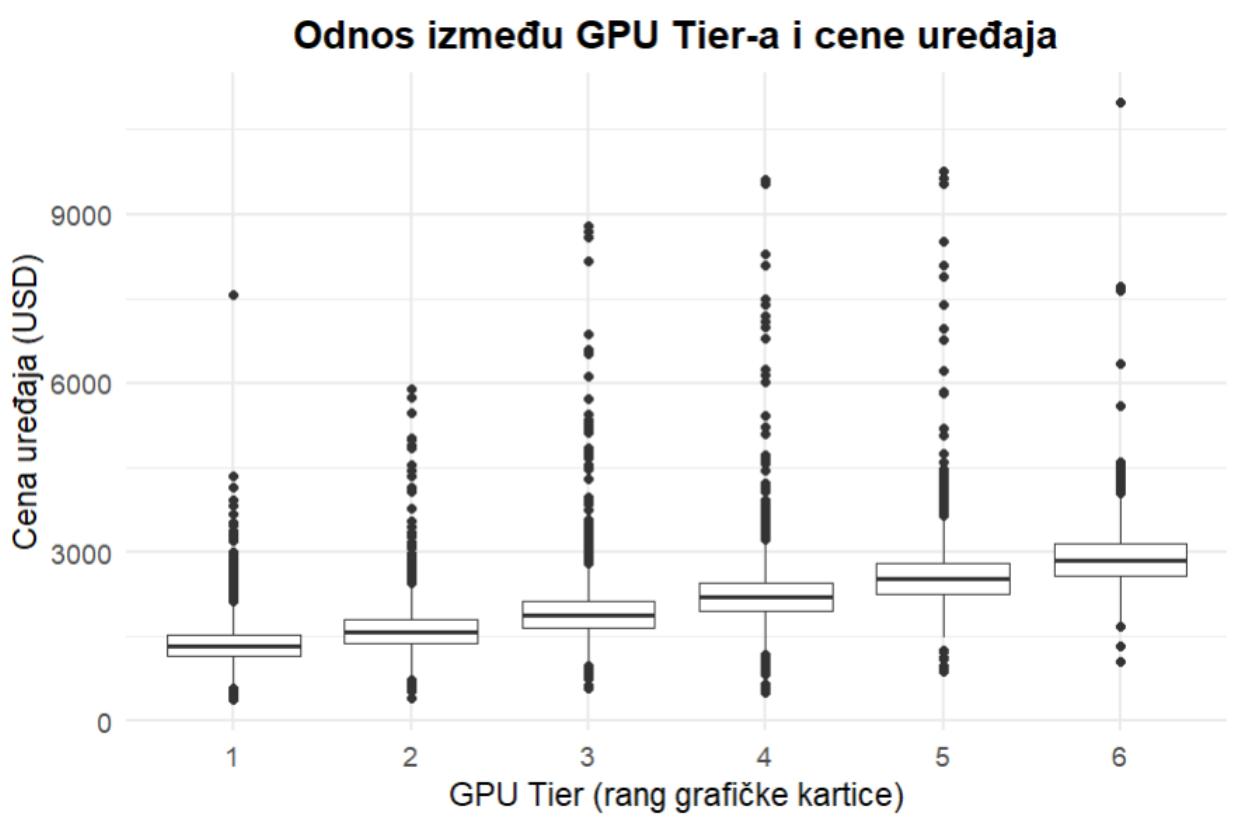
Sa grafika možemo videti boxplot-ove koji prikazuju cene uređaja u odnosu na klasu procesora koju imaju. 1 je najlošija, 6 je najbolja klasa. Medijana se porastom klase postepeno povećava, što je i logično, što je bolji procesor cena je veća. Za svaku kategoriju postoje potencijalni outlieri tj. cene dosta niže ili više od prosečnih za tu kategoriju, što je u redu, jer su druge komponente najverovatnije jeftinije. Postoje uređaji najviše klase, dosta jeftiniji od uređaja nižih klasa, to može značiti da je neka druga komponenta zaslužna za takvu cenu, a može biti i da su podaci pogrešno uneti.

```

ggplot(data, aes(x = as.factor(gpu_tier), y = price)) +
  geom_boxplot() +
  labs(
    title = "Odnos između GPU Tier-a i cene uređaja",
    x = "GPU Tier (rang grafičke kartice)",
    y = "Cena uređaja (USD)"
  ) +
  theme_minimal(base_size = 14) +
  theme(plot.title = element_text(hjust = 0.5, face = "bold"))

```

*Slika 26 R kod za crtanje boxplot-ova cene u zavisnosti od ranga grafičke kartice*



*Slika 27 Boxplot-ovi cene u zavisnosti od ranga grafičke kartice*

Sa grafika možemo videti kakve su cene uređaja u odnosu na grafičku karticu i takođe kao i kod procesora postoji 6 kategorija i još pravilnije sa porastom klase raste i cena. Potencijalnih outliera ima, ali biće ispitani, posebno ovaj uređaj sa cenom od 7500\$, a najlošijom klasom grafičke kartice. U globalu ova osobina je dosta bitna za krajnju cenu uređaja.

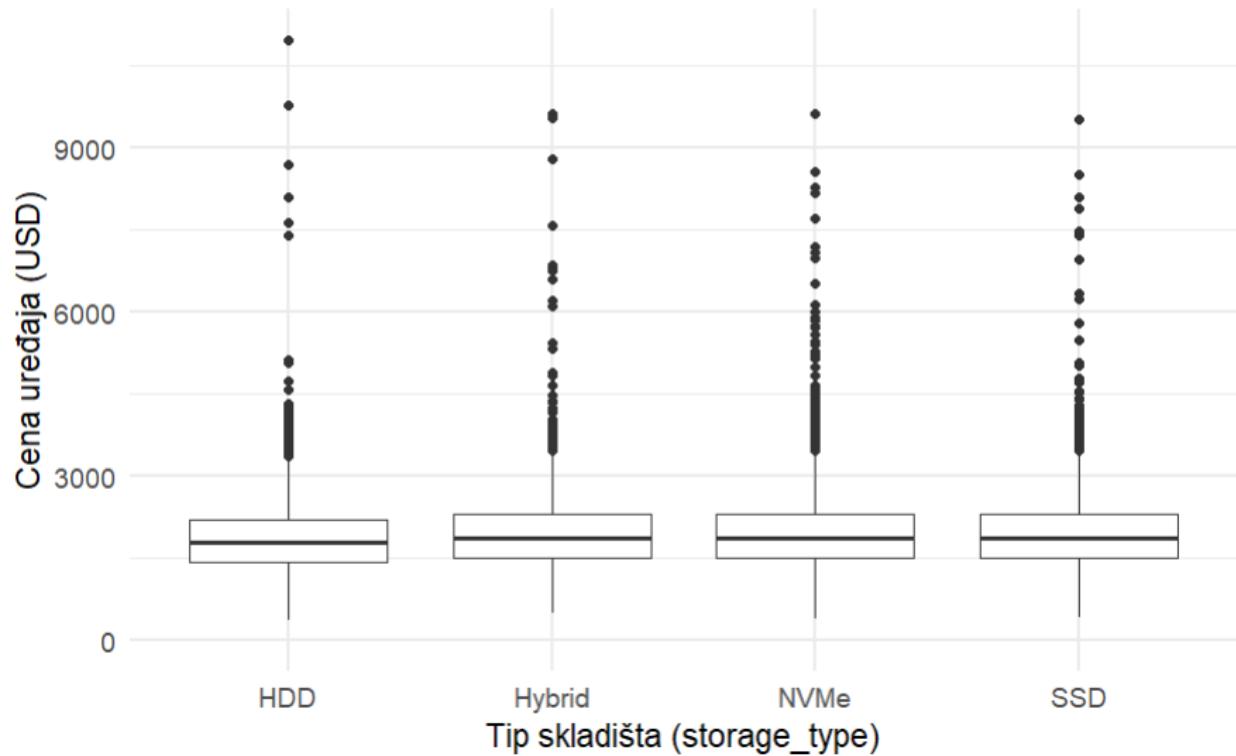
```

ggplot(data, aes(x = storage_type, y = price)) +
  geom_boxplot() +
  labs(
    title = "Cena uređaja u odnosu na tip skladišta podataka",
    x = "Tip skladišta (storage_type)",
    y = "Cena uređaja (USD)"
  ) +
  theme_minimal(base_size = 14) +
  theme(plot.title = element_text(hjust = 0.5, face = "bold"))

```

*Slika 28 R kod za crtanje boxplot-ova cene u odnosu na tip memorije*

### Raspodela cena u odnosu na tip skladišta podataka



*Slika 29 Boxplot-ovi cene u zavisnosti od tipa memorije*

Boxplot-ovi prikazuju zavisnost cene od tipa memorije, HDD je najstarija i najsporija, SSD novija od nje i brža, Hybrid je njihova kombinacija, a NVMe je najbrža i najbolja verzija SSD memorije. Ovo obeležje ne utiče značajno na cenu uređaja, medijane su skoro pa slične, malo je medijana niža kod HDD memorije što je i logično jer je to najsporija i najjeftinija memorija. Postoje uređaji sa velikom cenom, a najslabijom

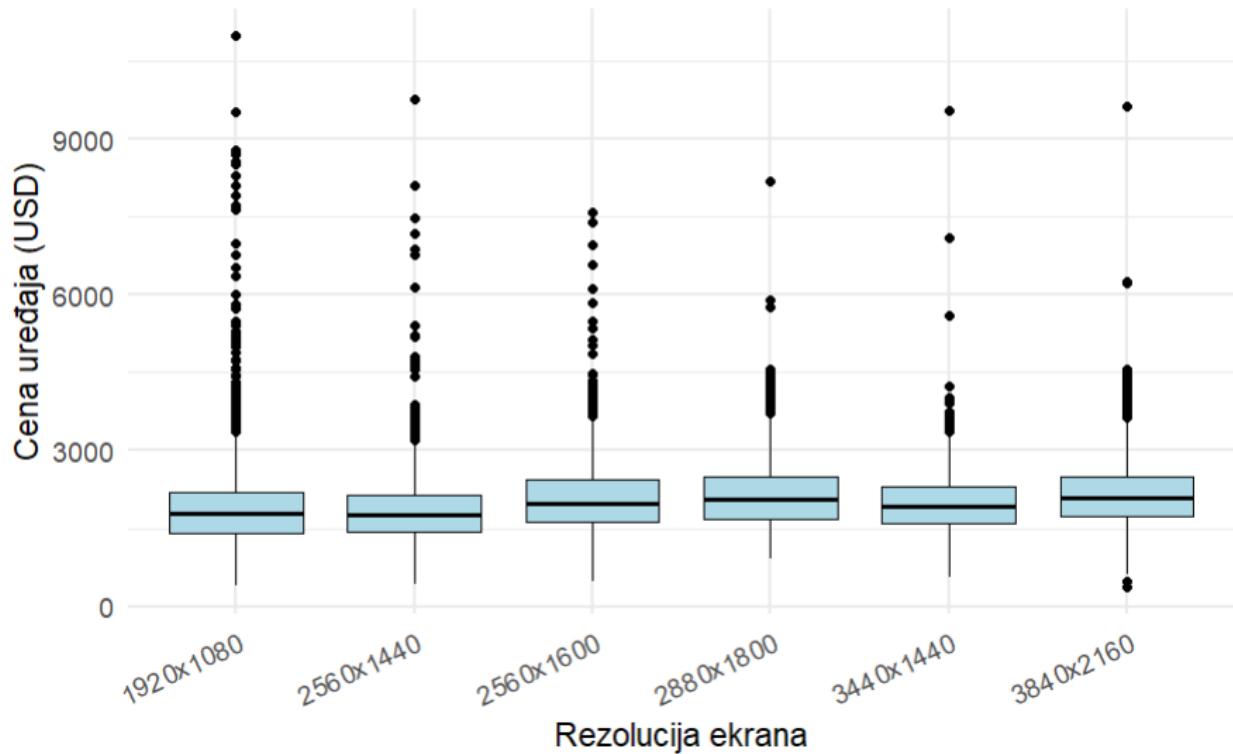
vrstom memorije, ali je verovatno ima dosta, jer su ostale komponente skuplje i jače. Dobar primer je RAM, ako imamo mnogo RAM-a, onda nam brzina eksterne memorije nije presudna, te je bolje uzeti sporiju i jeftiniju eksternu memoriju.

Prirodno pitanje koje se postavlja je da li bi nam multivarijanta analiza tipa eksterne memorije zajedno sa njenom veličinom u odnosu na cenu dala neke bolje rezultate. Ispostavlja se da nam ne bi dala nikakvo poboljšanje, najviše što bismo dobili time je potvrdu prethodno ustanovljenih trendova.

```
ggplot(data, aes(x = resolution, y = price)) +  
  geom_boxplot(fill = "lightblue", color = "black") +  
  labs(  
    title = "Raspodela cena u odnosu na rezoluciju ekrana",  
    x = "Rezolucija ekrana",  
    y = "Cena uređaja (USD)")  
  ) +  
  theme_minimal(base_size = 14) +  
  theme(  
    plot.title = element_text(hjust = 0.5, face = "bold"),  
    axis.text.x = element_text(angle = 25, vjust = 1, hjust = 1)  
  )
```

*Slika 30 R kod za crtanje boxplot-ova cene u zavisnosti od rezolucije*

## Raspodela cena u odnosu na rezoluciju ekrana



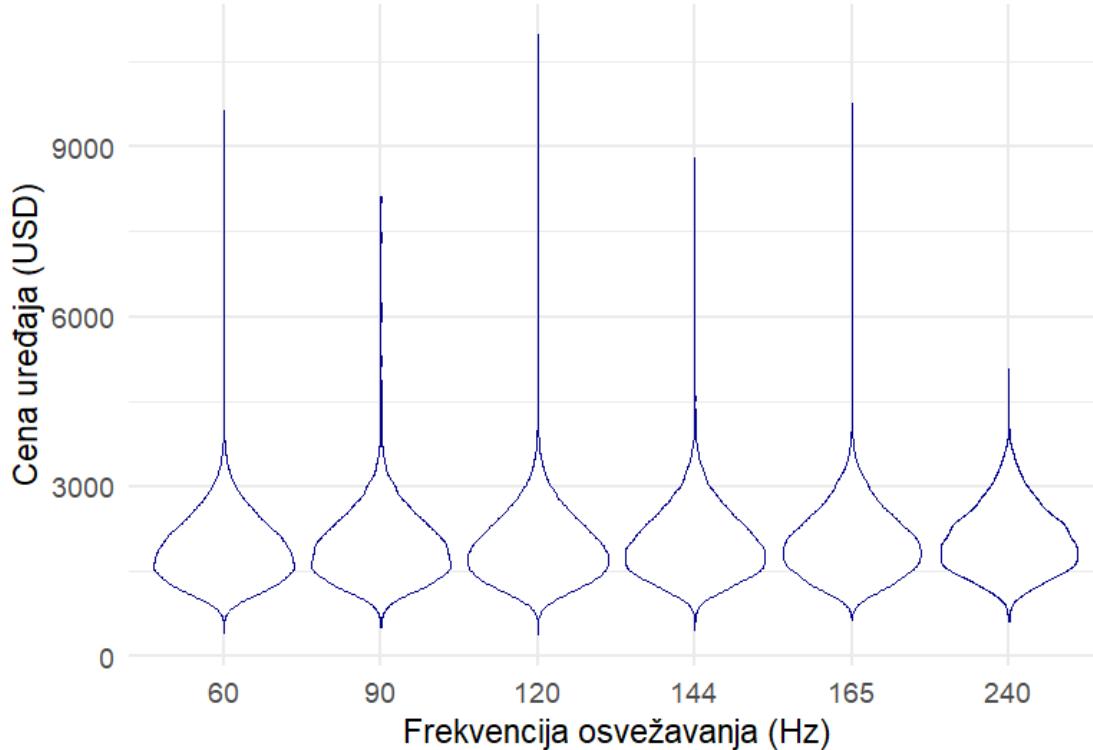
Slika 31 Boxplot-ovi cene u odnosu na rezoluciju ekrana

Sa grafika vidimo da medijana blago raste sve do rezolucije 2880x1800, nakon toga opada i na samom kraju ponovo raste. Kod svake rezolucije postoje jako skupi modeli što je sasvim u redu. Takođe je bitno napomenuti da ima nekoliko uređaja sa izuzetno malom cenom oko 500\$, a sa najvećom rezolucijom 3840X2160, što ne prati trend nikako i može predstaviti veliki problem u treniranju modela.

```
ggplot(data, aes(x = factor(refresh_hz), y = price)) +  
  geom_violin(color = "darkblue") +  
  labs(  
    title = "Cena uređaja u odnosu na frekvenciju osvežavanja",  
    x = "Frekvencija osvežavanja (Hz)",  
    y = "Cena uređaja (USD)")  
  ) +  
  theme_minimal(base_size = 14) +  
  theme(  
    plot.title = element_text(hjust = 0.5, face = "bold")  
  )
```

Slika 32 R kod za crtanje grafika zavisnosti cene od frekvencije osvežavanja

## Cena uređaja u odnosu na frekvenciju osvežavanja

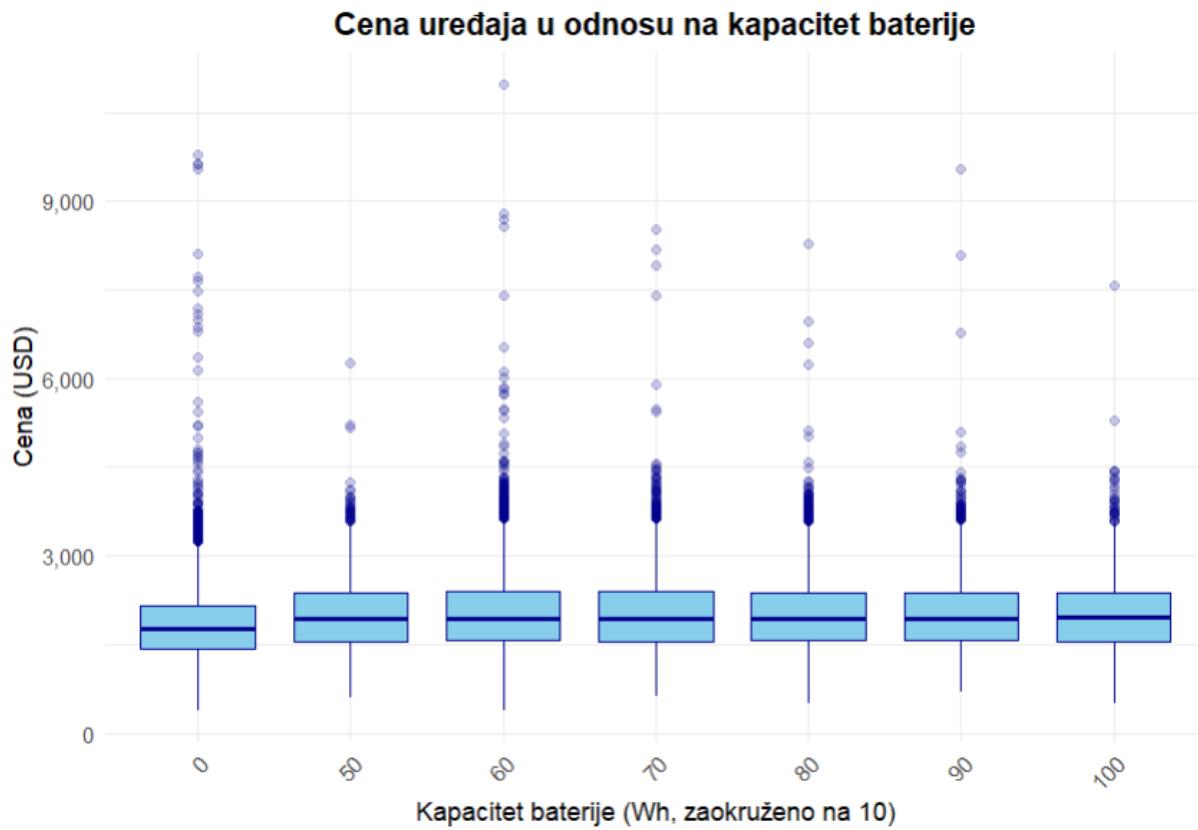


Slika 33 Grafik zavisnosti cene od frekvencije osvežavanja ekrana uređaja

Sa grafika se vidi da najveći broj uređaja ima 60Hz frekvenciju osvežavanja ekrana i cena je uglavnom u srednjem i nižem rangu. Sa povećanjem frekvencije ne raste cena, za svaku frekvenciju većina uređaja ima cenu do 1500\$, dosta većih vrednosti ima svugde, najviše kod 60 i 120 i 165Hz. Nema nekog pravilnog povećanja ili smanjenja, pa ovo obeležje i nije toliko bitno.

```
ggplot(data, aes(x = factor(round(battery_wh, -1)), y = price)) +  
  geom_boxplot(outlier.alpha = 0.2, fill = "skyblue", color = "darkblue") +  
  scale_y_continuous(labels = scales::comma) +  
  labs(  
    title = "Cena uređaja u odnosu na kapacitet baterije",  
    x = "Kapacitet baterije (Wh, zaokruženo na 10)",  
    y = "Cena (USD)")  
  ) +  
  theme_minimal() +  
  theme(  
    plot.title = element_text(hjust = 0.5, face = "bold"),  
    axis.text.x = element_text(angle = 45, hjust = 1)  
  )
```

Slika 34 R kod za crtanje boxplot-ova zavisnosti cene od kapaciteta baterije uređaja

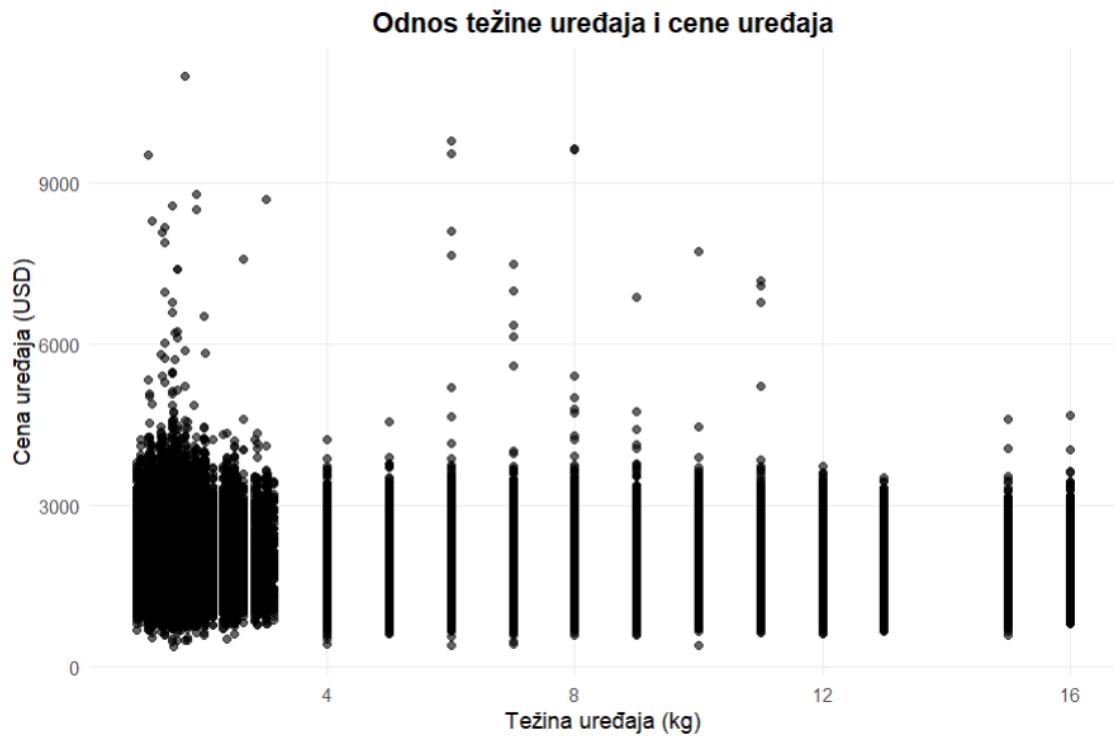


Slika 35 Boxplot-ovi zavisnosti cene od kapaciteta baterije uređaja

Znamo da su uređaji sa 0Wh baterije računari, a svi ostali su laptopovi. Sa grafika vidimo samo da je medijana za računare malo niža od medijane za laptopove, a sa povećanjem broja Wh medijane su skoro identične, tako da nam ovaj prediktor uopšte nije bitan.

```
ggplot(data, aes(x = weight_kg, y = price)) +
  geom_point(color = "black", alpha = 0.6) +
  theme_minimal() +
  labs(
    title = "Odnos težine uređaja i cene uređaja",
    x = "Težina uređaja (kg)",
    y = "Cena uređaja (USD)"
  ) +
  theme(
    plot.title = element_text(hjust = 0.5, face = "bold"),
    panel.grid.minor = element_blank()
  )
```

Slika 36 R kod za crtanje grafika zavisnosti cene uređaja od njegove težine



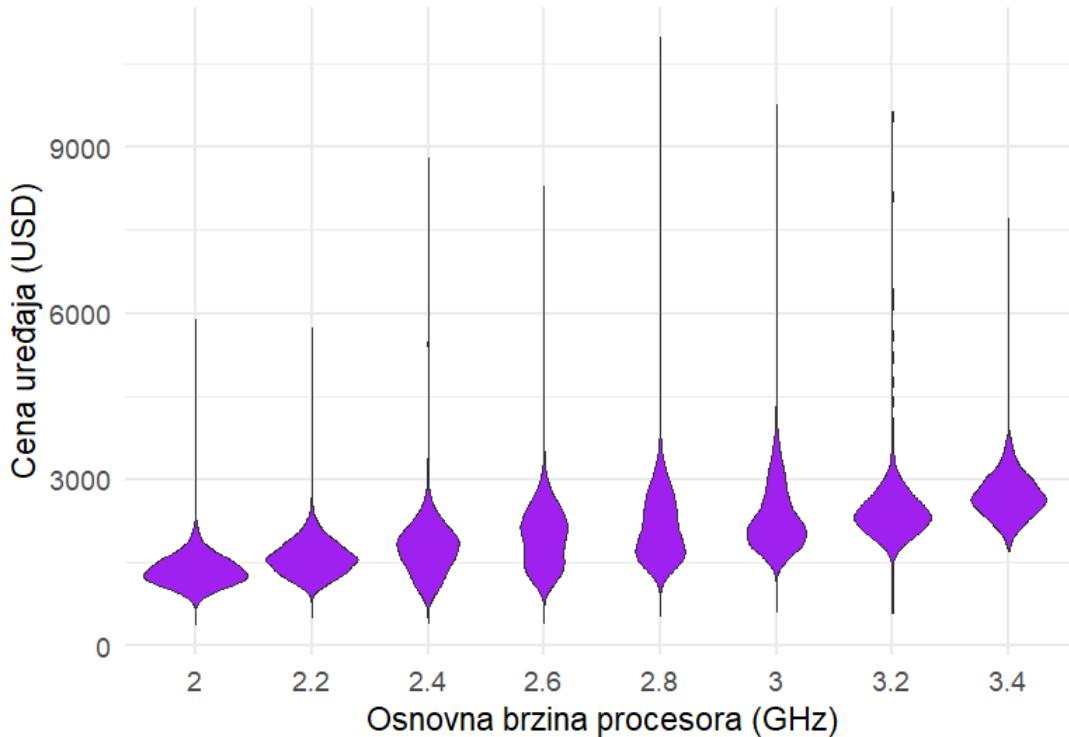
Slika 37 Grafik zavisnosti cene uređaja od njegove težine

Sa grafika vidimo da je većina uređaja težine od 1 do 3.5kg i ima ih sa raznim cenama. Uređaji manji od kilogram mogu potencijalno biti outlieri i potrebno ih je dodatno ispitati. Težina uglavnom i nije neko merilo cene, postoji dosta laganih uređaja raznih cena, isto važi i za dosta teže uređaje.

```
ggplot(data, aes(x = factor(cpu_base_ghz), y = price)) +
  geom_violin(fill = "purple") +
  labs(
    title = "Cena uređaja u odnosu na brzinu procesora",
    x = "Osnovna brzina procesora (GHz)",
    y = "Cena uređaja (USD)"
  ) +
  theme_minimal(base_size = 14) +
  theme(
    plot.title = element_text(hjust = 0.5, face = "bold")
  )
```

Slika 38 R kod za crtanje grafika zavisnosti cene od brzine procesora

## Cena uređaja u odnosu na brzinu procesora

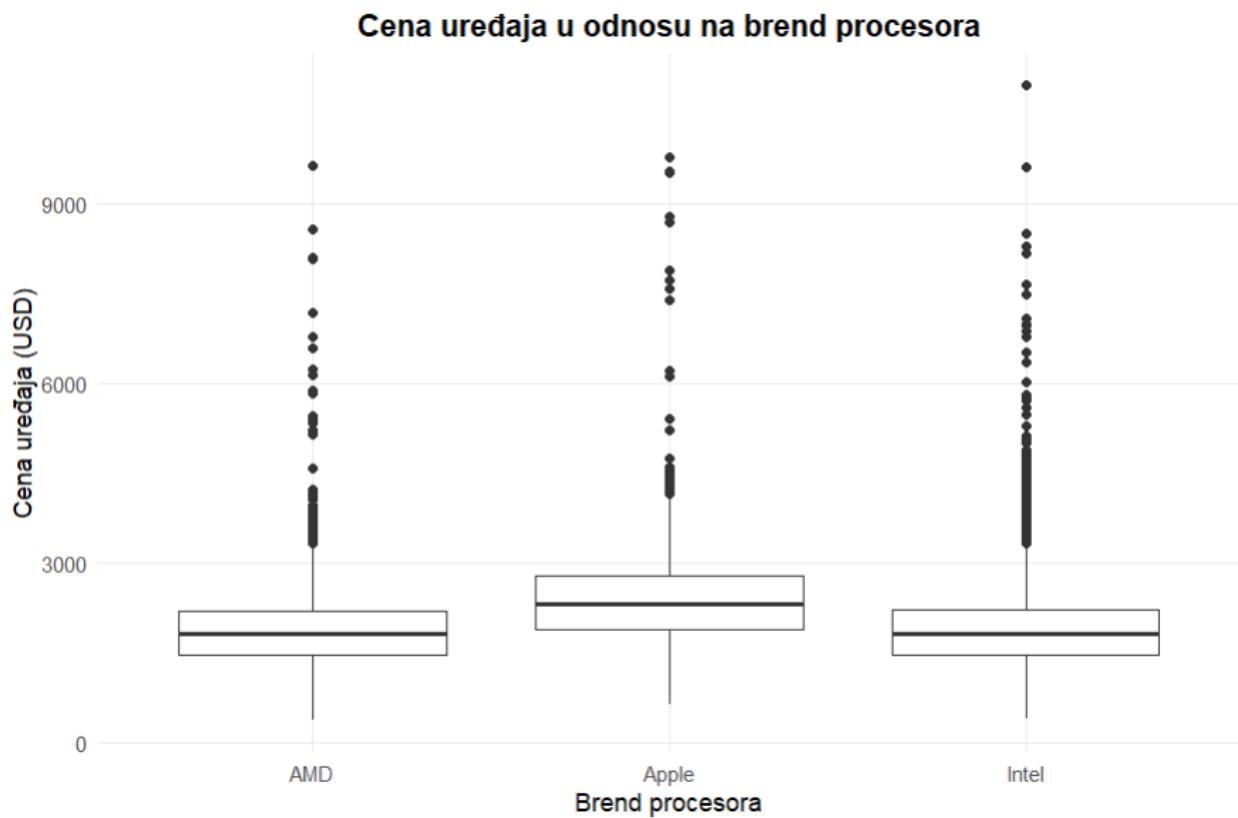


Slika 39 Grafik zavisnosti cene od brzine procesora

Sa grafika vidimo da veza cene i brzine procesora i nije baš linearan, postepeno raste do određenog dela, ali ima i dosta uređaja sa velikom brzinom procesora, a malom cenom, u redu je ako su im npr. druge komponente jeftinije ili može takođe zavisiti od proizvođača procesora.

```
ggplot(data, aes(x = cpu_brand, y = price)) +  
  geom_boxplot() +  
  labs(  
    title = "Cena uređaja u odnosu na brend procesora",  
    x = "Brend procesora",  
    y = "Cena uređaja (USD)"  
) +  
  theme_minimal() +  
  theme(  
    plot.title = element_text(hjust = 0.5, face = "bold"),  
    panel.grid.minor = element_blank()  
)
```

Slika 40 R kod grafika zavisnosti cene od proizvođača procesora

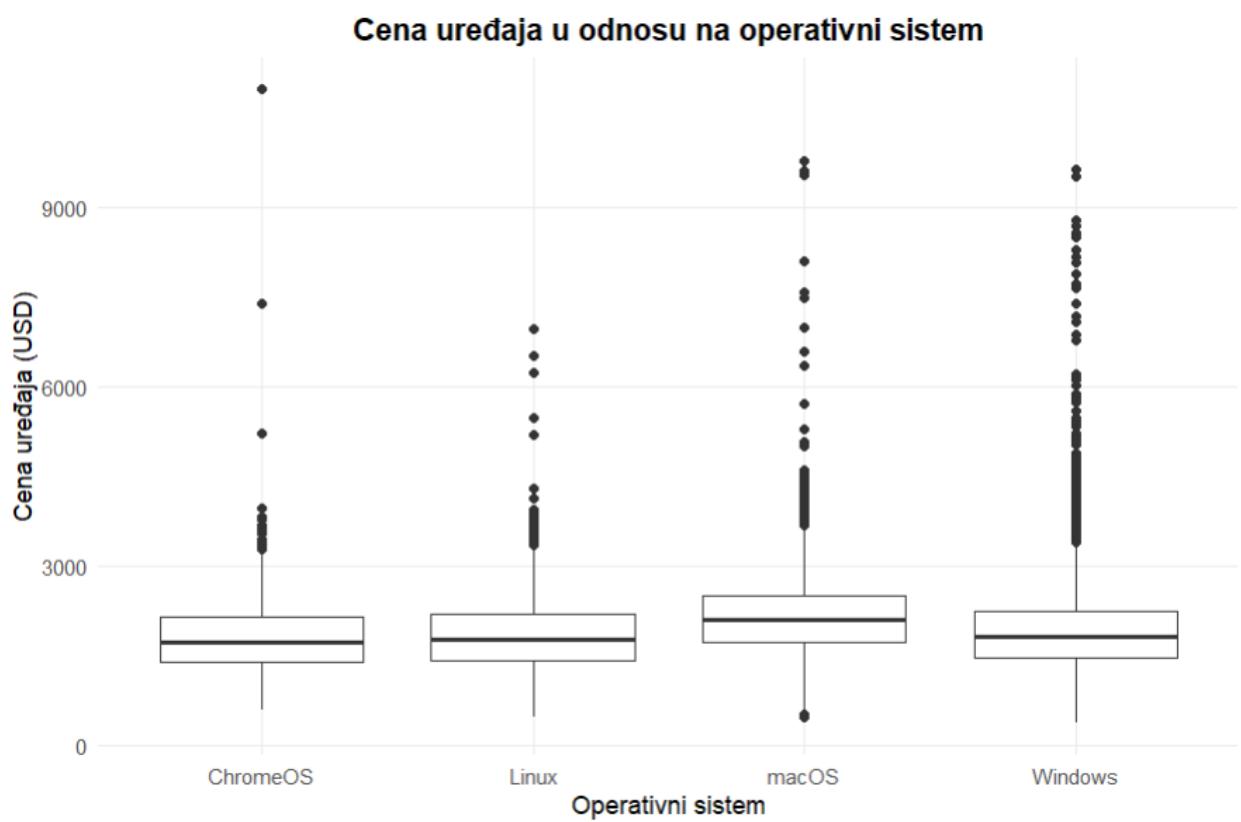


*Slika 41 Boxplot-ovi zavisnosti cene uređaja u odnosu na proizvođača procesora*

Možemo videti cene uređaja u odnosu na proizvođača procesora. Apple prednjači u odnosu na AMD i Intel što je i očekivano, jer oni obično i dolaze sa skupljim komponentama. Svi imaju dosta skupe uređaje što je potrebno ispitati, posebno uređaje iznad 9000\$, kojih je samo nekoliko.

```
ggplot(data, aes(x = os, y = price)) +
  geom_boxplot() +
  labs(
    title = "Cena uređaja u odnosu na operativni sistem",
    x = "Operativni sistem",
    y = "Cena uređaja (USD)"
  ) +
  theme_minimal() +
  theme(
    plot.title = element_text(hjust = 0.5, face = "bold"),
    panel.grid.minor = element_blank()
  )
```

*Slika 42 R kod za crtanje boxplot-ova zavisnosti cene od operativnog sistema uređaja*



*Slika 43 Boxplot-ovi zavisnosti cene od operativnog sistema uređaja*

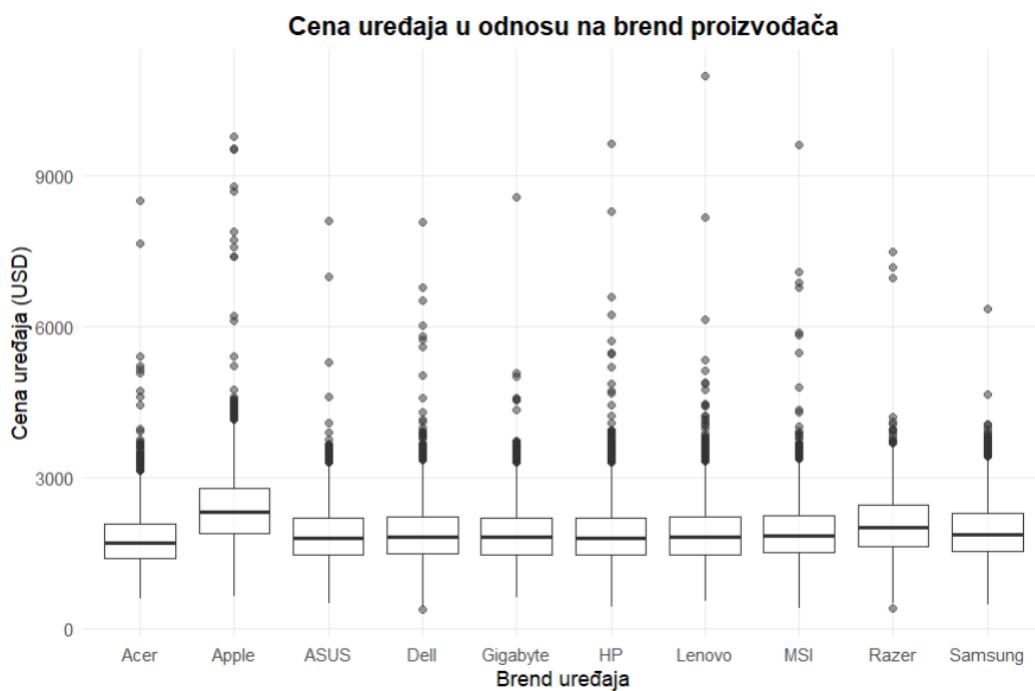
Medijana ChromeOS je najniža što je i očekivano jer je relativno mlad operativni sistem, sa ne toliko širokom upotrebom, a uređaji sa macOS su očekivano najskuplji zbog cene svojih komponenti i svog uticaja na tržištu. Uredaja sa dosta većim cenama ima svugde, najviše sa Windows operativnim sistemom, jer je on najpoznatiji i najkorišćeniji operativni sistem. Izdvaja se uređaj koji ima macOS i cenu ispod 1000\$. To su ekstremne vrednosti, koje bi mogle značajno da umanjuju preciznost modela.

```

ggplot(data, aes(x = brand, y = price)) +
  geom_boxplot() +
  theme_minimal() +
  labs(
    title = "Cena uređaja u odnosu na brend proizvođača",
    x = "Brend uređaja",
    y = "Cena uređaja (USD)"
  ) +
  theme_minimal() +
  theme(
    plot.title = element_text(hjust = 0.5, face = "bold"),
    panel.grid.minor = element_blank()
  )

```

Slika 44 R kod za crtanje boxplot-ova zavisnosti cene od proizvođača uređaja



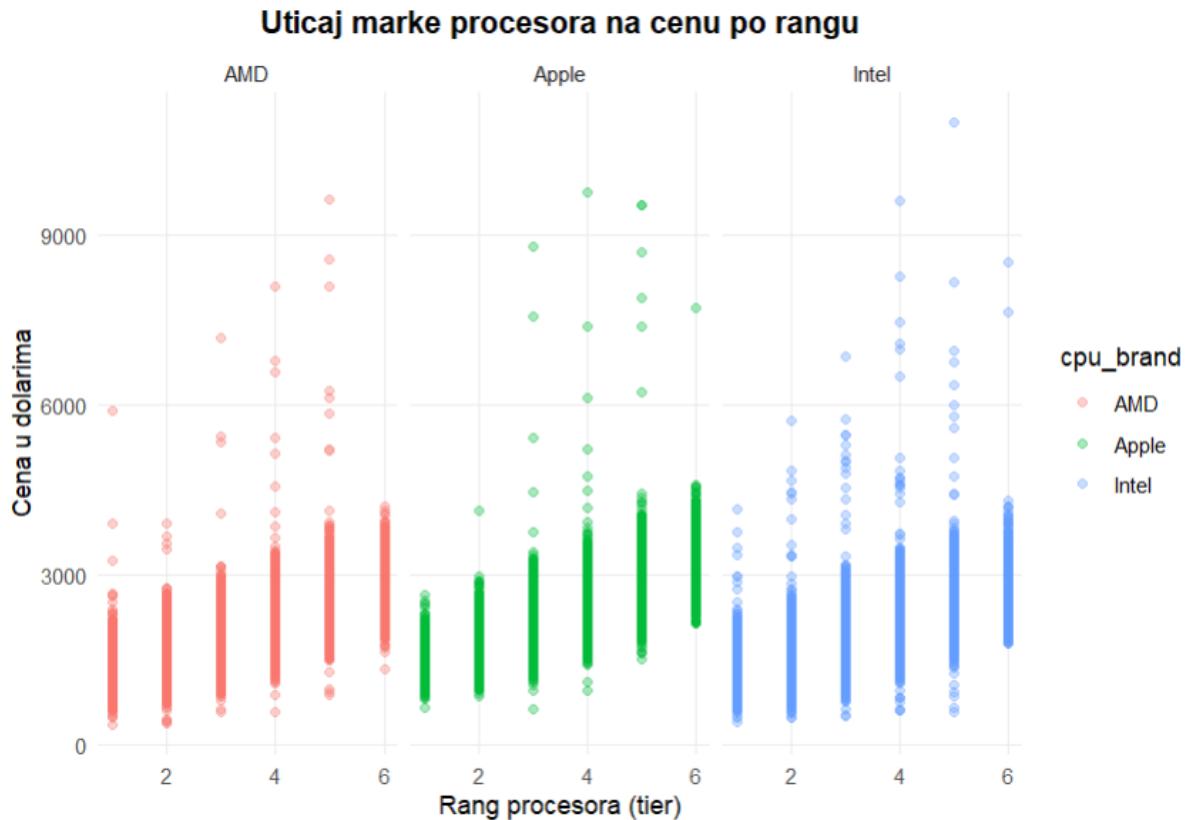
Slika 45 Boxplot-ovi zavisnosti cene od proizvođača uređaja

Medijane su uglavnom slične, Razer ima malo veću od ostalih, a Apple dosta ali i očekivano, Apple uređaji su uvek najskuplji. Kod HP, Lenovo i MSI brenda postoje modeli skuplji od 9000\$ što je potrebno ispitati i potencijalni outlieri jesu i modeli Dell i Razer marke koji su jeftiniji od 700\$ dolara. Ovo je važan prediktor, ali je uglavnom dosta bitniji u kombinaciji sa drugim prediktorima.

## Multivariantna analiza

```
ggplot(data, aes(x = cpu_tier, y = price, color = cpu_brand)) +  
  geom_point(alpha = 1/3) +  
  facet_wrap(~cpu_brand) +  
  theme_minimal() + labs(  
    title = "Uticaj marke procesora na cenu po rangu",  
    x = "Rang procesora (tier)",  
    y = "Cena u dolarima"  
) +  
  theme(  
    plot.title = element_text(hjust = 0.5, face = "bold"),  
    panel.grid.minor = element_blank()  
)
```

Slika 46 R kod za crtanje scatter dijagrama uticaja marke procesora na cenu po rangu



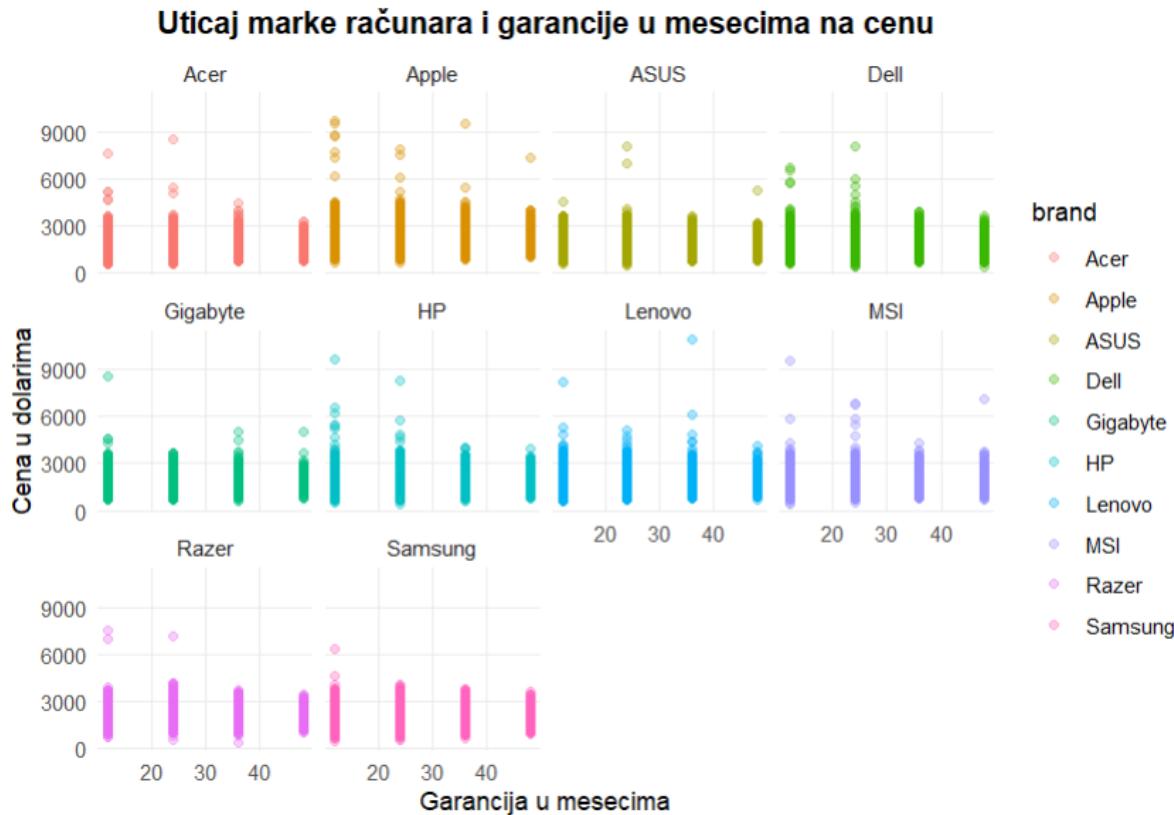
Slika 47 Scatter dijagram uticaja marke procesora na cenu po rangu

Na scatter grafiku iznad je prikazan uticaj marke po rangu procesora na cenu. Imamo tri marke procesora i šest nivoa procesora. Odmah na startu vidimo da je Apple najskuplji po svim nivoima. Vidimo da sve marke imaju dosta outliera, gde Intel ima najviše i to najčešće za tier vrednosti od 2 do 4. To nam govori, ono što smo već

mogli da naslutimo da je Intel najfleksibilniji procesor, koji se kombinuje sa dosta drugih komponenti, koje mogu uticati na cenu uređaja.

```
ggplot(data, aes(x = warranty_months, y = price, color = brand)) +
  geom_point(alpha = 1/3) +
  facet_wrap(~ brand) +
  theme_minimal() + labs(
    title = "Uticaj marke računara i garancije u mesecima na cenu",
    x = "Garancija u mesecima",
    y = "Cena u dolarima"
  ) +
  theme(
    plot.title = element_text(hjust = 0.5, face = "bold"),
    panel.grid.minor = element_blank()
  )
```

*Slika 48 R kod za crtanje scatter dijagrama uticaja marke računara i garancije u mesecima na cenu*



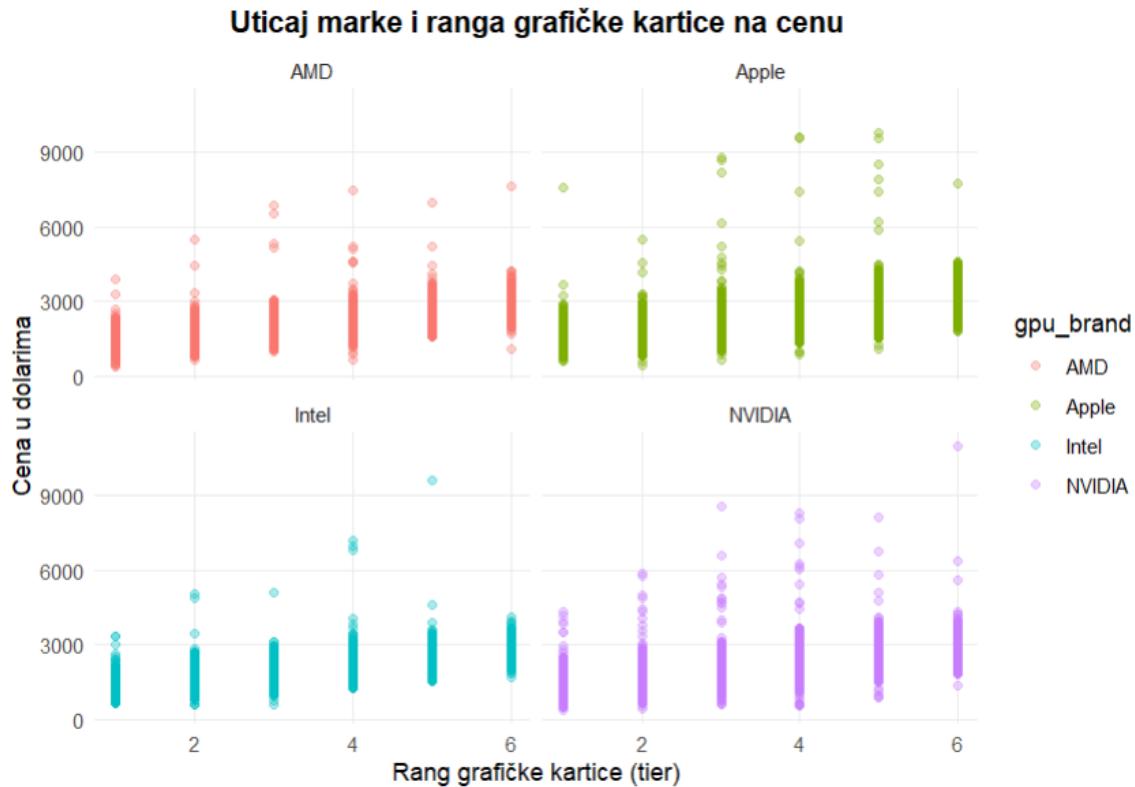
*Slika 49 Scatter dijagram uticaja marke računara i garancije u mesecima na cenu*

Na scatter dijagranu je prikazan uticaj marke računara i garancije u mesecima koje oni pružaju na cenu uređaja. Sa grafika možemo jasno videti da je Apple najskuplji po svim vrednostima garancije. Takođe, možemo videti da garancija ne utiče toliko

na cenu računara, kao što bismo prvo prepostavili. Kod svih brendova garancija od 40+ meseci ima uglavnom nižu cenu od svih ostalih. Najveća zarada kod svih brendova se postiže sa garancijom od 24 meseca, najverovatnije jer je to najčešća vrednost garancije i samim tim imamo najviše primera za tu vrednost. Možemo videti da kod svake marke računara imamo nekoliko outliera, gde Lenovo ima najekstremniju vrednost za garanciju od 36 meseci ima cenu od 9.000+ dolara.

```
ggplot(data, aes(x = gpu_tier, y = price, color = gpu_brand)) +  
  geom_point(alpha = 1/3) +  
  facet_wrap(~ gpu_brand) +  
  theme_minimal() + labs(  
    title = "Uticaj marke i ranga grafičke kartice na cenu",  
    x = "Rang grafičke kartice (tier)",  
    y = "Cena u dolarima"  
) +  
  theme(  
    plot.title = element_text(hjust = 0.5, face = "bold"),  
    panel.grid.minor = element_blank()  
)
```

Slika 50 R kod za crtanje scatter dijagrama uticaja marke i ranga grafičke kartice na cenu



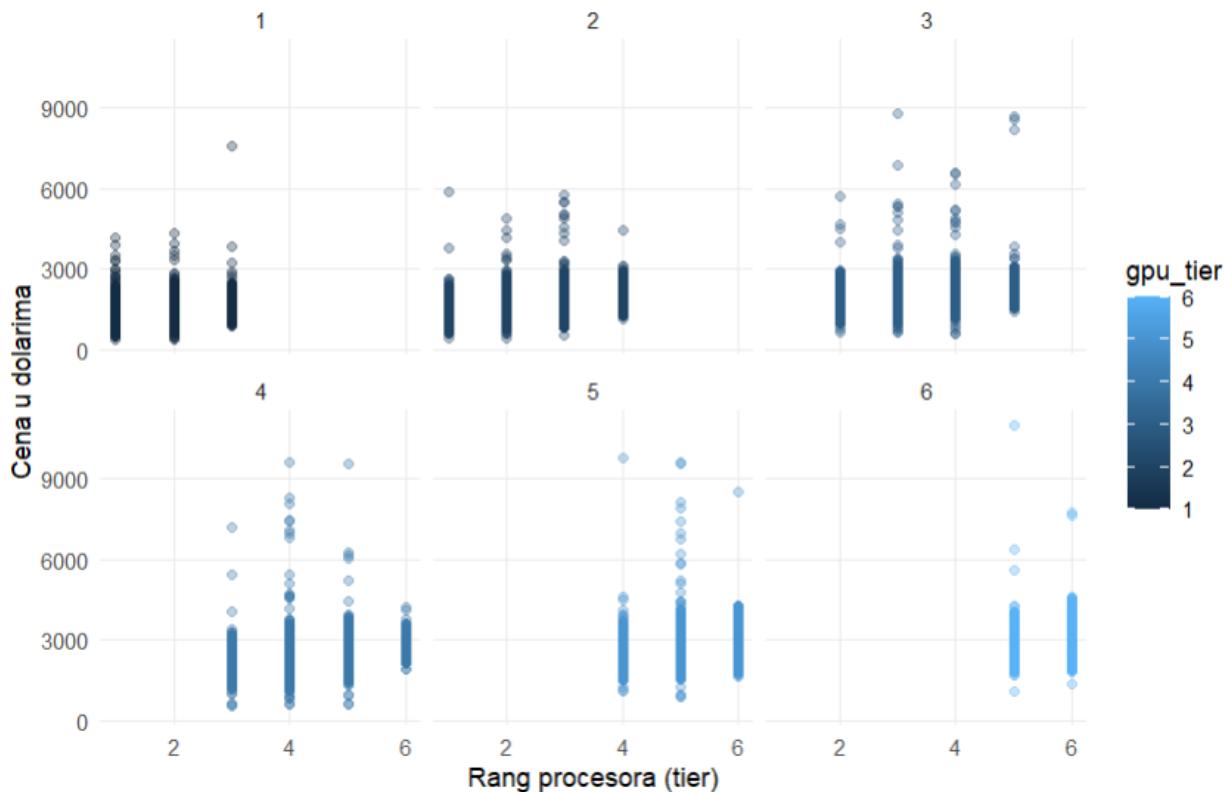
*Slika 51 Uticaj marke i ranga grafičke kartica na cenu*

Na scatter dijagramu sa slike, imamo prikazan uticaj marke i ranga grafičke kartice na cenu. Rang grafičke kartice nam predstavlja dobar prediktor sam po sebi, ali u kombinaciji sa markom bi mogao da postane još bolji prediktor. Vidimo sa slike, kao i do sad, da je Apple skuplji po svim rangovima od ostalih. AMD, Intel i NVIDIA imaju skoro identične cene po svim rangovima. Sa slike se jasno može primetiti da kod svake marke imamo outliere na skoro sve rangove. NVIDIA ima najviše outliera po svim rangovima i jednu ekstremnu vrednost za rang 6, gde je cena viša od 10.000 dolara.

```
ggplot(data, aes(x = cpu_tier, y = price, color = gpu_tier)) +
  geom_point(alpha = 1/3) +
  facet_wrap(~ gpu_tier) +
  theme_minimal() + labs(
    title = "Uticaj ranga procesora i ranga grafičke kartice na cenu",
    x = "Rang procesora (tier)",
    y = "Cena u dolarima"
  ) +
  theme(
    plot.title = element_text(hjust = 0.5, face = "bold"),
    panel.grid.minor = element_blank()
  )
```

*Slika 52 R kod za crtanje scatter dijagraama uticaja ranga procesora i grafičke kartice na cenu*

### Uticaj ranga procesora i ranga grafičke kartice na cenu



Slika 53 Scatter dijagram uticaja ranga procesora i grafičke kartice na cenu

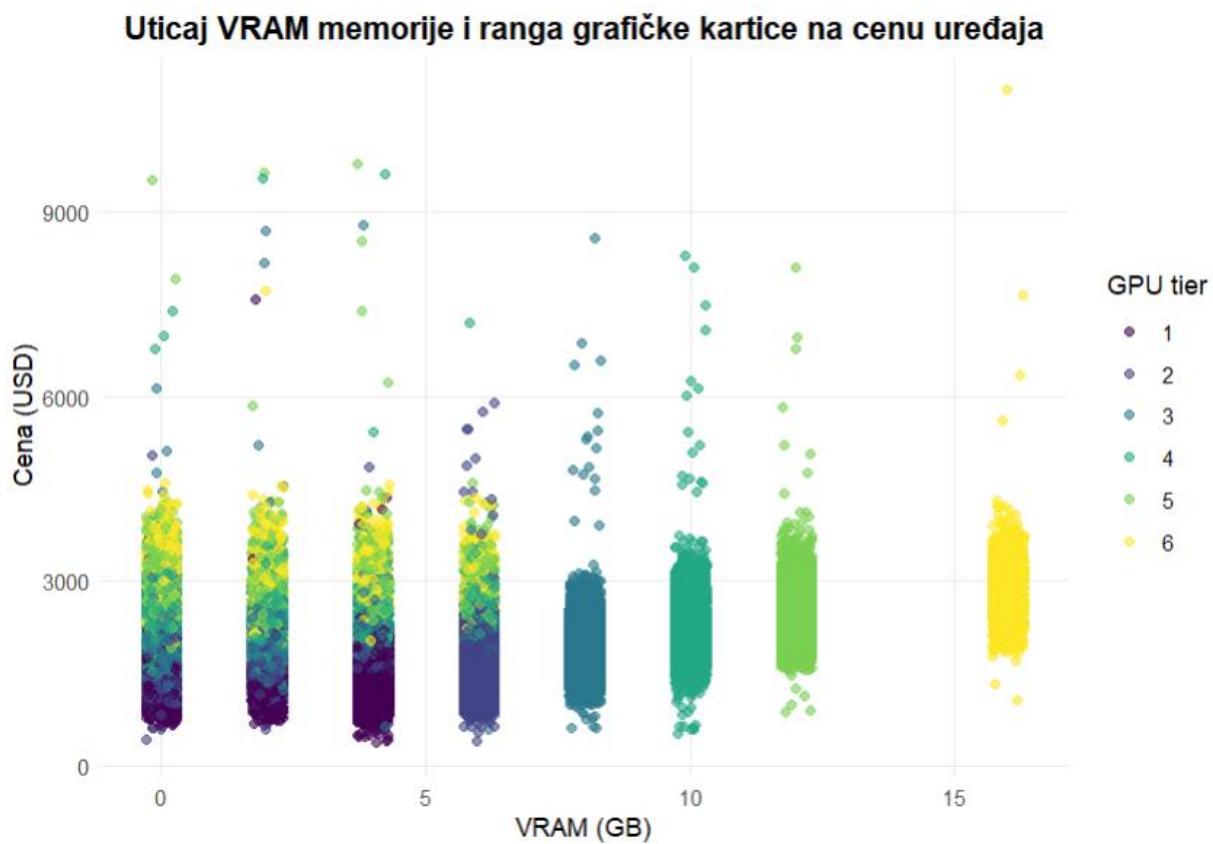
Na scatter dijagranu je prikazan odnos ranga procesora i ranga grafičke kartice i njihovog uticaja na cenu. Prvo što primećujemo sa slike je da neke vrednosti procesora za određene vrednosti grafičke kartice ne postoje. To nije greška, to nam u stvari govori o tome kako određene komponente komuniciraju jedna sa drugom i kako se kombinuju. Vidimo da npr. za rang 1 grafičke kartice imamo podatke samo do četvrtog ranga procesora, isto tako možemo videti da za rang 6 grafičke kartice imamo samo rangove 5 i 6 procesora. Možemo zaključiti da sa porastom ranga jedne komponente raste i rang druge komponente, a sa obzirom na to da su rangovi obe komponente dobar prediktor, možemo zaključiti da ćemo samo porastom jedne od te dve komponente povećati cenu i bez razmatranja druge komponente. Možemo takođe primetiti da imamo nekoliko outliera, ali ni jedan po X osi, što nam ponovno potvrđuje njihovu međusobnu vezu i kompatibilnost komponenti.

```

ggplot(data, aes(x = vram_gb, y = price, color = factor(gpu_tier))) +
  geom_jitter(alpha = 0.6, width = 0.3) +
  scale_color_viridis_d() +
  labs(
    title = "Uticaj VRAM memorije i ranga grafičke kartice na cenu uređaja",
    x = "VRAM (GB)",
    y = "Cena (USD)",
    color = "GPU tier"
  ) +
  theme_minimal() +
  theme(
    plot.title = element_text(hjust = 0.5, face = "bold"),
    panel.grid.minor = element_blank()
  )

```

*Slika 54 R kod za iscrtavanje grafika uticaja VRAM memorije i ranga grafičke kartice na cenu uređaja*



*Slika 55 Grafik uticaja VRAM memorije i ranga grafičke kartice na cenu uređaja*

Graf sa slike iznad pokazuje zajednički uticaj VRAM memorije i ranga grafičke kartice na cenu. Vidimo da cena raste sada pravilnije porastom ranga gpu-a i količine VRAM memorije, od 7GB VRAM-a, pa nadalje. Slabiji uređaji, sa malo ili 0GB VRAM-a se

nalaze uglavnom u najslabijoj kategoriji gpu-a, što ukazuje na integrisane (sa OGB VRAM-a) ili slabe (sa samo nekoliko GB VRAM-a) grafičke kartice kod ovih uređaja. Takođe pokazuje da ova dva prediktora zajedno imaju dobar uticaj na cenu. Ono što odskače jeste da postoje i skupi uređaji, sa malo ili nimalo VRAM-a, ali sa najvećom ili skoro najvećom klasom grafičke kartice, što opravdava njihovu cenu. Manji deo uređaja ima dosta visoku cenu, nizak rang grafičke kartice i malo ili nimalo VRAM-a, što pokazuje da za te uređaje postoji i neki treći prediktor (ili više njih) zbog koga su ove cene opravdane, npr. jak procesor, dosta RAM-a i slično.

```
ggplot(data, aes(x = ram_gb, y = price, color = release_year)) +
  geom_jitter(alpha = 0.4) +
  scale_color_viridis_c() +
  labs(
    title = "Uticaj RAM memorije i godine izlaska uređaja na cenu na cenu uređaja",
    x = "RAM (GB)",
    y = "Cena (USD)",
    color = "Godina izlaska"
  ) +
  theme_minimal() +
  theme(
    plot.title = element_text(hjust = 0.5, face = "bold"),
    panel.grid.minor = element_blank()
  )
```

*Slika 56 R kod za iscrtavanje grafika uticaja RAM memorije i godine izlaska uređaja na cenu*

#### **Uticaj RAM memorije i godine izlaska uređaja na cenu na cenu uređaja**



*Slika 57 Grafik uticaja RAM memorije i godine izlaska uređaja na cenu na cenu uređaja*

Grafik iznad je šaren i nema neke linearne povezanosti. Za svaku različitu količinu RAM memorije, postoje uređaji skoro svih godina izlaska i npr postoje uređaji sa 32GB RAM-a iz 2018 i iz 2025 koji su veoma skupi i koji su veoma jeftiniji. Iz ovoga vidimo da količina RAM memorije i nije toliko bitan prediktor, a pokušali smo to da vidimo preko ovog grafika, jer bi iz godine u godinu trebalo da raste frekvencija RAM memorije i da se na nekoliko godina poboljšava tip RAM memorije DDR3, DDR4, DDR5 i tako dalje, a nama je poznata samo količina RAM memorije, pa smo pokušali preko ovog grafa da vidimo da li ima neke povezanosti.

## Čišćenje i obrada podataka

### Uvod

Nakon analize grafika za većinu podataka i uočavanja njihovog efekta na ciljnu promenljivu, potrebno je da sklonimo određen broj podataka, naravno ukoliko je potrebno ili da damo opravdan komentar zbog čega bi ti podaci, iako odskaču od ostalih ili bilo šta slično, trebalo da ostanu netaknuti.

U ovom delu bavićemo se proverom nedostajućih vrednosti (NA, na, N/A, samo prazno ili bilo šta što signalizira da podaci fale), nelogičnih i nemogućih vrednosti, negativnim brojevima za nešto što ne može biti negativno, stvarima koje jedna u kombinaciji sa drugom ne mogu da postoje, pogrešno unetih vrednosti npr. za određeni podatak je uneto intel, a za neki drugi Intel, iako su iste stvari biće posmatrane različito. Ovakvi podaci ne smeju da postoje u našem skupu podataka i svi takvi podaci koji budu uočeni biće uklonjeni ili ispravljeni na odgovarajući način.

Za početak ćemo napraviti novu promenljivu „datav2“ koja će biti ista kao i naš skup podataka i nju ćemo menjati. Originalni skup podataka „data“ će ostati netaknut kako bismo, u slučaju da nešto pogrešimo, uvek imali sve podatke sačuvane.

## Nedostajuće vrednosti

```
colSums(is.na(datav2))
  device_type      brand        model   release_year       os    form_factor  cpu_brand
                0          0          0           0          0          0          0
  cpu_model      cpu_tier  cpu_cores  cpu_threads  cpu_base_ghz  cpu_boost_ghz  gpu_brand
                0          0          0           0          0          0          0
  gpu_model      gpu_tier     vram_gb    ram_gb    storage_type  storage_gb  storage_drive_count
                0          0          0           0          0          0          0
  display_type  display_size_in resolution refresh_hz  battery_wh  charger_watts  psu_watts
                0          0          0           0          0          0          0
  wifi           bluetooth  weight_kg warranty_months    price
                0          0          0           0          0          0
```

*Slika 58 Rezultat funkcije colSums()*

Uz pomoć is.na(datav2) dobijamo matricu gde se svuda, gde su NA vrednosti nalazi TRUE, a gde nisu FALSE. Funkcija colSums() sabira vrednosti po kolonama (TRUE posmatra kao 1, FALSE kao 0) i kao što se može videti na slici u skupu ne postoje vrednosti koje nedostaju.

## Pogrešno unete vrednosti

Analizom skupa podataka vizuelno i analizom grafika iz prethodnog koraka nisu uočene neke pogrešne vrednosti npr. tip uređaja da negde bude Device negde device ili uređaj od 100kg, negativna vrednost cene, memorije ili nečeg sličnog.

## Nelogične vrednosti

```
> nrow(filter(datav2, os == "macOS" & price < 1000))
[1] 35
```

*Slika 59 Broj uređaja koji imaju macOS i koštaju ispod 1000\$*

Sa grafika odnosa cene i os-a su nam bili sumnjivi uređaji sa macOS-om jeftiniji od 1000\$ i sada vidimo da su oni uglavnom od nekog drugog proizvođača što u stvarnosti nije moguće.

```
> macos_nije_apple = datav2 %>% filter(os == "macOS" & brand != "Apple")
> nrow(macos_nije_apple)
[1] 16032
```

*Slika 60 Broj nemogućih uređaja koji imaju macOS i nije ih proizvela Apple kompanija*

Sa slike vidimo da postoji 16032 reda kojima je proizvođač uređaja neka kompanija koja nije Apple, a imaju mac os što nije moguće i nije ni zakonski. Apple ne dozvoljava instaliranje mac os-a na proizvodima drugih kompanija. Ako se ne gleda

samo zvanično, postoje zajednice koje se bave podizanjem mac os-a na ne-Apple računarima i laptopovima, ali to ništa nije oficijalno i ovde su podaci samo o novim uređajima, imaju garanciju.

Zaključak je da iako je 16032 dosta dobar deo od 100k podaci su nerealni i netačni, pa će svi biti uklonjeni. Nakon uklanjanja ostaje 83968, što znači da je uklonjeno oko 16% postojećih podataka.



*Slika 61 Grafik nakon brisanja*

Zna se i da Apple uređaji ne mogu imati neki drugi os osim ako npr. imaju intel procesor što je i bio slučaj do pre nekoliko godina, pa ćemo proveriti i takve slučajeve.

```
> apple_intel_procesor <- datav2 %>% filter(brand == "Apple" & cpu_brand == "Intel")
> nrow(apple_intel_procesor)
[1] 0
```

*Slika 62 Broj uređaja koje je proizvela Apple kompanija i imaju Intel procesor*

Sa slike vidimo da apple uređaja sa intel procesorima nema.

```
> apple_apple_procesor <- datav2 %>% filter(brand == "Apple" & cpu_brand == "Apple")
> nrow(apple_apple_procesor)
[1] 11915
```

*Slika 63 Broj uređaja koje je proizveo Apple sa Apple procesorom*

Postoji 11915 apple uređaja i pošto ih nema sa intel procesorom, uklanjamo sve koji nemaju mac operativni sistem.

```
> apple_nije_macos = datav2 %>% filter(brand == "Apple" & os != "macOS")
> nrow(apple_nije_macos)
[1] 9740
```

*Slika 64 Broj nelogičnih uređaja koje je proizveo Apple koji nemaju mac os*

Postoji 9740 ovakvih redova, ti podaci takođe nisu realni i brišemo ih, uklonjeno je oko 11,5% postojećih podataka.



*Slika 65: Grafik nakon brisanja*

Za prethodne 2 stvari smo koristili domensko znanje, koje nam je mnogo pomoglo da uočimo nepravilnosti i da ih potom ispitamo.

Iako na prvi pogled deluje zabrinjavajuće to što je uklonjeno ~26% podataka, ova odluka je po nama opravdana, jer su ti zapisi bili logički i tehnički nemogući. Radi se o uređajima koji imaju macOS instaliran na hardveru koji nije proizveo Apple, što je ne samo nerealan scenario, već je to i zakonski zabranjeno, pa takvi uređaji ne mogu da postoje. Drugi deo nelogičnih zapisa činili su uređaji koje je navodno Apple proizveo, ali koji nemaju macOS, što takođe nije moguće, jer Apple ne isporučuje svoje proizvode ni sa jednim drugim operativnim sistemom. Mogli smo takve uređaje svrstati u posebnu kategoriju poput "unknown" ili popuniti vrednosti nekim pretpostavkama, ali bi to značilo da u model unosimo veštačke informacije koje ne postoje u stvarnosti i koje bi potencijalno narušile tačnost analize i predikcija. Upravo zato odlučili smo da ove podatke u potpunosti uklonimo, jer se ne radi o gubitku validnih informacija, već o eliminaciji netačnih i nelogičnih redova čije bi prisustvo verovatno smanjilo kvalitet modela.

```
> desktop_with_battery = datav2 %>% filter(device_type == "Desktop" & battery_wh > 0)
> nrow(desktop_with_battery)
[1] 0
> laptop_no_battery <- datav2 %>% filter(device_type == "Laptop" & battery_wh == 0)
> nrow(laptop_no_battery)
[1] 0
```

*Slika 66 Provera još nekoliko nelogičnosti*

Provera da li postoje laptopovi bez baterije ili računari sa baterijom, sa slike se može videti da ne postoje ni jedni ni drugi.

### Analiza i potencijalno izbacivanje outlier i high leverage tačaka

```
> preskupi_2021 = datav2 %>% filter(release_year == 2021 & price > 9000)
> nrow(preskupi_2021)
[1] 0
```

*Slika 67 Broj preskupih uređaja iz 2021*

Sa grafika zavisnosti cene od godine izdavanja uređaja u 2021 godini postoji uređaj sa cenom od preko 9000\$, a te godine još nisu postojale toliko skupe komponente, pa bi trebalo skloniti ovaj uređaj.

Pošto je ovde rezultat 0 znači da je ovaj uređaj već sklonjen ranije zbog nekih nelogičnih vrednosti, što dodatno potvrđuje da ovaj uređaj nije ni bio realan.

```

> preskupi_laptopovi = datav2 %>% filter(device_type == "Desktop" & price > 8000)
> preskupi_racunari = datav2 %>% filter(device_type == "Laptop" & price > 10000)
> nrow(peskupi_laptopovi)
[1] 3
> nrow(peskupi_racunari)
[1] 1

```

*Slika 68 Broj preskupih laptopova i računara*

Sa grafika zavisnosti cene od tipa uređaja ostavićemo laptopove koji koštaju do 10000\$, jer oni sadrže integrisane i skupe komponente, pa i mogu mnogo koštati (oni preko 10k su već nerealni sa bilo kakvim komponentama i brišemo ih – 3 uređaja).

Računari su jeftiniji od laptopova tako da do 8000\$ je maksimalna granica otprilike koliko mogu da koštaju, pa ćemo sve sa cenom preko 8000\$ ukloniti – 1 uređaj.

```

> skupi_slab_gpu = datav2 %>% filter(gpu_tier == 1 & price > 6000)
> nrow(skupi_slab_gpu)
[1] 1

```

*Slika 69 Broj uređaja koji su dosta skupi i imaju slabu grafičku karticu*

Sa grafika zavisnosti cene od ranga grafičke kartice postoje uređaji koji koštaju preko 6000\$, a najnižeg su nivoa grafičke kartice, što nije moguće, kakve god da su im druge komponente i ovaj podatak se dosta ističe od drugih, pa ćemo ga obrisati – 1 uređaj.

```

> jeftini_ogromna_rezolucija = datav2 %>% filter(resolution %in% c("3440x1440", "3840x2160") & price < 500)
> nrow(jeftini_ogromna_rezolucija)
[1] 2

```

*Slika 70 Broj uređaja koji su dosta jeftini i imaju ogromnu rezoluciju*

Sa grafika zavisnosti cene od rezolucije ekrana postoje uređaji sa maksimalnom rezolucijom i cenom ispod 500\$, što je nemoguće kakve god da su druge komponente, pa ćemo ih obrisati – 2 uređaja.

```

> prejeftini_macovi = datav2 %>% filter(os == "macOS" & price < 700)
> nrow(prejeftini_macovi)
[1] 0

```

*Slika 71 Broj prejeftinih uređaja sa mac os-om*

Sa grafika zavisnosti cene od os-a postoje uređaji sa macOS-om ispod 700\$ što je nerealno jeftino čak i za polovne modele, a ovde pričamo o novim uređajima.

Rezultat ovog koda će biti 0 takvih podataka, jer je to bio uređaj koji je prethodno uklonjen zbog nečeg drugog, što nam drugi put potvrđuje da ovaj uređaj nije realan da postoji.

```
> preskupi_chromeos = datav2 %>% filter(os == "ChromeOS" & price > 9000)
> nrow(peskupi_chromeos)
[1] 0
```

*Slika 72 Broj preskupih uređaja koji imaju chrome os*

Sa grafika zavisnosti cene od os-a vidimo da postoje uređaji koji imaju chromeOS, koji uglavnom imaju slabiji uređaji, koji koštaju preko 9000\$, što je nemoguće, kakve god da su im druge komponente, pa ćemo ih obrisati.

Rezultat ovog koda jeste 0, što nam govori da ovaj podatak i treba izbaciti, što smo ovde i drugi put dokazali, jer su uređaji sklonjeni prilikom nekog od prethodnih čišćenja podataka.

Nakon svih čišćenja podataka ostao je 74221 red, odnosno uklonjeno je oko 26% podataka od početnih 100000 redova.

## EDA (Exploratory Data Analysis)

### Uvod

Ova faza je jedna od najvažnijih u analizi podataka. Nakon početne faze, EDA služi za detaljniju analizu i za još bolje upoznavanje sa podacima. Biće nacrtana matrica i grafik korelacije za sva numerička obeležja međusobno. Određena numerička obeležja će biti pretvorena u kategorijska, neka čak i u ordinalna kategorijska, jer je bitan redosled kategorija. Na samom kraju biće ponovo nacrtani samo najbitniji grafici, što i jeste glavni cilj ove faze tj. da se otkriju najbitnija obeležja za budući model, kombinacijom domenskog znanja i svih informacija koje su dobijene dosadašnjom analizom podataka.

## Matrica korelacije

```
> numericke_kolone = datav2 %>% select_if(is.numeric)
> names(numericke_kolone)
[1] "release_year"           "cpu_cores"            "cpu_threads"          "cpu_base_ghz"
[5] "cpu_boost_ghz"          "vram_gb"              "ram_gb"               "storage_gb"
[9] "storage_drive_count"    "display_size_in"      "refresh_hz"           "battery_wh"
[13] "charger_watts"          "psu_watts"            "bluetooth"            "weight_kg"
[17] "warranty_months"        "price"
```

Slika 73 Izdvajanje numeričkih obeležja kako bi se napravila matrica korelacijske

Izdvajamo samo kolone koje su numeričke, jer samo između njih možemo videti korelaciju. Postoji 20 numeričkih obeležja.

Korelacija je statistička mera koja opisuje jačinu i smer linearne povezanosti između dve ili više numeričkih varijabli, tačnije da li se promenom jedne varijable menja i druga. Bliže 1 je sve više pozitivna korelacija tj. kako jedna raste i druga raste, a bliže -1, kako jedna raste druga opada, negativna korelacija, bliže 0, sve je manja povezanost.

```
> matrica_korelacie = cor(numericke_kolone, use = "complete.obs")
> matrica_korelacie
      release_year   cpu_cores   cpu_threads   cpu_base_ghz   vram_gb   ram_gb   storage_gb   storage_drive_count   display_size_in
release_year  1.0000000000 -0.0049151412 -0.0053996840 -0.007464735 -0.006395329 -0.005305932 -0.004379751 -0.0021320282 -0.0001312524 -0.005980845
cpu_cores     -0.0049151412  1.0000000000  0.9710522947  0.781831865  0.751651193  0.493102363  0.9046763742 -0.0007233065  0.0145386526  0.0325083828
cpu_threads    -0.0053996840  0.9710522947  1.0000000000  0.760812318  0.731313378  0.504665372  0.878182226 -0.0002783393  0.0136478804  0.031242966
cpu_base_ghz   -0.0074647349  0.7818318652  0.7608123177  1.0000000000  0.960542934  0.457533533  0.7579372820 -0.0026262519  0.3070673126  0.546495932
cpu_boost_ghz  -0.0063953294  0.7516511932  0.7313133784  0.960542934  1.000000000  0.439660639  0.7282024635 -0.0033801775  0.2965267755  0.524887143
vram_gb       -0.0053059323  0.4931023628  0.5406653724  0.457533533  0.439660639  1.000000000  0.5586344445  0.0054513629  0.0354120617  0.059371241
ram_gb        -0.0043379751  0.9046763742  0.878182226  0.757937282  0.728202463  0.558634444  1.0000000000 -0.0022216923  0.0220682216  0.045467030
storage_gb    -0.0021320282  -0.0007233065 -0.0002783393 -0.002626252 -0.003380178  0.005451363  -0.0022216923  1.0000000000 -0.0016862275 -0.002926738
storage_drive_count -0.0001312524  0.0145386526  0.0136478804  0.3070673138  0.296526776  0.035412062  0.0220682216 -0.0016862275  1.0000000000  0.500453587
display_size_in -0.0059808452  0.0325083828  0.0312429657  0.546495932  0.524887143  0.059371242  0.0454670296 -0.0029267378  0.5004535873  1.0000000000
refresh_hz     -0.0025447052  0.0008729120  0.0004970592  0.0032947403  0.003207602  0.000933873  0.0004439563  0.0019304090  0.021851473  0.003452860
battery_wh    -0.0047312803  -0.0336275558 -0.0323947403  0.556043058  0.533969704  -0.061974214  -0.0467971970  0.0071013743 -0.5082158791  -0.897426462
charger_watts -0.0037478541  -0.0320507647 -0.0296925266  0.469501768  0.450464328  -0.0427394731  0.0055529580  0.0048446690  0.4994640331  0.881999033
psu_watts      -0.0041913046  0.0333026057  0.032074621  0.545656513  0.5242718319  0.058757784  0.0455529580  -0.0001258576  0.0011382000  0.0005121266
bluetooth     0.0030303383  0.0004988790  0.0005384648  -0.000164341  0.002086091  0.0011382000  0.0005121266  -0.0001258576  0.0011382000  0.0005121266
weight_kg      -0.0042450163  0.0306741130  0.0293688337  0.498488239  0.48659955  0.053862768  0.0428785740  -0.0051747499  0.4520321889  0.808658394
warranty_months -0.0035952267  0.0011834408  0.00163885505 -0.001511402  0.001631755  0.001011388  -0.0015777659  -0.0050126700  0.0019029778  -0.002276035
price          0.0932419877  0.7294063915  0.6688685152  0.535839298  0.514351441  0.4435724844  0.091528954  -0.0830715253  0.0034498916  0.1476844413
```

```
      refresh_hz   battery_wh   charger_watts   psu_watts   bluetooth   weight_kg   warranty_months   price
release_year  -0.0025447056  0.004731280  0.0037478541 -0.0041913046  0.0030303383 -0.00427394731  0.0055529580  0.0011382000
cpu_cores      0.0008729120  -0.0336275558 -0.0320507647  0.0333026057  0.0004988790  0.0306741130  0.0011834408  0.7294063915
cpu_threads    -0.0004970592  -0.0323947403  0.0296925266  0.032076023  0.0005384648  0.0293688337  0.00163885505  0.6688685152
cpu_base_ghz   0.0032656443  -0.556043058  -0.469501768  0.5456556129 -0.0001643410  0.498488239  -0.0015114015  0.5358392983
cpu_boost_ghz  0.0032076023  0.533969704  -0.4504643278  0.5242718319  -0.0020860912  0.4786599550  -0.0016317548  0.5143514411
vram_gb        0.0009338730  -0.061974214  -0.0531077371  0.0587577842  0.0206516474  0.0538627676  0.0010113880  0.4435724844
ram_gb         0.0004339563  -0.046797197  -0.0427394731  0.0455529580  0.0011382000  0.0428785740  -0.0015777659  0.7692242645
storage_gb     0.0019330490  0.007101374  0.0015114015  0.0048446690  0.0005121266  -0.0051747499  -0.0050126700  0.091528954
storage_drive_count 0.0021851473  -0.508215879  -0.4268748710  0.4994640331  -0.0001258576  0.4520321889  -0.0019029778  -0.0830715253
display_size_in 0.0034528600  -0.897426462  -0.7525710906  0.8819990332  0.0013813083  0.8086583938  -0.0022760348  -0.1448438256
refresh_hz     1.0000000000  -0.0051988433  -0.0004826231  0.0067488075  0.0022926728  0.0032446104  0.0016365265  0.0792237145
battery_wh    -0.0051988428  1.0000000000  0.7659980643  -0.8958318939  -0.0011695198  -0.817474011  0.0034498916  0.1476844413
charger_watts -0.0040826231  0.765998064  1.0000000000  -0.7515749450  -0.0009602642  0.6856010204  0.0019020314  0.1240175716
psu_watts      0.0067488075  -0.895831894  -0.7515749450  1.0000000000  0.0002057415  0.8241576275  -0.0018446954  -0.1412671910
bluetooth     0.0022926728  -0.001169520  -0.0009602642  0.0002057415  1.0000000000  -0.0001910612  0.0016647573  0.0034327328
weight_kg      0.0032446104  0.817474011  -0.6856010204  0.8241576275  -0.0001910612  1.0000000000  -0.0018055525  -0.1311737207
warranty_months 0.0016365265  0.003449892  0.0019020314  -0.0018446954  -0.0016647573  -0.0018055525  1.0000000000  -0.0004075487
price          0.0792237145  0.147684441  0.1240175716  -0.1412671910  0.0034327328  -0.1311737207  -0.0004075487  1.0000000000
```

Slika 74 Matrica korelacijske

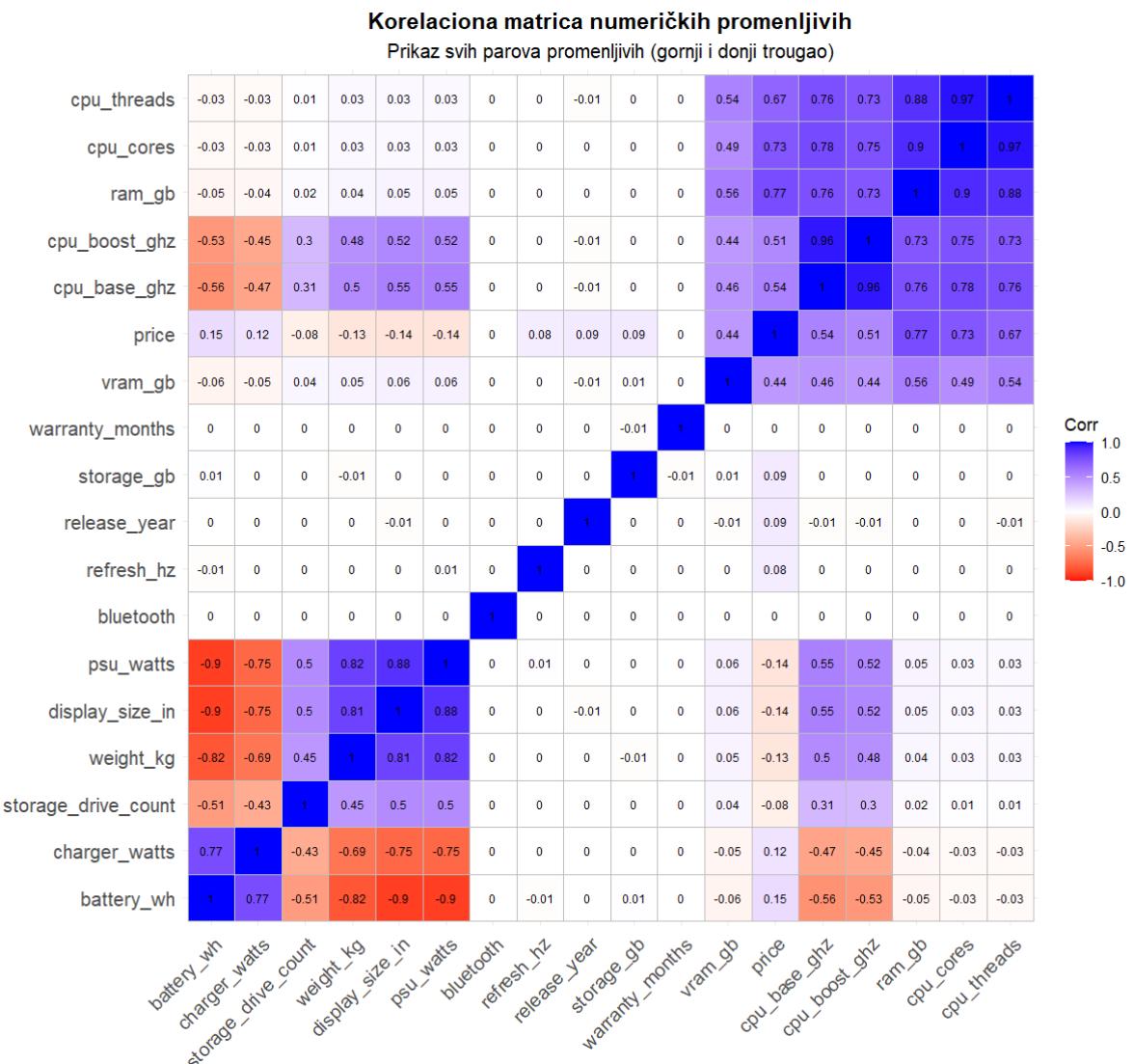
Na slikama iznad se može videti kompletna matrica korelacijske između svaka dva numerička obeležja.

```

ggcorrplot(
  matrica_korelaciije,
  hc.order = TRUE,
  type = "full",
  lab = TRUE,
  lab_size = 2.5,
  colors = c("red", "white", "blue"),
  outline.col = "gray",
  ggtheme = ggplot2::theme_minimal()
) +
  labs(
    title = "Korelaciona matrica numeričkih promenljivih",
    subtitle = "Prikaz svih parova promenljivih (gornji i donji trougao)"
  ) +
  theme(
    plot.title = element_text(hjust = 0.5, face = "bold", size = 14),
    plot.subtitle = element_text(hjust = 0.5, size = 12),
    axis.text.x = element_text(angle = 45, vjust = 1, hjust = 1)
  )
)

```

Slika 75 R kod za iscrtavanje matrice korelacija



Slika 76 Matrica korelacija prikazana grafički

Grafički prikaz malopre pomenute matrice, jer ovakav prikaz umesto gomile brojeva od malopre je mnogo pregledniji i mnogo se lakše mogu uočiti zavisnosti.

Što je više plavo polje jača je pozitivna korelacija, što je više crveno to je veća negativna korelacija. Donji i gornji trougao su preslikani, pa su isti, ali su nacrtana oba kako bi grafik bio lepši i naravno na dijagonali su sve jedinice, svako obeležje ima korelaciju 1 sa samim sobom.

Jake pozitivne korelacije sa cilnjom promenljivom su obeležja gpu\_tier, cpu\_tier, ram\_gb, cpu\_cores, cpu\_threads, cpu\_base\_ghz, cpu\_boost\_ghz. Cena najviše zavisi od hardverskih performansi CPU, GPU, RAM i slično i to će biti naši glavni prediktori za budući model.

Slabe pozitivne korelacije su sa storage\_gb, release\_year, refresh\_hz, bluetooth, warranty\_months. Same po sebi ne utiču mnogo, ali sa nečim u kombinaciji ovo bi se možda moglo promeniti. Negativne korelacije su sa weight\_kg, display\_size\_in, psu\_watts. Vrednosti su uglavnom dosta bliže 0, tako da i nema neke povezanosti.

Snažne međusobne korelacije jesu između cpu\_base\_ghz i cpu\_boost\_ghz, cpu\_tier sa bilo čim iz cpu dela, gpu\_tier i vram i još nekoliko stvari. Uzimaćemo po našoj proceni bitniju od svake dve kako ne bismo došli do multikolinearnosti. Kada su dve ili vise promenljive međusobno visoko korelisane kažemo da su multikolinearne.

```
> sort(matrica_korelacija[,"price"], decreasing = TRUE)
      price          ram_gb        cpu_cores      cpu_threads      cpu_base_ghz
1.000000000000  0.7692242645  0.7294063915  0.6688685152  0.5358392983
cpu_boost_ghz    vram_gb       battery_wh     charger_watts   release_year
0.5143514411  0.4435724844  0.1476844413  0.1240175716  0.0932419877
storage_gb      refresh_hz      bluetooth   warranty_months storage_drive_count
0.0915258954  0.0792237145  0.0034327328 -0.0004075487 -0.0830715253
weight_kg       psu_watts      display_size_in
-0.1311737207 -0.1412671910  -0.1448438256
```

*Slika 77 Korelacija numeričkih obeležja u odnosu na ciljnu promenljivu price*

Na slici iznad prikazane su, sortirane opadajuće, korelacije svih numeričkih obeležja sa cilnjom promenljivom price. Malopre je već rečeno koja obeležja imaju kakve korelacije, slika iznad služi za lepši i jednostavniji prikaz.

## Pretvaranje kategorijskih obeležja u factor obeležja

Sve kategoriske promenljive pretvaramo u factor. Factor je poseban tip promenljive koji služi da predstavi kategoriske promenljive, kažemo R-u da ova obeležja nemaju numeričko značenje već su podaci podeljeni po grupama. Umesto da kategoriske promenljive budu string-ovi pretvaraju se u factor kako bi grafici mogli da se pravilno iscrtaju, da se podaci lakše skladište i da bi moglo da se definišu i ordered promenljive. Ordered znači da postoji prirodan redosled i bitno je kojim redom ide koja kategorija.

```
datav2$device_type = as.factor(datav2$device_type)
datav2$brand = as.factor(datav2$brand)
datav2$os = as.factor(datav2$os)
datav2$storage_type = as.factor(datav2$storage_type)
datav2$cpu_brand = as.factor(datav2$cpu_brand)
datav2$gpu_brand = as.factor(datav2$gpu_brand)
datav2$display_type = as.factor(datav2$display_type)
datav2$resolution = as.factor(datav2$resolution)
datav2$wifi = as.factor(datav2$wifi)
datav2$form_factor = as.factor(datav2$form_factor)
datav2$cpu_tier = factor(
  datav2$cpu_tier,
  levels = sort(unique(datav2$cpu_tier)),
  ordered = TRUE
)
datav2$gpu_tier = factor(
  datav2$gpu_tier,
  levels = sort(unique(datav2$gpu_tier)),
  ordered = TRUE
)
```

*Slika 78 R kod za promenu promenljivih u factor*

Rang cpu-a i rang gpu-a pretvaramo u ordered promenljivu, jer je jako bitan redosled, najgori rang jeste broj 1 i raste sve do najboljeg sa brojem 6. Obeležja poput model ne pretvaramo, jer nisu previše bitna za budući model, ima hiljade i hiljade različitih kategorija, nema nekoliko određenih kategorija po kojima bi se grupisali. Numerička obeležja ostaju onakva kakva i jesu.

## Najbitniji grafici EDA faze i ANOVA statistički test

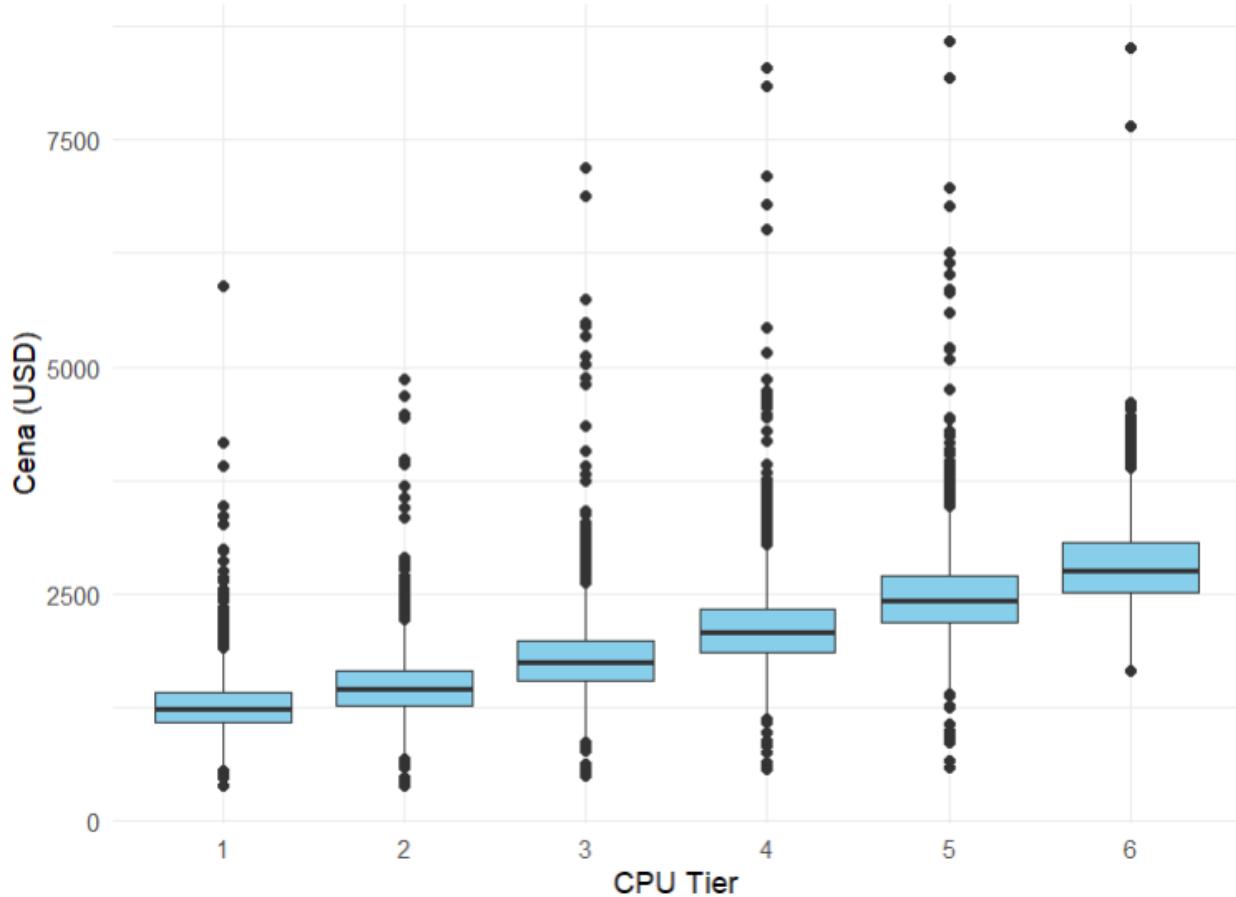
Nakon iscrtavanja početnih grafika i njihove analize, čišćenja i sređivanja podataka i naravno uz pomoć domenskog znanja u EDA fazi izdvojićemo neke grafika koje smatramo najbitnjima. Takođe uradićemo i ANOVA test kako bismo i statistički potvrdili ono što se vidi na grafiku. Test će biti urađen za obeležja `cpu_tier` i `gpu_tier`, jer su kategorisaka i imaju više od 3 kategorije, što je idealno za ovaj test (za obeležja sa dve kategorije bi npr. t-test bio bolji).

ANOVA (Analysis of Variance) je statistička metoda koja se koristi da testira da li bar jedna grupa ima drugačiji prosek od ostalih. Ako je vrednost  $p < 0.05$  to znači da postoje značajne razlike, u našem slučaju, u cenama između grupa.

```
ggplot(datav2, aes(x = cpu_tier, y = price)) +  
  geom_boxplot(fill = "skyblue") +  
  labs(title = "Cena u odnosu na CPU Tier",  
       x = "CPU Tier", y = "Cena (USD)") +  
  theme_minimal()
```

Slika 79 R kod za iscrtavanje grafika zavisnosti cene od ranga CPU-a

### Cena u odnosu na CPU Tier



Slika 80 Boxplot zavisnosti cene od ranga CPU-a

Boxplot-ovi iznad pokazuju da cena jasno raste sa povećanjem ranga procesora, medijana se postepeno povećava. Svaki rang takođe može imati i skuplje i jeftinije uređaje, u zavisnosti od drugih komponenti, procesor je svakako jedna od najbitnijih komponenti uređaja, ali i jačina ostalih komponenti može znatno da smanji ili poveća cenu. Kod većih rangova veći je i raspon cena, skuplji uređaji mogu biti raznih vrsta: gaming uređaji, premium brendovi i slično, negde je jak procesor i ostale komponente su slabije, negde je obrnuto, tako da i cene koje odskaču su sasvim realne. Cene koje odskaču kod nižih rangova su uglavnom gaming računari sa jakom grafičkom karticom ili sa mnogo memorije ili velikim RAM-om.

```

> anova_cpu = aov(price ~ cpu_tier, data = datav2)
> summary(anova_cpu)
      Df    Sum Sq  Mean Sq F value Pr(>F)
cpu_tier      5 1.338e+10 2.676e+09   22135 <2e-16 ***
Residuals 74215 8.973e+09 1.209e+05
---
Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

```

*Slika 81 ANOVA test za cpu\_tier obeležje*

Vrednost p je veoma mala, mnogo manja od praga značajnosti 0.05, što znači da se prosečne cene uređaja među grupama značajno razlikuju i to nije slučajno. Takođe F vrednost je veoma velika, što je ova vrednost veća, razlike između grupa su jače.

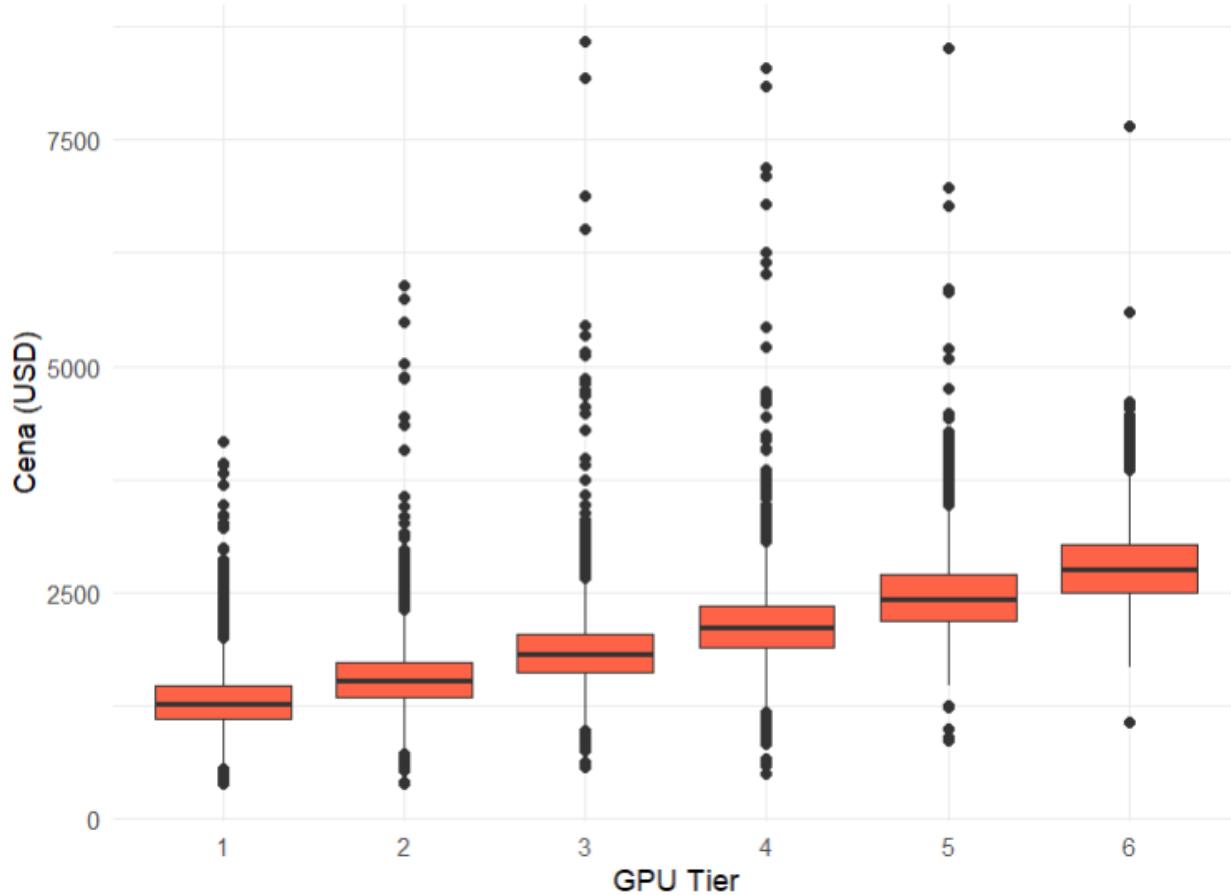
```

ggplot(datav2, aes(x = gpu_tier, y = price)) +
  geom_boxplot(fill = "tomato") +
  labs(title = "Cena u odnosu na GPU Tier",
       x = "GPU Tier", y = "Cena (USD)") +
  theme_minimal()

```

*Slika 82 R kod za iscrtavanje grafika zavisnosti cene od ranga procesora*

Cena u odnosu na GPU Tier



Slika 83 Boxplot zavisnosti cene od ranga procesora

Boxplot-ovi iznad pokazuju da je ovaj prediktor još važniji za cenu, medijane takođe pravilno rastu sa povećanjem ranga procesora. Niži rangovi (1 i 2) uglavnom ne prelaze neki srednji cenovni rang, kao i za prethodno uređaji većih rangova sa velikim cenama su neke profesionalne radne stanice ili neki jako dobri gaming uređaji. Takođe kod nižih rangova nema uređaja koji su mnogo skupi, jer je gpu uglavnom i najskuplja komponenta uređaja (naravno uz procesor).

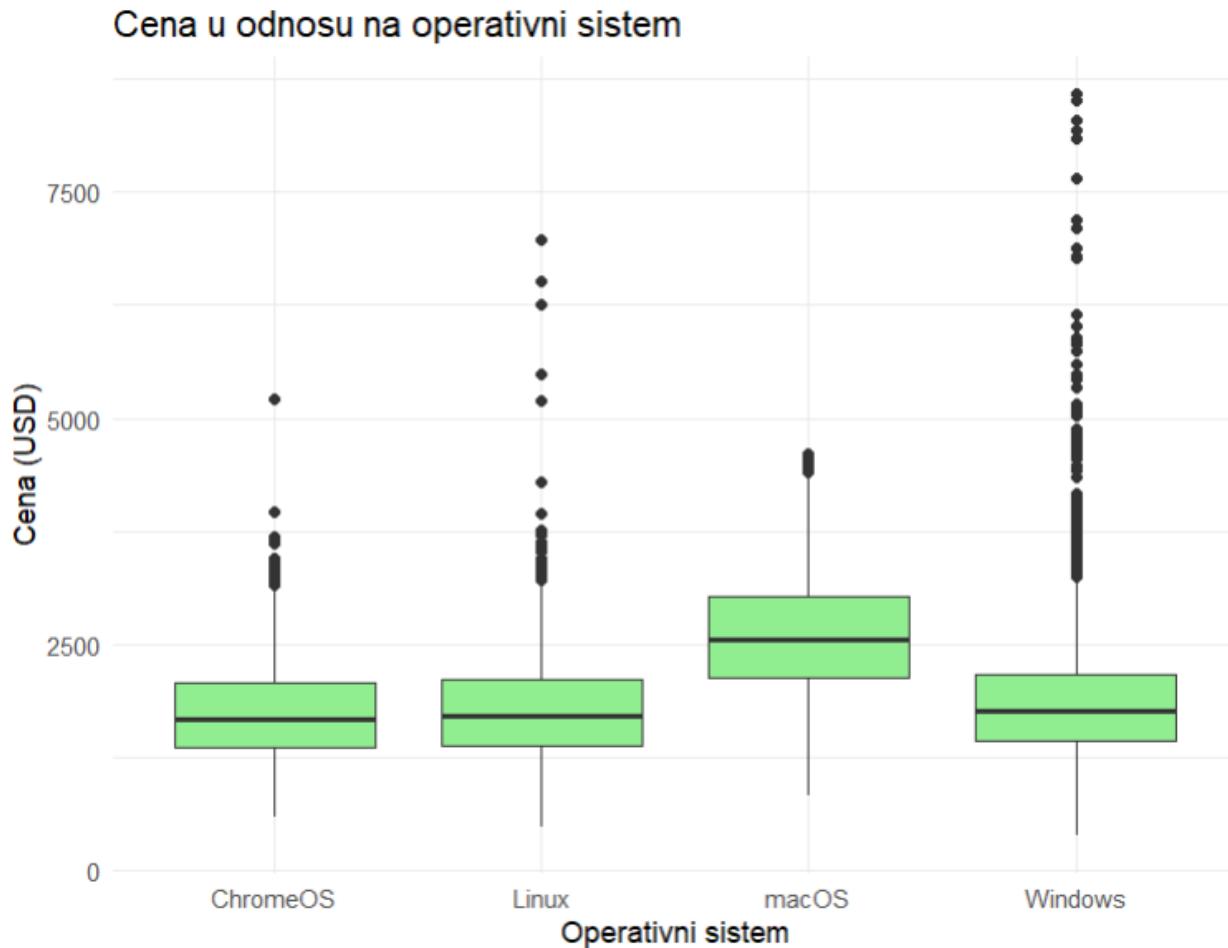
```
> anova_gpu = aov(price ~ gpu_tier, data = datav2)
> summary(anova_gpu)
      Df   Sum Sq  Mean Sq F value Pr(>F)
gpu_tier     5 1.352e+10 2.704e+09  22722 <2e-16 ***
Residuals 74215 8.833e+09 1.190e+05
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Slika 84 ANOVA test za gpu\_tier obeležje

Rezultati testa su slični kao i za prethodno obeležje, čak je F value malo veća, što nam i za ovo obeležje pokazuje da ima statistički značajan uticaj na cenu uređaja.

```
ggplot(datav2, aes(x = os, y = price)) +  
  geom_boxplot(fill = "lightgreen") +  
  labs(title = "Cena u odnosu na operativni sistem",  
       x = "Operativni sistem", y = "Cena (USD)") +  
  theme_minimal()
```

Slika 85 R kod za iscrtavanje grafika zavisnosti cene od vrste operativnog sistema



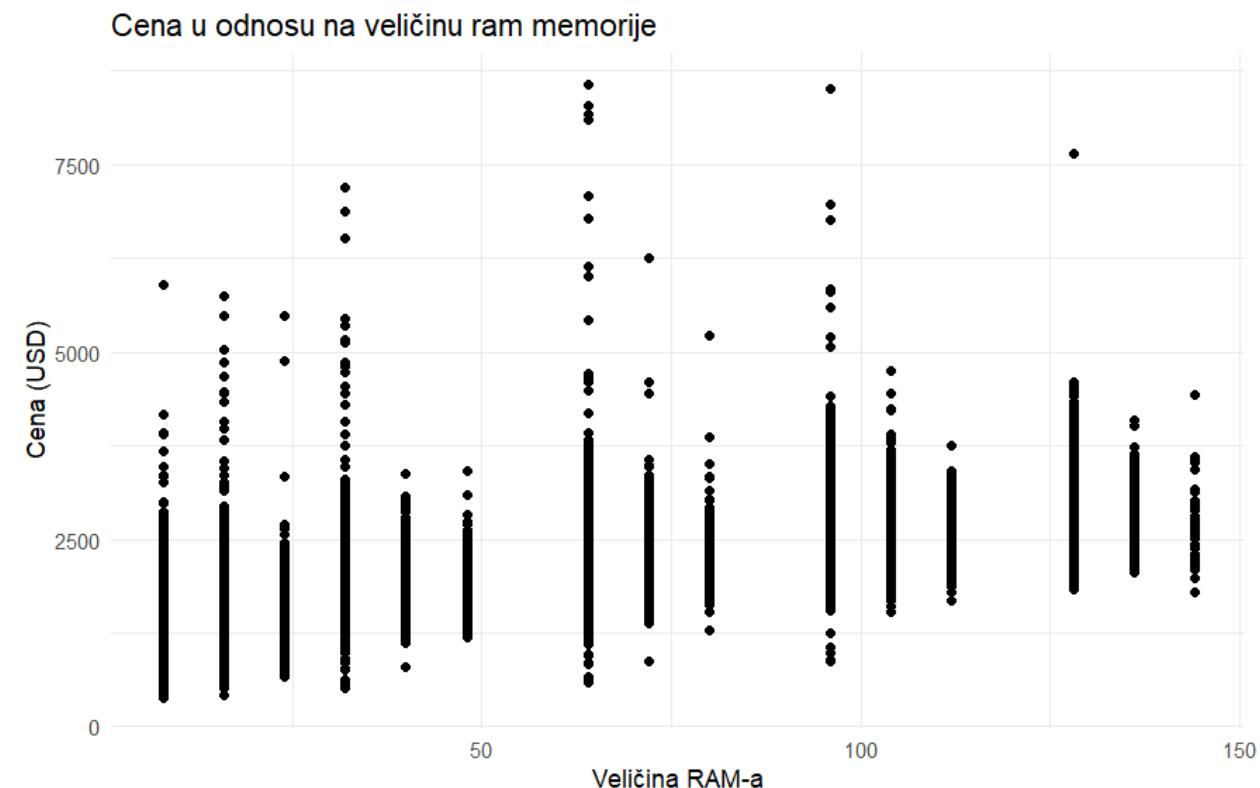
Slika 86 Boxplot zavisnosti cene od vrste operativnog sistema

Boxplot-ovi iznad pokazuju da uređaji sa mac operativnim sistemom imaju ubedljivo najvišu medijanu i takođe imaju i veliki cenovni opseg tako da su njihovi uređaji ubedljivo najskuplji. Uređaji sa windows operativnim sistemom imaju i najveći cenovni opseg, najviše uređaja i koristi ovaj os i postoji gomila uređaja od

najjeftinijih do najskupljih. Uređaji sa chrome operativnim sistemom imaju najmanju medijanu i najmanji broj cena koje odskaču i to su uglavnom uređaji sa slabijim i jeftinijim komponentama i služe za obavljanje osnovnih zadataka, obrazovanje i slično. Uređaji sa linux os-om su uglavnom stabilniji i nižih cena, većih od chrome os-a, ali ne sa nešto prezahtevnim hardverom. Ovaj prediktor i nije toliko jako povezan sa cenom kao prethodni, ali je dobar i solidan je pokazatelj kakva je cena u odnosu na neku kategoriju.

```
ggplot(datav2, aes(x = ram_gb, y = price)) +
  geom_point() +
  labs(title = "Cena u odnosu na veličinu ram memorije",
       x = "Veličina RAM-a", y = "Cena (USD)") +
  theme_minimal()
```

*Slika 87 R kod za iscrtavanje grafika ispod*



*Slika 88 Scatter grafik odnosa veličine RAM-a i cene*

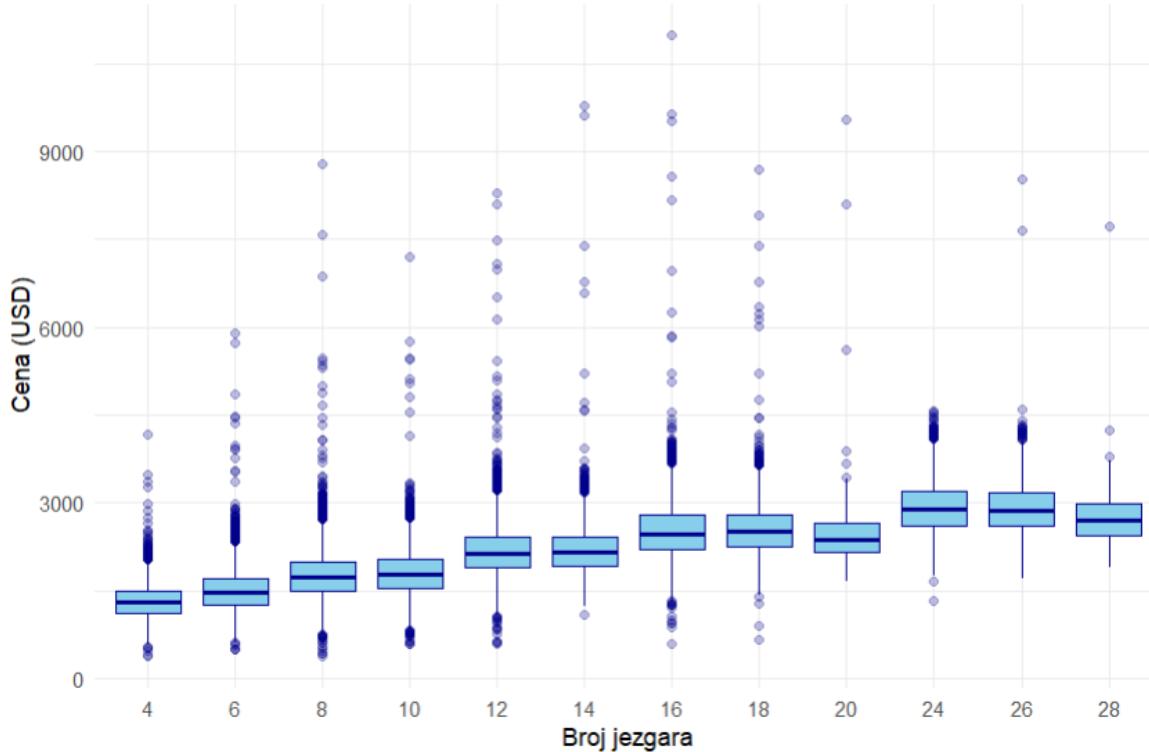
RAM nam predstavlja dobar prediktor cene, jer RAM prati trend što više RAM-a to više para. Takvu odliku možemo videti i sa grafika, gde možemo jasno primetiti da

donja granica raste sa porastom RAM-a. Možemo takođe primetiti da smo ostavili određeni broj outliera iz razloga što RAM nije najbitniji prediktor i te ekstremnije vrednosti nam znače za predviđanje cene kroz druge prediktore.

```
ggplot(datav2, aes(x = factor(cpu_cores), y = price)) +
  geom_boxplot(fill = "skyblue", color = "darkblue", outlier.alpha = 0.25) +
  labs(
    title = "Cena uređaja u odnosu na broj jezgara procesora",
    x = "Broj jezgara",
    y = "Cena (USD)"
  ) +
  theme_minimal() +
  theme(
    plot.title = element_text(hjust = 0.5, face = "bold")
  )
```

*Slika 89 R kod za iscrtavanje grafika ispod*

**Cena uređaja u odnosu na broj jezgara procesora**



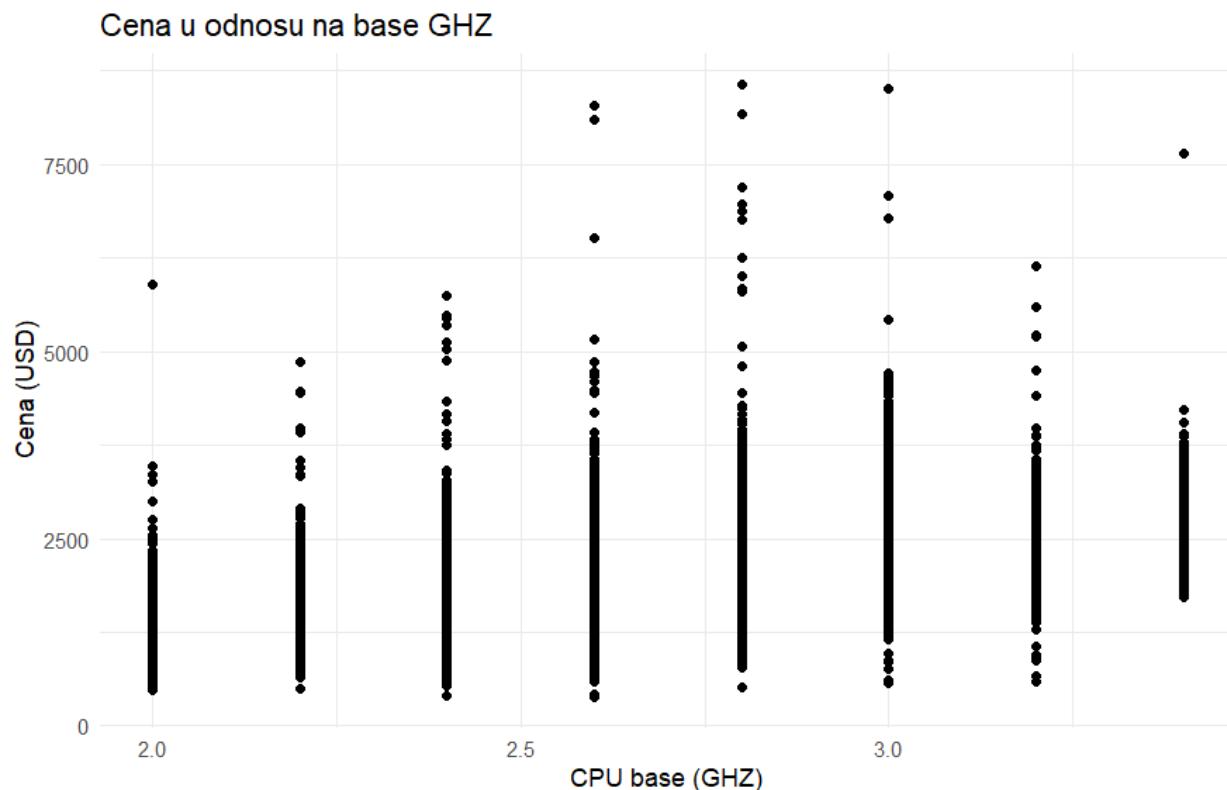
*Slika 90 Scatter grafik odnosa broja jezgara procesora i cene*

Broj jezgara unutar procesora nam predstavlja dobar prediktor jer ima dobru linearnu korelaciju 0.73, što se može videti iz matrice korelacija. Takođe, na grafiku isto možemo primetiti da sa porastom broja jezgara cena uglavnom raste, ali to nije uvek slučaj, medijane od uređaja sa 20, 26 ili 28 jezgara baš i ne prate trend rasta

cene. Vidimo da imamo outliere koje nismo sklonili, jer je potrebno ispitati od kojih još prediktora zavisi cena i outlier-i su uglavnom u normalnim granicama. Sa grafika se može primetiti da sve vrednosti idu za po 2 (4, 6, 8, 10, ...) do 20 i onda imamo skok na 24. Razlog tome nije što ne postoje procesori sa 22 jezgra, nego u ovom konkretnom dataset-u nije zabeležen ni jedan procesor sa 22 jezgra.

```
ggplot(datav2, aes(x = cpu_base_ghz, y = price)) +
  geom_point() +
  labs(title = "Cena u odnosu na base GHZ",
       x = "CPU base (GHZ)", y = "Cena (USD)") +
  theme_minimal()
```

*Slika 91 R kod za iscrtavanje grafika ispod*



*Slika 92 Scatter dijagram odnosa brzine procesora i cene*

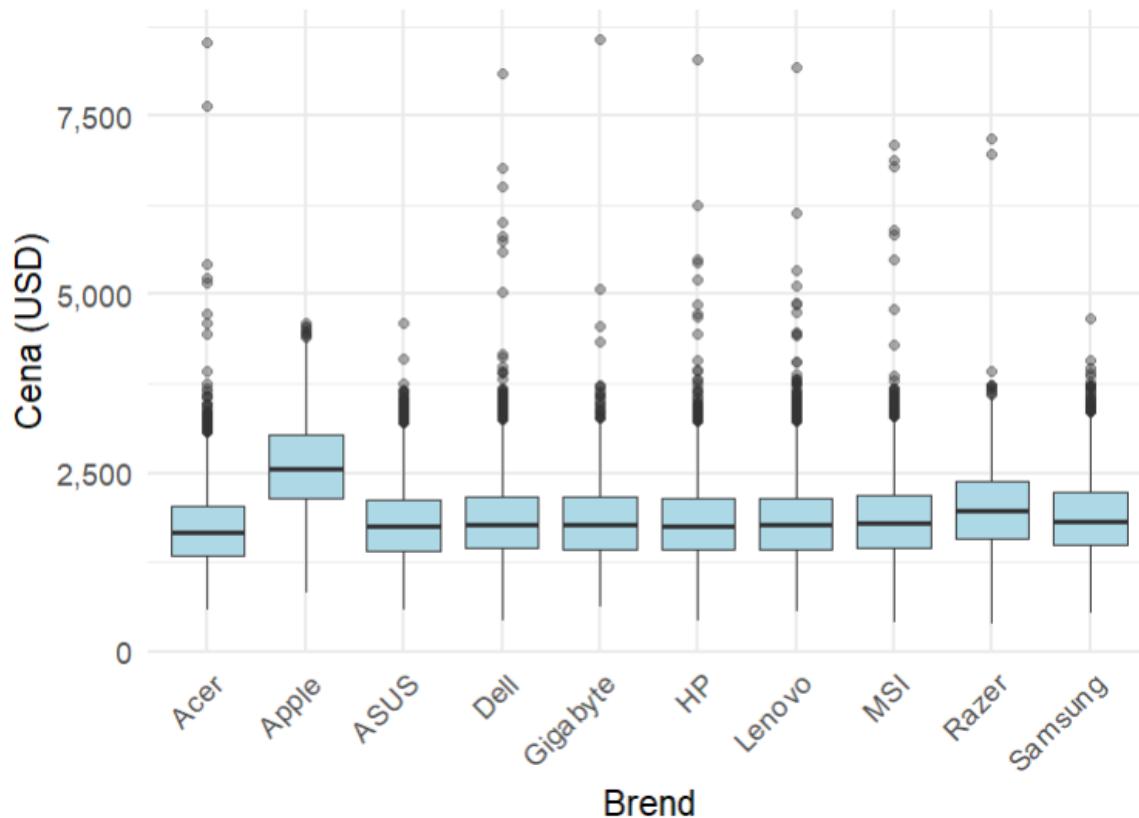
Na grafiku iznad imamo odnos brzine procesora u GHZ i cene. Ako malo bolje pogledamo, videćemo da postoji blage pozitivne linearne korelacijske između brzine procesora i cene. To nije uopšte neočekivano ponašanje, jer brzina često diktira cenu nekog uređaja, ne samo kod računara. Što se tiče outliera, ostali su određeni

outlieri jer su nam potrebni u drugim komponentama. Ono što je zanimljivo kod brzine procesora je to što donja granica do 3.0 GHZ gotovo da i ne raste, što nam govori da brzina procesora ne mora biti preterano dobar prediktor.

```
ggplot(datav2, aes(x = brand, y = price)) +
  geom_boxplot(fill = "lightblue", outlier.alpha = 0.4) +
  scale_y_continuous(labels = scales::comma) +
  labs(
    title = "Cena u odnosu na brend uređaja",
    x = "Brend",
    y = "Cena (USD)"
  ) +
  theme_minimal(base_size = 14) +
  theme(
    plot.title = element_text(hjust = 0.5, face = "bold"),
    axis.text.x = element_text(angle = 45, hjust = 1)
  )
```

*Slika 93 R kod za iscrtavanje grafika ispod*

### Cena u odnosu na brend uređaja

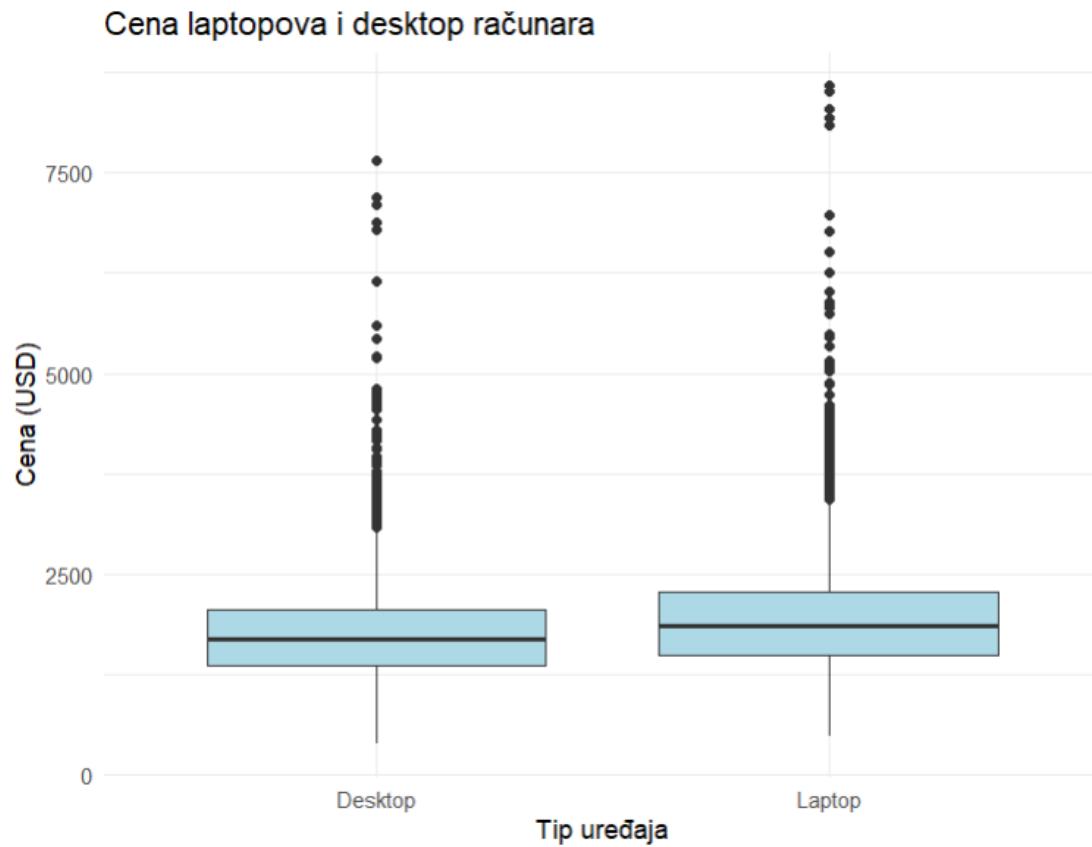


*Slika 94 Boxplot zavisnosti cene od proizvođača uređaja*

Boxplot-ovi sa slike iznad pokazuju da se proizvođač Apple dosta izdvaja od ostalih i njegova medijana je znatno veća. Apple dosta pored komponenti naplaćuje i brand, pa je i logično da bude ovako, pored ove i Razer malo odskače sa medijanom, kao jeftinija marka od Apple, ali skuplja od svih ostalih. Razer uređaji su uglavnom gejmerski pa je i logično što imaju veću cenu, ostale marke nude uređaje raznih cena od najjeftinijih do najskupljih, što je najviše povezano sa komponentama, a ne samim brendom. Acer je jedini koji ima manju medijanu, jer on najčešće nudi uređaje niske i srednje klase.

```
ggplot(datav2, aes(x = device_type, y = price)) +
  geom_boxplot(fill = "lightblue") +
  labs(title = "Cena laptopova i desktop računara",
       x = "Tip uređaja", y = "Cena (USD)") +
  theme_minimal()
```

*Slika 95 R kod za iscrtavanje grafika ispod*



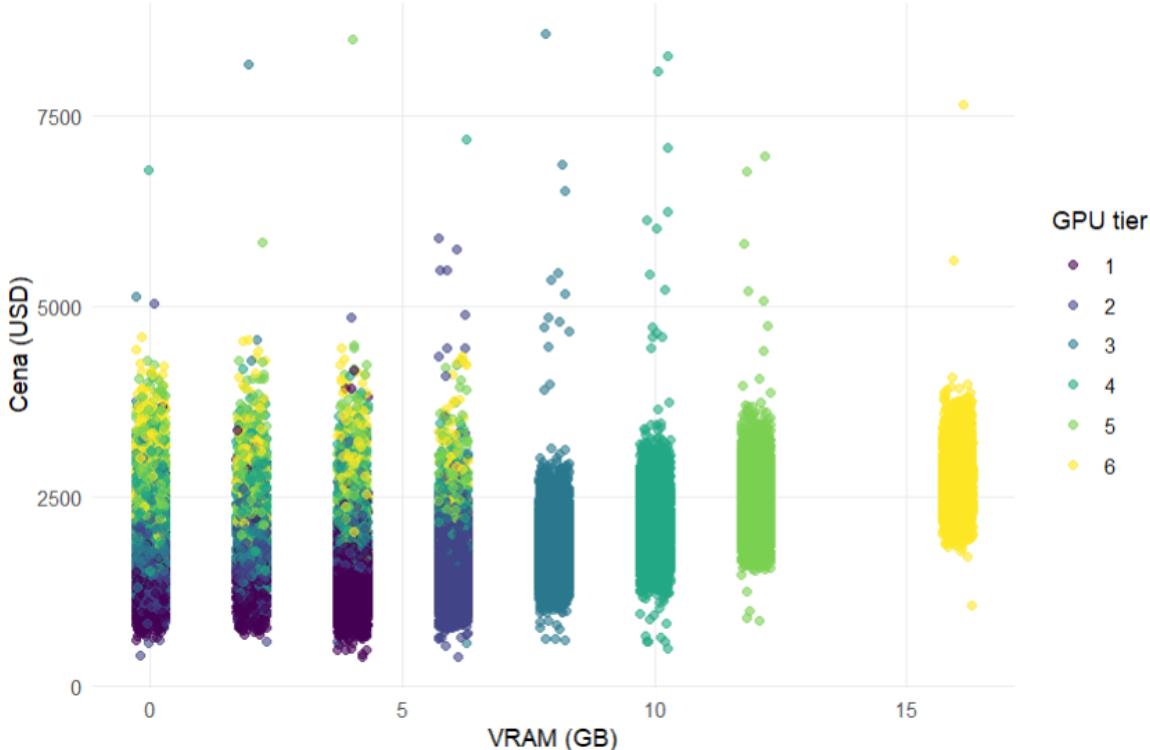
*Slika 96 Boxplot zavisnosti cene od vrste uređaja*

Boxplot-ovi iznad pokazuju da su laptopovi skuplji od računara, što smo i ranije zaključili. Medijana im je veća i imaju više cena koje odskaču nego računari, ali je to sasvim normalno, laptopovi imaju integrisane komponente, drugačiji sistem hlađenja, prenosivi su pa je cena opravdano veća. Jednak nivo performansi kod laptopova i računara uvek će laptopovi biti skuplji, kod oba tipa cene koje odskaču su uglavnom gaming uređaji i profesionalni modeli i u sasvim su realnom opsegu.

```
ggplot(datav2, aes(x = vram_gb, y = price, color = factor(gpu_tier))) +
  geom_jitter(alpha = 0.6, width = 0.3) +
  scale_color_viridis_d() +
  labs(
    title = "Uticaj VRAM memorije i ranga grafičke kartice na cenu uređaja",
    x = "VRAM (GB)",
    y = "Cena (USD)",
    color = "GPU tier"
  ) +
  theme_minimal() +
  theme(
    plot.title = element_text(hjust = 0.5, face = "bold"),
    panel.grid.minor = element_blank()
  )
)
```

Slika 97 R kod za iscrtavanje grafika ispod

**Uticaj VRAM memorije i ranga grafičke kartice na cenu uređaja**

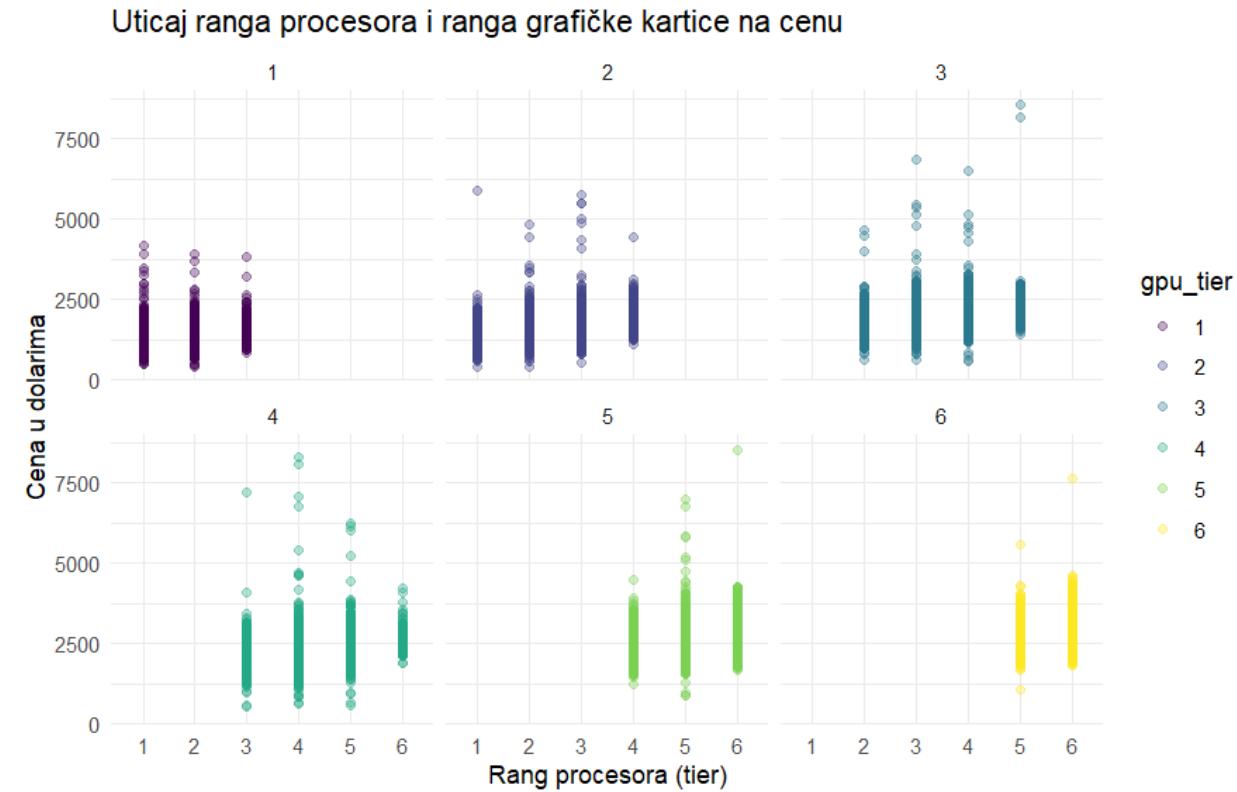


Slika 98 Dijagram uticaja VRAM memorije i ranga grafičke kartice na cenu uređaja

Sa grafika iznad možemo videti da su u odnosu na isti ovaj grafik pre brisanja podataka, sklonjeni jeftiniji uređaji sa većim rangovima gpu-a i da su sklonjeni dosta skupi uređaji najvišeg ranga grafičke kartice, a sa veoma malo VRAM-a. Ovaj grafik je pregledniji i pokazuje veću povezanost između ova dva obeležja.

```
ggplot(datav2, aes(x = cpu_tier, y = price, color = gpu_tier)) +
  geom_point(alpha = 1/3) +
  facet_wrap(~ gpu_tier) +
  theme_minimal() + labs(
    title = "Uticaj ranga procesora i ranga grafičke kartice na cenu",
    x = "Rang procesora (tier)",
    y = "Cena u dolarima"
  )
```

*Slika 99 R kod za iscrtavanje grafika ispod*



*Slika 100 Dijagram uticaja ranga procesora i ranga grafičke kartice na cenu*

Na grafiku iznad je prikazan uticaj ranga procesora i ranga grafičke kartice na cenu. Možemo videti da posle čišćenja podataka imamo manje outliera. Znamo da su rang procesora i rang grafičke kartice dosta dobri prediktori. To možemo videti sa matrici korelacija, gde rang procesora ima linearnu zavisnost od 0.77 i rang grafičke kartice

0.78. Na ovom grafiku možemo videti da imaju i međusobnu zavisnost i to nam takođe potvrđuje i matrica zavisnosti 0.86. Tako da možemo razmatrati korišćenje samo jednog od ova 2 prediktora za predikciju cene u modelu.

## Feature engineering

### Feature Selection

Feature selection se bavi odabirom prediktora (feature). U ovom delu ćemo prikazati koji su to najbitniji prediktori i takođe ćemo izbaciti kolone koje nam neće pomoći u daljem radu.

Sa prethodnih grafika smo mogli da primetimo koji feature-i su nam najbitniji. To su kolone kao što su:

1. CPU\_tier
2. GPU\_tier
3. RAM
4. Tip računara
5. Marka računara
6. Operativni sistem

Međutim, mogli smo da primetimo da nam dosta ostalih kolona skoro ništa ne znače u predikciji cene. Ti podaci su nam bili od velike važnosti za pripremu podataka i prepoznavanje trendova, ali sada nam samo popunjavaju prostor.

Prva kolona koja nam ne pomaže kod predikcije je model. Ta kolona sadrži nazive modela nekog uređaja. Tu vrstu podataka ne možemo iskoristiti za treniranje i predikciju cene, jer modela ima mnogo. Kada bismo trenirali model nad tim podacima, on ne bi mogao ništa da zaključi i kada bi se pojavio model koji nije bio u trening skupu, naš model bi napravio veliku grešku. Bolji prediktor nam predstavlja brend, iako je nepreciznija predikcija samo preko brenda, barem neće naš model da „preuči“, tj. da bude overfittovan na trening skupu.

```
# uklanjanje modela  
datav3$model = NULL
```

*Slika 101 R kod uklanjanja modela*

Sledeća kolona za uklanjanje je gpu\_model. Analogno modelu uređaja, gpu\_model nam ne daje dodatno znanje i ne predstavlja dobar prediktor, samo će dovesti do overfittovanja. Tako da ćemo i njega ukloniti iz našeg skupa.

```
# uklanjanje gpu_modela  
datav3$gpu_model = NULL
```

*Slika 102 R kod uklanjanja modela grafičke kartice*

Dalje uklanjamo form\_factor. Podaci su dosta razbacani, tj. podaci ne prate nikakav trend, i ima mnogo outlier-a. Ne možemo izvući dobre zaključke i znanje iz te kolone. Biće uklonjen u nastavku rada.

```
# uklanjanje form_factora  
datav3$form_factor = NULL
```

*Slika 103 R kod uklanjanja form\_factor-a*

Display\_type je sledeća kolona koju uklanjamo. Nema značajnih trendova i ima dosta outlier-a. Zajedno sa njom uklonićemo i display\_size\_in. Kolona koja ima mnogo outlier-a i dosta malo se razlikuju cene po svim veličinama.

```
# uklanjanje display_type  
datav3$display_type = NULL
```

*Slika 104 R kod uklanjanja tipa display-a*

```
# uklanjanje display_size  
datav3$display_size_in = NULL
```

*Slika 105 R kod uklanjanja veličine display-a*

Psu\_watts ima slabu korelaciju sa cenom, što se vidi sa matrice korelacija i takođe vizuelno sa grafika. Nije dobar prediktor i biće uklonjen iz skupa.

```
# uklanjanje psu_watts  
datav3$psu_watts = NULL
```

*Slika 106 R kod uklanjanja napajanja*

Wifi ne utiče mnogo na cenu. To znamo iz domenskog znanja i takođe nam grafik potvrđuje. Iako ima određenu korelaciju sa cenom, znamo da svi novi uređaji imaju najjači wifi interfejs i da će retko ko imati mogućnost kupovine uređaja sa slabijim wifi interfejsom. Biće uklonjen.

```
# uklanjanje wifi  
datav3$wifi = NULL
```

*Slika 107 R kod uklanjanja wifi*

Bluetooth nema direktnu korelaciju sa cenom i ne prati nikakav trend. Ne dobijamo nikakvo znanje iz bluetooth-a i zato ćemo ukloniti tu kolonu u potpunosti.

```
# uklanjanje bluetooth  
datav3$bluetooth = NULL
```

*Slika 108 R kod uklanjanja bluetooth*

Poslednja kolona, koju ćemo ukloniti je weight\_kg. Ova kolona ne prati nikakav trend i nije dobar prediktor za cenu uređaja. Sa porastom težine, cena niti raste niti opada. Iz domenskog znanja takođe znamo da kada kupujemo desktop računar nikad ne gledamo težinu, a kod kupovine laptop računara gledamo težinu samo ako putujemo često, i tada težina dosta zavisi od veličine celokupnog laptopa i sa smanjenjem veličine ne moraju se nužno smanjiti performanse. Tako da vidimo da cena ne zavisi direktno od težine, te ćemo ukloniti tu kolonu.

## #uklanjanje težine

```
datav3$weight_kg = NULL
```

Slika 109 R kod uklanjanja težine u kg

Od sada pa nadalje će naš skup podataka imati strukturu kao što je prikazano na sledećoj slici.

```
'data.frame': 74221 obs. of 23 variables:
 $ device_type      : Factor w/ 2 levels "Desktop","Laptop": 1 2 1 2 1 2 2 2 1 1 ...
 $ brand            : Factor w/ 10 levels "Acer","Apple",...: 10 10 4 5 8 4 7 6 10 6 ...
 $ release_year     : int 2022 2022 2024 2024 2025 2024 2025 2023 2021 2022 ...
 $ os               : Factor w/ 4 levels "ChromeOS","Linux",...: 4 4 4 2 4 4 4 4 4 4 ...
 $ cpu_brand        : Factor w/ 3 levels "AMD","Apple",...: 3 3 1 1 3 3 1 3 1 3 ...
 $ cpu_model        : chr "Intel i5-11129" "Intel i7-11114" "AMD Ryzen 5 7550" "AMD Ryzen 7 6230" ...
 $ cpu_tier         : Ord.factor w/ 6 levels "1"<"2"<"3"<"4"<...: 3 4 2 5 5 6 1 2 2 4 ...
 $ cpu_cores        : int 12 12 6 16 16 26 4 6 14 ...
 $ cpu_threads       : int 24 24 12 32 32 52 8 12 10 28 ...
 $ cpu_base_ghz     : num 2.8 2.6 2.6 2.8 3.2 3 2 2.2 2.6 3 ...
 $ cpu_boost_ghz    : num 3.8 3.6 3.6 3.9 4.3 4.1 2.9 3.1 3.6 3.9 ...
 $ gpu_brand        : Factor w/ 4 levels "AMD","Apple",...: 4 4 1 4 4 4 4 4 4 4 ...
 $ gpu_tier         : Ord.factor w/ 6 levels "1"<"2"<"3"<"4"<...: 2 4 2 5 6 6 1 2 2 4 ...
 $ vram_gb          : int 6 10 6 12 16 16 4 6 6 10 ...
 $ ram_gb           : int 16 64 16 96 96 128 8 16 16 64 ...
 $ storage_type      : Factor w/ 4 levels "HDD","Hybrid",...: 3 3 1 3 3 4 3 3 3 1 ...
 $ storage_gb        : int 1024 512 512 256 512 1024 512 1024 1024 512 ...
 $ storage_drive_count: int 1 1 2 1 2 1 1 1 3 1 ...
 $ resolution        : Factor w/ 6 levels "1920x1080","2560x1440",...: 2 1 5 3 2 3 6 1 2 1 ...
 $ refresh_hz        : int 90 90 120 90 90 60 120 165 120 60 ...
 $ battery_wh        : int 0 56 0 80 0 80 60 70 0 0 ...
 $ warranty_months   : int 36 12 36 12 36 48 24 36 36 24 ...
 $ price             : num 1384 2275 1332 2682 2752 ...
```

Slika 110 Nova struktura skupa podataka

## Feature Engineering

### Cpu power score

Kao prvi feature koji bismo mogli dodati jeste upravo cpu\_power\_score koji predstavlja kolika je zapravo sirova snaga procesora. To ćemo dobiti kombinacijom broja jezgara procesora i osnovne frekvencije procesora.

```
datav2$cpu_power_score = datav2$cpu_cores * datav2$cpu_base_ghz
```

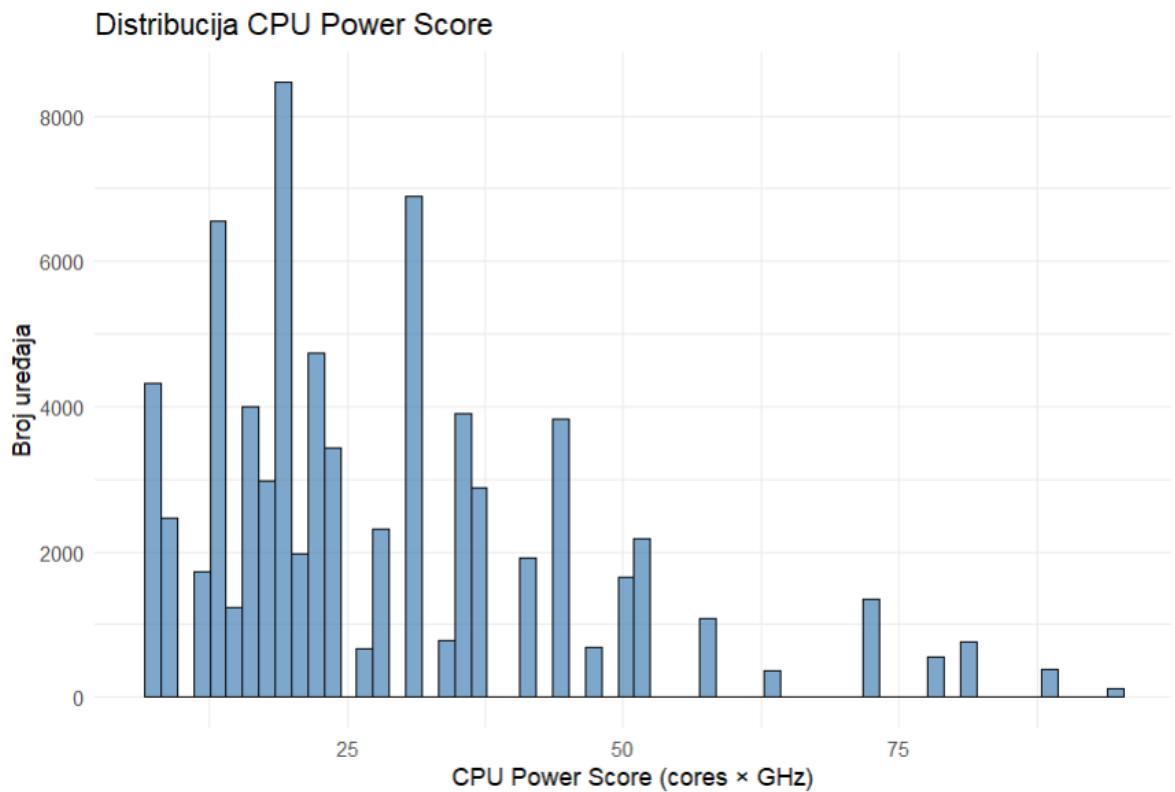
Slika 111 Dodavanje novog feature-a cpu power score

```

ggplot(datav2, aes(x = cpu_power_score)) +
  geom_histogram(bins = 60, fill = "steelblue", alpha = 0.7, color = "black") +
  labs(
    title = "Distribucija CPU Power Score",
    x = "CPU Power Score (cores x GHz)",
    y = "Broj uređaja"
  ) +
  theme_minimal()

```

*Slika 112 R kod za iscrtavanje grafika za cpu power score*



*Slika 113 Grafik novog feature-a cpu power score*

Grafik iznad pokazuje da se, kako je i očekivano, najveći broj računara nalazi u delu do 40 score-a, najčešće uređaji imaju po 8 ili 16 jezgara, sa 2.5 do 2.8 GHz snage procesora. Manji broj uređaja ima score preko 60 i to su high-end računari, profesionalni laptopovi i slično.

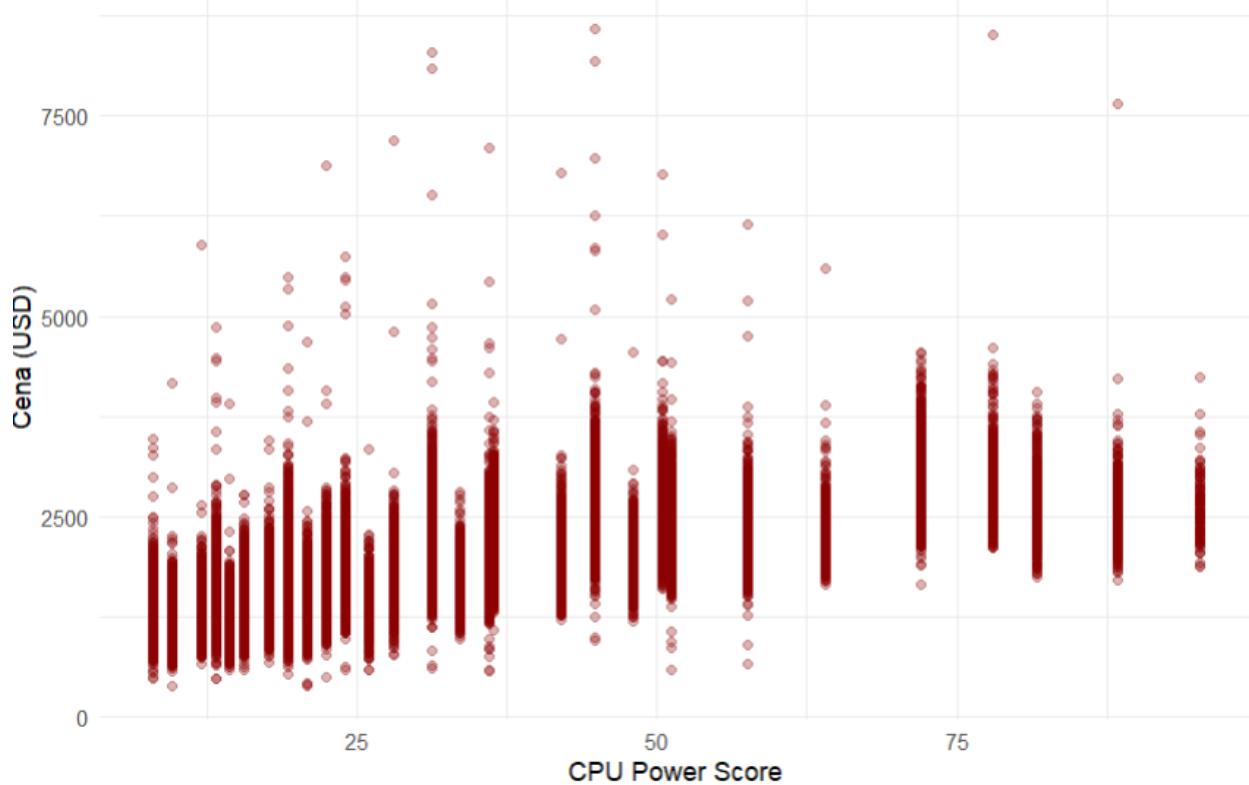
```

ggplot(datav2, aes(x = cpu_power_score, y = price)) +
  geom_point(alpha = 0.3, color = "darkred") +
  labs(
    title = "Odnos CPU Power Score-a i cene uređaja",
    x = "CPU Power Score",
    y = "Cena (USD)"
  ) +
  theme_minimal()

```

*Slika 114 R kod za iscrtavanje grafika zavisnosti cene u odnosu na cpu power score*

Odnos CPU Power Score-a i cene uređaja



*Slika 115 Grafik zavisnosti cene u odnosu na cpu power score*

Iznad se nalazi grafik zavisnosti cene od novog feature-a. Može se primetiti da cena blago raste sa povećanjem score-a, ali ne preterano, postoji dosta outlier-a posebno u delu od 25 do 50 score-a. Moglo bi se sve podeliti u nekoliko kategorija, što bi se svelo na kolonu cpu\_tier, pa to nećemo raditi.

```
> cor(datav2$cpu_power_score, datav2$price)
[1] 0.7002545
```

*Slika 116 Korelacija cpu power score sa cenom*

Korelacija je jaka i dobra, malo manja od cpu\_tier pre pretvaranja u factor. Ovaj prediktor je svakako dobar i nećemo ovo pretvarati u kategorije pošto već imamo cpu tier koji je ordinal factor promenljiva. Cpu power score nije zamena za cpu tier, već je komplementaran numerički pokazatelj procesorske snage, za svaku od 6 kategorija iz cpu tier postoje uređaji i sa većom i sa manjom vrednošću novog prediktora, tako da se podaci ne dupliraju.

Pored osnovnih vizuelnih pokazatelja, CPU Power Score je značajan jer unosi kontinuiranu meru procesorske snage, dok cpu\_tier predstavlja samo kategorizovanu verziju performansi. Ovaj novi atribut omogućava razlikovanje uređaja unutar iste kategorije npr. dva računara mogu da budu svrstavana u isti cpu tier i da jedan ima 4 jezgra na 2.4 GHz (score = 9.6), a drugi 8 jezgara na 3.0 GHz (score = 24). Ovakve razlike CPU Tier ne može da predstavi, dok CPU Power Score to čini vrlo jasno.

Zbog toga CPU Power Score predstavlja dobar feature, koji unosi suptilnije informacije o sirovim performansama procesora, pa ćemo ovaj feature zadržati.

### Cgt score

Kao drugi feature u okviru fe-a bismo mogli dodati kombinaciju dva, verovatno najbitnija, obeležja na osnovu dosadašnjih analiza i na osnovu domenskog znanja, a to su cpu tier i gpu tier. Procesor je osnovna komponenta u uređaju sa kojom je sve povezano, a grafička kartica je najbitnija za performanse uređaja, pošto imamo raspoređene uređaje po rangu za oba ova pojedinačno zanima nas kakav će ishod biti kada se spoje.

Pošto su cpu tier i gpu tier ranije još promenjeni u ordinalne factor promenljive koristimo njihovu numeričku vrednost kako bismo ih pomnožili.

```
datav2$cgt_score = as.numeric(datav2$cpu_tier) * as.numeric(datav2$gpu_tier)
```

*Slika 117 Dodavanje novog feature-a cgt score*

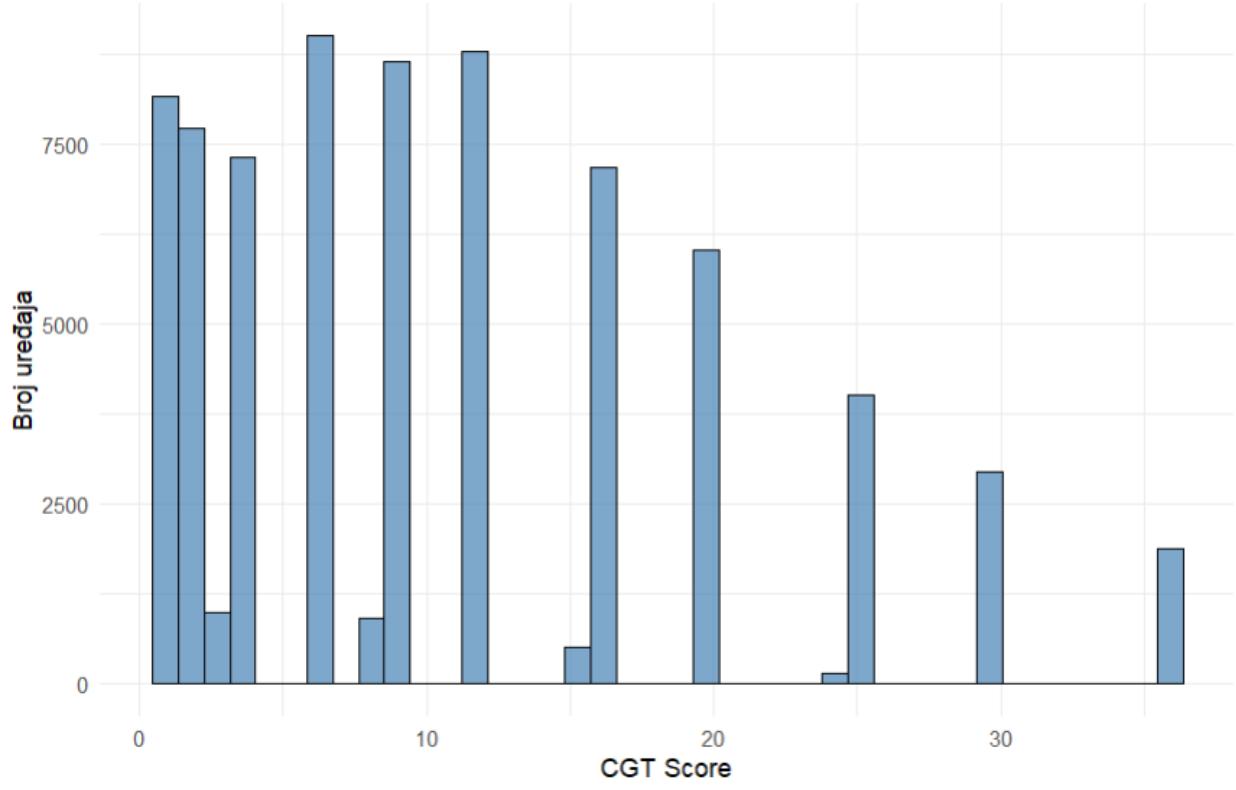
```

ggplot(datav2, aes(x = cgt_score)) +
  geom_histogram(fill = "steelblue", bins = 40, color = "black", alpha = 0.7) +
  labs(
    title = "Distribucija Combined GPU-CPU Tier Score (CGT)",
    x = "CGT Score",
    y = "Broj uređaja"
  ) +
  theme_minimal()

```

*Slika 118 R kod za iscrtavanje grafika za cgt score*

Distribucija Combined GPU–CPU Tier Score (CGT)



*Slika 119 Grafik novog feature-a cgt score*

Pošto su za oba obeležja vrednosti klase od 1 do 6, ovaj score ima vrednosti od 2 do 36, izuzetno je rupičast, tačnije dosta vrednosti nema, što je u redu, većina vrednosti je grupisana oko nekih delova. Vrednosti sa dosta velikim score-om ima malo, to su uglavnom profesionalni uređaji, a i većina našeg skupa podataka jesu upravo računari srednje i niže klase, zato njih i ima najviše.

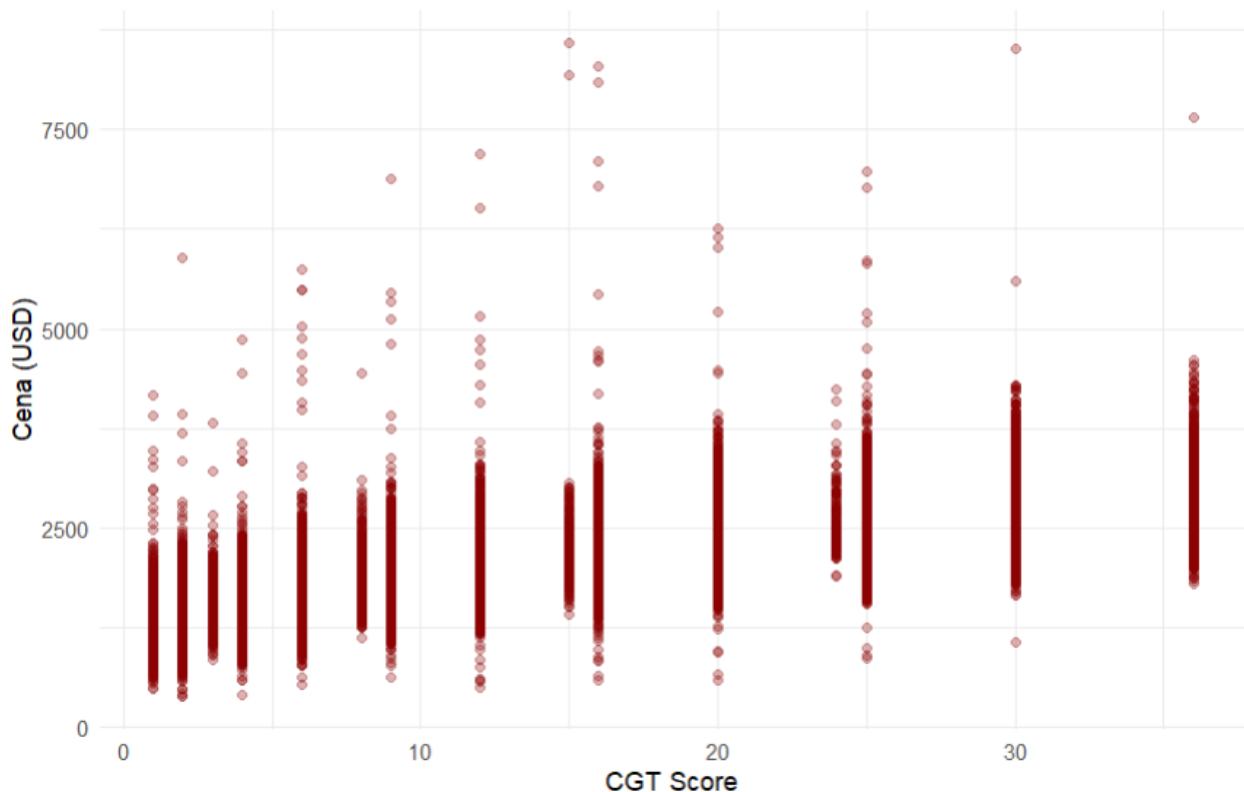
```

ggplot(datav2, aes(x = cgt_score, y = price)) +
  geom_point(alpha = 0.3, color = "darkred") +
  labs(
    title = "Odnos CGT Score-a i cene uređaja",
    x = "CGT Score",
    y = "Cena (USD)"
  ) +
  theme_minimal()

```

Slika 120 R kod za iscrtavanje grafika zavisnosti cene u odnosu na cgt score

Odnos CGT Score-a i cene uređaja



Slika 121 Grafik zavisnosti cene u odnosu na cgt score

Na grafiku zavisnosti cene od cgt score-a cena postepeno raste kako raste i cgt score. Outlieri su prisutni u većini score-ova i sasvim su realni, outlieri koji se malo više ističu u srednjoj klasi, najčešće je dosta veća cena posledica količine RAM memorije i njenog tipa.

```
> cor(datav2$cgt_score, datav2$price)
[1] 0.7934817
```

Slika 122 Korelacija cgt score sa cenom

Korelacija je izuzetno visoka, najveća je od svih prediktora do sad, najbolji je pojedinačni prediktor, tako da itekako ga zadržavamo.

Cgt Score predstavlja verovatno najkorisniji inženjerski feature celog skupa podataka, jer kombinuje dve najuticajnije komponente računara u domenu performansi: procesor i grafičku karticu. Na osnovu prethodne analize, oba ova prediktora pojedinačno snažno utiču na cenu i cpu tier i gpu tier su među najboljim kategorijskim obeležjima, kao što je malopre spomenuto.

Njihovim kombinovanjem dobija se indikator ukupne klase performansi uređaja. Na osnovu svega, CGT Score predstavlja jako dobar feature koji će značajno povećati kvalitet modela predikcije i zadržaćemo ga.

### Cpu generation

Kada smo uklanjali višak kolone većinom smo sve kolone tipa „model“ uklonili jer postoji mnogo modela i nemoguće je izvući znanje iz njih. Međutim, ostavili smo jednu kolone te vrste, a to je cpu\_model. Ostavili smo ovu kolonu iz razloga što uz malo kreativnosti možemo izvući generaciju svakog procesora i napraviti neku vrstu tier-a tj. ranga tih procesora na osnovu svoje generacije.

Imamo tri proizvođača procesora, a to su Intel, AMD i Apple. Kod Intel-a se generacije označavaju sa „i“ nakon čega sledi broj koji označava generaciju, npr. i3, i5, i7, i9. AMD označava svoje generacije samo jednim brojem, kao što su 3, 5, 7, 9. Apple za generaciju koristi oznaku M, i onda broj iz intervala od 1 do 3, nakon toga ima dodatnu oznaku Pro ili Max, koji bliže označavaju taj model i svoje sposobnosti.

Kako su tri različita proizvođača, nije baš moguće direktno uporediti generacije, ali je moguće to uraditi približno. U suštini, Intel i AMD koriste sličnu oznaku za generaciju i moguće ih je lako uporediti. Intel i5 je po performansama sličan AMD Ryzen 5, Intel i3 je sličan AMD Ryzen 3, i tako dalje. Budući da Apple drugačije označava generacije svojih procesora, moraćemo malo kreativnije poređenje da

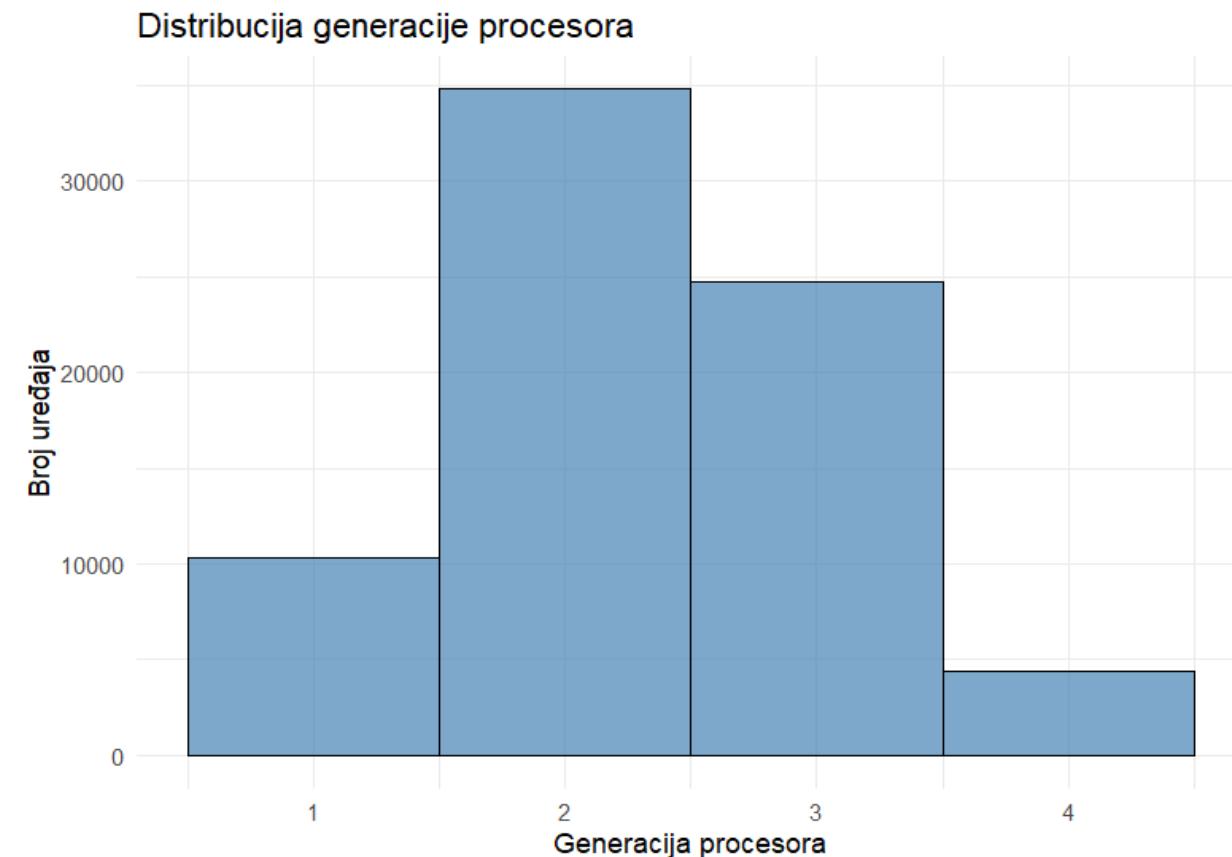
napravimo. Prepostavimo da  $M1 = i3 = 3$ ,  $M2 = i5 = 5$ ,  $M3 = i7 = 7$  i procesori sa oznakama Pro ili Max su jednake  $i9$  i  $9$ .

```
datav3 <- datav3 %>% mutate(cpu_generation = case_when(
  str_detect(cpu_model, regex("i9|Ryzen 9|Pro|Max", ignore_case = FALSE)) ~ 4,
  str_detect(cpu_model, regex("i3|Ryzen 3|M1", ignore_case = FALSE)) ~ 1,
  str_detect(cpu_model, regex("i5|Ryzen 5|M2", ignore_case = FALSE)) ~ 2,
  str_detect(cpu_model, regex("i7|Ryzen 7|M3", ignore_case = FALSE)) ~ 3,
  TRUE ~ NA_real_
))
```

Slika 123 R kod inženjeringa generacije procesora

```
ggplot(datav3, aes(x = cpu_generation)) +
  geom_histogram(fill = "steelblue", bins = 4, color = "black", alpha = 0.7) +
  labs(
    title = "Distribucija generacije procesora",
    x = "Generacija procesora",
    y = "Broj uređaja"
  ) +
  theme_minimal()
```

Slika 124 R kod iscrtavanja dijagrama distribucije generacije procesora

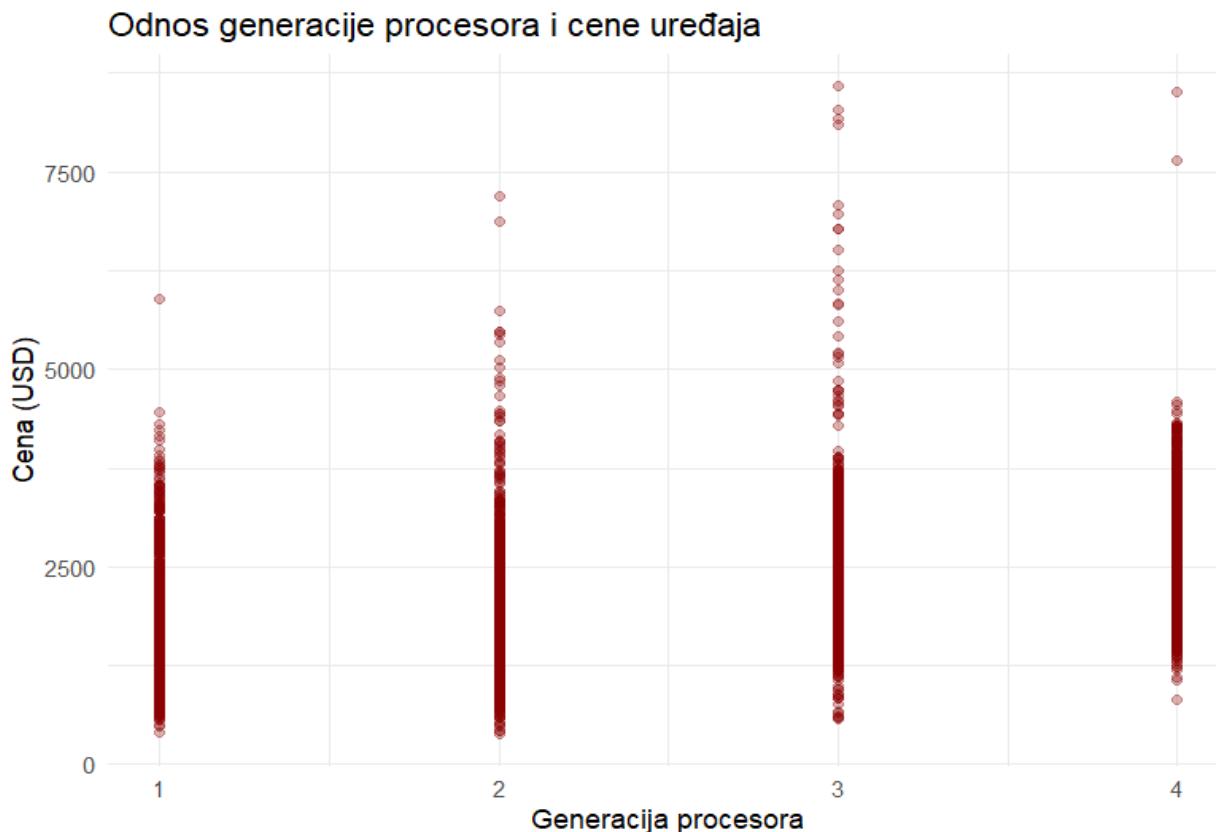


Slika 125 Dijagram distribucije generacije procesora

Na grafiku iznad možemo videti raspodelu generacije procesora. Na osnovu raspodele zaključujemo da je naša prepostavka bila pun pogodak. Vidimo da generacije 2 i 3 ima najviše, što prati praksu, gde ljudi najčešće kupuju uređaje sa procesorima iz srednjeg nivoa. To su npr. Intel i5 i u malo ređim slučajevima Intel i7. Generacije 1 i 4 dosta manje ljudi kupuje, što je i potpuno logično. Procesor kao što je Intel i3 je dosta slab i polako gubi svoju upotrebnu, dok procesor Intel i9 je ipak previše skup za većinu ljudi.

```
ggplot(datav3, aes(x = cpu_generation, y = price)) +
  geom_point(alpha = 0.3, color = "darkred") +
  labs(
    title = "Odnos generacije procesora i cene uređaja",
    x = "Generacija procesora",
    y = "Cena (USD)"
  ) +
  theme_minimal()
```

*Slika 126 R kod iscrtavanja scatter dijagrama*



*Slika 127 Dijagram odnosa generacije procesora i cene uređaja*

Sa Scatter dijagrama iznad možemo dodatno potvrditi da smo pravilnu prepostavku napravili pri inženjeringu kolone o generaciji procesora. Na osnovu donjih granica vidimo da generacije prate pozitivni trend, gde viša generacija ima skuplju cenu. Ono što takođe možemo primetiti je da generacije 2 i 3 imaju nekoliko netipičnih vrednosti, što nam može govoriti o tome da je možda bio malo precizniji način da razdvojimo podatke ili nam takođe može govoriti da cena u tim rangovima zavisi dosta više od ostalih komponenti.

```
> cor(datav3$cpu_generation, datav3$price)
[1] 0.7087494
```

*Slika 128 Korelacija između generacije procesora i cene*

Vidimo da je korelacija između, novokreiranog prediktora, cpu\_generation i cene 0.71. To je odlična korelacija i označava nam da bi generacija procesora bila dobar prediktor za cenu uređaja.

Na osnovu svih iscrtanih grafika i korelacije, zaključujemo da nam je cpu\_generation dovoljno dobar prediktor i da ćemo ga iz tog razloga zadržati za fazu treniranja i testiranja modela mašinskog učenja.

```
datav3$cpu_generation = factor(
  datav3$cpu_generation,
  levels = sort(unique(datav3$cpu_generation)),
  ordered = TRUE
)
```

*Slika 129 R kod pretvaranja generacije procesora u kategorijsku promenljivu*

Poslednje što treba uraditi je prebaciti generaciju procesora u kategorijsku promenljivu. Vidimo da ima četiri nivoa (1, 2, 3, 4), tako da ćemo napraviti ordinalnu kategorijsku promenljivu, gde će 1 biti najniža vrednost i 4 biti najviša. Na slici iznad je prikazan R kod za pretvaranje u kategorijsku promenljivu.

## Struktura skupa

```
> str(datav3)
'data.frame': 74221 obs. of 26 variables:
 $ device_type      : Factor w/ 2 levels "Desktop","Laptop": 1 2 1 2 1 2 2 2 1 1 ...
 $ brand            : Factor w/ 10 levels "Acer","Apple",...: 10 10 4 5 8 4 7 6 10 6 ...
 $ release_year     : int  2022 2022 2024 2024 2025 2024 2025 2023 2021 2022 ...
 $ os               : Factor w/ 4 levels "ChromeOS","Linux",...: 4 4 4 2 4 4 4 4 4 4 ...
 $ cpu_brand        : Factor w/ 3 levels "AMD","Apple",...: 3 3 1 1 3 3 1 3 1 3 ...
 $ cpu_model         : chr  "Intel i5-11129" "Intel i7-11114" "AMD Ryzen 5 7550" "AMD Ryzen 7 6230" ...
 $ cpu_tier          : Ord.factor w/ 6 levels "1"<"2"<"3"<"4"<..: 3 4 2 5 5 6 1 2 2 4 ...
 $ cpu_cores         : int  12 12 6 16 16 26 4 6 14 ...
 $ cpu_threads       : int  24 24 12 32 32 52 8 12 10 28 ...
 $ cpu_base_ghz     : num  2.8 2.6 2.6 2.8 3.2 3 2 2.2 2.6 3 ...
 $ cpu_boost_ghz    : num  3.8 3.6 3.6 3.9 4.3 4.1 2.9 3.1 3.6 3.9 ...
 $ gpu_brand         : Factor w/ 4 levels "AMD","Apple",...: 4 4 1 4 4 4 4 4 4 4 ...
 $ gpu_tier          : Ord.factor w/ 6 levels "1"<"2"<"3"<"4"<..: 2 4 2 5 6 6 1 2 2 4 ...
 $ vram_gb           : int  6 10 6 12 16 16 4 6 6 10 ...
 $ ram_gb            : int  16 64 16 96 96 128 8 16 16 64 ...
 $ storage_type      : Factor w/ 4 levels "HDD","Hybrid",...: 3 3 1 3 3 4 3 3 3 1 ...
 $ storage_gb         : int  1024 512 512 256 512 1024 512 1024 1024 512 ...
 $ storage_drive_count: int  1 1 2 1 2 1 1 1 3 1 ...
 $ resolution         : Factor w/ 6 levels "1920x1080","2560x1440",...: 2 1 5 3 2 3 6 1 2 1 ...
 $ refresh_hz         : int  90 90 120 90 90 60 120 165 120 60 ...
 $ battery_wh         : int  0 56 0 80 0 80 60 70 0 0 ...
 $ warranty_months    : int  36 12 36 12 36 48 24 36 36 24 ...
 $ price              : num  1384 2275 1332 2682 2752 ...
 $ cpu_power_score    : num  33.6 31.2 15.6 44.8 51.2 78.8 13.2 15.6 42 ...
 $ cgt_score          : num  6 16 4 25 30 36 1 4 4 16 ...
 $ cpu_generation     : Ord.factor w/ 4 levels "1"<"2"<"3"<"4": 2 3 2 3 3 4 1 2 2 3 ...
> |
```

Slika 130 Nova struktura skupa podataka

Na slici iznad možemo videti kako izgleda finalna struktura našeg skupa podataka.

## Treniranje modela

### Priprema skupa

Pre početka treniranja, potrebno je cenu uređaja pretvoriti u logaritamski prikaz. Logaritamska transformacija nije neophodna, ali je dosta dobro odraditi je. Cena sadrži dosta niskih vrednosti i samo mali broj ekstremno velikih, što smo mogli zaključiti sa grafika na početku dokumenta. Te ekstremne vrednosti dosta povlače cenu na „desnu stranu“, što pravi normalnu raspodelu, a linearna regresija podrazumeva linearnost. Iz tog razloga je pre treniranja modela linearne regresije cena transformisana u logaritamski prikaz.

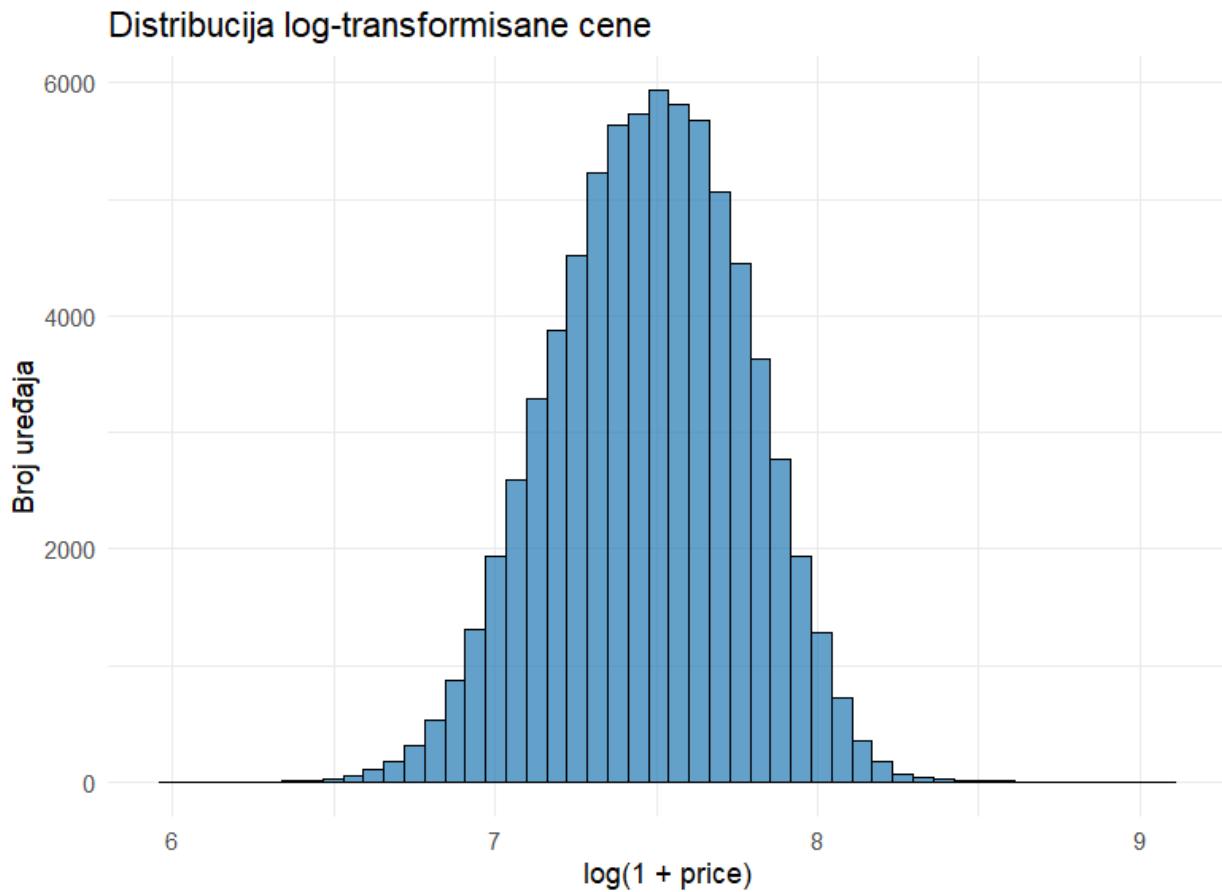
```

datav4$log_price <- log1p(datav4$price)

ggplot(datav4, aes(x = log_price)) +
  geom_histogram(bins = 50, fill = "#1f78b4", color = "black", alpha = 0.7) +
  labs(
    title = "Distribucija log-transformisane cene",
    x = "log(1 + price)",
    y = "Broj uređaja"
  ) +
  theme_minimal()

```

*Slika 131 R kod logaritamske transformacije cene i iscrtavanja grafika*



*Slika 132 Grafik distribucije cene nakon transformacije*

Na grafiku iznad se sada može videti kako distribucija cene uređaja izgleda. Ako to uporedimo sa slikom raspodele cene od pre, možemo primetiti da sada ekstremno male i ekstremno veliki podaci imaju mnogo manji uticaj na njenu raspodelu. Takođe, možemo videti da grafik više nije „povučen na desnu stranu“ tj. right

skewed, već je vrednost koje ima najviše postavljena u centru i sve ostale su oko nje raspoređene.

```
set.seed(123)

n = nrow(datav4)

train_index = sample(seq_len(n), size = 0.8 * n)
train_data = datav4[train_index, ]
test_data = datav4[-train_index, ]
```

*Slika 133 R kod podele skupa na trening i testni skup*

Sledeće što je potrebno uraditi, je podeliti skup na trening i test skup. Skup ćemo podeliti po standardnoj raspodeli: 80:20, gde će 80% biti skup za trening podataka, dok će 20% skupa biti deo za testiranje, tj. predikciju. Kako bi podela bila najrealnija, postavićemo određeni seed, u ovom slučaju je to 123. Nakon toga ćemo samo podeliti indekse na osnovu tog seed-a i uz pomoć tih indeksa napraviti trening i test skup.

## Linearna regresija

### Uvod

Koristićemo linearu regresiju, kako bismo procenili cenu uređaja na osnovu njegovih hardverskih karakteristika. Linearna regresija je jedan od najjednostavnijih, ali u isto vreme i najkorisnijih metoda za predviđanje. Jasno se može videti kako svaki pojedinačni faktor utiče na promenu zavisne promenljive. Linearna regresija je bila naš prvi izbor, a kasnije će biti obrađeni još neki modeli. Postoje naravno i mane ovog metoda, jer pretpostavlja da su svi odnosi između promenljivih linearни, što uglavnom nije slučaj.

Iako se na scatter grafikonu ne može povući prava linija, to nije problem linearog modela već prirode podataka: većina promenljivih je kategorijska ili diskretna, pa se vrednosti grupišu u klastere umesto da čine kontinuiranu dijagonalu. Linearna regresija ne zahteva da podaci budu raspoređeni u "pravoj liniji", već procenjuje

prosečan efekat svake kategorije ili jedinice na ciljnu vrednost. Zato model dobro funkcioniše čak i kada grafički prikaz ne izgleda linearan.

Odlučili smo da krenemo od promenljive koja je po nama najbitnija i da nakon tog modela pravimo sledeći dodavanjem novog obeležja, kako bismo postepeno videli koje obeležje koliko utiče na model.

Tokom analize svakog modela posmatraćemo pre svega rezidualne vrednosti, kako bi proverili da li su greške u predikciji približno simetrične i bez ekstremnih odstupanja, što ukazuje na stabilnost modela. Zatim analiziramo p-value svakog pojedinačnog prediktora, jer nizak p-value (ispod 0.05) znači da taj prediktor statistički značajno utiče na cenu. Posebnu pažnju obratili smo i na F-statistiku i njen globalni p-value, koji pokazuju da li je čitav model statistički značajan u celini, odnosno da li bar neki od uključenih prediktora doprinoсе objašnjenu cenu. Na kraju smo uporedili Multiple R<sup>2</sup> i Adjusted R<sup>2</sup>: Multiple R<sup>2</sup> govori koliki procenat varijanse cena model objašnjava, dok Adjusted R<sup>2</sup> koriguje tu vrednost u odnosu na broj prediktora i pokazuje da li se model zaista poboljšava dodavanjem novih promenljivih. U skupu podataka postoji baš mnogo redova, pa su R<sup>2</sup> metrike uglavnom skoro pa identične, da je malo podataka razlikovale bi se minimalno.

```
true_price = test_data$price
```

Slika 134 R komanda za dobijanje stvarne vrednosti cene

Za svaki model prilikom predviđanja će biti korišćenja funkcija expm1() zbog koje ćemo dobiti predviđanje cene u stvarnom opsegu. Bez ovoga dobili bismo vrednosti predviđanja cene u opsegu od 6.5 do 8.5, a to nam ništa ne znači.

## Model 1

```
model_1 <- lm(log_price ~ cpu_tier, data=train_data)
pred_1 <- expm1(predict(model_1, test_data))

m1_rmse <- sqrt(mean((pred_1 - true_price)^2))
m1_mae <- mean(abs(pred_1 - true_price))
m1_R2 <- 1 - sum((pred_1 - true_price)^2) / sum((true_price - mean(true_price))^2)
```

Slika 135 Implementacija modela 1 i evaluacionih mera (RMSE, MAE, R<sup>2</sup>) nad test skupom

Za prvi model koristi se samo promenljiva `cpu_tier` u odnosu na cenu. Ima jednu od najvećih korelacija sa cenom i iz domenskog znanja je rang procesora odnosno kojoj klasi pripada je veoma bitan prediktor i za početak biće i jedini.

```
> summary(model_1)

Call:
lm(formula = log_price ~ cpu_tier, data = train_data)

Residuals:
    Min      1Q  Median      3Q     Max 
-1.41628 -0.12038  0.00274  0.12208  1.56473 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 7.537865  0.000914 8246.728 < 2e-16 ***
cpu_tier.L   0.690471  0.002716  254.185 < 2e-16 ***
cpu_tier.Q  -0.030483  0.002514  -12.126 < 2e-16 ***
cpu_tier.C  -0.019464  0.002217   -8.778 < 2e-16 ***
cpu_tier^4   0.005880  0.001925    3.054  0.00226 **  
cpu_tier^5  -0.002514  0.001656   -1.518  0.12903  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.1857 on 59370 degrees of freedom
Multiple R-squared:  0.6077,    Adjusted R-squared:  0.6077 
F-statistic: 1.839e+04 on 5 and 59370 DF,  p-value: < 2.2e-16
```

Slika 136 Statistički rezime Modela 1

Reziduali uglavnom ne maše mnogo, ali znaju promašiti i do 1.4, 1.5 u log skali što je malo više. Multiple  $R^2$  i adjusted  $R^2$  imaju vrednosti  $\sim 0.6$  što znači da `cpu_tier` objašnjava oko 60% varijanse cene, što je sasvim okej, sa obzirom da u modelu imamo samo jednu promenljivu, ali model može biti mnogo bolji. F statistika je ogromna i p-value je jako mali, što znači da je model kao celina značajan.

```
> m1_rmse; m1_mae
[1] 346.5089
[1] 261.6485
```

Slika 137 RMSE i MAE prvog modela

## Model 2

```
model_2 <- lm(log_price ~ cpu_tier + gpu_tier, data=train_data)  
pred_2 <- expm1(predict(model_2, test_data))  
  
m2_rmse <- sqrt(mean((pred_2 - true_price)^2))  
m2_mae <- mean(abs(pred_2 - true_price))
```

*Slika 138 Implementacija modela 2 i evaluacionih mera (RMSE, MAE, R<sup>2</sup>) nad test skupom*

U model 2 dodajemo i gpu\_tier, jer je uz rang procesora, jako bitan i rang grafičke kartice koju uređaj poseduje. Korelacija sa cenom je jako visoka i treba naglasiti da je ovaj deo i najbitniji kod gaming laptopova i računara.

```

> summary(model_2)

Call:
lm(formula = log_price ~ cpu_tier + gpu_tier, data = train_data)

Residuals:
    Min      1Q  Median      3Q     Max 
-1.36066 -0.11000  0.00117  0.11261  1.48535 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 7.553e+00 8.751e-04 8631.336 < 2e-16 ***
cpu_tier.L  3.651e-01 4.478e-03   81.533 < 2e-16 ***
cpu_tier.Q -1.877e-02 2.944e-03  -6.374 1.86e-10 ***
cpu_tier.C  7.336e-04 2.233e-03   0.329   0.743    
cpu_tier^4  2.000e-03 1.843e-03   1.085   0.278    
cpu_tier^5  3.561e-05 1.556e-03   0.023   0.982    
gpu_tier.L  3.560e-01 4.241e-03   83.939 < 2e-16 ***
gpu_tier.Q -3.009e-02 2.795e-03  -10.768 < 2e-16 ***
gpu_tier.C -2.956e-03 2.179e-03  -1.357   0.175    
gpu_tier^4  2.700e-03 1.861e-03   1.451   0.147    
gpu_tier^5  -1.773e-03 1.634e-03  -1.085   0.278    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.1734 on 59365 degrees of freedom
Multiple R-squared:  0.658,    Adjusted R-squared:  0.6579 
F-statistic: 1.142e+04 on 10 and 59365 DF,  p-value: < 2.2e-16

```

Slika 139 Statistički rezime Modela 2

Reziduali su nešto bolji tj. uži u odnosu na prethodni model, ali i dalje ima većih odstupanja. Multiple  $R^2$  i adjusted  $R^2$  su se poboljšali za otprilike 5% i to je to značajan napredak. Za F statistiku i p-value se može doneti isti zaključak kao u prethodnom modelu.

```

> m2_rmse; m2_mae
[1] 323.7605
[1] 241.9086

```

Slika 140 RMSE i MAE drugog modela

## Model 3

```
model_3 <- lm(log_price ~ cpu_tier + gpu_tier + ram_gb, data=train_data)

pred_3 <- expm1(predict(model_3, test_data))

m3_rmse <- sqrt(mean((pred_3 - true_price)^2))
m3_mae <- mean(abs(pred_3 - true_price))
```

Slika 141 Implementacija modela 3 i evaluacionih mera (RMSE, MAE,  $R^2$ ) nad test skupom

```
> summary(model_3)

Call:
lm(formula = log_price ~ cpu_tier + gpu_tier + ram_gb, data = train_data)

Residuals:
    Min      1Q  Median      3Q     Max 
-1.35854 -0.11009  0.00119  0.11238  1.48619 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 7.5415241  0.0056912 1325.130 < 2e-16 ***
cpu_tier.L  0.3545243  0.0069130   51.284 < 2e-16 ***
cpu_tier.Q -0.0218880  0.0033272  -6.579 4.79e-11 ***
cpu_tier.C  0.0010847  0.0022398   0.484   0.628  
cpu_tier^4  0.0022693  0.0018478   1.228   0.219  
cpu_tier^5  -0.0001457  0.0015587  -0.093   0.926  
gpu_tier.L  0.3454299  0.0067431   51.227 < 2e-16 ***
gpu_tier.Q -0.0320581  0.0029600  -10.830 < 2e-16 ***
gpu_tier.C -0.0022358  0.0022078  -1.013   0.311  
gpu_tier^4  0.0028483  0.0018621   1.530   0.126  
gpu_tier^5  -0.0020772  0.0016412  -1.266   0.206  
ram_gb      0.0002141  0.0001063   2.014   0.044 *  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.1734 on 59364 degrees of freedom
Multiple R-squared:  0.658,    Adjusted R-squared:  0.658 
F-statistic: 1.038e+04 on 11 and 59364 DF,  p-value: < 2.2e-16
```

Slika 142 Statistički rezime Modela 3

Sledeći logičan korak u pravljenju modela jeste dodavanje ram\_gb tj. količine RAM memorije koju uređaj poseduje, kao prediktor. RAM ima dobru pozitivnu korelaciju sa cenom i obično skuplji uređaji dolaze sa većom količinom RAM memorije.

Vrednosti reziduala, obe  $R^2$  metrike, F statistike i p-value su dosta slične prethodnom modelu, što znači da nam RAM ne donosi nikakvu novu informaciju

što već ne donose prethodna dva prediktora, ipak ima dobru korelaciju sa cenom i njegovo p je veoma malo 0.044, pa ćemo ga zadržati u našem modelu.

```
> m3_rmse; m3_mae
[1] 323.7903
[1] 241.9057
```

*Slika 143 RMSE i MAE trećeg modela*

## Model 4

```
model4 <- lm(log_price ~ cpu_tier + gpu_tier + ram_gb +
               cpu_power_score, data=train_data)

pred4 <- expm1(predict(model4, test_data))

m4_rmse <- sqrt(mean((pred4 - true_price)^2))
m4_mae <- mean(abs(pred4 - true_price))
```

*Slika 144 Implementacija modela 4 i evaluacionih mera (RMSE, MAE, R<sup>2</sup>) nad test skupom*

```
> summary(model4)

Call:
lm(formula = log_price ~ cpu_tier + gpu_tier + ram_gb + cpu_power_score,
    data = train_data)

Residuals:
    Min      1Q   Median      3Q     Max 
-1.34352 -0.10753 -0.00026  0.10842  1.49763 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 7.8760851  0.0081901 961.664  <2e-16 ***
cpu_tier.L  0.8418522  0.0110644  76.086  <2e-16 ***
cpu_tier.Q  0.1265267  0.0042029  30.105  <2e-16 ***
cpu_tier.C  0.0405490  0.0022965  17.657  <2e-16 ***
cpu_tier^4  0.0156126  0.0018176   8.590  <2e-16 ***
cpu_tier^5  0.0169382  0.0015506  10.924  <2e-16 ***
gpu_tier.L  0.3706740  0.0065902  56.246  <2e-16 ***
gpu_tier.Q -0.0288706  0.0028866 -10.002  <2e-16 ***
gpu_tier.C  0.0005403  0.0021532   0.251   0.8019  
gpu_tier^4  0.0031356  0.0018155   1.727   0.0842 .  
gpu_tier^5 -0.0017504  0.0016002  -1.094   0.2740  
ram_gb     0.0003477  0.0001037   3.353   0.0008 ***  
cpu_power_score -0.0096269  0.0001733 -55.539  <2e-16 ***  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.169 on 59363 degrees of freedom
Multiple R-squared:  0.6749,    Adjusted R-squared:  0.6749 
F-statistic: 1.027e+04 on 12 and 59363 DF,  p-value: < 2.2e-16
```

*Slika 145 Statistički rezime Modela 4*

U naredni model dodajemo promenljivu koju smo napravili u okviru feature engineering faze, a to je cpu\_power\_score koja predstavlja kako kombinacija broja jezgara i frekvencije procesora utiču na cenu.

Reziduali se sužavaju u odnosu na prethodni model, p vrednost za ram je sada još manja nego u prethodnom modelu što govori, da je dobro što je ram ostao u modelu. Obe  $R^2$  metrike su se povećale za ~2% i za F statistiku i p-value donosi se isti zaključak kao i ranije. Ovaj prediktor je značajan.

```
> m4_rmse; m4_mae
[1] 313.0251
[1] 233.2059
```

*Slika 146 RMSE i MAE četvrtog modela*

## Model 5

```
model_5 <- lm(log_price ~ cpu_tier + gpu_tier + ram_gb +
                  cpu_power_score + cgt_score, data=train_data)

pred_5 <- expm1(predict(model_5, test_data))

m5_rmse <- sqrt(mean((pred_5 - true_price)^2))
m5_mae <- mean(abs(pred_5 - true_price))
```

*Slika 147 Implementacija modela 5 i evaluacionih mera (RMSE, MAE,  $R^2$ ) nad test skupom*

```
> summary(model_5)

Call:
lm(formula = log_price ~ cpu_tier + gpu_tier + ram_gb + cpu_power_score +
    cgt_score, data = train_data)

Residuals:
    Min      1Q      Median      3Q      Max 
-1.34618 -0.10764 -0.00008  0.10819  1.49000 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 8.0826319  0.0310257 260.514 < 2e-16 ***
cpu_tier.L  1.0299154  0.0294071  35.023 < 2e-16 ***
cpu_tier.Q  0.1707065  0.0076567  22.295 < 2e-16 ***
cpu_tier.C  0.0417517  0.0023022  18.135 < 2e-16 ***
cpu_tier^4  0.0160196  0.0018178   8.812 < 2e-16 ***
cpu_tier^5  0.0166661  0.0015505  10.749 < 2e-16 ***
gpu_tier.L  0.5901958  0.0324811  18.170 < 2e-16 ***
gpu_tier.Q  0.0121287  0.0066040   1.837  0.0663 .  
gpu_tier.C  0.0010864  0.0021538   0.504  0.6140    
gpu_tier^4  0.0039408  0.0018186   2.167  0.0302 *  
gpu_tier^5  -0.0023761  0.0016021  -1.483  0.1381    
ram_gb     0.0007424  0.0001184   6.271 3.61e-10 ***
cpu_power_score -0.0095407 0.0001737 -54.920 < 2e-16 ***
cgtscore   -0.0154554  0.0022393  -6.902 5.18e-12 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.169 on 59362 degrees of freedom
Multiple R-squared:  0.6752, Adjusted R-squared:  0.6751 
F-statistic: 9492 on 13 and 59362 DF,  p-value: < 2.2e-16
```

*Slika 148 Statistički rezime Modela 5*

U naredni model dodajemo još jednu promenljivu koju smo napravili u FE fazi koja predstavlja kombinaciju ranga procesora i ranga grafičke kartice. Zbog ovog obeležja

upravo vidimo kakav je uticaj kombinovane snage uređaja, dosta se često dešava da uređaj imaju slabiji proces i jaku grafičku karticu ili obrnuto.

Reziduali su slični onima iz prethodnog modela. P vrednost za cgt score je izuzetno mala što pokazuje da je ovo značajan prediktor. F statistika je nešto manja nego kod prethodnih modela, ali i dalje poprilično visoka što nam govori da je ovaj model u celini dosta značajan. Obe  $R^2$  metrike su minimalno porasle.

```
> m5_rmse; m5_mae  
[1] 312.9573  
[1] 233.1662
```

Slika 149 RMSE i MAE petog modela

## Model 6

```
model_6 <- lm(log_price ~ cpu_tier + gpu_tier + ram_gb +  
                  cpu_power_score + cgt_score + storage_gb,  
                  data=train_data)  
  
pred_6 <- expm1(predict(model_6, test_data))  
  
m6_rmse <- sqrt(mean((pred_6 - true_price)^2))  
m6_mae <- mean(abs(pred_6 - true_price))
```

Slika 150 Implementacija modela 6 i evaluacionih mera (RMSE, MAE,  $R^2$ ) nad test skupom

U ovom modelu dodali smo i promenljivu storage\_gb. Korelacija sa cenom je minimalna, ali zbog domenskog znanja znamo da je ovaj faktor bitan, posebno ako se posmatra i tip memorije u uređaju.

```
> m6_rmse; m6_mae  
[1] 309.7857  
[1] 230.3027
```

Slika 151 RMSE i MAE šestog modela

```

> summary(model_6)

Call:
lm(formula = log_price ~ cpu_tier + gpu_tier + ram_gb + cpu_power_score +
    cgt_score + storage_gb, data = train_data)

Residuals:
    Min      1Q  Median      3Q     Max 
-1.35132 -0.10592 -0.00036  0.10580  1.48472 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 8.052e+00 3.056e-02 263.513 < 2e-16 ***  
cpu_tier.L   1.032e+00 2.896e-02  35.629 < 2e-16 ***  
cpu_tier.Q   1.712e-01 7.539e-03 22.703 < 2e-16 ***  
cpu_tier.C   4.110e-02 2.267e-03 18.131 < 2e-16 ***  
cpu_tier^4   1.577e-02 1.790e-03  8.808 < 2e-16 ***  
cpu_tier^5   1.628e-02 1.527e-03 10.664 < 2e-16 ***  
gpu_tier.L   5.965e-01 3.198e-02 18.650 < 2e-16 ***  
gpu_tier.Q   1.285e-02 6.503e-03  1.975 0.0482 *    
gpu_tier.C   7.438e-04 2.121e-03  0.351 0.7258    
gpu_tier^4   3.837e-03 1.791e-03  2.143 0.0321 *    
gpu_tier^5   -2.324e-03 1.578e-03 -1.473 0.1407    
ram_gb       7.508e-04 1.166e-04  6.441 1.20e-10 ***  
cpu_power_score -9.513e-03 1.711e-04 -55.616 < 2e-16 ***  
cgf_score    -1.583e-02 2.205e-03 -7.178 7.16e-13 ***  
storage_gb   3.808e-05 8.812e-07 43.210 < 2e-16 ***  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.1664 on 59361 degrees of freedom
Multiple R-squared:  0.6851,    Adjusted R-squared:  0.685 
F-statistic:  9224 on 14 and 59361 DF,  p-value: < 2.2e-16

```

*Slika 152 Statistički rezime Modela 6*

Vrednost reziduala je nešto manja u odnosu na prethodni model. P za storage\_gb je poprilično mala što nam govori da je ovo dobar prediktor. F statistika i p vrednost je slična kao i pre, a obe  $R^2$  metrike su se povećale za 1%.

### Model 7

```

model_7 <- lm(log_price ~ cpu_tier + gpu_tier + ram_gb +
    cpu_power_score + cgf_score + storage_gb + brand,
    data=train_data)

pred_7 <- expm1(predict(model_7, test_data))

m7_rmse <- sqrt(mean((pred_7 - true_price)^2))
m7_mae <- mean(abs(pred_7 - true_price))

```

*Slika 153 Implementacija modela 7 i evaluacionih mera (RMSE, MAE,  $R^2$ ) nad test skupom*

U ovaj model dodajemo promenljivu brand. Iako smo se do sada bavili uglavnom hardverskim cenama, brend uređaja je izuzetno bitan. Neki brendovi proizvode samo specifične uređaji, neki pored realno cene naplaćuju i ime samog brenda.

```
> summary(model_7)

Call:
lm(formula = log_price ~ cpu_tier + gpu_tier + ram_gb + cpu_power_score +
    cgt_score + storage_gb + brand, data = train_data)

Residuals:
    Min      1Q  Median      3Q     Max 
-1.33339 -0.09876  0.00258  0.10212  1.46828 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 7.981e+00  2.890e-02 276.165 < 2e-16 ***
cpu_tier.L  1.017e+00  2.733e-02  37.227 < 2e-16 ***
cpu_tier.Q  1.666e-01  7.116e-03 23.415 < 2e-16 ***
cpu_tier.C  3.706e-02  2.140e-03 17.317 < 2e-16 ***
cpu_tier^4  1.535e-02  1.689e-03  9.086 < 2e-16 ***
cpu_tier^5  1.595e-02  1.441e-03 11.067 < 2e-16 ***
gpu_tier.L  5.918e-01  3.018e-02 19.607 < 2e-16 ***
gpu_tier.Q  1.261e-02  6.137e-03  2.054  0.0400 *  
gpu_tier.C  1.529e-03  2.001e-03  0.764  0.4449    
gpu_tier^4  3.921e-03  1.690e-03  2.320  0.0203 *  
gpu_tier^5 -1.845e-03  1.489e-03 -1.239  0.2152    
ram_gb      7.192e-04  1.100e-04  6.537  6.31e-11 ***
cpu_power_score -9.515e-03 1.615e-04 -58.932 < 2e-16 ***
cgtscore   -1.549e-02  2.081e-03 -7.442 1.01e-13 *** 
storage_gb  3.833e-05  8.316e-07 46.085 < 2e-16 ***
brandApple 3.318e-01  4.268e-03 77.751 < 2e-16 ***
brandASUS   4.623e-02  2.740e-03 16.876 < 2e-16 ***
brandDell   6.651e-02  2.551e-03 26.074 < 2e-16 ***
brandGigabyte 4.874e-02  3.407e-03 14.304 < 2e-16 ***
brandHP     4.772e-02  2.545e-03 18.750 < 2e-16 ***
brandLenovo  5.939e-02  2.479e-03 23.957 < 2e-16 ***
brandMSI    7.719e-02  2.929e-03 26.350 < 2e-16 ***
brandRazer   1.668e-01  4.023e-03 41.462 < 2e-16 ***
brandSamsung 8.969e-02  2.908e-03 30.843 < 2e-16 ***

---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.157 on 59352 degrees of freedom
Multiple R-squared:  0.7196,    Adjusted R-squared:  0.7195 
F-statistic: 6622 on 23 and 59352 DF,  p-value: < 2.2e-16
```

Slika 154 Statistički rezime Modela 7

Reziduali se sužavaju, p za sve brendove je jako niska, F statistika malo niža nego ranije, ali i dalje dosta visoka, p-value dosta nizak kao i ranije, ali najznačajniji je porast obe  $R^2$  metrike za 5%, što nam govori da je ovo izuzetno biran prediktor.

```
> m7_rmse; m7_mae  
[1] 285.0065  
[1] 214.8316
```

Slika 155 RMSE i MAE sedmog modela

## Model 8

```
model_8 <- lm(log_price ~ cpu_tier + gpu_tier + ram_gb +  
                  cpu_power_score + cgt_score + storage_gb +  
                  brand + os,  
                  data=train_data)  
  
pred_8 <- expm1(predict(model_8, test_data))  
  
m8_rmse <- sqrt(mean((pred_8 - true_price)^2))  
m8_mae <- mean(abs(pred_8 - true_price))
```

Slika 156 Implementacija modela 8 i evaluacionih mera (RMSE, MAE,  $R^2$ ) nad test skupom

```
> m8_rmse; m8_mae  
[1] 284.4012  
[1] 214.5646
```

Slika 157 RMSE i MAE osmog modela

U model 8 dodajemo i os tj. operativni sistem. Mac operativni sistem je vezan isključivo za Apple, a ostale kompanije proizvode uređaje sa drugim operativnim sistemima. Želimo da vidimo koliko koji operativni sistem utiče na cenu.

```

> summary(model_8)

Call:
lm(formula = log_price ~ cpu_tier + gpu_tier + ram_gb + cpu_power_score +
    cgt_score + storage_gb + brand + os, data = train_data)

Residuals:
    Min      1Q  Median      3Q     Max 
-1.33772 -0.09847  0.00245  0.10220  1.46409 

Coefficients: (1 not defined because of singularities)
              Estimate Std. Error t value Pr(>|t|)    
(Intercept) 7.936e+00 2.897e-02 273.927 < 2e-16 ***  
cpu_tier.L   1.016e+00 2.725e-02  37.288 < 2e-16 ***  
cpu_tier.Q   1.664e-01 7.095e-03  23.451 < 2e-16 ***  
cpu_tier.C   3.691e-02 2.134e-03  17.298 < 2e-16 ***  
cpu_tier^4   1.535e-02 1.684e-03   9.112 < 2e-16 ***  
cpu_tier^5   1.588e-02 1.437e-03  11.054 < 2e-16 ***  
gpu_tier.L   5.909e-01 3.010e-02  19.631 < 2e-16 ***  
gpu_tier.Q   1.234e-02 6.119e-03   2.016  0.0438 *    
gpu_tier.C   1.464e-03 1.996e-03   0.734  0.4632    
gpu_tier^4   4.043e-03 1.685e-03   2.399  0.0164 *    
gpu_tier^5   -2.078e-03 1.485e-03  -1.400  0.1615    
ram_gb       7.214e-04 1.097e-04   6.576  4.86e-11 ***  
cpu_power_score -9.516e-03 1.610e-04 -59.108 < 2e-16 ***  
cgtscore     -1.542e-02 2.075e-03  -7.433 1.08e-13 ***  
storage_gb   3.835e-05 8.293e-07  46.247 < 2e-16 ***  
brandApple   3.759e-01 5.158e-03  72.877 < 2e-16 ***  
brandASUS    4.643e-02 2.732e-03  16.998 < 2e-16 ***  
brandDell    6.676e-02 2.544e-03  26.244 < 2e-16 ***  
brandGigabyte 4.920e-02 3.398e-03  14.478 < 2e-16 ***  
brandHP      4.811e-02 2.538e-03  18.957 < 2e-16 ***  
brandLenovo   5.984e-02 2.472e-03  24.205 < 2e-16 ***  
brandMSI     7.741e-02 2.921e-03  26.501 < 2e-16 ***  
brandRazer   1.668e-01 4.011e-03  41.588 < 2e-16 ***  
brandSamsung  9.006e-02 2.900e-03  31.055 < 2e-16 ***  
osLinux      2.182e-02 3.804e-03   5.737  9.70e-09 ***  
osmacOS       NA        NA        NA        NA      
osWindows    4.801e-02 3.046e-03  15.760 < 2e-16 ***  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.1565 on 59350 degrees of freedom
Multiple R-squared:  0.7212,    Adjusted R-squared:  0.7211 
F-statistic:  6141 on 25 and 59350 DF,  p-value: < 2.2e-16

```

Slika 158 Statistički rezime Modela 8

Reziduali su kao i sa svakim novim modelom malo uži, F statistika i P su slični kao i ranije. Obe  $R^2$  metrike su minimalno porasle, ali na 100k redova i malo povećanje je značajno. Treba naglasiti da su NA vrednosti za sve kod mac os-a zato što samo

Apple uređaji imaju ovaj operativni sistem, pa se ne donosi absolutno nikakav novi zaključak.

## Model 9

```
model_9 <- lm(log_price ~ cpu_tier + gpu_tier + ram_gb +
                 cpu_power_score + cgt_score + storage_gb +
                 brand + os + device_type,
                 data=train_data)

pred_9 <- expm1(predict(model_9, test_data))

m9_rmse <- sqrt(mean((pred_9 - true_price)^2))
m9_mae <- mean(abs(pred_9 - true_price))
```

Slika 159 Implementacija modela 9 i evaluacionih mera (RMSE, MAE, R<sup>2</sup>) nad test skupom

```
> summary(model_9)

Call:
lm(formula = log_price ~ cpu_tier + gpu_tier + ram_gb + cpu_power_score +
    cgt_score + storage_gb + brand + os + device_type, data = train_data)

Residuals:
    Min      1Q  Median      3Q     Max 
-1.27481 -0.09540  0.00195  0.09660  1.41123 

Coefficients: (1 not defined because of singularities)
              Estimate Std. Error t value Pr(>|t|)    
(Intercept) 7.469e+00  2.849e-02 262.196 < 2e-16 ***
cpu_tier.L  4.246e-01  2.734e-02 15.530 < 2e-16 ***
cpu_tier.Q  1.038e-02  7.127e-03 1.456  0.14539    
cpu_tier.C  2.763e-03  2.098e-03 1.317  0.18774    
cpu_tier^4  2.741e-03  1.623e-03 1.688  0.09133 .  
cpu_tier^5  4.540e-04  1.393e-03 0.326  0.74444    
gpu_tier.L  5.402e-01  2.885e-02 18.728 < 2e-16 ***
gpu_tier.Q  -1.086e-02  5.872e-03 -1.850  0.06429 .  
gpu_tier.C  3.032e-03  1.912e-03 1.586  0.11285    
gpu_tier^4  2.953e-03  1.615e-03 1.829  0.06742 .  
gpu_tier^5  -1.624e-03  1.422e-03 -1.142  0.25344    
ram_gb      6.614e-04  1.051e-04 6.293  3.14e-10 ***
cpu_power_score -6.348e-04  1.966e-04 -3.229  0.00124 ** 
cgtscore   -9.647e-03  1.990e-03 -4.849  1.25e-06 ***
storage_gb  3.814e-05  7.945e-07 48.001 < 2e-16 ***
brandApple 3.772e-01  4.941e-03 76.337 < 2e-16 ***
brandASUS   4.760e-02  2.617e-03 18.186 < 2e-16 ***
brandDell   6.820e-02  2.437e-03 27.981 < 2e-16 ***
brandGigabyte 4.999e-02  3.256e-03 15.356 < 2e-16 ***
brandHP     4.916e-02  2.432e-03 20.215 < 2e-16 ***
brandLenovo  6.002e-02  2.369e-03 25.341 < 2e-16 ***
brandMSI    7.798e-02  2.799e-03 27.863 < 2e-16 ***
brandRazer  1.693e-01  3.843e-03 44.062 < 2e-16 ***
brandSamsung 9.020e-02  2.778e-03 32.467 < 2e-16 ***
osLinux    2.229e-02  3.645e-03 6.115  9.70e-10 *** 
osmacOS      NA        NA        NA        NA      
osWindows   4.771e-02  2.919e-03 16.345 < 2e-16 *** 
device_typeLaptop 1.194e-01  1.638e-03 72.848 < 2e-16 *** 
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.15 on 59349 degrees of freedom
Multiple R-squared:  0.7441,    Adjusted R-squared:  0.744 
F-statistic: 6637 on 26 and 59349 DF,  p-value: < 2.2e-16
```

Slika 160 Statistički rezime Modela 9

U model 9 dodajemo i promenljivu device\_type. Laptopovi i računari uglavnom imaju malo drugačiji raspored cena u odnosu na druge komponente i želimo da vidimo koliki to ima uticaj na model.

Reziduali su se standardno smanjili, ali je P za tip uređaja izuzetno mala i obe  $R^2$  metrike su se povećale za više od 2% što ovaj prediktor čini poprilično dobrim.

```
> m9_rmse; m9_mae  
[1] 272.0706  
[1] 204.7432
```

Slika 161 RMSE i MAE devetog modela

## Model 10

```
model_10 <- lm(log_price ~ cpu_tier + gpu_tier + ram_gb +  
                  cpu_power_score + cgt_score + storage_gb +  
                  brand + os + device_type + cpu_generation,  
                  data=train_data)  
  
pred_10 <- expm1(predict(model_10, test_data))  
  
m10_rmse <- sqrt(mean((pred_10 - true_price)^2))  
m10_mae <- mean(abs(pred_10 - true_price))
```

Slika 162 Implementacija modela 10 i evaluacionih mera (RMSE, MAE,  $R^2$ ) nad test skupom

```
> m10_rmse; m10_mae  
[1] 272.2064  
[1] 204.7727
```

Slika 163 RMSE i MAE desetog modela

U model 10 dodajemo i poslednju promenljivu dobijenu u feature engineering fazi, a to je cpu\_generation, koju smo dobili upravo izdvajanjem zajedničkih imena u modelima uređaja iz kojih se vidi kojoj generaciji, tog brenda, pripada procesor koji se nalazi u njima.

```

> summary(model_10)

Call:
lm(formula = log_price ~ cpu_tier + gpu_tier + ram_gb + cpu_power_score +
    cgt_score + storage_gb + brand + os + device_type + cpu_generation,
    data = train_data)

Residuals:
    Min      1Q  Median      3Q     Max 
-1.27439 -0.09520  0.00203  0.09663  1.41168 

Coefficients: (1 not defined because of singularities)
              Estimate Std. Error t value Pr(>|t|)    
(Intercept) 7.471e+00 2.851e-02 262.025 < 2e-16 ***
cpu_tier.L   3.984e-01 2.812e-02 14.165 < 2e-16 ***
cpu_tier.Q   1.940e-03 1.030e-02 0.188 0.850616    
cpu_tier.C   1.812e-02 6.944e-03 2.609 0.009076 **  
cpu_tier^4  -2.412e-04 3.050e-03 -0.079 0.936961    
cpu_tier^5  -9.855e-03 2.686e-03 -3.669 0.000244 *** 
gpu_tier.L   5.401e-01 2.884e-02 18.727 < 2e-16 ***
gpu_tier.Q  -1.091e-02 5.871e-03 -1.859 0.063054 .  
gpu_tier.C   2.965e-03 1.912e-03 1.551 0.120970    
gpu_tier^4  2.869e-03 1.614e-03 1.777 0.075558 .  
gpu_tier^5  -1.621e-03 1.422e-03 -1.140 0.254396    
ram_gb       6.593e-04 1.051e-04 6.274 3.55e-10 *** 
cpu_power_score -6.319e-04 1.966e-04 -3.214 0.001308 ** 
cgtscore    -9.638e-03 1.989e-03 -4.845 1.27e-06 *** 
storage_gb   3.811e-05 7.944e-07 47.978 < 2e-16 *** 
brandApple  3.664e-01 6.608e-03 55.449 < 2e-16 *** 
brandASUS   4.760e-02 2.617e-03 18.191 < 2e-16 *** 
brandDell   6.821e-02 2.437e-03 27.991 < 2e-16 *** 
brandGigabyte 4.998e-02 3.255e-03 15.357 < 2e-16 *** 
brandHP     4.916e-02 2.431e-03 20.219 < 2e-16 *** 
brandLenovo  6.002e-02 2.368e-03 25.346 < 2e-16 *** 
brandMSI    7.798e-02 2.798e-03 27.870 < 2e-16 *** 
brandRazer  1.694e-01 3.843e-03 44.073 < 2e-16 *** 
brandSamsung 9.022e-02 2.778e-03 32.478 < 2e-16 *** 
osLinux     2.230e-02 3.644e-03 6.120 9.42e-10 *** 
osmacOS      NA        NA        NA        NA      
osWindows   4.771e-02 2.918e-03 16.350 < 2e-16 *** 
device_typeLaptop 1.194e-01 1.638e-03 72.864 < 2e-16 *** 
cpu_generation.L 2.155e-02 6.634e-03 3.249 0.001159 ** 
cpu_generation.Q 8.271e-03 7.073e-03 1.169 0.242300    

cpu_generation.C -1.458e-02 5.237e-03 -2.785 0.005355 ** 
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.15 on 59346 degrees of freedom
Multiple R-squared:  0.7442,    Adjusted R-squared:  0.7441 
F-statistic: 5953 on 29 and 59346 DF,  p-value: < 2.2e-16

```

Slika 164 Statistički rezime Modela 10

Metrike su uglavnom ostale skoro iste i ovaj prediktor nam ne daje ništa toliko značajno što već nije moglo da se zaključi od ranije, jer već imamo 10 prediktora i većina od njih je izuzetno važna za model. P za svaku generaciju procesora, osim jedne, su dosta male tako da ćemo ipak ostaviti ovaj prediktor, već je dosta stvari poznato zato i minimalno utiče na povećanje tj. smanjenje metrika.

## Model 11

```
model_11 <- lm(log_price ~ cpu_tier + gpu_tier + ram_gb +
                  cpu_power_score + cgt_score + storage_gb +
                  brand + os + device_type + cpu_generation + vram_gb,
                  data=train_data)

pred_11 <- expm1(predict(model_11, test_data))

m11_rmse <- sqrt(mean((pred_11 - true_price)^2))
m11_mae <- mean(abs(pred_11 - true_price))
```

*Slika 165 Implementacija modela 11 i evaluacionih mera (RMSE, MAE, R<sup>2</sup>) nad test skupom*

```

> summary(model1_11)

Call:
lm(formula = log_price ~ cpu_tier + gpu_tier + ram_gb + cpu_power_score +
    cgt_score + storage_gb + brand + os + device_type + cpu_generation +
    vram_gb, data = train_data)

Residuals:
    Min      1Q   Median      3Q     Max 
-1.27857 -0.09542  0.00185  0.09633  1.40866 

Coefficients: (1 not defined because of singularities)
                Estimate Std. Error t value Pr(>|t|)    
(Intercept) 7.451e+00  2.852e-02 261.238 < 2e-16 ***
cpu_tier.L  4.044e-01  2.809e-02 14.397 < 2e-16 ***
cpu_tier.Q  2.847e-03  1.029e-02  0.277  0.78197    
cpu_tier.C  1.517e-02  6.940e-03  2.185  0.02886 *  
cpu_tier^4 -1.188e-04  3.046e-03 -0.039  0.96889    
cpu_tier^5 -7.787e-03  2.688e-03 -2.896  0.00378 ** 
gpu_tier.L  5.216e-01  2.885e-02 18.083 < 2e-16 ***
gpu_tier.Q  -1.298e-02  5.866e-03 -2.213  0.02692 *  
gpu_tier.C  1.386e-03  1.914e-03  0.724  0.46899    
gpu_tier^4  2.234e-03  1.613e-03  1.385  0.16621    
gpu_tier^5 -1.860e-03  1.421e-03 -1.309  0.19040    
ram_gb     6.636e-04  1.050e-04  6.323  2.58e-10 ***
cpu_power_score -6.282e-04  1.963e-04 -3.199  0.00138 ** 
cgtscore   -9.696e-03  1.987e-03 -4.880  1.06e-06 *** 
storage_gb 3.804e-05  7.935e-07 47.940 < 2e-16 *** 
brandApple 3.824e-01  6.734e-03 56.791 < 2e-16 *** 
brandASUS  4.762e-02  2.614e-03 18.218 < 2e-16 *** 
brandDell  6.805e-02  2.434e-03 27.962 < 2e-16 *** 
brandGigabyte 4.995e-02  3.251e-03 15.366 < 2e-16 *** 
brandHP    4.898e-02  2.428e-03 20.171 < 2e-16 *** 
brandLenovo 5.994e-02  2.365e-03 25.343 < 2e-16 *** 
brandMSI   7.779e-02  2.795e-03 27.833 < 2e-16 *** 
brandRazer 1.692e-01  3.838e-03 44.092 < 2e-16 *** 
brandSamsung 9.021e-02  2.774e-03 32.516 < 2e-16 *** 
osLinux   2.216e-02  3.640e-03  6.089 1.14e-09 *** 
osmacOS    NA        NA        NA        NA        
oswindows  4.765e-02  2.915e-03 16.347 < 2e-16 *** 
device_typeLaptop 1.195e-01  1.636e-03 73.014 < 2e-16 *** 
cpu_generation.L 1.706e-02  6.637e-03  2.571  0.01015 *  
cpu_generation.Q 7.832e-03  7.065e-03  1.109  0.26760    
cpu_generation.C -1.163e-02  5.236e-03 -2.220  0.02641 *  
vram_gb    2.564e-03  2.138e-04 11.990 < 2e-16 *** 

```

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.1498 on 59345 degrees of freedom

Multiple R-squared: 0.7448, Adjusted R-squared: 0.7447

F-statistic: 5774 on 30 and 59345 DF, p-value: < 2.2e-16

### Slika 166 Statistički rezime Modela 11

U modelu 11 dodajemo i vram\_gb promenljiva koja predstavlja koliko RAM memorije se nalazi u grafičkoj kartici, što je jako bitno, posebno kod gaming uređaja.

Reziduali su još manji nego pre, obe  $R^2$  metrike su se malo povećale što je pozitivno i samo P za ovaj prediktor je izuzetno malo, zajedno sa svim do sad navedenim i domenskim znanjem, dobija dodatne informacije o gpu koji je jedna od najbitnijih komponenti bilo kog uređaja, ovaj prediktor je značajan za naš model.

```
> m11_rmse; m11_mae  
[1] 271.7866  
[1] 204.5186
```

*Slika 167 RMSE i MAE jedanaestog modela*

## Model 12

```
model_12 <- lm(log_price ~ cpu_tier + gpu_tier + ram_gb +  
                  cpu_power_score + cgt_score + storage_gb +  
                  brand + os + device_type + cpu_generation + vram_gb + release_year,  
                  data=train_data)  
  
pred_12 <- expm1(predict(model_12, test_data))  
  
m12_rmse <- sqrt(mean((pred_12 - true_price)^2))  
m12_mae <- mean(abs(pred_12 - true_price))
```

*Slika 168 Implementacija modela 12 i evaluacionih mera (RMSE, MAE, R<sup>2</sup>) nad test skupom*

Za finalni model dodajemo i godinu izdavanja uređaja. Godinu izdavanja dodajemo zato što noviji uređaji gotovo uvek imaju veću cenu. Hardver brzo zastareva, a svake godine izlazi nova generacija komponenti. Iako korelacija nije visoka, release\_year je važan indikator tehnološke svežine uređaja i zato ima statistički značajan doprinos predikciji cene.

```

> summary(model_12)

Call:
lm(formula = log_price ~ cpu_tier + gpu_tier + ram_gb + cpu_power_score +
    cgt_score + storage_gb + brand + os + device_type + cpu_generation +
    vram_gb + release_year, data = train_data)

Residuals:
    Min      1Q  Median      3Q     Max 
-1.25870 -0.09422  0.00075  0.09385  1.43985 

Coefficients: (1 not defined because of singularities)
                Estimate Std. Error t value Pr(>|t|)    
(Intercept) -2.287e+01  6.004e-01 -38.098 < 2e-16 ***
cpu_tier.L   3.962e-01  2.751e-02  14.405 < 2e-16 ***
cpu_tier.Q   3.160e-03  1.007e-02   0.314  0.75374  
cpu_tier.C   1.222e-02  6.795e-03   1.798  0.07224 .  
cpu_tier^4   5.268e-04  2.982e-03   0.177  0.85979  
cpu_tier^5   -7.908e-03 2.632e-03  -3.004  0.00266 ** 
gpu_tier.L   5.191e-01  2.824e-02  18.381 < 2e-16 *** 
gpu_tier.Q   -1.398e-02 5.744e-03  -2.434  0.01495 *  
gpu_tier.C   7.704e-04  1.874e-03   0.411  0.68105  
gpu_tier^4   2.014e-03  1.580e-03   1.275  0.20231  
gpu_tier^5   -1.945e-03 1.391e-03  -1.398  0.16212  
ram_gb       6.695e-04  1.028e-04   6.515  7.34e-11 *** 
cpu_power_score -5.470e-04 1.923e-04  -2.845  0.00444 ** 
cgtscore     -9.582e-03 1.945e-03  -4.926  8.43e-07 *** 
storage_gb   3.821e-05  7.769e-07  49.177 < 2e-16 *** 
brandApple   3.826e-01  6.594e-03  58.025 < 2e-16 *** 
brandASUS    4.798e-02  2.559e-03  18.750 < 2e-16 *** 
brandDell    6.809e-02  2.383e-03  28.574 < 2e-16 *** 
brandGigabyte 5.002e-02  3.183e-03  15.715 < 2e-16 *** 
brandHP      4.896e-02  2.378e-03  20.589 < 2e-16 *** 
brandLenovo   5.965e-02  2.316e-03  25.754 < 2e-16 *** 
brandMSI     7.784e-02  2.737e-03  28.445 < 2e-16 *** 
brandRazer   1.692e-01  3.758e-03  45.013 < 2e-16 *** 
brandSamsung 8.967e-02  2.717e-03  33.008 < 2e-16 *** 
osLinux      2.284e-02  3.564e-03   6.410  1.46e-10 *** 
osmacOS       NA        NA        NA        NA      
osWindows    4.781e-02  2.854e-03  16.752 < 2e-16 *** 
device_typeLaptop 1.197e-01  1.602e-03  74.709 < 2e-16 *** 
cpu_generation.L 1.994e-02  6.498e-03   3.069  0.00215 ** 
cpu_generation.Q 5.791e-03  6.918e-03   0.837  0.40249  
cpu_generation.C -9.992e-03 5.127e-03  -1.949  0.05133 .  
vram_gb       2.583e-03  2.094e-04  12.339 < 2e-16 *** 
release_year  1.499e-02  2.965e-04  50.564 < 2e-16 *** 
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.1466 on 59344 degrees of freedom
Multiple R-squared:  0.7553,    Adjusted R-squared:  0.7552 
F-statistic:  5910 on 31 and 59344 DF,  p-value: < 2.2e-16

```

Slika 169 Statistički rezime Modela 12

Suženost reziduala je slična prethodnih nekoliko modela, ali je P za ovaj prediktor izuzetno malo i obe  $R^2$  metrike su se povećale za preko 1%, a već su i do ovog modela metrike bile izuzetno velike, a na ~74k redova ovo je znatno poboljšanje i ovo naš prediktor čini izuzetno dobrom.

```
> m12_rmse; m12_mae
[1] 266.3943
[1] 199.8705
```

*Slika 170 RMSE i MAE dvanaestog modela*

U odnosu na početni model 1 greške su uspele dosta da se smanje RMSE sa 346.5 na 266.4 i MAE sa 261.6 na 199.9\$.  $R^2$  metrike su se povećale sa ~0.6077 na ~0.7553. Ovi podaci nam govore da je i naš početni model bio dobar, ali daleko od toga da se samo sa jednim prediktorom, iako je veoma bitan, može napraviti sjajan model. Finalni model (Model 12) pokazuje da je cena računara visoko predvidiva na osnovu kombinacije hardverskih karakteristika i nekoliko kategorijskih indikatora i ovo će biti naš finalni model.

### Model samo laptopovi

Izdvojićemo još dva dodatna modela, prvi je model koji u skupu podataka sadrži samo laptopove.

```
laptopovi = datav4[datav4$device_type == "Laptop", ]
```

*Slika 171 Izdvajanje laptopova u poseban skup*

```
set.seed(123)

n_lap = nrow(laptopovi)
train_idx_lap = sample(seq_len(n_lap), size = 0.8 * n_lap)

train_lap = laptops[train_idx_lap, ]
test_lap = laptops[-train_idx_lap, ]

nrow(train_lap)
nrow(test_lap)

model_lap = lm(log_price ~ cpu_tier + gpu_tier + ram_gb + cgt_score + storage_gb +
  brand + os + cpu_generation + vram_gb + release_year +
  charger_watts, data = train_lap)

pred_lap = expm1(predict(model_lap, test_lap))
true_lap = test_lap$price

lap_rmse = sqrt(mean((pred_lap - true_lap)^2))
lap_mae = mean(abs(pred_lap - true_lap))
```

*Slika 172 Implementacija modela koji sadrži samo laptopove i evaluacionih mera (RMSE, MAE,  $R^2$ ) nad test skupom*

Kao i za prethodne modele liniarne regresije delimo skup podataka na train i test, treniramo model, predviđamo cenu i računamo RMSE i MAE. Za model smo koristili neke najbitnije prediktore, kao što su bili i za ceo skup podataka, a ideja je bila da se fokusiramo na prediktore vezane samo za laptop, charger\_watts se dobro pokazao, za razliku od battery\_wh koji nije neki bitan prediktor.

```
> lap_rmse  
[1] 263.3885  
> lap_mae  
[1] 201.47
```

*Slika 173 RMSE i MAE modela koji sadrži samo laptopove*

```

> summary(model_lap)

Call:
lm(formula = log_price ~ cpu_tier + gpu_tier + ram_gb + cgt_score +
    storage_gb + brand + os + cpu_generation + vram_gb + release_year +
    charger_watts, data = train_lap)

Residuals:
    Min      1Q  Median      3Q     Max 
-1.15149 -0.09215  0.00040  0.09182  1.44011 

Coefficients: (1 not defined because of singularities)
                Estimate Std. Error t value Pr(>|t|)    
(Intercept) -2.146e+01  7.596e-01 -28.251 < 2e-16 ***
cpu_tier.L   3.877e-01  2.869e-02 13.511 < 2e-16 ***
cpu_tier.Q   1.740e-02  1.200e-02  1.449  0.14729  
cpu_tier.C   1.234e-02  8.880e-03  1.390  0.16460  
cpu_tier^4    8.861e-03  3.798e-03  2.333  0.01964 *  
cpu_tier^5   -1.092e-02  3.389e-03 -3.223  0.00127 ** 
gpu_tier.L   5.510e-01  3.298e-02 16.704 < 2e-16 ***
gpu_tier.Q   -8.887e-03 6.549e-03 -1.357  0.17482  
gpu_tier.C   3.577e-03  2.412e-03  1.483  0.13814  
gpu_tier^4    2.945e-03  2.042e-03  1.442  0.14918  
gpu_tier^5   -3.059e-04  1.802e-03 -0.170  0.86523  
ram_gb       6.013e-04  1.245e-04  4.830  1.37e-06 ***
cgtscore    -1.035e-02  2.163e-03 -4.785  1.72e-06 *** 
storage_gb   3.747e-05  9.766e-07 38.372 < 2e-16 *** 
brandApple   3.943e-01  8.391e-03 46.989 < 2e-16 *** 
brandASUS    4.559e-02  3.223e-03 14.146 < 2e-16 *** 
brandDell    6.848e-02  2.994e-03 22.875 < 2e-16 *** 
brandGigabyte 4.767e-02  4.006e-03 11.901 < 2e-16 *** 
brandHP      4.844e-02  2.998e-03 16.159 < 2e-16 *** 
brandLenovo   5.939e-02  2.910e-03 20.410 < 2e-16 *** 
brandMSI     7.459e-02  3.468e-03 21.510 < 2e-16 *** 
brandRazer   1.636e-01  4.750e-03 34.443 < 2e-16 *** 
brandSamsung  8.737e-02  3.401e-03 25.689 < 2e-16 *** 
osLinux      2.395e-02  4.559e-03  5.254 1.50e-07 *** 
osmacOS        NA       NA       NA       NA      
osWindows    4.679e-02  3.665e-03 12.769 < 2e-16 *** 
cpu_generation.L 1.344e-02  8.314e-03  1.617  0.10589  
cpu_generation.Q -1.347e-02  8.932e-03 -1.508  0.13150  
cpu_generation.C -1.380e-02  6.769e-03 -2.039  0.04149 *  
vram_gb      2.472e-03  2.730e-04  9.054 < 2e-16 *** 
release_year  1.435e-02  3.753e-04 38.234 < 2e-16 *** 
charger_watts 5.967e-05  1.567e-05  3.808  0.00014 *** 
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1432 on 35351 degrees of freedom
Multiple R-squared:  0.7653,    Adjusted R-squared:  0.7651 
F-statistic: 3842 on 30 and 35351 DF,  p-value: < 2.2e-16

```

*Slika 174 Statistički rezime modela koji sadrži samo laptopove*

Ovaj model je pokazao veoma dobre rezultate, čak šta više ima najveću vrednost  $R^2$  metrike do sada (ali naravno ovo je samo deo podataka nije ceo skup kao malopre).

F statistika je velika i p vrednost je mala što nam ukazuje da smo izabrali uglavnom dobre prediktore. Možemo primetiti i da ovaj model daje najmanju RMSE  $\sim 263,4$ .

## Model samo računari

```
racunari = datav4[datav4$device_type == "Desktop",]
```

*Slika 175 Izdvajanje računara u poseban skup*

```
set.seed(123)
```

```
n_rac = nrow(racunari)
train_idx_rac = sample(seq_len(n_rac), size = 0.8 * n_rac)

train_rac = racunari[train_idx_rac, ]
test_rac = racunari[-train_idx_rac, ]

nrow(train_rac)
nrow(test_rac)

model_rac = lm(log_price ~ cpu_tier + gpu_tier + ram_gb + storage_gb +
    brand + os + vram_gb + release_year + psu_watts, data = train_rac)

pred_rac = expm1(predict(model_rac, test_rac))
true_rac = test_rac$price

rac_rmse = sqrt(mean((pred_rac - true_rac)^2))
rac_mae = mean(abs(pred_rac - true_rac))
```

*Slika 176 Implementacija modela koji sadrži samo računare i evaluacionih mera (RMSE, MAE, R<sup>2</sup>) nad test skupom*

```
> rac_rmse
[1] 256.5342
> rac_mae
[1] 195.7146
```

*Slika 177 RMSE i MAE modela koji sadrži samo računare*

```

> summary(model_rac)

Call:
lm(formula = log_price ~ cpu_tier + gpu_tier + ram_gb + storage_gb +
    brand + os + vram_gb + release_year + psu_watts, data = train_rac)

Residuals:
    Min      1Q  Median      3Q     Max 
-1.25062 -0.09606  0.00168  0.09573  1.38508 

Coefficients: (1 not defined because of singularities)
              Estimate Std. Error t value Pr(>|t|)    
(Intercept) -2.530e+01  9.757e-01 -25.925 < 2e-16 ***
cpu_tier.L   2.268e-01  1.056e-02  21.485 < 2e-16 ***
cpu_tier.Q   -3.671e-02  4.930e-03 -7.447 9.90e-14 ***
cpu_tier.C   2.355e-03  3.210e-03  0.733  0.4633    
cpu_tier^4   5.397e-05  2.589e-03  0.021  0.9834    
cpu_tier^5   -2.204e-05  2.162e-03 -0.010  0.9919    
gpu_tier.L   3.633e-01  9.389e-03 38.694 < 2e-16 ***
gpu_tier.Q   -3.432e-02  4.204e-03 -8.164 3.41e-16 ***
gpu_tier.C   -3.456e-03  3.073e-03 -1.124  0.2608    
gpu_tier^4   4.570e-03  2.519e-03  1.814  0.0697 .  
gpu_tier^5   -2.928e-03  2.192e-03 -1.336  0.1816    
ram_gb       7.452e-04  1.451e-04  5.136  2.83e-07 ***
storage_gb   4.005e-05  1.267e-06 31.609 < 2e-16 ***
brandApple   4.103e-01  8.009e-03 51.225 < 2e-16 ***
brandASUS    5.418e-02  4.180e-03 12.962 < 2e-16 ***
brandDell    7.413e-02  3.887e-03 19.069 < 2e-16 ***
brandGigabyte 5.690e-02  5.124e-03 11.105 < 2e-16 ***
brandHP      5.473e-02  3.894e-03 14.054 < 2e-16 ***
brandLenovo   6.501e-02  3.788e-03 17.162 < 2e-16 ***
brandMSI     8.734e-02  4.470e-03 19.539 < 2e-16 ***
brandRazer   1.759e-01  6.181e-03 28.462 < 2e-16 ***
brandSamsung  9.736e-02  4.511e-03 21.582 < 2e-16 ***
osLinux      2.364e-02  5.825e-03  4.059  4.94e-05 ***
osmacOS        NA         NA         NA         NA        
osWindows    4.892e-02  4.697e-03 10.415 < 2e-16 ***
vram_gb      2.945e-03  3.277e-04  8.985 < 2e-16 ***
release_year  1.609e-02  4.824e-04 33.351 < 2e-16 ***
psu_watts    2.889e-05  5.037e-06  5.736  9.83e-09 ***

---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.1515 on 23967 degrees of freedom
Multiple R-squared:  0.7273,    Adjusted R-squared:  0.727 
F-statistic: 2458 on 26 and 23967 DF,  p-value: < 2.2e-16

```

Slika 178 Statistički rezime modela koji sadrži samo računare

Slično kao i prethodni, ovaj model je dao dobre rezultate metrika, malo niže  $R^2$  od ostalih, ali zato ubedljivo najniža vrednost RMSE i MAE. F statistika i p vrednost su sličnih vrednosti kao i u prošlom modelu što nam pokazuje da je ovo dobar model.

## Random Forest

### Uvod

Sledeći model, koji ćemo koristiti je Random Forest. Random forest je model mašinskog učenja, koji radi pomoću stabla odlučivanja. On sadrži više stabala unutar sebe i „pravi“ šumu, tj. povezuje stabla i kroz njih propušta podatke.

Ovaj model deluje kao dobra opcija za predviđanje cene uređaja jer su podaci kod nas takvi da je veoma lako pretpostaviti da li je uređaj skuplji ili jeftiniji samo na osnovu marke komponente ili operativnog sistema računara. To se podudara radom stabla odlučivanja.

### Treniranje modela

```

install.packages("ranger")
library(ranger)

# Treniranje modela

rf_model <- ranger(
  log_price ~ cpu_tier + gpu_tier + ram_gb +
    cpu_power_score + cgt_score + storage_gb +
    brand + os + device_type + cpu_generation +
    vram_gb + release_year + battery_wh + warranty_months ,
  data = train_data,
  num.trees = 500,
  mtry = floor(sqrt(13)),
  min.node.size = 5,
  sample.fraction = 0.75,
  importance = "impurity",
  seed = 123
)

```

*Slika 179 R kod treniranja random forest*

Na slici iznad možemo videti kod treniranja modela random forest. Korišćena je biblioteka „ranger“. Za treniranje modela korišćeni su prediktori iz najboljeg modela linearne regresije. Broj stabala je postavljena na 500, sample.fraction je postavljena na 0.75, to znači da će svako drvo videti samo nasumičnih 75% podataka iz trening skupa, zajedno sa tim seed je postavljen na 123. Takvom postavkom smo dosta smanjili mogućnost overfittovanja.

## Predikcija

```
# Predikcije na test skupu  
  
rf_pred <- expm1(predict(rf_model, test_data)$predictions)  
true_price <- test_data$price  
  
# Metrike  
  
rf_rmse <- sqrt(mean((rf_pred - true_price)^2))  
rf_mae <- mean(abs(rf_pred - true_price))  
rf_r2 <- 1 - sum((rf_pred - true_price)^2) /  
    sum((true_price - mean(true_price))^2)  
  
rf_rmse  
rf_mae  
rf_r2
```

Slika 180 R kod predikcije i izračunavanja metrika

```
> rf_rmse  
[1] 271.044  
> rf_mae  
[1] 203.1626  
> rf_r2  
[1] 0.755676
```

Slika 181 metrike random forest modela

Sa slike iznad možemo videti koliko je naš model bio precizan u predviđanju cene uređaja. Kao i kod linearne regresije, koristili smo 3 metrike:

1. RMSE
2. MAE
3.  $R^2$

Na osnovu slike vidimo da je RMSE malo veći od RMSE kod linearne regresije (266) i takođe je MAE malo veći, nego što je to bio slučaj kod linearne regresije (200). Ovo nam ukazuje da naš model generalno pravi malo veću grešku u odnosu na linearnu regresiju, ali greška nije mnogo velika.

Poslednja metrika je  $R^2$  score, sa slike vidimo da model ima  $R^2$  score od 0.756, što je malo više nego što je bio slučaj kod najboljeg modela linearne regresije (0.755). Zaključak je da je ovaj model random forest-a zanemarljivo precizniji od najboljeg modela linearne regresije i pravi minimalno veću grešku pri predviđanju. Takođe, možemo zaključiti da je ovaj model veoma dobar i prepoznaće trendove u dovoljno preciznoj meri.

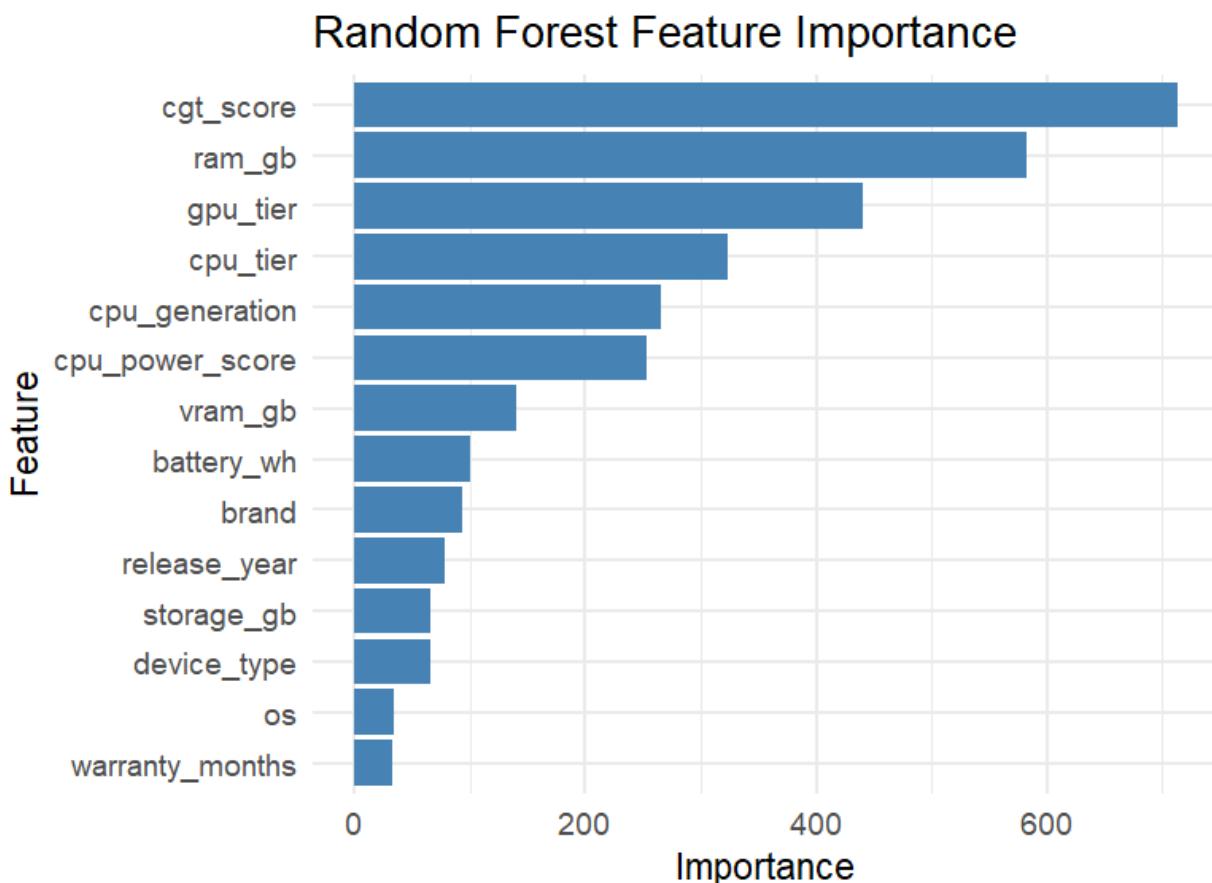
## Feature importance

```
imp <- data.frame(
  feature = names(rf_model$variable.importance),
  importance = rf_model$variable.importance
) %>% arrange(desc(importance))

print(imp)

ggplot(imp, aes(x = reorder(feature, importance), y = importance)) +
  geom_col(fill = "steelblue") +
  coord_flip() +
  labs(
    title = "Random Forest Feature Importance",
    x = "Feature",
    y = "Importance"
  ) +
  theme_minimal(base_size = 14)
```

Slika 182 R kod feature importance



Slika 183 Grafik važnosti prediktora

Poslednji deo kod ispitivanja modela je prikazivanje važnosti prediktora, tj. koliko su prediktori u stvari imali uticaja u predviđanju cene. Kod linearne regresije je to moguće videti komandom *summary()*, gde bi smo onda gledali one zvezdice, koje bi nam rekle koliko je taj feature imao uticaja. Ovde smo to prikazali grafički.

Sa grafika možemo videti da je *cgt\_score* imao najveći uticaj i bio je proglašen za najboljeg prediktora, što je i za očekivati, sa obzirom na to da je *cgt\_score* imao najveću korelaciju sa cenom. Važno je još navesti da operativni sistem i brand nisu imali tolikog uticaja, kao što smo u početku prepostavili da će imati.

## XGBoost

### Uvod

Sledeći model koji ćemo ispitivati je XGBoost. XGBoost je veoma sličan Random Forest-u po principu rada, sa određenim promenama. Dok Random Forest koristi više stabala odlučivanja da naprave predikciju, XGBoost koristi među stabla koja se bave ispravljanjem greške prethodnog stabla odlučivanja. Time se uglavnom dobija bolja preciznost, ali je dosta zahtevnije.

### Treniranje modela

```
install.packages("xgboost")
library(xgboost)

# priprema

numeric_cols <- names(datav4)[sapply(datav4, is.numeric)]
numeric_cols <- setdiff(numeric_cols, c("price", "log_price"))

train_matrix <- as.matrix(train_data[, numeric_cols])
test_matrix <- as.matrix(test_data[, numeric_cols])

train_label <- train_data$log_price
test_label <- test_data$price

# treniranje

xgb_model <- xgboost(
  data = train_matrix,
  label = train_label,
  nrounds = 100,
  objective = "reg:squarederror",
  tree_method = "hist",
  verbose = 0
)
```

*Slika 184 R kod treniranja modela xgboost*

Na slici iznad možemo videti kod, koji je korišćen za treniranje modela XGBoost-a. Prvo što se primećuje je da postoji deo pripreme podataka, što do sada nije

postojalo kod linearne regresije i Random Forest-a. U toj pripremi se izdvajaju samo numerički podaci. Razlog tome je što XGBoost zahteva da se sve kategoriske promenljive enkodiraju. Međutim, pošto nas obojica imamo samo laptopove sa samo 16GB RAM-a, a naš skup ima oko 70.000 podataka i 10-ak kategoriskih feature-a, nismo bili u mogućnosti da enkodiramo sve podatke i pokrenemo algoritam za trening XGBoost. Odlučili smo da koristimo samo numeričke podatke u predikciji cene.

## Predikcija

```
# predikcija

xgb_pred_log <- predict(xgb_model, newdata = test_matrix)

xgb_pred <- expm1(xgb_pred_log)

true_price <- test_data$price

xgb_rmse <- sqrt(mean((xgb_pred - true_price)^2))
xgb_mae <- mean(abs(xgb_pred - true_price))
xgb_r2 <- 1 - sum((xgb_pred - true_price)^2) /
sum((true_price - mean(true_price))^2)

xgb_rmse
xgb_mae
xgb_r2
```

Slika 185 R kod predikcije i izračunavanja metrika

```
> xgb_rmse
[1] 272.2438
> xgb_mae
[1] 204.3575
> xgb_r2
[1] 0.7535081
```

Slika 186 metrike XGBoost

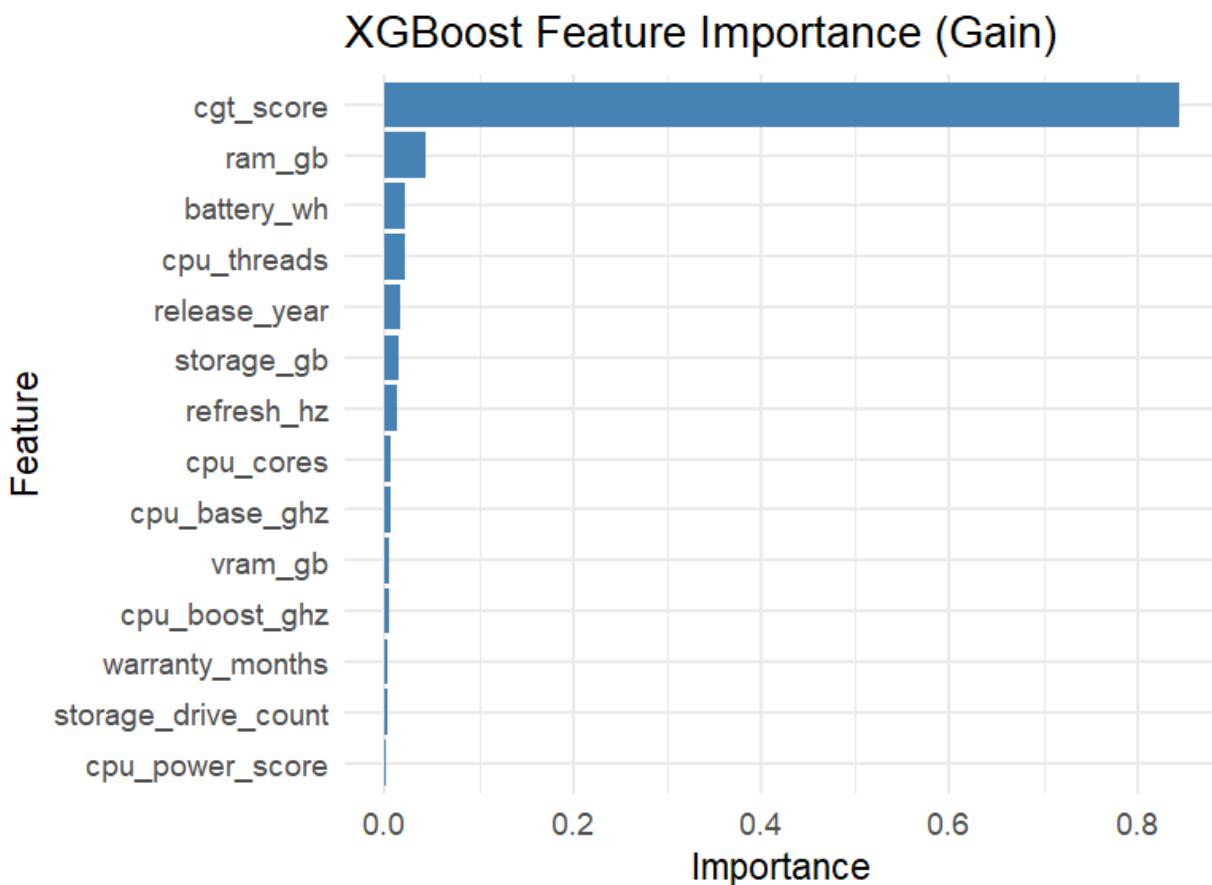
Na slikama iznad vidimo iste 3 metrike, koje smo koristili i kod linearne regresije i kod Random Forest-a. Vidimo da je RMSE i MAE veoma sličan RMSE i MAE Random Forest-a, gde je XGBoost pravio samo malo veću grešku u odnosu na Random Forest.

$R^2$  score, tj. preciznost našeg modela je 0.753, što je za nijansu niža u odnosu na linearne regresiju i Random Forest. To nije čudno, sa obzirom na to da XGBoost nije imao pristup kategoriskim promenljivama. Kada sagledamo sve metrike, zaključujemo da je model XGBoost pravio malo višu grešku i samim tim imao za nijansu neprecizniju predikciju, ali je model i dalje veoma dobar jer prepozna trendove i može biti pouzdan u većini slučajeva.

## Feature importance

```
xgb_imp_raw <- xgb.importance(  
    model = xgb_model,  
    feature_names = colnames(train_matrix)  
)  
  
xgb_imp <- xgb_imp_raw %>%  
    select(feature = Feature, importance = Gain) %>%  
    arrange(desc(importance))  
  
print(xgb_imp)  
  
# grafik  
  
ggplot(xgb_imp, aes(x = reorder(feature, importance), y = importance)) +  
    geom_col(fill = "steelblue") +  
    coord_flip() +  
    labs(  
        title = "XGBoost Feature Importance (Gain)",  
        x = "Feature",  
        y = "Importance"  
) +  
    theme_minimal(base_size = 14)
```

*Slika 187 R kod prikazivanja važnosti prediktora*



*Slika 188 Grafik važnosti prediktora*

Kod XGBoost-a smo takođe prikazali važnost prediktora grafički. Sa grafika se vidi veoma čudna i neuobičajena situacija. Cgt\_score je prikazan kao najbitniji feature, koji sam predstavlja 85% bitnosti, dok su svi ostali daleko manje bitni. Tu sad najbolje vidimo razliku između XGBoost-a i Random Forest-a. XGBoost je prepoznao da je cgt\_score dovoljan prediktor i sve ostale je ostavio po strani, tj. koristio je ostale samo kada su mu bili neophodni, u onim specifičnim slučajevima.

## Lasso regresija

### Uvod

Poslednji model koji ćemo ispitivati je Lasso. Lasso radi slično linearnoj regresiji, ali za razliku od linearne regresije, Lasso može sam da odabere najbolje prediktore,

dodavanjem penala (kazni). Lasso dodaje L1 penal smanjuje koeficijent određenih feature i u nekim slučajevima može da ih postavi na 0, tj. da ih potpuno izbaci.

## Treniranje

```
library(glmnet)

x = model.matrix(
  log_price ~ cpu_tier + gpu_tier + ram_gb +
  cpu_power_score + cgt_score + storage_gb +
  brand + os + device_type + cpu_generation + vram_gb + release_year,
  data = train_data
)[, -1]

y = train_data$log_price

lasso_model = cv.glmnet(
  x, y,
  alpha = 1,
  nfolds = 10,
  type.measure = "mse"
)

lasso_model$lambda.min

x_test = model.matrix(
  log_price ~ cpu_tier + gpu_tier + ram_gb +
  cpu_power_score + cgt_score + storage_gb +
  brand + os + device_type + cpu_generation + vram_gb + release_year,
  data = test_data
)[, -1]
```

Slika 189 R kod treniranja modela Lasso regresije

Na slici iznad se nalazi R kod za treniranje modela Lasso regresije. Korišćena je biblioteka „glmnet“. Pri pozivu *glmnet* funkcije, potrebno je postaviti *alpha* na 1, kako bi se radila Lasso regresija i takođe smo podesili da radi sa 10 foldova.

## Predikcija

```
lasso_pred = predict(lasso_model, newx = x_test, s = "lambda.min")
lasso_pred_real = expm1(lasso_pred)

lasso_rmse = sqrt(mean((lasso_pred_real - test_data$price)^2))
lasso_mae = mean(abs(lasso_pred_real - test_data$price))
lasso_r2 = 1 - sum((lasso_pred_real - test_data$price)^2) /
  sum((test_data$price - mean(test_data$price))^2)

lasso_rmse; lasso_mae; lasso_r2
```

Slika 190 R kod predikcije i računanja metrika modela

```
> lasso_rmse
[1] 266.4547
> lasso_mae
[1] 199.8371
> lasso_r2
[1] 0.7638797
```

Slika 191 metrike modela

Sa slike iznad možemo videti da su RMSE i MAE dosta slične linearnoj regresiji. Vrednosti su toliko marginalne da nema smisla upoređivati ih na tim metrikama.  $R^2$  score je zato dosta bolji od svih modela koje smo testirali do sad. To nam govori o tome da je Lasso mnogo bolje prepoznao koji prediktori su stvarno važni.

## Feature importance

```
lasso_coef = coef(lasso_model, s = "lambda.min")

lasso_imp = data.frame(
  feature = rownames(lasso_coef),
  coefficient = as.numeric(lasso_coef)
)

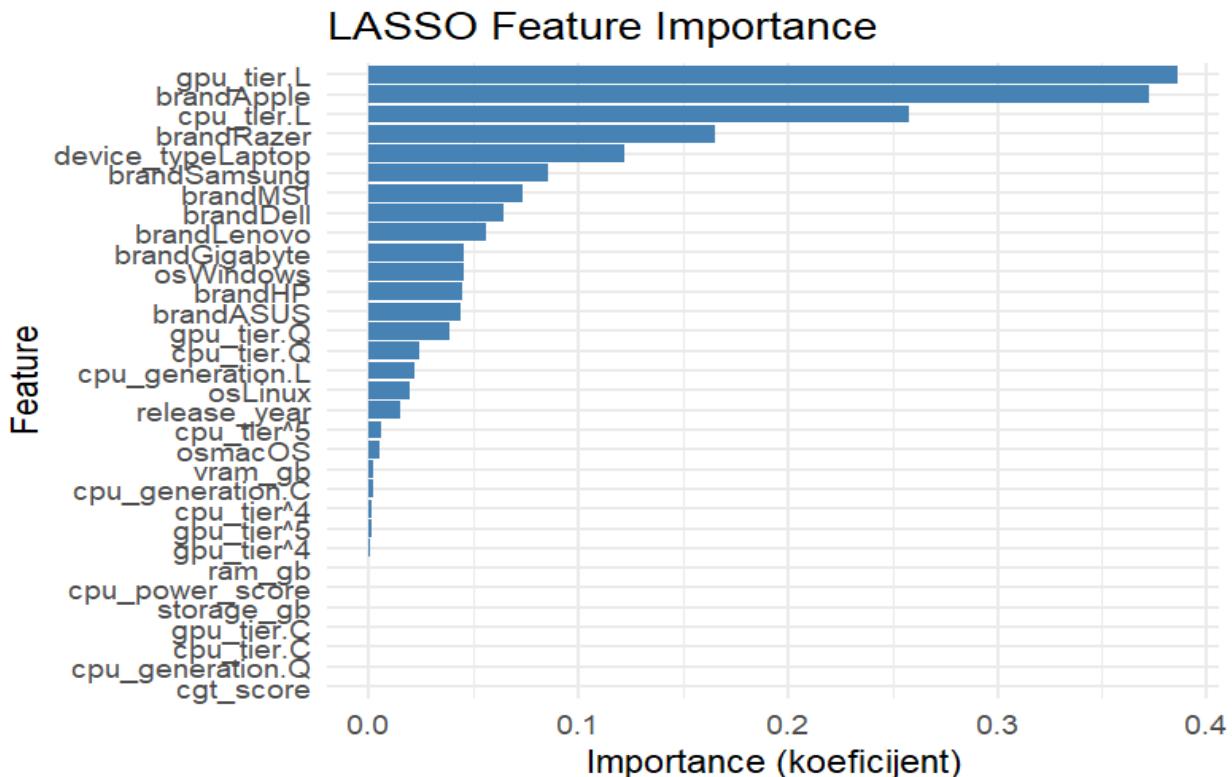
lasso_imp = lasso_imp %>% filter(feature != "(Intercept)")

lasso_imp$importance = abs(lasso_imp$coefficient)

lasso_imp = lasso_imp %>% arrange(desc(importance))

ggplot(lasso_imp, aes(x = reorder(feature, importance), y = importance)) +
  geom_col(fill = "steelblue") +
  coord_flip() +
  labs(
    title = "LASSO Feature Importance",
    x = "Feature",
    y = "Importance (koeficijent)"
) +
  theme_minimal(base_size = 14)
```

Slika 192 R kod računanja značajnosti feature



Slika 193 Grafik značajnosti feature

Sa grafika vidimo koji su feature-i bili najbitniji kod Lasso regresije. Ono što je interesantno kod ovog grafika je to što je cgt\_score postavljen na 0, tj. nije uopšte bitan. Tu primećujemo pravu moć Lasso regresije, gde je prepoznala da je to samo mešavina gpu\_tier-a i cpu\_tier-a i odbacila ga. Takođe, vidimo da je dosta posmatralo brand da li je Apple, što ostali modeli nisu toliko razmatrali. Na osnovu ovog grafika možemo razumeti kako je Lasso imao bolju predikciju na testnom skupu od ostalih modela.

## Poređenje modela

Model/Metric	RMSE	MAE	R <sup>2</sup>
Linearna regresija	266.3943	199.8705	0.7553
Random Forest	271.044	203.1626	0.7557
XGBoost	272.2438	204.3575	0.7535
LASSO regresija	266.4547	199.8371	0.7639
LR(samo laptopovi)	263.3885	201.47	0.7653
LR(samo računari)	256.5342	195.7146	0.7273

U poređenju četiri korišćena modela – linearna regresija, Random Forest, XGBoost i Lasso regresija – pokazalo se da model lasso regresije daje najbolje ukupne rezultate, ali su svakako metrike uglavnom slične.

Linearna regresija postiže najniži RMSE (266.39) i najniži MAE (199.87), što znači da predviđa cene sa najmanjim prosečnim odstupanjem. Ostvaruje solidan koeficijent determinacije R<sup>2</sup> = 0.7553, kao kod Random Forest-a. Linearna regresija se pokazala kao skoro najbolji model zato što su podaci linearni i imaju visoku korelaciju sa cenom.

Random Forest postiže nešto viši RMSE i MAE, ali ima skoro najveći R<sup>2</sup> (0.7557). Ovo znači da Random Forest objašnjava skoro najveći procenat varijanse cene, ali uz nešto veće prosečne greške. Razlog tome je što RF previše prati strukturu podataka (overfitting), pa je dobro prilagođen treniranju, ali nešto slabiji u predikciji na test skupu.

XGBoost je u ovom slučaju dao najlošiji rezultat od svih modela, iako razlika nije drastična. Njegovi RMSE i MAE su nešto veći, a  $R^2$  nešto niži. XGBoost je prepoznao cgt\_score kao ubedljivo najvažniji prediktor, dok je sve ostale feature-e posmatrao kao manje bitne, što ukazuje da model nije uspeo u potpunosti da iskoristi kompleksnost podataka.

LASSO regresija, kao regularizovana varijanta linearne regresije, dala je rezultate veoma slične klasičnoj linearnej regresiji, ali uz dodatnu prednost u selekciji najvažnijih karakteristika. Njeni RMSE, MAE i  $R^2$  su skoro isti kao u finalnom linearnom modelu, što potvrđuje da podaci imaju dominantno linearnu strukturu koju LASSO može dobro da iskoristi. Najveća prednost LASSO-a jeste to što eliminiše slabe i redundantne prediktore, čime model postaje jednostavniji i stabilniji bez gubitka performansi.

Na samom kraju dodata su i još dva dodatna modela linearne regresije, koji sadrže samo laptopove tj. samo računare i pokazali su se kao veoma dobri. Prvi je zabeležio najveću  $R^2$  vrednost, a drugi najmanju RMSE i MAE. Treba naglasiti da ovi modeli obuhvataju samo deo podataka, ne sve kao ovi ostali.

## Zaključak

Cilj projekta je bio da se skup podataka o računarima i laptopovima i njihovoj ceni, sa kaggle-a, analizira, pripremi i na kraju isproba predviđanje cene na osnovu komponenti uređaja kroz odabранe modela mašinskog učenja. Prvo što je urađeno bila je analiza celokupnog skupa podataka, gde smo dobili ideju o strukturi skupa i kako određene komponente pojedinačno i zajedno utiču na cenu. Nakon toga smo pregledali skup u potrazi za nedostajućim i nelogičnim vrednostima. Zaključili smo da nije bilo nedostajućih vrednosti, ali je bilo dosta nelogičnih i te smo odmah uklonili.

Sledeći korak u pripremi skupa, bio je EDA. U tom koraku smo iscrtavali najbitnije grafike i pokušali da razumemo koja obeležja predstavljaju najbolje prediktore. Na osnovu grafika i domenskog znanja, izdvojili smo najbitnije prediktore, a one nedovoljno dobre, smo uklonili iz skupa. Unutar Feature Engineering-a smo

pokušali da „napravimo“ nove prediktore iz postojećih i sagledali njihovu korelaciju sa cenom.

Sa FE-om smo završili pripremu podataka i sledeći korak je bio treniranje i testiranje modela mašinskog učenja. Skup podataka smo podelili na trening skup i testni skup i započeli treniranje modela. Koristili smo 4 modela mašinskog učenja i na kraju smo zaključili da je lasso regresija dala najbolje rezultate.

Sve u svemu, rad pokazuje da se cena računara može uspešno predvideti kombinovanjem tehničkih parametara i nekoliko dodatnih indikatora kao što su brend i godina izdavanja. Modeli su se pokazali dovoljno preciznim da se koriste za procenu cene uređaja. Iako postoji prostor za dalja poboljšanja, ovaj projekat predstavlja kompletan i zaokružen primer primene metoda nauke o podacima na konkretan i praktičan problem.

## Literatura

1. <https://www.kaggle.com/datasets/paperxd/all-computer-prices>
2. Microsoft Teams, kanal Uvod u nauku o podacima
3. James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An Introduction to Statistical Learning*. Springer.
4. Wickham, H., & Grolemund, G. (2017). *R for Data Science: Import, Tidy, Transform, Visualize, and Model Data*. O'Reilly Media.