Универзитет у Београду

Електротехнички факултет



Проналажење скривеног знања

Пројектни задатак за јунско-јулски рок 2024. године

Данило Станишић 2023/3332

Београд, јун 2024.

Садржај

[Задатак 1: Прикупљање података 3](#_Toc168102473)

[Задатак 2: Анализа података 4](#_Toc168102474)

[а) излистати колики је број књига за продају по категоријама књиге (жанровима) 4](#_Toc168102475)

[б) излистати колико књига се продаје од стране сваког издавача 4](#_Toc168102476)

[ц) излистати колико књига у опису садржи реч „љубав“ 4](#_Toc168102477)

[д) излистати колико књига је издато по годинама, у последњих 7 година 5](#_Toc168102478)

[е) приказати ранг листу првих 30 најскупљих књига које се продају 5](#_Toc168102479)

[ф) приказати ранг листу свих књига издатих у 2023. или 2024. години, и излистати их растуће према цени 6](#_Toc168102480)

[г) приказати књиге (Топ30) које имају: 7](#_Toc168102481)

[Задатак 3: Визуелизација података 10](#_Toc168102482)

[a) 10 најзаступљенијих издавача који имају највећи број књига у понуди 10](#_Toc168102483)

[б) број књига по категоријама (жанровима) 11](#_Toc168102484)

[ц) број издатих књига по декадама (1961-1970, 1971-1980, 1981-1990, 1991-2000, 2001-2010, 2011-данас) 11](#_Toc168102485)

[д) број (и процентуални однос) књига које се продају, за првих 5 издавачких кућа са највећим бројем књига 12](#_Toc168102486)

[е) број (и процентуални однос) свих књига за продају, које по цени припадају једном од следећих опсега (<= 500, 501-1500, 1501-3000, 3001-5000, 5001-10000, 10001-15000, > 15000) 13](#_Toc168102487)

[ф) број (и процентуални однос) књига за продају које имају тврд повез, у односу на укупан број књига за продају (издатих у последње 3 године) 14](#_Toc168102488)

[Задатак 4: Имплементација регресије 16](#_Toc168102489)

[Задатак 5: Имплементација класификације 17](#_Toc168102490)

[Задатак 6: Имплементација кластеризације 18](#_Toc168102491)

# Задатак 1: Прикупљање података

Веб индексер и веб парсер су имплементирани у *Python* програмском језику. Коришћена је библиотека *Scrapy* и прикупљени су подаци са странице <https://www.knjizare-vulkan.rs/>. Резултати овог дела пројекта чувани су у *.json* фајлу *„books.json“* са 24716 записа. Због ограниченог броја захтева ка серверима са исте *IP* адресе, захтеви су се слали на сваких 5 секунди.

Запис сваке књиге садржи следеће колоне: наслов књиге, аутор/аутори, категорија, издавач, година издања, број страна, повез, формат, опис, цена, тежина.

Пример записа:

{"title": "SAVREMENE PROMENE U NARODNOM GOVORU ", "author": "Toni Hogland", "category": "POEZIJA", "publisher": "KONTRAST", "year": "2023", "pages": "106", "binding": "Broš", "format": "14,8x21", "description": "Hoglandova neposrednost, živahnost njegovog kolokvijalnog, ali klinički preciznog izraza, njegova sposobnost da čitaoca već od prvog stiha uvuče u mehanizam svoje pjesme, pri čemu sa svakom novom strofom postepeno pojačava nivo naše pozornosti – sve to upućuje na jednu temeljnu karakteristiku povezanu s njegovim izvanrednim narativnim taktom, na onu „životvornu sposobnost za povezivanje” koju autor naziva glasom pjesnika. Nije slučajno što Hoglandova definicija za pjesnika glasi „ranjeni građanin“. U istoimenom ogledu on piše kako zadatak pjesnika „za razliku od uloge dodijeljene većini nas, nije da prikrije ili zamaskira svoju ranjenost, već da je učini upadljivo očiglednom. Pjesnici su korisni društvu tačno u tolikoj mjeri do koje je njihovo iskustvo reprezentativno – reprezentativno i ubojito iskreno”. Upravo na tom pjesničkom području ispovijesti i brutalne emotivne iskrenosti koja zalazi duboko u psihu savremenog zapadnog čovjeka, autor ostavlja najjači utisak. Hogland nas uz svoje diskretne poetičke nagovještaje neposredno podučava kako rigoroznije, ali i saosjećajnije, da sagledamo svijet oko sebe, i pokazuje na koji način možemo početi da odgonetamo tajanstvenu knjigu sopstva. (iz pogovora Marije Bergman Pelikani)", "price": "629,00", "weight": "0.5 kg"},

# Задатак 2: Анализа података

Пре саме анализе, подаци су филтрирани. Избачени су исти записи и они који имају неко од поља празно. Након филтрирања чувани су у *.json* фајлу *„books\_filter.json“* са 15744 записа. За креирање упита и саму анализу података коришћена је библиотека *Pandas*. Решење овог дела налази се у фајлу „*data\_analysis.ipynb*“.

### а) излистати колики је број књига за продају по категоријама књиге (жанровима)

Упит: combined\_data.main\_category.value\_counts()

Излаз:

main\_category

ROMANI 5819

DEČJE KNJIGE 3771

STRUČNA LITERATURA 2353

JEZIK I KNJIŽEVNOST 1085

OPŠTA INTERESOVANJA 867

STRIPOVI I MANGE 575

POPULARNA PSIHOLOGIJA I LIČNI RAZVOJ 524

TINEJDŽ I YA 420

UMETNOST 307

UDŽBENICI 23

Name: count, dtype: int64

### б) излистати колико књига се продаје од стране сваког издавача

Упит: combined\_data.publisher.value\_counts()

Излаз:

publisher

VULKAN IZDAVAŠTVO 2263

LAGUNA 2042

KREATIVNI CENTAR 611

SLUŽBENI GLASNIK 566

AKADEMSKA KNJIGA 417

...

NOVOSTI I GLAS CRKVE 1

RTS I MUZEJ VOJVODINE 1

RTS I RTV 1

LYCEUM IURIS 1

ZMAJ SEZAM BUK 1

Name: count, Length: 497, dtype: int64

### ц) излистати колико књига у опису садржи реч „љубав“

Упит: combined\_data['description'].str.contains('ljubav', case=False, na=False).sum()

Излаз:

2358

### д) излистати колико књига је издато по годинама, у последњих 7 година

Упит: combined\_data[(combined\_data['year'] >= pd.Timestamp.now().year - 6) & (combined\_data['year'] <= pd.Timestamp.now().year)].groupby('year').size()

Излаз:

year

2018 959

2019 1367

2020 1343

2021 1887

2022 1802

2023 1889

2024 346

dtype: int64

### е) приказати ранг листу првих 30 најскупљих књига које се продају

Упит: combined\_data.sort\_values(by='price', ascending=False).head(30)

Излаз:

A white sheet with black text

Description automatically generated

A white text with black text

Description automatically generated with medium confidence

### ф) приказати ранг листу свих књига издатих у 2023. или 2024. години, и излистати их растуће према цени

Упит: combined\_data[combined\_data['year'].isin([2023, 2024])].sort\_values(by='price', ascending=True)

Излаз:

A white sheet with black text

Description automatically generated

### г) приказати књиге (Топ30) које имају:

* највећи број страна

Упит: combined\_data.sort\_values(by='pages', ascending=False).head(30)

Излаз:

A white and black text

Description automatically generated with medium confidence A white and black text

Description automatically generated with medium confidence

* највећу цену

Упит: combined\_data.sort\_values(by='price', ascending=False).head(30)

Излаз:



A white and black text

Description automatically generated with medium confidence

* највећи формат

Упит: combined\_data['format\_area'] = combined\_data['format'].apply(lambda expression: str(eval(expression)))

Излаз:

A white background with black text

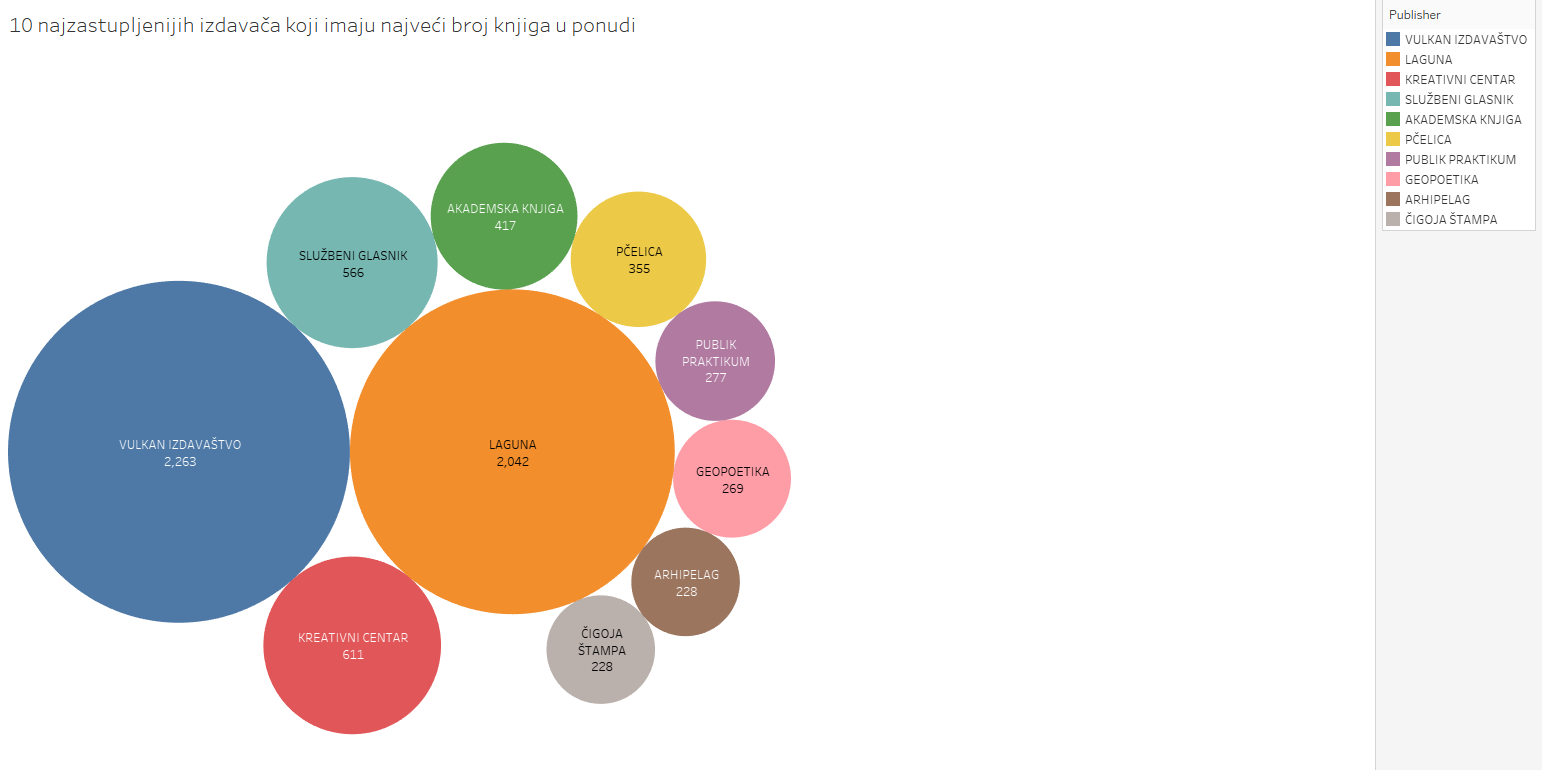
Description automatically generated A white sheet with black text

Description automatically generated

# Задатак 3: Визуелизација података

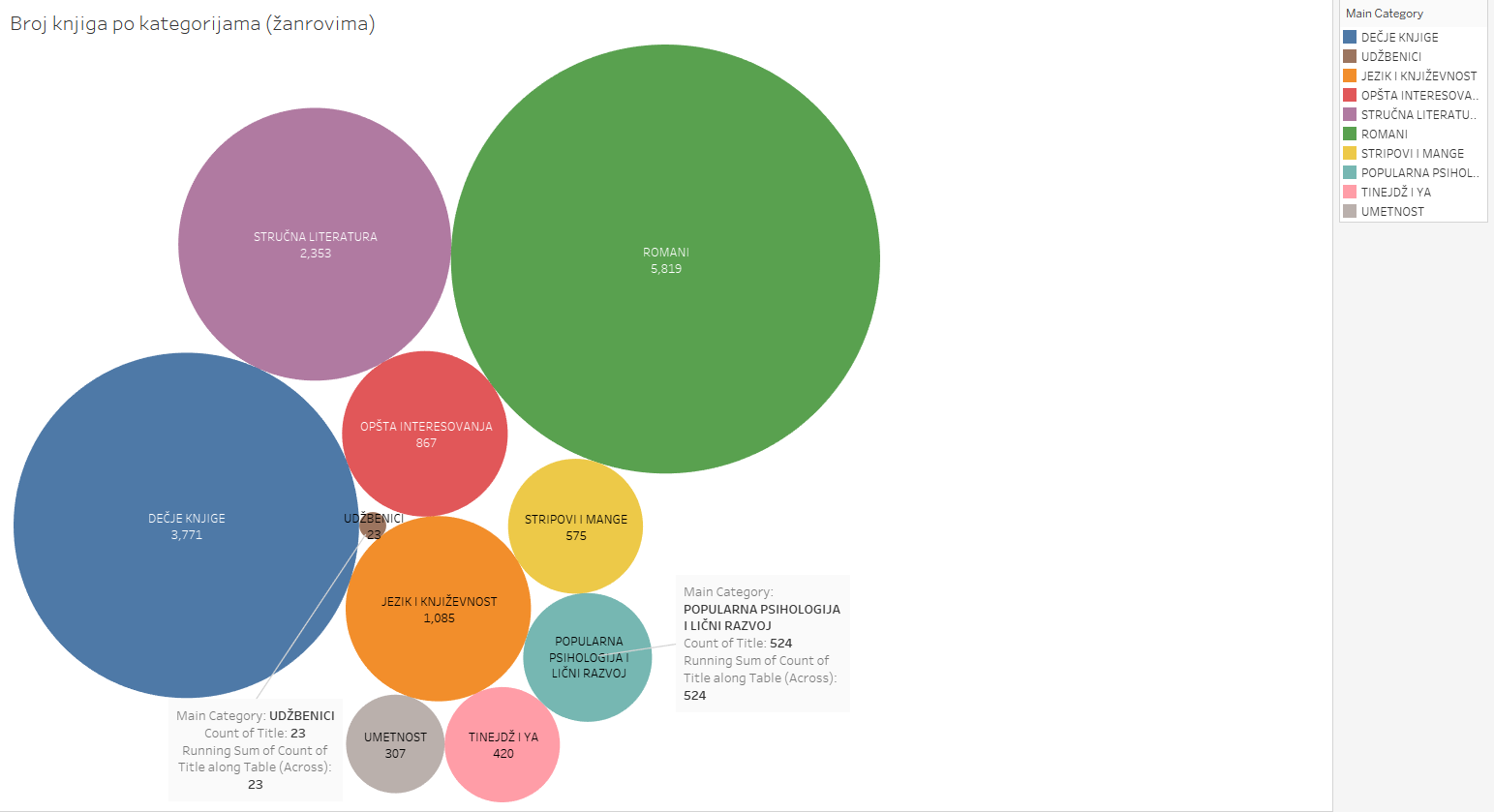
За визуелизацију података коришћен је програм *Tableau*. Слике овог дела задатка налазе се у фолдеру „*tableau*“.

### a) 10 најзаступљенијих издавача који имају највећи број књига у понуди

Излаз: 

### б) број књига по категоријама (жанровима)

Излаз:



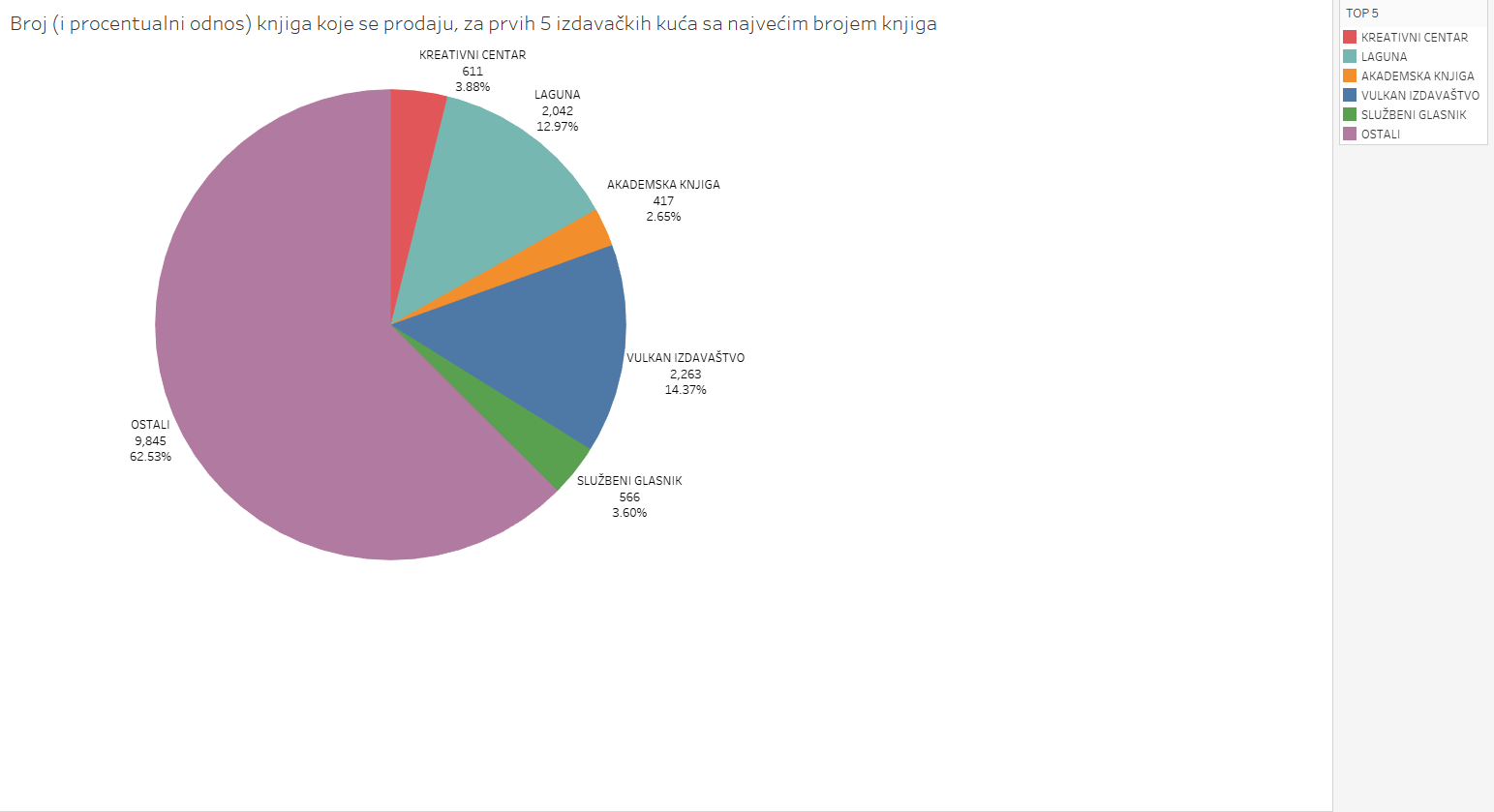
### ц) број издатих књига по декадама (1961-1970, 1971-1980, 1981-1990, 1991-2000, 2001-2010, 2011-данас)

Излаз:



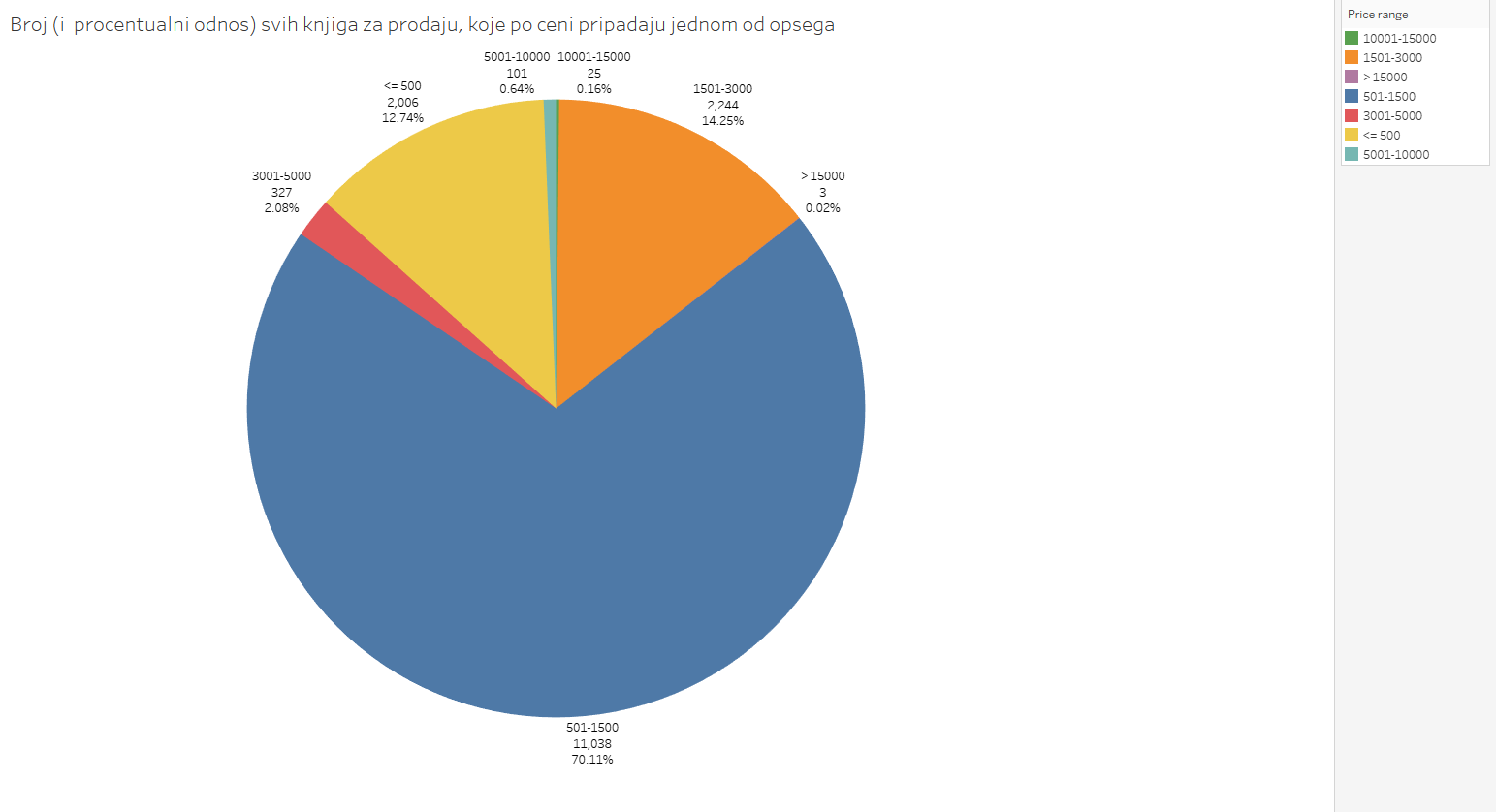
### д) број (и процентуални однос) књига које се продају, за првих 5 издавачких кућа са највећим бројем књига

Излаз:



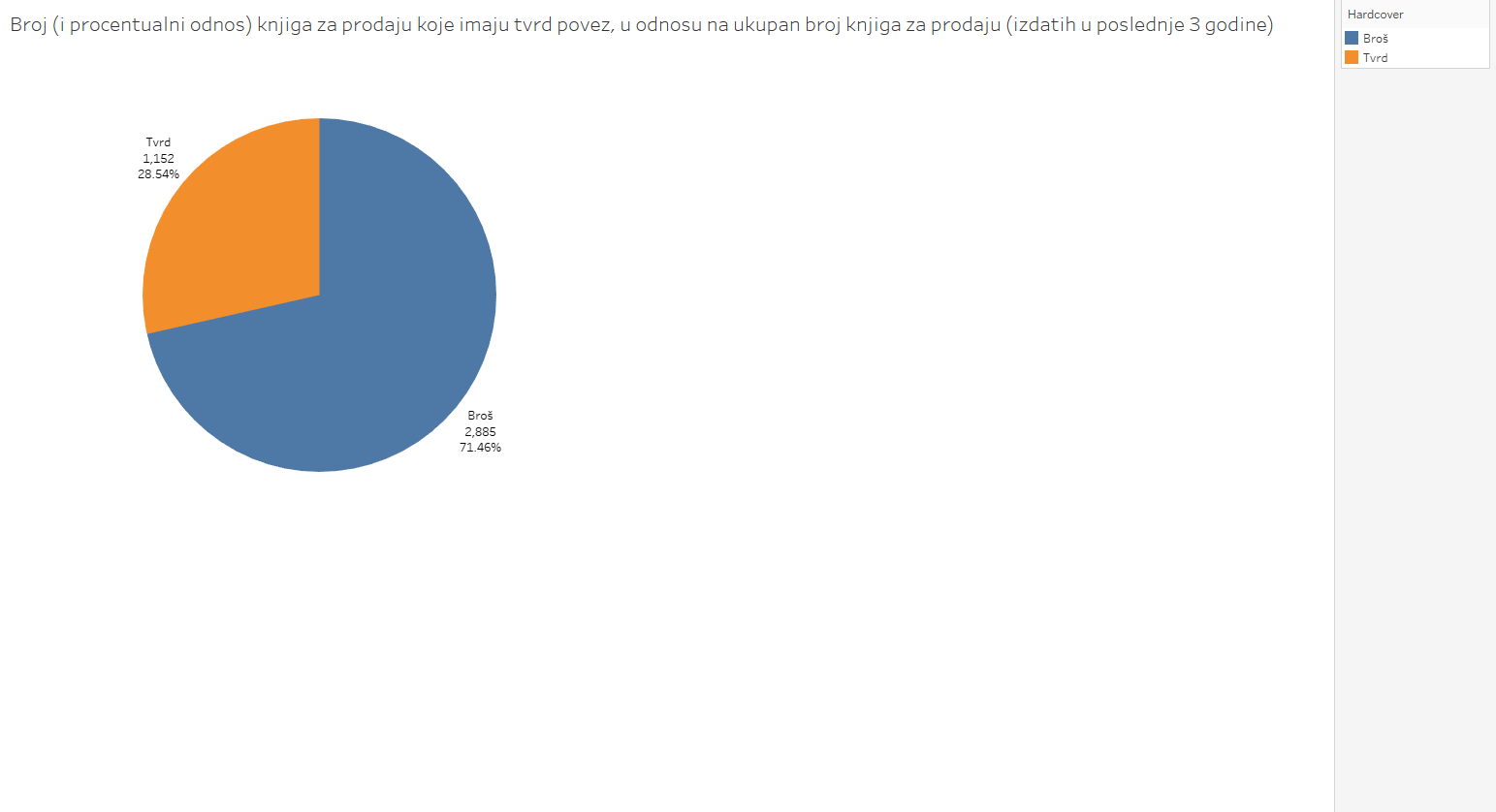
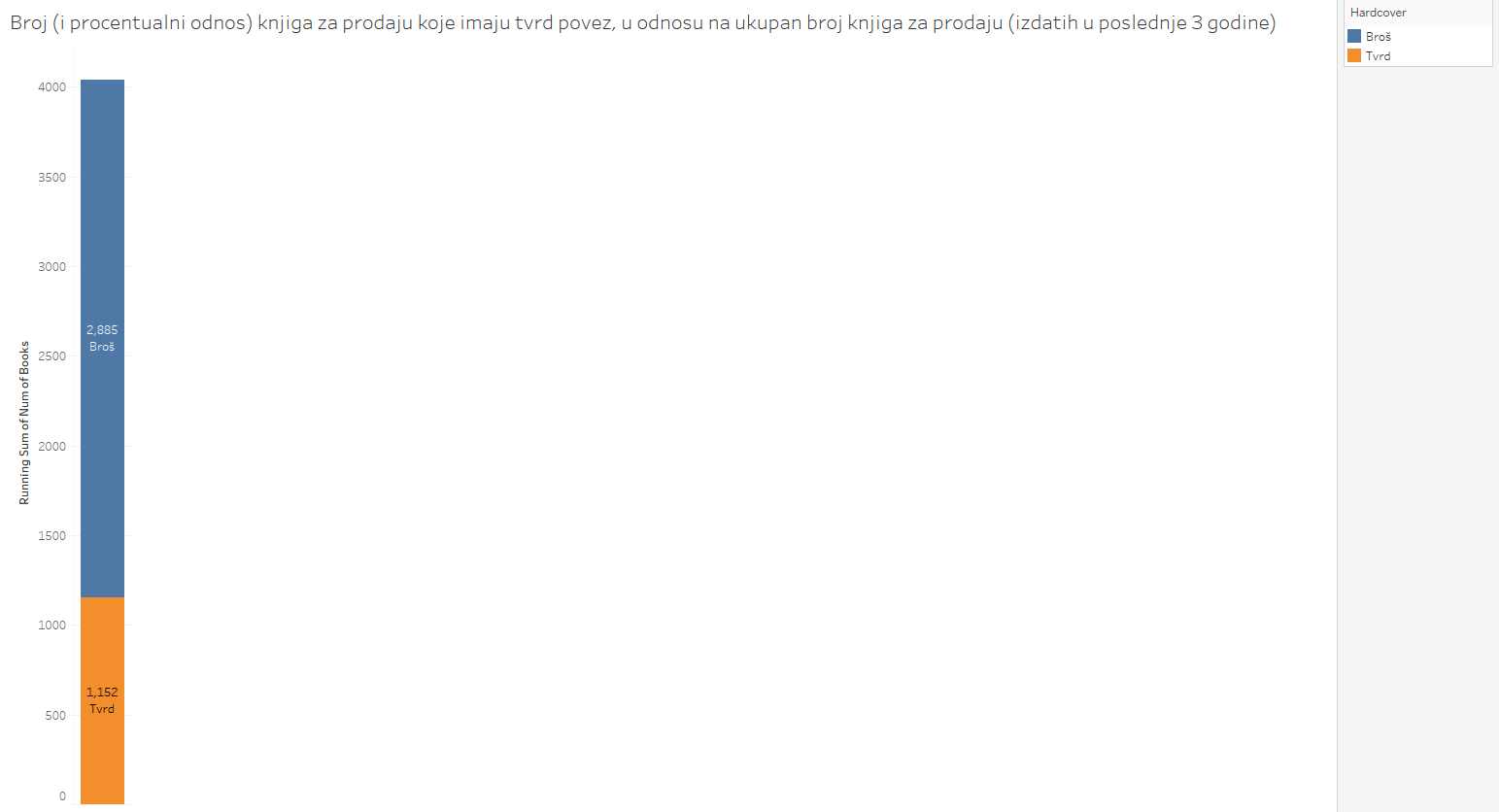
### е) број (и процентуални однос) свих књига за продају, које по цени припадају једном од следећих опсега (<= 500, 501-1500, 1501-3000, 3001-5000, 5001-10000, 10001-15000, > 15000)

Излаз:



### ф) број (и процентуални однос) књига за продају које имају тврд повез, у односу на укупан број књига за продају (издатих у последње 3 године)

Излаз:

# Задатак 4: Имплементација регресије

За реализацију овог дела пројекта коришћене су библиотеке *Pandas, Numpy*, *Customtkinter*, *Tkinter*, *Matplotlib* и *Threading.* Решење овог дела налази се у фајловима „*main\_linear.py*“, „*scaler.py*“ и „*lin.py*“ који се налазе у прилогу.

Апликација има једноставан графички интерфејс у којем корисник уноси податке књиге и као излаз добије њену предиктивну вредност.

Улазни подаци који се користе за предикцију су повез, главна категорија, подкатегорија, број страница и формат. Вредности су имплементацијом *scaler* класе скалиране. Модел даје добре резултате за јефтиније књиге док за скупље не толико због мањка података. На цену књиге значајно утиче повез, док подкатегорија и не толико. Такође, исробавањем више комбинација брзине учења и итерација (брзина учења од 0,1 до 0,0001, итерације од 500 до 50000), оптимална решења су добијена за вредности 0,01 и 10000.

На примеру књиге „ЗАГОР Шума укрштених судбина“ са ценом 3240,00 РСД добије се следећа предикција. Са друге стране, може да се види колико тврд повез утиче на саму цену књиге.

A screenshot of a computer

Description automatically generated A screenshot of a computer

Description automatically generated

# Задатак 5: Имплементација класификације

За реализацију овог дела пројекта коришћене су библиотеке *Pandas, Numpy*, *Customtkinter*, *Tkinter*, *Matplotlib* и *Threading.* Решење овог дела налази се у фајловима „*main\_log.py*“, „*scaler.py*“, „*log\_multinom.py*“ и „*log\_bin.py*“ који се налазе у прилогу.

Апликација опет има једноставан графички интерфејс у којем корисник уноси податке књиге, бира врсту класификације и као излаз добије њен опсег вредности.

Улазни подаци који се користе за предикцију су повез су као у прошлом делу. Вредности су имплементацијом *scaler* класе скалиране. Модел као и у прошлом делу даје добре резултате за јефтиније књиге док за скупље не толико. На цену књиге значајно утиче повез, док подкатегорија и не толико. Такође, исробавањем више комбинација брзине учења и итерација (брзина учења од 0,1 до 0,0001, итерације од 500 до 50000), оптимална решења су добијена за вредности 0,01 и 1000 за „један насупрот једног“, и 0.1 и 3000 за мултиномијалну логистичку регресију. Разлике у предикцији између ова два модела могу се видети на следећем примеру.

A screenshot of a computer

Description automatically generated A screenshot of a computer

Description automatically generated

Такође се и овде види приметна разлика између тврдог и брош повеза.

# Задатак 6: Имплементација кластеризације

За реализацију овог дела пројекта коришћене су библиотеке *Pandas, Numpy*, *Customtkinter*, *Tkinter*, *IPython* и *Sklearn.* Решење овог дела налази се у фајлу „*main\_kmeans.ipynb*“ који се налази у прилогу.

Улазни подаци који се користе за кластеризацију су повез, формат, број страница и цена. Све вредности су скалиране у опсегу од 1 до 5. За рачунање растојања коришћена је Еуклидска, Менхетн и Чебишијева дистанца. Формула која је најбрже конвергирала у већини случајева је била Еуклидска. Број кластера је променљив, док најбоље резултате показује на 3 или 4. Уколико за 100 итерација не конвергира, завршиће рад. Углавном је потребно око 20 итерација до конвергенције.

A screen shot of a graph

Description automatically generated A screenshot of a calculator

Description automatically generated

На слици се види пример кластеризације за к=4 кластера. Љубичаста је 0. кластер, плава 1., зелена 2. и жута 3. Оно што можемо да видимо из табеле је да 0. и 3. кластер имају тврди повез, док 1. и 2. имају брош. Разлика између кластера са истим типом повеза јесте, као што се и види у табели, цена и величина књиге.