

Univerzitet u Sarajevu
Elektrotehnički fakultet
Drugi ciklus studija
Odsjek za računarstvo I informatiku

SEMINARSKI RAD

Sistemi za podršku odlučivanju

Studenti:
Imamović Nasiha
Novalić Neira
Pirija Lejla
Rovčanin Nejra
Rudalija Ema
Salihović Mirnesa

Sarajevo, april 2023.

Sadržaj

1. Definicija problema	3
1.1.a Opis odluke i procesa donošenja odluke koji je potrebno podržati budućim sistemom. Identifikacija donosioca odluke	3
1.1.b Grafički prikaz procesa odlučivanja	3
1.2. Opis najviše korištene metode za rješavanje datog problema	4
2. Analiza literature	11
2.1. House Price Prediction: Hedonic Price Model vs Artificial Neural Network (2004)	11
2.2. House Price Prediction Using Machine Learning And Neural Networks (2018)	14
2.3. Prediction of House Pricing Using Machine Learning with Python (2020)	16
2.4. House Pricing Prediction (2019)	17
2.5. Neural Network Hyperparameter Optimization For Prediction of Real Estate Prices in Helsinki (2021)	20
3. Skup podataka	24
3.1. Opis skupa podataka	24
3.2. Obrada nedostajućih podataka	26
3.3. Korelacija između atributa	26
3.4. Vizualizacija skupa podataka	27
4. Struktura sistema i faze procesa odlučivanja	27
4.1. Prilagoditi "Opštu strukturu sistema za podršku odlučivanju"	27
4.2. Objasniti kojem tipu problema pripada problem	31
5. Preprocesiranje podataka	32
5.1. Dodatna analiza skupa podataka	32
5.2. Učitavanje ključnih biblioteka	32
5.3. Učitavanje skupa podataka	33
5.4. Profiliranje podataka	33
5.5. Čišćenje podataka	47
5.6. Transformacija podataka	50
5.7. Smanjenje podataka	51
5.8. Razdvajanje skupa podataka na skup za treniranje, testiranje i validaciju	51
6. Reference	53

1. Definicija problema

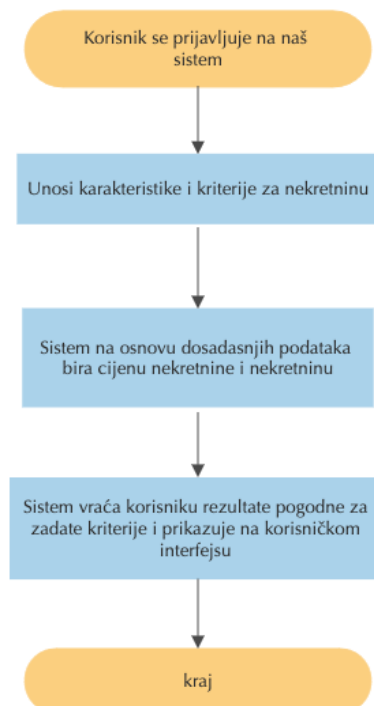
1.1.a Opis odluke i procesa donošenja odluke koji je potrebno podržati budućim sistemom. Identifikacija donosioca odluke

Budućim sistemom podržat ćemo proces donošenja odluke za predikciju cijene nekretnine, poznavajući neke od njenih osobina, kao jedine dostupne podatke. Dakle, korisnik unosi sve podatke o nekretnini, kao što je pozicija, broj prozora, ima li garažu, i slično, te na osnovu unesenih podataka dobiva estimaciju cijene nekretnine.

Odluku donosi naš sistem, na osnovu datih informacija, i modela kojeg smo istrenirali, a korisnici našeg sistema je dosta osoba naših godina, koji trenutno počinju razmišljati o svom stambenom pitanju, tačnije o kupovini nekretnine. Želimo napraviti nešto što može pomoći nama i drugim osobama koje su se odlučile za kupovinu stana ili kuće na prostoru Bosne i Hercegovine.

1.1.b Grafički prikaz procesa odlučivanja

Na slici ispod se nalazi dijagram toka procesa za sistem podrške odlučivanju cijene za kupovinu nekretnine.



Slika 1 – Proces odlučivanja

Najprije se vrši prijava korisnika na naš sistem. Zatim, korisnik unosi svoje kriterije za nekretninu, uključujući budžet, veličinu, lokaciju i druge specifikacije. Nakon unesenih specifikacija, sistem vrši procjenu koristeći znanje stečeno kroz treniranje modela, te vrši procjenu cijene i nekretnine koja je idealna za kriterije unesene od korisnika. Te za kraj, sistem prikazuje korisniku pregled nekretnina koje odgovaraju njegovim specifikacijama na korisnički interfejs.

Eventualni problemi kreću već od korisničkog unosa karakteristika i kriterija za nekretninu koju želi da pretraži i za koju želi znati cijenu. Moguće je da neće navesti dovoljno mnogo ili dovoljno jasne kriterije, te da odluka bude netačna ili nedeterministična. Drugi problem već nalazimo od samog procesa odlučivanja našeg sistema - zahtjevano je napraviti sistem otporan na loše unose, sistem koji je brz i efikasan, kao i to da ostane aktuelan sa vanjskim promjenama, npr inflacija kod cijene kuće. I za kraj, problem kod vraćanja rješenja korisniku, kako ih napraviti razumnim i kako sažeti od čitave nekretnine, informaciju koja zaista interesuje korisika.

1.2. Opis najviše korištene metode za rješavanje datog problema

Analizirajući korištene modele u radovima koji će biti predstavljeni u nastavku, zaključili smo da su se najviše pojavljivali modeli linearne regresije, random forest i neuralne mreže, te smo odlučili da isprobamo sva tri na našem datasetu. Finalni rezultati našeg rada su izuzetno dobri i sa svim modelima postignuta je tačnost od preko **99%**.

Linearna regresija

Kreirali smo model linearne regresije kao što je prikazano u kodu ispod.

```
# importing module
from sklearn.linear_model import LinearRegression
# creating an object of LinearRegression class
LR = LinearRegression()
#fitting the training data
LR.fit(x_train1,y_train1)
y_prediction = LR.predict(x_test1)
```

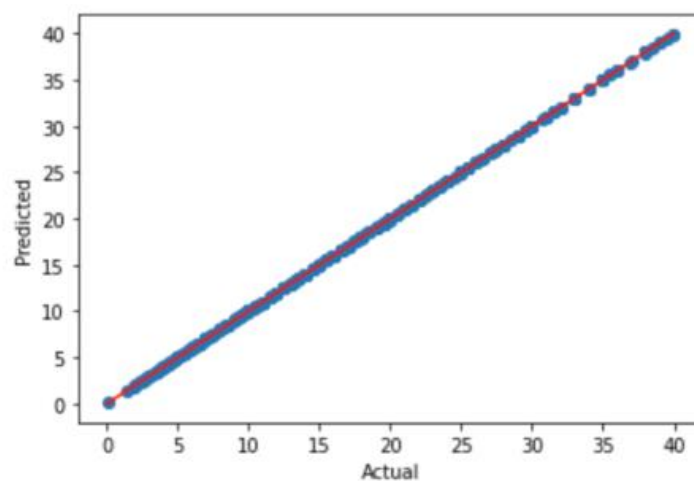
Ovaj model linearne regresije ostvario je R2 score 1. Predviđene vrijednosti odgovaraju stvarnim vrijednostima u potpunosti pri čemu smo zaključile da su podaci veoma dobro pripremljeni i obrađeni. Rezultati linearne regresije su prikazani na slici ispod:

```
r2 score: 1.0
MSE: 6.089408244066005e-28
RMSE: 2.4676726371352432e-14
```

Slika 2 – Tačnost modela

```
plt.scatter(y_test1,y_prediction)
plt.plot([0, 40], [0, 40], color = 'red')
```

```
plt.xlabel('Actual')
plt.ylabel('Predicted')
```



Slika 3 – Predviđene i tačne vrijednosti (linearna regresija)

```
pred_df=pd.DataFrame({'Actual Value':y_test1,'Predicted Value':y_prediction})
pred_df
```

	Actual Value	Predicted Value
2534	9.6	9.6
1283	13.0	13.0
616	39.0	39.0
2862	2.5	2.5
2730	30.0	30.0
...
488	12.5	12.5
1009	5.9	5.9
2188	15.0	15.0
1999	20.0	20.0
502	38.0	38.0

625 rows x 2 columns

Slika 4 – Predviđene i tačne vrijednosti (linearna regresija)

Random Forest

Nakon uspješno obavljene linearne regresije odlučile smo se i na primjenu Random Forest algoritma. Model je prikazan u kodu ispod:

```
rfr = RandomForestRegressor()

RandomForestRegressor(bootstrap=True, criterion='mse', max_depth=None,
max_features='auto', max_leaf_nodes=None, min_impurity_decrease=0.0,
min_samples_leaf=1, min_samples_split=2, min_weight_fraction_leaf=0.0,
n_estimators=10, n_jobs=1, oob)
```

Ovim modelom je postignuta tačnost je 99.99% što vidimo na slici ispod.

```
rfr.fit(x_train, y_train)

score = rfr.score(x_train, y_train)
print("Accuracy:", score)
```

Accuracy: 0.9999971525525557

Slika 5 – Tačnost za Random Forest

```
ypred = rfr.predict(x_test)
mse = mean_squared_error(y_test, ypred)
print("MSE: ", mse)
print("RMSE: ", np.sqrt(mse))
```

MSE: 0.001161047541835255
RMSE: 0.0340741477051922

Slika 6 – MSE i RMSE greške za Random Forest

Nekoliko predviđenih i stvarnih vrijednosti su prikazane u tabeli ispod.

```
y_test_np = y_test.to_numpy()

from tabulate import tabulate

headers = ["Sold price", "Predicted price"]
```

```

m = np.column_stack((y_test_np, ypred))

# tabulate data
table = tabulate(m, headers, tablefmt="fancy_grid")

# output
print(table)

```

Sold price	Predicted price
9.6	9.6325
13	13
39	38.999
2.5	2.502
30	30
36.9	36.986
20	20
30	30
10.5	10.5

Slika 7 – Predviđene i tačne vrijednosti (Random Forest)

Neuronska mreža

Nakon uspješno obavljene linearne regresije i primjene Random Forest algoritma, pristupile smo kreiranju neuronske mreže. Pri eksperimentisanju sa atributima definisanim u data setu zaključile smo da kolone „Bazen“, „Alarm“, „Okućnica“, „Podrum/Tavan“ nepovoljno utiču na tačnost te smo na samom početku odlučile ukloniti te podatke. U ovom dijelu smo još djelomično odradile obradu podataka pri čemu smo pored uklanjanja ovih kolona, izvršile normalizaciju svih podataka u data set-u uz pomoć StandardScaler-a.

```

from sklearn.preprocessing import StandardScaler

sc_X = StandardScaler()
X = sc_X.fit_transform(X)

sc_Y = StandardScaler()
y = sc_Y.fit_transform(y.values.reshape(-1, 1))

```

Nakon normalizacije podataka, kreirale smo model neuronske mreže koji je prikazan u kodu ispod.

```

import scikeras.wrappers
import numpy as np
from scikeras.wrappers import KerasRegressor
from sklearn.model_selection import cross_val_score
from sklearn.preprocessing import StandardScaler
from keras.models import Sequential
from keras.layers import Dense
from keras import models
from keras import layers
from tensorflow import keras
from keras.layers import Dense, Dropout, Flatten
import tensorflow as tf

from keras.models import Sequential
from keras.optimizers import RMSprop, Adam
from keras.callbacks import ReduceLROnPlateau

def baseline_model():
    # create model
    model = Sequential()

    # The Input Layer :
    model.add(Dense(64, kernel_initializer='normal', input_dim = train.shape[1],
activation='relu'))

    # The Hidden Layers :
    model.add(Dense(32, activation='relu'))
    model.add(Dense(16, activation='relu'))
    model.add(Dense(8, activation='relu'))

    # The Output Layer :
    model.add(Dense(1, kernel_initializer='normal', activation='linear',
kernel_regularizer = tf.keras.regularizers.l1(l=0.01) ))
    # Compile model
    model.compile(loss='mse', optimizer= keras.optimizers.Adam(lr=0.01)) # mse
    loss
    return model

```

Naš model se sastoji od jednog input sloja, tri skrivena sloja i jednog izlaznog sloja. Korištene aktivacijske funkcije su relu i linear. Stopa učenja je 0.01 a definisana funkcija gubitka je mse. Trening i testni skup podataka su podijeljeni u omjeru 80-20 pri čemu je 10% izdvojeno i za validacijski skup. Za fitting je korišten KerasRegressor.

```
#build model
```



```

seed = 7
np.random.seed(seed)

estimator = KerasRegressor(build_fn=baseline_model, epochs=120, batch_size=64,
verbose=2, validation_split=0.1) # KerasRegressor for regression problem

#cross validation
train, test, train_label, test_label = train_test_split(X,y,test_size=0.2,
random_state = seed)

estimator.fit(train, train_label)

```

Nakon treniranja neuronske mreže dobivene metrike izvršena je predikcija cijena na testnom skupu podataka i dobivene metrike su:

```

d = test_label - prediction
mse_f = np.mean(d**2)
mae_f = np.mean(abs(d))
rmse_f = np.sqrt(mse_f)
r2_f = 1-(sum(d**2)/sum((y-np.mean(y))**2))

print("Results:")
print("MAE:", mae_f)
print("MSE:", mse_f)
print("RMSE:", rmse_f)
print("R-Squared:", r2_f)

```

```

Results:
MAE: 0.02439923686374222
MSE: 0.001036987342578159
RMSE: 0.03220228784695521
R-Squared: [0.99979234]

```

Slika 8 – MAE, MSE i RMSE greške za neuronske mreže

Rezultati koji smo dobili su izuzetno dobri. Ukoliko izvršimo prikaz predviđenih i stvarnih vrijednosti vidimo da naša mreže izuzetno dobro vrši predikcije cijena kuće na osnovu deinisanih ulaznih parametara.

```

from tabulate import tabulate

```

```

headers = ["Sold price", "Predicted price"]

m = np.column_stack((sc_Y.inverse_transform(test_label),
sc_Y.inverse_transform(prediction)))

# tabulate data
table = tabulate(m, headers, tablefmt="fancy_grid")

# output
print(table)

```

Sold price	Predicted price
177	178.602
126	126.862
273	267.478
273	273.131
304	304.486
141	141.256
188	187.869
74	75.3413
304	303.309

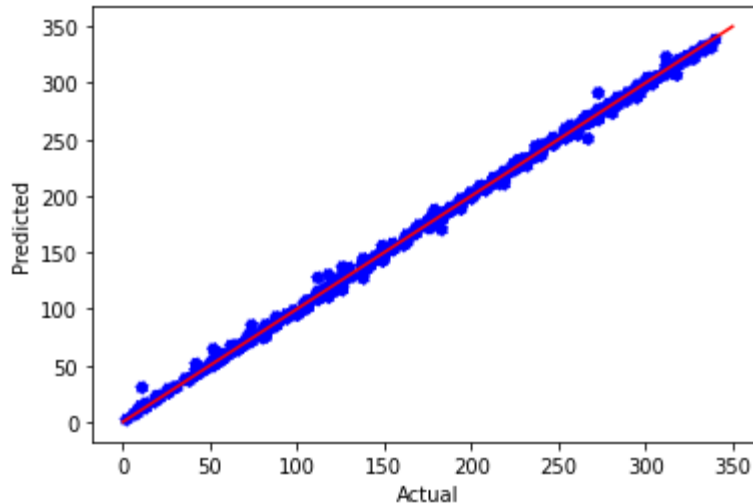
Slika 9 – Predviđene i tačne vrijednosti (neuronske mreže)

Predviđene i stvarne cijene smo prikazali i na grafiku ispod.

```

plt.scatter(sc_Y.inverse_transform(test_label),
sc_Y.inverse_transform(prediction), color = 'blue', ls='dotted')
plt.plot([0, 350], [0, 350], color = 'red')
plt.xlabel('Actual')
plt.ylabel('Predicted')

```



Slika 10 – Predviđene i tačne vrijednosti (neuronske mreže)

2. Analiza literature

2.1. House Price Prediction: Hedonic Price Model vs Artificial Neural Network (2004) ^[1]

Glavni cilj ovog rada je komparacija moći predikcije cijene kuća hedonic modela i neuralnih mreža. Za prikupljanje podataka za dataset kao referentna država uzet je Novi Zeland. Autori rada su smatrali da je ova država veoma zanimljiva za istraživanje jer oko 70% njenih stanovnika posjeduje svoju nekretninu. Korišteni data set se sastoji od podatka za 200 nekretnina

Faktori koji su uzeti u razmatranje su:

- Veličina kuće
- Starost kuće
- Vrsta kuće
- Broj spavaćih soba
- Broj kupaonica
- Broj garaža
- Geografska pozicija
- Pogodnosti u neposrednoj blizini kuće

Rezultati ovog istraživanja pokazuju veliki potencijal neuronskih mreža u oblasti predikcije cijena kuća iako je u nekim drugim radovima istaknuta njihova black-box priroda.

U radu se ističe važnost precizne predikcije cijena nekretnina i benefita koje bi različite uloge u procesu kupoprodaje ostvarile. Tradicionalni model predikcije cijena se većinom zasniva na međusobnom poređenju nekretnina a ne na različitim faktorima koji kreiraju cijenu. Iz navedenog razloga analiza ovog problema se gleda kroz prizmu dva različita modela. Hedonic model ima pretpostavke da se objekti poput kuća mogu gledati kao agregacija individualnih komponenti koja ih čini. Iako je ovaj model široko poznat i korišten, ima puno problema poput specifikacija za model procedure, multikolinearnost, nezavisnu interakciju ulaznih parametara, nelinearnost i veliki broj outlier-a zbog čega se značajno narušavaju performanse kod procjene stvarnih cijena nekretnina. Sa druge strane neuralne mreže su dale soluciju na većinu ovih problema a naročito ako podatkovni uzorci pokazuju nelinearnost. Ovu informaciju potvrđuje i istraživanje rađeno sa podacima iz Singapura gdje su neuralne mreže dale znatno bolje rezultate od višestrukog modela regresije za estimaciju vrijednosti. Neuralne mreže su u suštini prvobitno i dizajnirane da budu replikacija ljudskog mozga te svojom strukturom i načinom obrade podataka imaju mogućnost učenja određenih paterna i izuzetaka koji se javljaju u prodajnom sektoru.

Pri kreiranju Hedonic modela cijena je definisana kao funkcija i prikazana je na narednoj slici:

$$\text{PRICE} = f(\text{LAND, AGE, TYPE, BEDROOMS, BATHROOMS, GARAGES, AMENITIES, INNER CHRISTCHURCH, NORTH CHRISTCHURCH, SOUTH CHRISTCHURCH, EAST CHRISTCHURCH, WEST CHRISTCHURCH, NORTHWEST CHRISTCHURCH, } \epsilon)$$

Slika 11 – Funkcija za predikciju cijene kuće ^[1]

Varijable u modelu su definisane kao:

PRICE =	Price of house in Christchurch in NZD
LAND (+) =	Land size (in square meters)
AGE (-) =	Age of the house (in years)
TYPE (+) =	Type of house; 1 if the house has a garden, 0 otherwise
BEDROOMS (+) =	Number of bedrooms
BATHROOMS (+) =	Number of bathrooms
GARAGES (+) =	Number of garages
AMENITIES (+) =	Amenities around the house; 1 if the house is close to two or more public facilities (i.e. bus stop, school, public park and so on), 0 otherwise
ϵ =	Error term

Slika 12 – Varijable u Hedonic modelu ^[1]

Priori hipoteze su indicirane sa + i - u navedenoj specifikaciji. Prema literaturi koju su istraživali autori, varijable cijena, zemlja, tip, broj spavaćih soba, broj kupatila, broj garaža i pogodnosti oko kuće (blizina škole, parka, javnog prevoza) imaju pozitivnu korelaciju sa cijenom kuće.

Sa druge strane za neuralne mreže kreiran je dataset sa sličnim atributima te je podijeljen u omjeru 80-20 na trening i testni skup.

Krierana su tri Hedonic modela i postignuti rezultati ukazuju na činjenicu da cijena kuće značajno zavisi od tipa kuće (da li ima vrt ili ne) te da je cijena kuće koja ima vrt veća bez obzira na lokaciju na kojoj se nalazi.

Naredna agregacija modela ukazuje na činjenicu da veličina zemlje i broj garaža respektivno su najznačajniji faktori za određivanje cijene kuće sa vrtom dok pogodnosti u okolini kuće nisu pokazale veliku korelaciju sa cijenom.

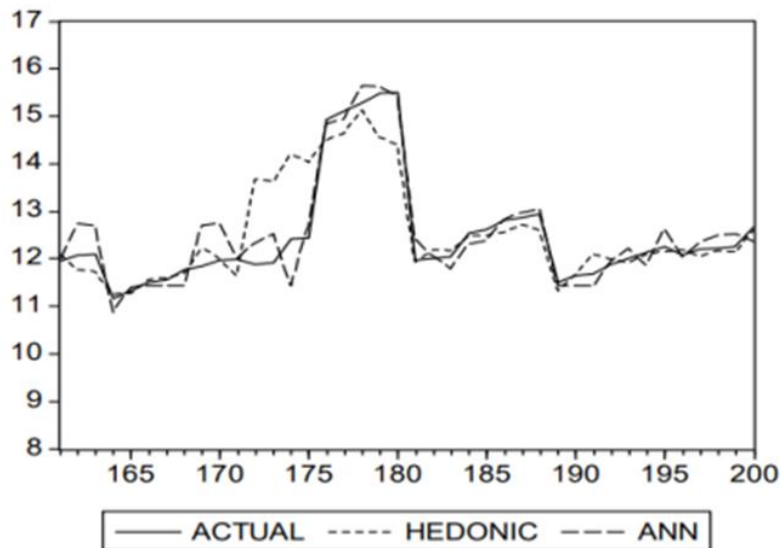
Za kuće koje nemaju vrt pokazalo se da atributi godina izgradnje kuće i broj garaža imaju veliki uticaj na cijenu kuće dok veličina zemlje za kuće bez bašte je manje važan faktor. Također faktor koji utiče na određivanje cijene kuće bez bašte je i broj soba.

Za neuralne mreže pokazalo se da najznačajniji faktor za određivanje cijene kuće je godina izgradnje kuće te broj garaža- respektivno. Također geografska lokacija kuće ima jaku korelaciju sa cijenom kuće u odnosu na ostale parametre poput veličine zemljišta, broj spavaćih soba, vrste kuće i ostaloga.

Poređenjem rezultata zaključeno je da model neuralnih mreža može bolje procijeniti cijenu kuće od hedonic modela ali nije pokazana značajna superiornost jednog modela nad drugim u pogledu mogućnosti predikcije.

	Model 1	Model 2	Model 3
Hedonic price model			
- R ²	0.6192	0.7499	0.3807
- RMSE	876,215.63	642,580.05	1,435,810.81
Neural network model			
- R ²	0.9000	0.8408	0.6907
- RMSE	449,111.46	512,614.99	1,014,721.92
	n = 40	n = 31	n = 9
Note: Model 1: house with and without garden. Model 2: house with garden. Model 3: house without garden.			

Slika 13 – Poređenje Hedonic modela i modela neuralnih mreža^[1]



Slika 14 – Rezultati stvarne i predviđene vrijednosti ^[1]

2.2. House Price Prediction Using Machine Learning And Neural Networks (2018) ^[2]

Tržište nekretnina je veoma konkurentna ali i netransparentna industrija te iz tog razloga rudarenje i obrada dostupnih podataka u cilju predviđanja cijena pomaže da se naprave povoljne odluke zasnovane na znanju.

Autori ovog rada stavljaju fokus na razvijanje modela koji predviđa cijenu nekretnina za kupca prema njegovim/njenim željama.

U cilju realizacije tog sistema ne primijenjuje se samo jedan algoritam već serija algoritama:

- Klasična linearna regresija
- Forest regresija
- Boosted regresija

Pored ovih tehnika neuralne mreže su korištene za povećanje tačnosti algoritma.

U osnovi ovog sistema se koristi data mining. Data set je obrađen ,svi outlier-i su uklonjeni i izvršena je prioritizacija parametara.

Istraživanje je dovelo do zaključka da cijena nekretnine značajno zavisi od pogodnosti koje se nalaze u njenom okruženju poput bolnica, škola,supermarketa,parkova itd.Iz tog razloga u

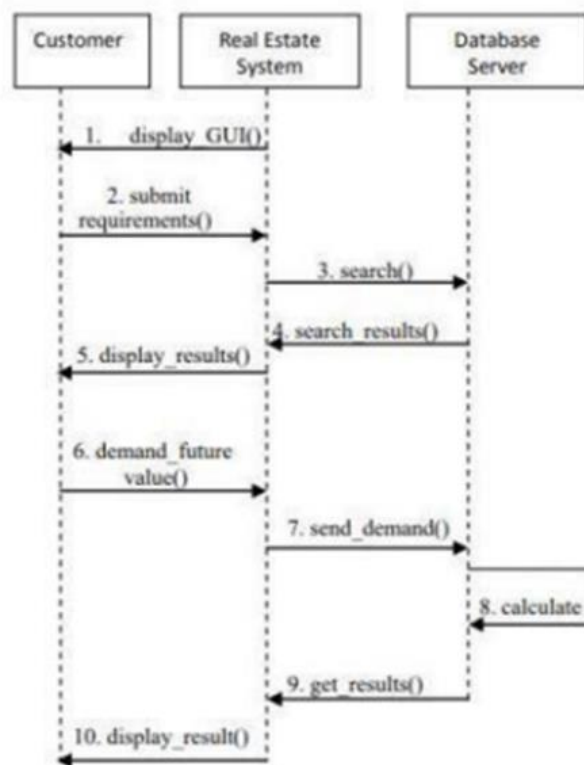
algoritmu se koristi Google maps API sa postavljenim lokalnim radijusom od 0.5 km te ukoliko se bilo koja od navedenih ustanova nađe u blizini cijena nekretnine se poveća u skladu s tim.

Neki od parametara koji se koriste u modelima su:

- Broj spavaćih soba
- Veličina zemljišta
- Dostupnost lifta
- Dostupnost parkinga
- Broj spratova

Autori nisu uključili rezultate kreiranog modela u rad ali su naveli da je ovim pristupom postignuta izuzetna tačnost. Kreirani sistem optimalno koristi Linearnu regresiju, Forest i Boosted regresiju. Također ističu da je sistem kreiran tako da je dodavanje novih parametara veoma jednostavno i da se tim neće narušiti glavna funkcionalnost sistema.

Rad sistema predstavljen je dijagramom sekvence na narednoj slici:



Slika 15 – Dijagram sekvence sistema za predikciju cijena kuća [2]

2.3. Prediction of House Pricing Using Machine Learning with Python (2020) [3]

U ovom radu je izvršena analiza predikcije cijena kuća korištenjem različitih metoda regresije i uz pomoć Python biblioteka.

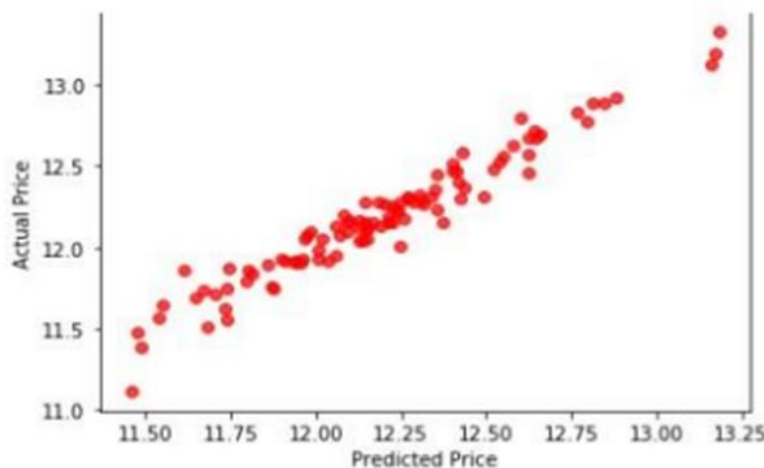
Istraživanjem autori ovog rada su ustanovili da postoji nekoliko komponenti koje utiču na troškove kuće i to:

- State of being - uključuje attribute koji su mjerljivi i vidljivi ljudskim okom poput kvadrature kuće, broj soba, dostupnost kuhinje i parking mjesta, starost kuće itd.
- State of thought - potencijal nekretnine u vidu mogućnost koja se pružaju agencijama/arhitektima da je urede i predstave u određenom svjetlu u svjetskom poretku
- State of territory - stanje teritorije na kojoj se nalazi nekretnina koja se očituje u mogućnosti ostvaranja vrijednih objekata u njenoj neposrednoj blizini (mogućnost izgradnja škole, mjesta za zaposlenje) ali i potencijal dijela za turizam

Nakon prikupljana podataka i kreiranja dataset-a pristupljeno je obradi podataka pri čemu je izvršeno čišćenje a zatim k-fold cross validacija. Da bi se postigli dobri i precizni rezultati, korištene su različite tehnike rudarenja podataka. Korišteni su regresijski algoritmi a pored njih i neki klasifikacijski algoritmi poput:

- SVM
- Drvo odlučivanja
- Random Forest

Postignuti rezultati su prikazani na narednoj slici:



Slika 16 – Predikcija cijena kuća korištenjem mašinskog učenja [3]

2.4. House Pricing Prediction (2019) ^[4]

U ovom radu korišten je Ames Housing dataset u svrhu predviđanja finalnih prodajnih cijena za kuće u datasetu. Korišteno je 79 opisnih varijabli, te one opisuju skoro svaku aspekt kuće.

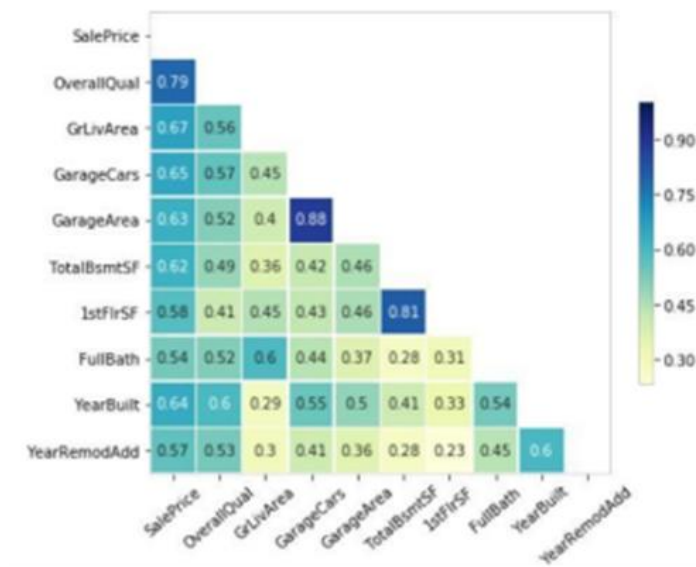
Metodologije:

Za prvi dio projekta u fokusu je bilo pretprocesiranje podataka, odabir značajki te predstavljanje istog pomoću korelacionih matrica, dijagrama raspršenosti, histograma itd.

Izvršeno je:

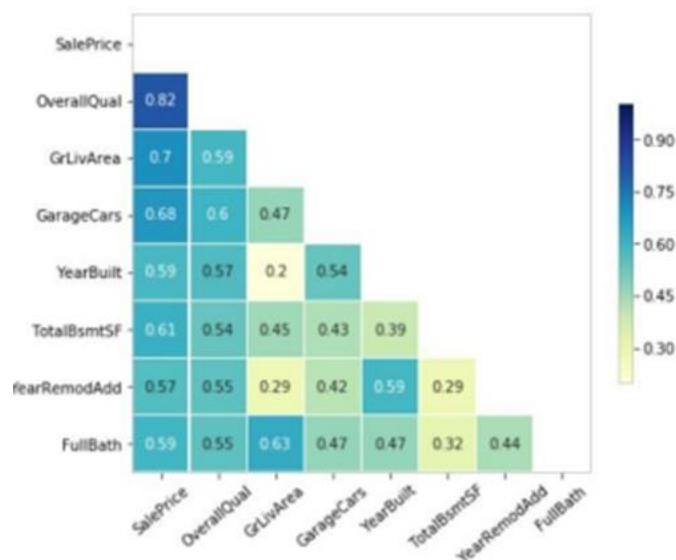
- Otkrivanje i rješavanje nedostajućih vrijednosti
- Outlieri
- Normalizacija distribucijskih varijabli
- Provjera regresijskih pretpostavki

Nakon uklanjanja outliera i normalizacije podataka dobivena je mapa kolinearnosti:



Slika 17 – Mapa kolinearnosti nakon normalizacije podataka ^[4]

Ovdje je odlučeno da se uklone varijable Garage Area i 1stFlrSF:



Slika 18 – Mapa kolinearnosti nakon uklanjanja varijabli Garage Area i 1stFlrSF^[4]

Multikolinearnost je znatno smanjena.

Za drugi dio projekta poređene su četiri vrste regresija: linearna regresija, Ridge regresija, Lasso regresija i random forest. Linearna regresija je osnovni model koji je korišten za prediktivnu analizu u cilju dobivanja prvog pregleda podataka u jednostavnom modelu. Ridge regresija je napredni regresijski model koji uzima kaznu ('penalty') svakog pokrenutog faktora prilikom kalkulacija. Tuningom hiperparametara pokazano je da performanse ovog modela nisu toliko dobre, pa je korišten Lasso regresijski model. Lasso model koristi kvadratnu sumu greške zbog njegove sposobnosti da izvrši odabir i regulaciju podskupa. Random forest model je robusan na greške i outliers pa neće doći do overfitting-a jer greška u forestu konvergira što je broj drveća u šumi veći.

Za treći dio konstruisana je 3-slojna NN (feedward network) i urađen je tuning hiperparametara u cilju smanjenja validacijske greške. Pokazano je da je ovakva neuronska mreža dosta robusna za predikciju cijena kuća.

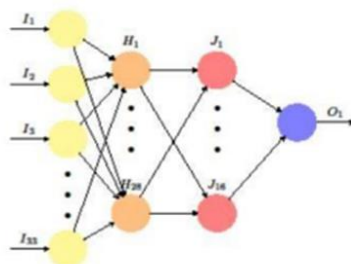
Autori su vjerovali da sve numeričke vrijednosti (osim id-a) sadrže informacije o kući, tako da su sve one uzete u obzir prilikom pravljenja training seta, te je izvršena MinMax normalizacija za svaku značajku. Nakon toga dobiven je dataset sa 33 ulaza u neuralne jedinice i 1 izlazna neuralna jedinica. Nakon tuninga hiperparametara, pokazano je da najbolje performanse ima model sa 28 neuralnih jedinica u prvoj skrivenoj jedinici i 16 neuralnih jedinica u drugoj neuralnoj jedinici. ReLU je korištena kao aktivacijska funkcija.

Formulacija NN:

$$\begin{aligned}h_1 &= \text{Relu}(W_1 \odot X_{in}) \\h_2 &= \text{Relu}(W_2 \odot h_1) \\\hat{y} &= \text{Relu}(W_3 \odot h_2)\end{aligned}$$

Slika 19 - Formulacija NN^[4]

Dijagram NN:



Slika 20 - Dijagram NN^[4]

Kao funkcija gubitka korištena je MSE. Na kraju su poređene performanse modela te je korišten onaj model sa najboljim performansama za finalnu predikciju.

Nakon poređenja modela:

model	Linear	Ridge	Lasso	RF	NN
MSE	0.0205	0.014	0.013	0.0289	2e-5

Slika 21- Poređenje modela^[4]

NeuralNet je bio najtačniji te je korišten za predikciju. Ovaj rad je pobijedio 50% timova na Kaggleleaderboard-u.

Unaprijeđenje:

Za random forest model, trebalo bi smanjiti dimenzionalnost dataseta koristeći PCA i druge metodologije.

Također bi se mogli uvesti drugi modeli kao što je npr SVM, za kojeg se pokazalo da dobro radi na datasetivima većih dimenzionalnosti.

2.5. Neural Network Hyperparameter Optimization For Prediciton of Real Estate Prices in Helsinki (2021) ^[5]

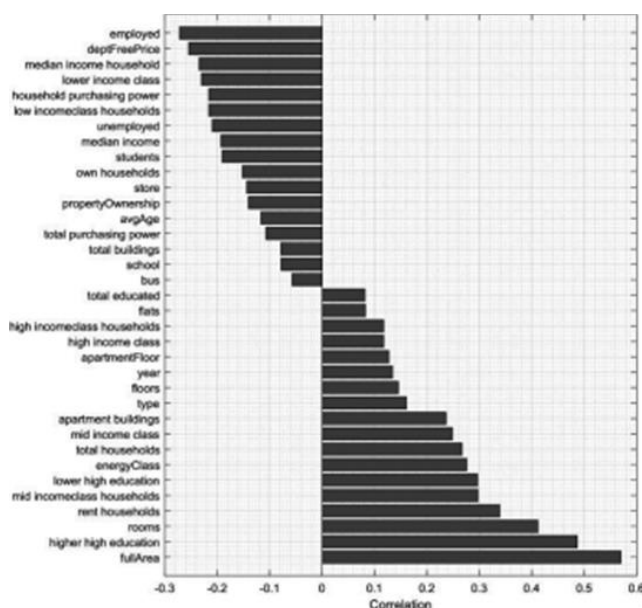
U ovom radu je prilagođen MLP (multilayer perceptron) neuronska mreža za predviđanje cijena nekretnina u Helsinkiju. Predstavljene su i metodologije za prilagođavanje hiperparametara u cilju poboljšanja performansi.

Metodologije:

Korišten je osnovni ANN model koji se bazira na ekspertnom znanju te je on bio osnov za optimizaciju ANN modela koji su poređeni i ocjenjivani. Zatim je primijenjena optimizacija hoperparametara na ANN modelu.

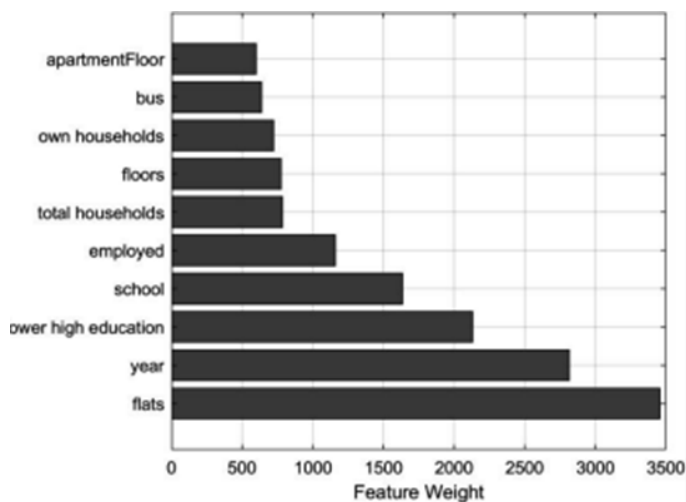
Dataset:

ANN zahtijeva da dataset bude pripremljen na idući način: 2 subseta koja će biti korištena za treniranje i testiranje modela. Dataset sadrži podatke o prodanim nekretninama u Helsinkiju u 2019. godini. Korišten je web-crawler u cilju pribavljanja podataka i crawlano je nekoliko različitih web stranica sa nekreninama. Prilikom analiziranja dataseta ustanovljeno je da je godina izgrdnje veoma bitan faktor koji uvodi nelinearnost u ovaj problem. Starogradnja može biti dosta skuplja u odnosu na novogradnju u istoj regiji. Veličina stana je također veoma bitan faktor. Outlieri su se najčešće pojavljivali u ovoj kategoriji. Pearce korelacija je korištena u cilju analize značajki dataseta.



Slika 22 – Analiza dataset-a ^[5]

Izvršena je također i analiza susjedstva u kojem se nalazi nekretnina. Na slici se može vidjeti koliko su ti faktori bitni:



Slika 23 – Feature Weight^[5]

Nad datasetom je izvršeno uklanjanje outliera.

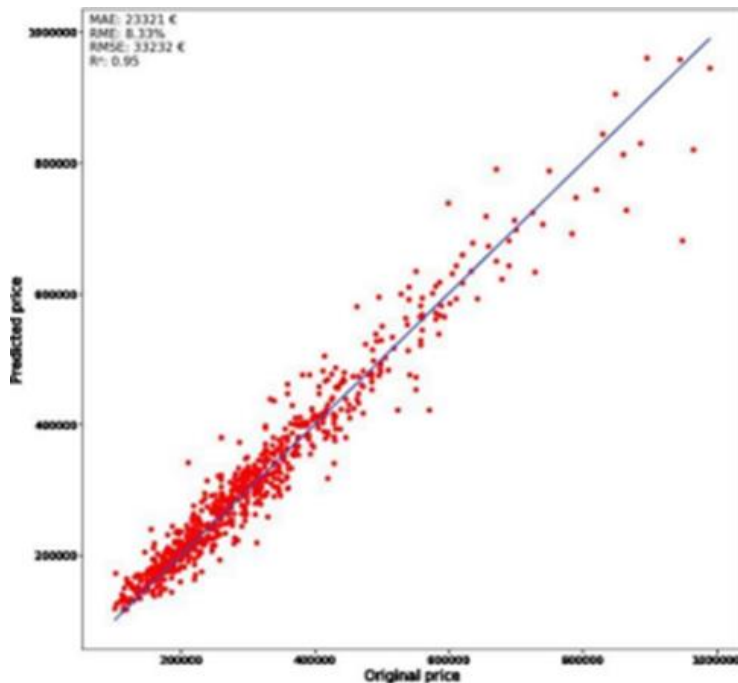
ANN:

Nakon 2003 pokretanja sa različitim hiperparametrima, pokazano je da je oprimaan model pronađen koristeći reLU aktivaijsku finsiku, Adam optimizacioni algoritam batch veličina 550, dropout learning rate 0,002, kao i validadacijski split 8%. Najbolja arhitektura modela je koriđtena sa jednim ulaznim slojem , 42 čvora, 6 skrivenih slojeva i jednim izlaznim čvorom. Zatim je urađeno poređenje osnovnog i opimizovanog modela sa 9 evaluacijskih metrika.

Model	Training results					Testing results			
	MSE	MSE Train	MSE Difference	Val loss	Loss	R ²	RMSE	RME	MAE
Baseline	0.0027	0.0023	0.0004	0.00899	0.0019	0.90	46,815.7	10.8%	30,968.6
Optimized	0.0018	0.0015	0.0003	0.00669	0.0011	0.95	33,232.2	8.3%	23,320.9
Difference	-0.0009	-0.0008	-0.0001	-0.0023	-0.008	0.05	-13,583.5	-2.5%	-7647.7
Improvement	33.3%	34.8%	25.0%	25.6%	42.1%	5.56%	29.0%	23.2%	24.7%

Slika 24 – Evaluacijske metrike^[5]

Na idućoj slici je predstavljena greška između predviđene i originalne cijene. Niže cijene su bolje predviđane u odnosu na veće. Metrke su pokazele dobre performanse na čitavom datasetu.



Slika 25 – Originalna vs Predviđena cijena

2.6. House Price Prediction Approach Based on Deep Learning and ARIMA Model (2019) ^[6]

Dizajn i implementacija:

Mnogo faktora utiču na cijenu kuće. U ovom radu odabrano je i obrazloženo 13 faktora koji su utiču na cijenu kuće. Ti faktori su: veličinu, lokaciju, orijentaciju, brojevi spavaćih soba, brojevi soba, brojevi od kupaonice, vrste podova, vrste dekoracije, dostupnost škola, dostupnost lifta, godina izgradnje, dostupnost metroa i metro udaljenost.

Podaci su preuzeti sa interneta, te je zatim izvršena obrada i modeliranje podataka. Numeričke vrijednosti koje nedostaju popunjene su sa srednom vrijednošću podataka. Izvršena je normalizacija numeričkih varijabli te su outlieri uklonjeni.

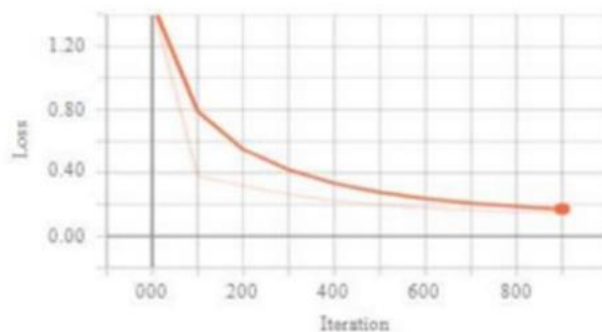
Korišten je model dubokog učenja za predviđanje cijena kuće. Model se sastoji od ulaznog data sloja, 4 skrivena sloja, te jednog izlaznog sloja. Za aktivacijsku funkciju odabrana je ReLU funkcija. Mreža je prvo trenirana, zatim je za testiranje tj. za tačnost korištena vijednost MSE (mean square error) koja se također uzima kao funkcija gubitka gdje je P_i je stvarna vijednost kuće, a \hat{P}_i je predviđena.

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (P_i - P'_i)^2,$$

Slika 26 – MSE formula ^[6]

U ovom radu je fokus na predviđanje trenda cijena kuća te je korišten ARIMA model (Autoregressive Integrated Moving Average) koji je poznat za predviđanje vremenskih serija.

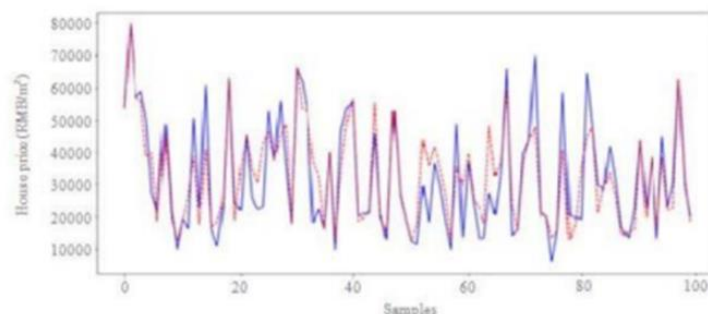
Prilikom treniranja learning rate je postavljen na 0,01 i keep_prob hiperparametar je postavljen na 0,9 radi smanjenja overfittinga.



Slika 27 – Trening i test gubitak ^[6]

Pokazano je da su se gubici tokom treninga i testiranja smanjivali sa povećanjem iteracija, te da su im krive slične.

Za evaluaciju performansi korišten je SVR (support vector regression) model koji je prilagođen eksperimentima poređenja.



Slika 28 – SVR model na trening dataset-u ^[6]

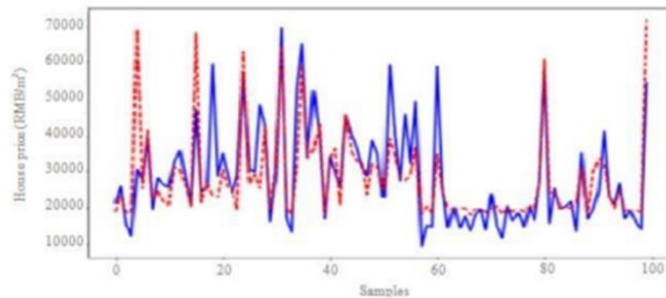


Fig. 5. The ground truth and predicted value of the SVR model on test dataset.

Slika 29 – SVR model na testnom dataset-u^[6]

Na slikama su prikazana poređenja između SVR modela i stvarne cijene u trening i testnom datasetu.

Root mean square error (RMSE) i mean relative error (MRE) su korišteni za evaluaciju performansi predikcije ova 2 modela.

Model	RMSE /Training	RMSE /Test	MRE /Training	MRE /Test
Proposed	5393	7671	0.19	0.22
SVR	9024	10216	0.20	0.28

Slika 30 – Evaluacija performansi predikcije^[6]

Pokazano je da predloženi model dubokog učenja ima bolje rezultate nego SVM model za predikciju cijena nekretnina u ovom slučaju, te je on korišten za dalju analizu u ovom radu.

3. Skup podataka

3.1. Opis skupa podataka

Mi smo odlučili kreirati naš dataset, koji će se zasnivati na podacima dostupnim na stranici OLX.ba za čitavu Bosnu i Hercegovinu, koristeći našu skriptu za crawling podataka koji su dostupni na OLX stranici (oko 3400 nekretnina za koje je navedena cijena) sa svim stavkama koje su navedene u oglasima (kvadratura, kanton, općina, broj soba, sprat, dostupnost lifta, stanje stana, vrsta grijanja, itd).

Dataset se sastoji od 4479 redova i 31 kolone.

Kolone koje su obuhvatile neke od najvažnijih osobina za procjenu cijene nekretnine jesu: cijena, lokacija, broj kvadrata, broj soba, broj spratova, vrsta grijanja, kvadratura okućnice, vrsta poda, vrsta kanalizacije, alarm, balkon, internet, kablovska TV, namještena, ostavu/špajz, parking, tavan/podrum, struja, telefonski priključak, uknjižena/ZK, video nadzor, voda, garaža, primarna orijentacija, blindirana vrata, klima, plin, godina izgradnje, nedavno adaptirana, iznajmljena i bazen.

Na slici ispod vidimo i pregled dataseta u Google Collab-u.

	price	Lokacija	Kvadrata	Broj soba	Broj spratova	\
0	450.000 KM	Ilidža	260	5	3	
1	159.000 KM	Tuzla	240	4	3	
2	195.000 KM	Hadžići	140	7	3	
3	199.000 KM	Grad Mostar	238	6	1	
4	350.000 KM	Novo Sarajevo	150	7	2	
...	
4474	499.000 KM	Novo Sarajevo	320	8+	2	
4475	290.000 KM	Banja Luka	212	4	2	
4476	65.000 KM	Sarajevo, Novi Grad	NaN	NaN	NaN	
4477	170.000 KM	Brčko	92	3	1	
4478	280.000 KM	Sarajevo, Novi Grad	300	8+	3	

	Vrsta grijanja	Okućnica (kvadratura)	Vrsta poda	Kanalizacija	\
0	Centralno (Kotlovnica)	200	Laminat	Standardna	
1	Plin	NaN	NaN	NaN	
2	Centralno (Kotlovnica)	NaN	NaN	NaN	
3	Drva	603	NaN	NaN	
4	Centralno (Plin)	400	Parket	Standardna	
...	
4474	Centralno (Plin)	840	Parket	NaN	
4475	Ostalo	NaN	NaN	NaN	
4476	NaN	NaN	NaN	NaN	
4477	Centralno (Kotlovnica)	575	Laminat	NaN	
4478	Centralno (Kotlovnica)	1000	Parket	Standardna	

	Alarm	...	Voda	Garaža	Primarna orjentacija	Blindirana vrata	Klima	Plin	\
0	1	...	1	0	NaN	0	0	0	
1	0	...	1	0	NaN	0	0	0	
2	0	...	0	0	NaN	0	0	0	
3	0	...	1	1	NaN	0	0	0	
4	1	...	1	1	Sjeveroistok	1	1	1	
...	
4474	0	...	1	1	NaN	0	1	1	
4475	0	...	1	0	NaN	0	0	0	
4476	0	...	0	0	NaN	0	0	0	
4477	0	...	1	0	NaN	0	0	0	
4478	0	...	1	1	Jug	1	0	0	

	Godina izgradnje	Nedavno adaptirana	Iznajmljeno	Bazen
0	NaN	0	0.0	0
1	NaN	0	0.0	0
2	NaN	0	0.0	0
3	NaN	0	0.0	0
4	NaN	0	0.0	0
...
4474	NaN	0	NaN	NaN
4475	NaN	0	NaN	NaN
4476	NaN	0	NaN	NaN
4477	NaN	0	NaN	NaN
4478	Novogradnja	1	NaN	NaN

[4479 rows x 31 columns]

Slika 31 – Pregled dataseta

3.2. Obrada nedostajućih podataka

Prilikom obrade podataka, odbacili smo svega nekoliko redova našeg dataseta, i dvije klone koje su nosile nepotrebne podatke za naš sistem, što znači da se nije odrazilo na naš model, njegovu tačnosti niti je uticalo na cjelokupni skup podataka. Isto važi i za zamjenu nedostajućih vrijednosti, jer ih nije bio velik broj.

3.3. Korelacija između atributa

Nakon obrade svih numeričkih i kategoričkih atributa izvršile smo enkodiranje labela i kreirale smo heat mapu na kojoj je jasno prikazana korelacija između atributa.

Enkodiranje labela prikazano je u sljedećem isječku:

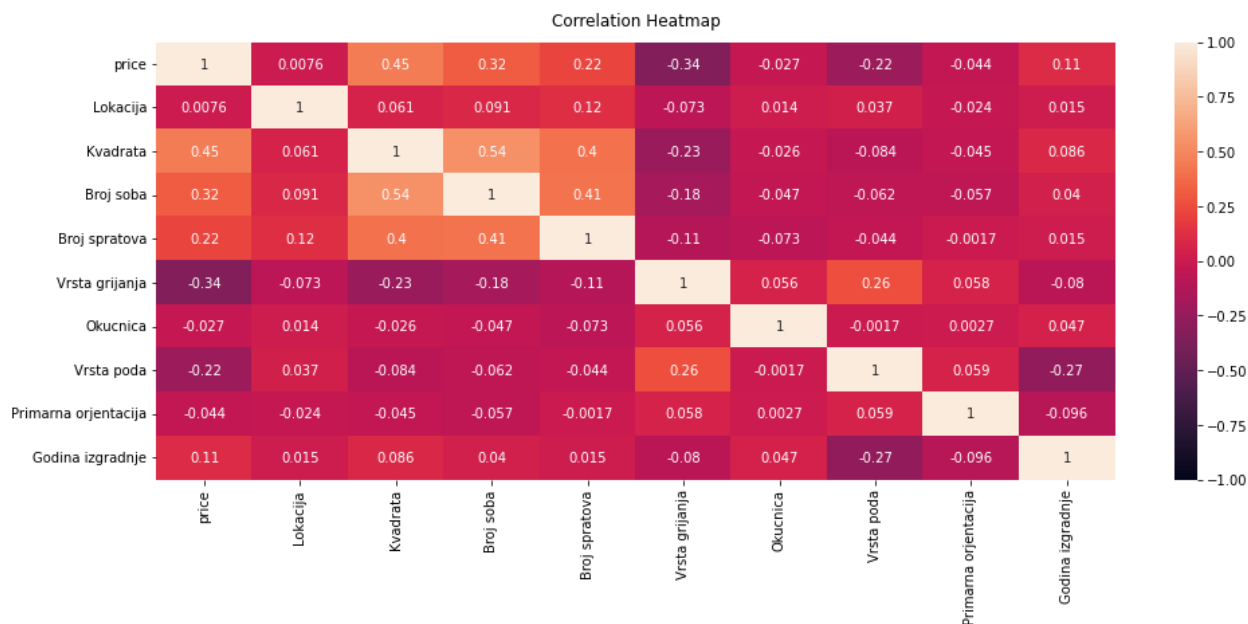
```
# Label encoding
from sklearn.preprocessing import LabelEncoder
from sklearn import preprocessing
label_encoder = preprocessing.LabelEncoder()

df['Vrsta poda'] = label_encoder.fit_transform(df['Vrsta poda'])
df['Vrsta grijanja'] = label_encoder.fit_transform(df['Vrsta grijanja'])
df['Primarna orijentacija'] = label_encoder.fit_transform(df['Primarna
orijentacija'])
df['Lokacija'] = label_encoder.fit_transform(df['Lokacija'])
df['Godina izgradnje'] = label_encoder.fit_transform(df['Godina izgradnje'])
```

Heat mapa korelacije varijabli je prikazana na sljedećoj slici:

```
import seaborn as sns
import matplotlib.pyplot as plt

plt.figure(figsize=(16, 6))
heatmap = sns.heatmap(df.corr(), vmin=-1, vmax=1, annot=True)
heatmap.set_title('Correlation Heatmap', fontdict={'fontsize':12}, pad=12)
```



Slika 32 - Prikaz korelacijske matrice

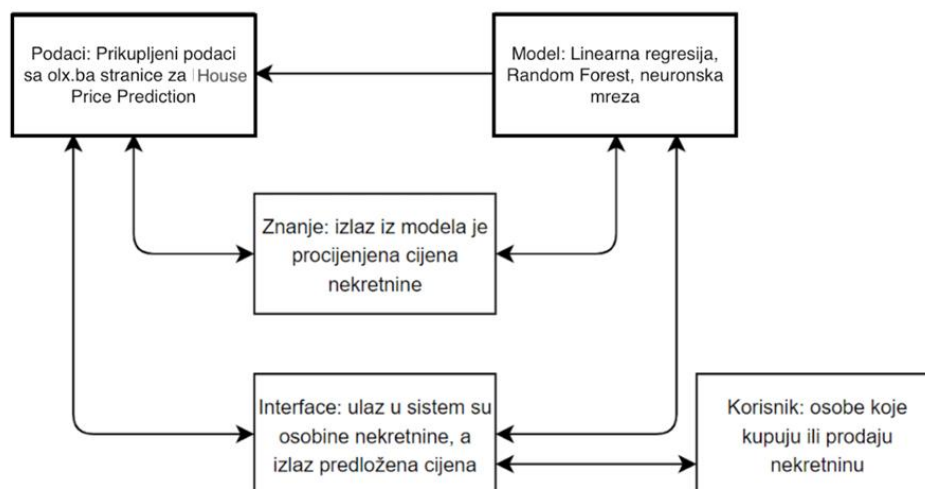
3.4. Vizualizacija skupa podataka

Vizualizacija skupa podataka je detaljno obrađena u poglavlju koje se odnosi na profiliranje podataka. U tom dijelu nalaze se boxplotovi i barplotovi za sve razmatrane attribute. Radi jednostavnosti i preglednosti poglavlja 5.4. odlučili smo da navedene prikaze prezentujemo u sklopu istog.

4. Struktura sistema i faze procesa odlučivanja

4.1. Prilagoditi “Opštu strukturu sistema za podršku odlučivanju”

Cilj projekta je da napravimo algoritam takav da pomaže pri odlučivanju cijene nekretnine. Naš dataset će se zasnivati na podacima dostupnim na stranici OLX.ba za čitavu Bosnu i Hercegovinu. Prvobitni cilj je crawlati sve podatke koji su dostupni na OLX-u (oko 3400 nekretnina za koje je navedena cijena) sa svim stavkama koje su navedene u oglasima (kvadratura, kanton, općina, broj soba, sprat, dostupnost lifta, stanje stana, vrsta grijanja itd..) te zatim trebamo analizirati dobiveni dataset i uraditi obradu podataka. Nakon toga cilj projekta je napraviti model takav da vrši oslucivanje cijene za datu nekretninu na osnovu dostupnih podataka.



Slika 33 – Dijagram strukture sistema za podršku odlučivanju

a) Korisnik / korisnici kojim pomažemo pri donošenju odluke

Korisnici kojima pomažemo pri donošenju odluke su osobe koje žele da stave neku nekretninu na prodaju tj. prodavaoci nekretnina , i osobe koje kupuju nekretnine kupci nekratnina.

b) Predikcione / deskriptivne modele koji se planiraju implementirati a koji su prepoznati putem analize relevantne literature (LV2)

Analizom postojećih radova došli smo do zaključka da bi za ovaj tip problema najbolje bilo implementirati model linearne regresije. Nakon uspješno obavljene linearne regresije odlučile smo se i na primjenu Random Forest algoritma. Te nakon uspješno obavljene linearne regresije i primjene Random Forest algoritma, planiramo pristupiti kreiranju neuronske mreže.

c) Podaci i njihov oblik

WEB SCRAPING

- Napravljena je Node.js skripta za web scraping podataka sa stranice OLX.ba
- Povučeni su svi dostupni podaci koji su korišteni udaljem radu
- Povučeno je nešto više od 4000 podataka sa 32 atributa

id	url	price	Lokacija	Kvadrata	Broj soba	Broj spratova	Vrsta grijanja	Okućnica (kvadratura)	Vrsta poda	Kanalizacija	Alarm	Balkon	Internet
art_50559511	https://www.olix.ba/objekt/450.000-KM	450.000 KM	Enfite	260	5	3	Centralno (Kuhovnici)	200	Laminat	Standardna	1	1	1
art_42776538	https://www.olix.ba/objekt/159.000-KM	159.000 KM	Tuzla	240	4	3	Plin	NA	NA	NA	0	1	1
art_48182478	https://www.olix.ba/objekt/195.000-KM	195.000 KM	Hadžići	140	7	3	Centralno (Kuhovnici)	NA	NA	NA	0	0	0
art_47387158	https://www.olix.ba/objekt/199.000-KM	199.000 KM	Grad Mostar	238	6	1	Drva	603	NA	NA	0	1	0
art_45110499	https://www.olix.ba/objekt/350.000-KM	350.000 KM	Novo Sarajevo	150	7	2	Centralno (Plin)	400	Parfet	Standardna	1	1	1
art_40763448	https://www.olix.ba/objekt/650.000-KM	650.000 KM	Star Grad	250	7	2	Centralno (Plin)	80	Parfet	Standardna	1	0	1
art_50387468	https://www.olix.ba/objekt/85.000-KM	85.000 KM	Zemica	73	5	3	Oslobo	2227	NA	NA	0	1	0
art_49090466	https://www.olix.ba/objekt/530.000-KM	530.000 KM	Star Grad	240	7	3	Centralno (Plin)	705	Laminat	Standardna	0	1	1
art_42753803	https://www.olix.ba/objekt/75.000-KM	75.000 KM	Banovci	178	5	2	Oslobo	NA	NA	NA	0	0	0
art_49016467	https://www.olix.ba/objekt/340.000-KM	340.000 KM	Banja Luka	168	8+	1	Drva	458	Oslobo	Standardna	0	0	1
art_50122267	https://www.olix.ba/objekt/105.000-KM	105.000 KM	Prijedor	140	5	1	Centralno (Kuhovnici)	6000	NA	NA	0	0	0
art_39823861	https://www.olix.ba/objekt/300.000-KM	300.000 KM	Tuzla	110	7	2	Centralno (Kuhovnici)	477	Laminat	Standardna	0	1	1
art_47273586	https://www.olix.ba/objekt/130.000-KM	130.000 KM	Zemica	120	3	2	Centralno (Plin)	62	Parfet	Standardna	0	1	1
art_50348031	https://www.olix.ba/objekt/79.000-KM	79.000 KM	Banja Luka	98	2	1	Drva	9083	NA	NA	0	0	0
art_40202220	https://www.olix.ba/objekt/891.000-KM	891.000 KM	Banja Luka	396	8+	4	Šinje	235	Oslobo	Standardna	1	1	1
art_49937479	https://www.olix.ba/objekt/385.000-KM	385.000 KM	Lukavci	385	8+	3	Drva	400	Laminat	Standardna	0	1	0
art_33943829	https://www.olix.ba/objekt/200.000-KM	200.000 KM	Brčko	75	2	2	Šinje	621	Parfet	Standardna	0	1	0
art_49765675	https://www.olix.ba/objekt/1.600.000-KM	1.600.000 KM	Novo Sarajevo	500	8+	4	Centralno (Plin)	700	Parfet	Standardna	1	1	1
art_49134429	https://www.olix.ba/objekt/210.000-KM	210.000 KM	Zemica	120	5	1	Centralno (Kuhovnici)	230	NA	NA	0	0	1
art_48558937	https://www.olix.ba/objekt/100.000-KM	100.000 KM	Novo Sarajevo	100	4	1	Plin	250	NA	NA	0	1	0
art_49259878	https://www.olix.ba/objekt/80.000-KM	80.000 KM	Lukavci	117	8+	2	Oslobo	521	Oslobo	Standardna	0	0	0
art_49216363	https://www.olix.ba/objekt/85.000-KM	85.000 KM	Brčko	89	4	1	Drva	511	Broškal	Standardna	0	1	0
art_48854587	https://www.olix.ba/objekt/200.000-KM	200.000 KM	Lukavci	1150	8+	3	Centralno (Kuhovnici)	1200	Laminat	Standardna	0	1	0
art_35577790	https://www.olix.ba/objekt/600.000-KM	600.000 KM	Brčko	171	4	1	Šinje	465	Laminat	NA	0	1	0
art_49303994	https://www.olix.ba/objekt/69.000-KM	69.000 KM	Doboj	120	4	1	Centralno (Kuhovnici)	NA	NA	NA	0	1	0
art_48704773	https://www.olix.ba/objekt/259.000-KM	259.000 KM	Hadžići	550	8+	3	Drva	1200	Broškal	Standardna	0	1	0
art_43296752	https://www.olix.ba/objekt/270.000-KM	270.000 KM	Tuzla	58	3	1	Drva	18231	NA	NA	0	0	0

Slika 34 - Podaci dataseta preuzetih sa olix-a

Inicijalni data set je bio sačinjen od 4478 instanci pri čemu su definisani ključni atributi po kojima će se vršiti analiza. U prvom koraku su uklonjene kolone "Iznajmljeno" i "url" s obzirom da one nisu relevantne ovom kontekstu te u nastavku rada se vrši obrada podataka u 30 preostalih kolona. Za svaku kolonu je izvršeno detaljno pretprocesiranje podataka pri čemu su uklonjeni svi neadekvatno uneseni podaci i popunjene su nedostajuće vrijednosti. Pored toga uklonjeni su i outlieri te je izvršeno balansiranje podataka

<ul style="list-style-type: none"> • Price (Numericka) • Lokacija (Kategoricka) • Broj kvadrata (Numericka) • Broj soba (Numericka) • Broj spratova (Numericka) • Vrsta grijanja (Kategoricka) • Okućnica (kvadratura) (Numericka) <ul style="list-style-type: none"> • Vrsta poda (Kategoricka) • Kanalizacija (Kategoricka) • Video nadzor (Numericka) <ul style="list-style-type: none"> • Alarm (Numericka) • Balkon (Numericka) • Internet (Numericka) • Kablovska (Numericka) • Namještena (Numericka) • Ostava/Špajz (Numericka) 	<ul style="list-style-type: none"> • Struja (Numericka) • Telefonski priključak (Numericka) • Uknjižena (Numericka) <ul style="list-style-type: none"> • Voda (Numericka) • Garaža (Numericka) • Adresa (Numericka) • Primarna orijentacija (Kategoricka) <ul style="list-style-type: none"> • Blindirana vrata (Numericka) • Klima (Numericka) • Plin (Numericka) • Godina izgradnje (Numericka) • Nedavno adapturana (Numericka) <ul style="list-style-type: none"> • Bazen (Numericka) • Parking (Numericka) • Podrum/Tavan (Numericka)
---	--

Slika 35

d) Analizirati koji izlazi iz sistema će posebno biti korisni korisniku sistema pri procesu donošenja odluke

Price - predstavlja numerčku varijabla intervala u kojem se nalazi price nekretnine.. Izlaz je jedinstven i ograničava preporučenu cijenu nekretnine na osnovu zahtjeva kupca iste. Očekivani izlaz modela je procijenjena cijena nekretnine. Za korisnika koji prodaje nekretnine daje preporuku za cijenu koju bi korisnik trebao tražiti, a za korisnika koji kupuje poređenje tražene cijene sa tržištem nekretnina.

e) Na osnovu prepoznatih izlaza, analizirati koji su to ulazi koje će korisnik unositi u sistem

Ulazi u sistem su predstavljeni sljedećim varijablama:

- Lokacija - lokacije na kojoj se nekretnina nalazi (Kategorička)
- Broj kvadrata - kvadratura određene nekretnine (Numerička)
- Broj soba - broj prostorija koje ulaze u cijenu i kvadraturu (Numerička)
- Broj spratova - broj spratova koje ulaze u nekretninu (Numerička)
- Vrsta grijanja - vrsta grijanja koja je prisutna (Kategorička)
- Okućnica (kvadratura) - kvadratura okućnice koja dolazi uz kuću (Numerička)
- Vrsta poda - vrsta poda koja je urađena u kući materijal (Kategorička)
- Kanalizacija - posjedovanje kanalizacije ili ne (Kategorička)
- Video nadzor - posjedovanje video nadzora ili ne (Numerička)
- Alarm - posjedovanje alarma ili ne (Numerička)
- Balkon - posjedovanje balkona ili ne (Numerička)
- Internet - posjedovanje interneta ili ne (Numerička)
- Kablovska - posjedovanje kablovske TV ili ne (Numerička)
- Namještena - posjedovanje namještaja ili ne (Numerička)
- Parking - posjedovanje parkinga ili ne (Numerička)
- Struja - posjedovanje struje ili ne (Numerička)
- Telefonski priključak - posjedovanje telefonskog priključka ili ne (Numerička)
- Uknjižena - posjedovanje uknjiženih dokumenata ili ne (Numerička)
- Voda - posjedovanje vode ili ne (Numerička)
- Garaža (Numerička)
- Adresa (Numerička)
- Primarna orijentacija (Kategorička)
- Blindirana vrata - posjedovanje blindiranih vrata ili ne (Numerička)
- Klima (Numerička)
- Plin (Numerička)
- Godina izgradnje - godina izgradnje nekretnine, njena starost (Numerička)
- Nedavno adaptirana (Numerička)
- Bazen (Numerička)

f) Da li su postojeći podaci iz skupa podataka relevantni za uspješno donošenje odluke? Zašto?

Svi podaci, izuzev generičkih podataka kao što je url i sl., su relevantni za proces odlučivanja - podaci o nekretnini kao što su: kvadratura, raspoloživost soba, broj spratova itd, ali i podaci o lokaciji i uključenosti dodatnih životnih potreba kao što su voda, struha, grijanje i sl. Zašto? Podaci o lokaciji, kvadraturi, broju soba nam govore o udaljenosti/blizini nekretnine od naseljenog dijela kao i o njenoj veličini i prostranosti i mogućnosti analize da li je odabrana nekretnina pogodna i prilagođena za životne potrebe porodice, a s druge strane, podaci o godini izgradnje, dostupnosti interneta, vrste podova i slično pomažu u analizi nekretnine o njenom trenutnom stanju starosti kao i uslugama koje su potrebne kupcu.

Analizom posmatranog dataseta zaključili smo da bi za proces odlučivanja bilo korisnije da skup podataka sadrži više detalja o socijalnoj slici kupca, obzirom da je u skupu trenutno prisutno više podataka o samim nekretninama, a nešto manje o samom kupcu. Podaci o njegovom stanju bi dodatno utjecali na ispravnost odlučivanja. Podaci (koji se ne nalaze na OLX oglasima) a koji bi mogli poboljšati model jer utiču na cijenu nekretnina:

- Prosječna plata stanovništva kantona/općine u kojoj se nalazi nekretnina
- Prosječna primanja domaćinstava kantona/općine u kojoj se nalazi nekretnina
- Broj škola/vrtića kantona/općine u kojoj se nalazi nekretnina
- Postotak/broj zaposlenih/nezaposlenih po kantonu
- Postotak/broj obrazovanih osoba u kantonu/općini u kojoj se nalazi nekretnina (visoko obrazovanje)
- Postotak/broj obrazovanih osoba u kantonu/općini u kojoj se nalazi nekretnina (srednje obrazovanje)
- Postotak/broj neobrazovanih osoba u kantonu/općini u kojoj se nalazi nekretnina
- Index zagađenosti kantona/općine u kojoj se nalazi nekretnina

Iako ovi podaci nisu dostupni na OLX-u (u sklopu oglasa za određenu nekretninu) npr prosječna plata se može pronaći na stranici PIO MIO. Postotak zaposlenih / nezaposlenih osoba na stranicama biroa za zapošljavanje, broj vrtića/škola se može pronaći na stranicama ministarstava obrazovanja kantona, broj obrazovanih / neobrazovanih osoba se može pronaći na popisu stanovništva ([link](#))

4.2. Objasniti kojem tipu problema pripada problem

Tip problema: Djelimično struktuirani

Po samoj definiciji djelimično struktuiranog problema, sistem za podršku određivanja cijene nekretnine spada u takvu skupinu. Ovakav tip problema se vodi određenim iskustvom unutar nestruktuiranog dijela problema, što preslikano na problem određivanja cijene nekretnine predstavlja iskustvo osobe koja prodaje nekretninu, ako problem posmatramo iz perspektive prodavca nekretnine, odnosno osobe koja kupuje nekretninu ako problem posmatramo iz

perspektive osobe koja kupuje nekretninu, a sve to s ciljem ispravnog određivanja cijene nekretnine.

Zašto smatramo da problem koji rješavamo nije potpuno struktuirani problem?

Bez obzira na sve stavke koje su navedene, a koje utječu na cijenu nekretnine, ne može se napraviti generalizirani model po kojem bi bile određene cijene kojem bi se “slijepo vjerovalo”. Razlog za to je što npr. dvije nekretnine mogu zadovoljavati/posjedovati iste stavke, ali npr. “kvalitet” pojedinih stavki nije isti. Naprimjer, dvije nekretnine imaju bazen, pri čemu jedna nekretnina ima puno bolji i veći bazen od druge nekretnine ili da su naprimjer, dvije nekretnine namještene, ali da jedna ima puno bolji, kvalitetniji, namještaj od druge. Zbog svega navedenog, cijene ove dvije nekretnine ne mogu biti iste. Nužno je uvesti faktor procjene koji je temeljen na iskustvu prodavca/kupca nekretnine što čini svaki problem na neki način nestruktuiranim. Iako dobijamo rezultat za cijenu nekretnine od strane modela, ipak je čovjek na kraju dužan odlučiti da li iskoristiti takav rezultat.

5. Preprocesiranje podataka

5.1. Dodatna analiza skupa podataka

Osim već dosadašnje analize, detaljnija analiza podataka će biti obrađena u poglavlju 5.4. u sklopu profiliranja podataka.

5.2. Učitavanje ključnih biblioteka

Za potrebe preprocesiranja podataka u Google Collab-u smo koristili funkcije iz biblioteka *pandas*, *numpy*, *seaborn*, *scipy.stats*, *matplotlib*. Od ostalih biblioteka koje se nalaze u sklopu već navedenih korišteni su i *percentile*, *LabelEncoder*, *preprocessing*, *pyplot* i *chi2*.

Biblioteke *seaborn*, *matplotlib* i *chi2* su nam poslužile za grafičko predstavljanje podataka i korelacije među njima, *pandas* za učitavanje skupa podataka i *numpy* za matematičke operacije.

```
from google.colab import drive

import pandas as pd
import numpy as np
from numpy import percentile

from sklearn.preprocessing import LabelEncoder
from sklearn import preprocessing

import seaborn as sns
import matplotlib.pyplot as plt
```



```
import scipy.stats
from scipy.stats import chi2
```

5.3. Učitavanje skupa podataka

Skup podataka se nalazi na Google Drive, u folderu namjenjenom za izradu projekta. Zahvaljujući mogućnosti Google Collab okruženja da se direktno poveže na Google Drive, importovali smo biblioteku *drive* i na taj način smo mogli učitati skup podataka pomoću funkcije *read_csv*. Dataset se nalazi u varijabli *df*.

```
from google.colab import drive
drive.mount('/content/drive')

df = pd.read_csv('/content/drive/MyDrive/SPO projekat/Updated
dataset/house_pricing_dataset_final.csv')
```

5.4. Profiliranje podataka

Da bismo bolje izanalizirali vrijednosti našeg dataset-a, iskoristili smo funkciju *info*, kako bismo dobili uvid u to koliko null i notnull vrijednosti imamo, te kojeg su tipa kolone našeg dataseta. Na slici ispod vidimo poziv funkcije, i njen rezultat.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4479 entries, 0 to 4478
Data columns (total 31 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   price                                4479 non-null   object
1   Lokacija                             4479 non-null   object
2   Kvadrata                             4380 non-null   object
3   Broj soba                            4424 non-null   object
4   Broj spratova                        4424 non-null   object
5   Vrsta grijanja                       4424 non-null   object
6   Okučnica (kvadratura)                3362 non-null   object
7   Vrsta poda                           2285 non-null   object
8   Kanalizacija                         2157 non-null   object
9   Alarm                                4479 non-null   object
10  Balkon                              4479 non-null   object
11  Internet                             4479 non-null   object
12  Kablovska TV                         4479 non-null   object
13  Namještena                           4479 non-null   object
14  Ostava/špajz                         4479 non-null   object
15  Parking                              4479 non-null   object
16  Podrum/Tavan                         4479 non-null   object
17  Struja                                4479 non-null   object
18  Telefonski priključak                 4479 non-null   object
19  Uknjiženo / ŽK                       4479 non-null   object
20  Video nadzor                         4479 non-null   object
21  Voda                                  4479 non-null   object
22  Garaža                                4479 non-null   object
23  Primarna orijentacija                 1226 non-null   object
24  Blindirana vrata                     4479 non-null   object
25  Klima                                4479 non-null   object
26  Plin                                  4479 non-null   object
27  Godina izgradnje                      2074 non-null   object
28  Nedavno adaptirana                   4479 non-null   object
29  Iznajmljeno                          4440 non-null   float64
30  Bazen                                4440 non-null   object
dtypes: float64(1), object(30)
memory usage: 1.1+ MB
None
```

Slika 36 - Prikaz notnull vrijednosti dataseta

Vidimo da varijabla *price* i lokacija, od 4479 mogućih redova, imaju isti toliko notnull vrijednosti, što znači da su potpune, a to smo i potvrdili pozivanjem funkcije *isna* nad varijablom *price*, što vidimo ispod.

```
[ ] df['price'].isna().sum()
```

0

Slika 37- Prikaz broja null vrijednosti varijable *price*

Ostale varijable imaju null vrijednosti, a njihov broj jednak je ukupnom broju redova 4479, umanjen za broj na početku druge kolone na slici.

Analiza numeričkih varijabli

Podalje ćemo analizirati kolonu *price*. Najprije ćemo izvršiti njen kratak pregled, koristeći funkciju *print*, što vidimo na sljedećoj slici.

```
[ ] print(df['price'])
```

0 450000

1 159000

2 195000

3 199000

4 350000

...

4474 499000

4475 290000

4476 65000

4477 170000

4478 280000

Name: price, Length: 4469, dtype: object

Slika 38 - Prikaz vrijednosti varijable *price*

Zatim, da bismo imali jasniji uvid u varijablu *price*, iskoristili smo funkciju *describe*, koja vraća statističke osobine kolone cijena, što vidimo na sljedećoj slici.

```
[ ] df['price'].describe()
```

```
count          4479
unique          542
top      150.000 KM
freq           130
Name: price, dtype: object
```

Slika 39 - Prikaz statističkih osobina varijable price

Imamo uvid u attribute:

count: broj neNaN vrijednosti u koloni *price*, u ovom slučaju 4479.

unique: broj jedinstvenih vrijednosti u koloni *price*, u ovom slučaju 542.

top: najčešća (mod) vrijednost u koloni *price*, u ovom slučaju 150.000 KM.

freq: broj pojavljivanja najčešće (mod) vrijednosti u koloni *price*, u ovom slučaju 130.

Detaljnije statističke vrijednosti za kolonu *price* smo dobili pozivima metoda: *percentile*, *min*, *max* i *mean*. Prikaz poziva metoda i rezultata vidimo na sljedećoj slici:

```
from numpy import percentile
quartiles = percentile(df['price'], [25, 50, 75])
# calculate min/max
data_min, data_max = df['price'].min(), df['price'].max()
# print 5-number summary
print('Min: %.3f' % data_min)
print('Q1: %.3f' % quartiles[0])
print('Median: %.3f' % quartiles[1])
print('Mean: %.3f' % df["price"].mean())
print('Q3: %.3f' % quartiles[2])
print('Max: %.3f' % data_max)
```

```
Min: 50.000
Q1: 90000.000
Median: 150000.000
Mean: 168934.478
Q3: 240000.000
Max: 399999.000
```

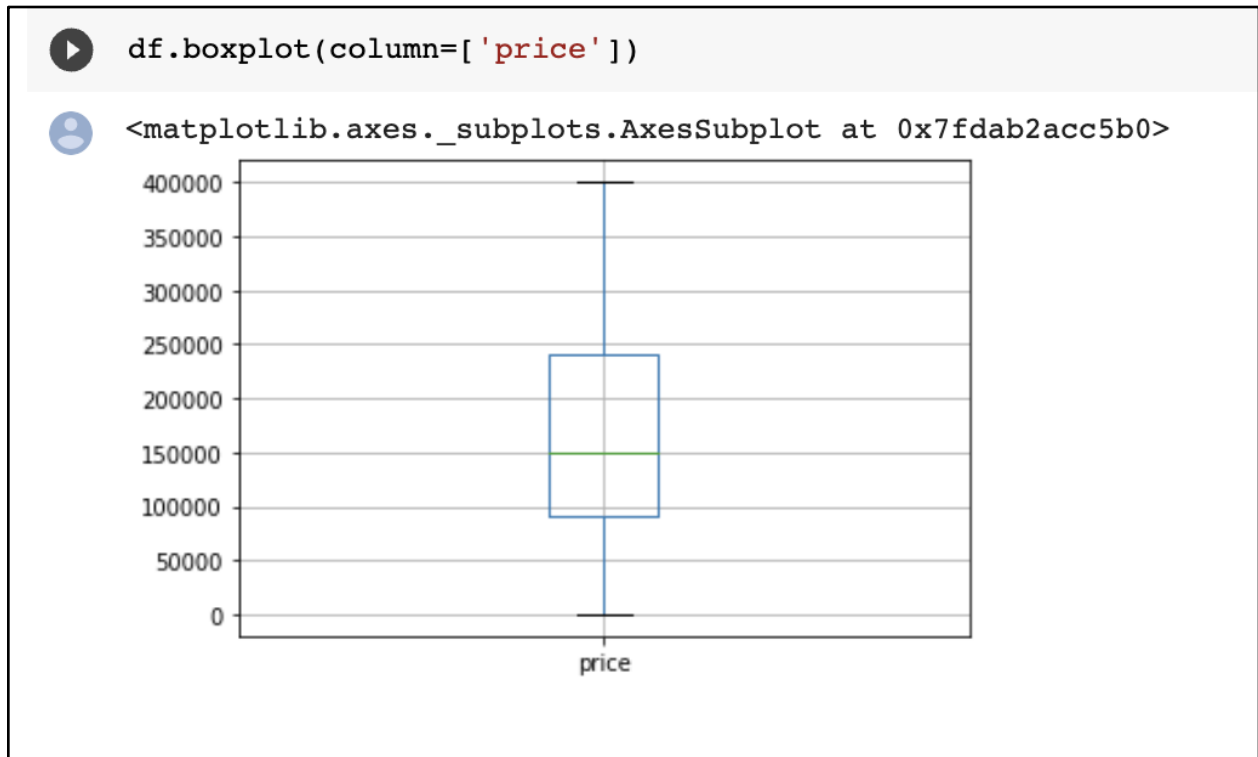
Slika 40 - Prikaz statističkih vrijednosti varijable price

Prvo se računaju kvantili, tj. vrijednosti koje dijele skup podataka na četiri jednaka dijela. Konkretno, računaju se kvantili na 25%, 50% i 75% kvantilima distribucije *price* varijable. Zatim se

računaju minimalna i maksimalna vrijednost *price* varijable. Nakon toga se ispisuje 5-numerički sažetak (minimum, Q1 kvartil, medijan, srednja vrijednost, Q3 kvartil i maksimum) za varijablu *price*.

Svaka vrijednost je formatirana da bude prikazana s tri decimalna mjesta pomoću `%.3f`.

Kako bismo vizualizirali raspodjelu numeričkih podataka i prikazali maloprije spomenutih pet osnovnih statističkih mjera: minimum, maximum, kvartile Q1, Q2 i Q3, napraviti ćemo i boxplot varijable *price*. Također ćemo dobiti i bolji uvid u outliere i ekstremne vrijednosti.



Slika 41 - Prikaz boxplota varijable *price*

Analizom boxplota vidimo da outlier-a i ekstremnih vrijednosti nema – kasnije biti će objašnjeno zašto, a vrijednosti nekretnine se kreću od 99 000 do 240 000 KM. Medijana je 15 000.

Na sličan način kao i varijablu *price*, analizirat ćemo i ostale numeričke varijable, pri čemu ćemo u izvještaju pokazati samo rezultate primjenjive analize.

Broj spratova

```
[ ] df['Broj spratova'].isna().sum()
```

49

```
[ ] df['Broj spratova'].describe()
```

```
count      3803
unique        5
top          2
freq       1928
Name: Broj spratova, dtype: object
```

```
df.boxplot(column=['Broj spratova'])
```

<matplotlib.axes._subplots.AxesSubplot at 0x7:

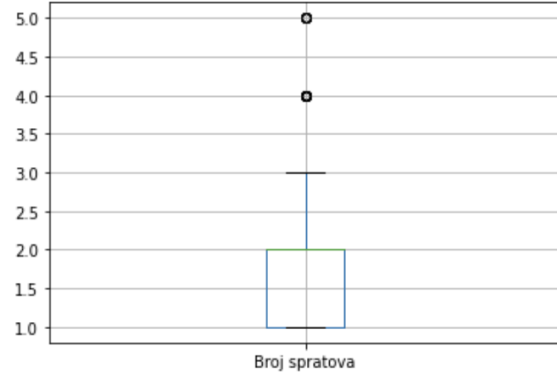


Tabela 1: Prikaz statistike i boxplota varijable Broj spratova

Broj soba

```
[ ] df['Broj soba'].isna().sum()
```

0

```
[ ] df['Broj soba'].describe()
```

```
count      3804
unique        8
top          4
freq       747
Name: Broj soba, dtype: object
```

```
df.boxplot(column=['Broj soba'])
```

<matplotlib.axes._subplots.AxesSubplot at 0x7:

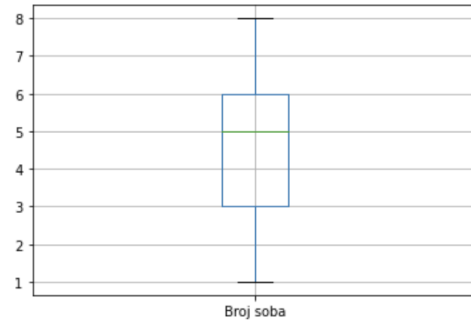


Tabela 2: Prikaz statistike i boxplota varijable Broj soba

Kvadratura

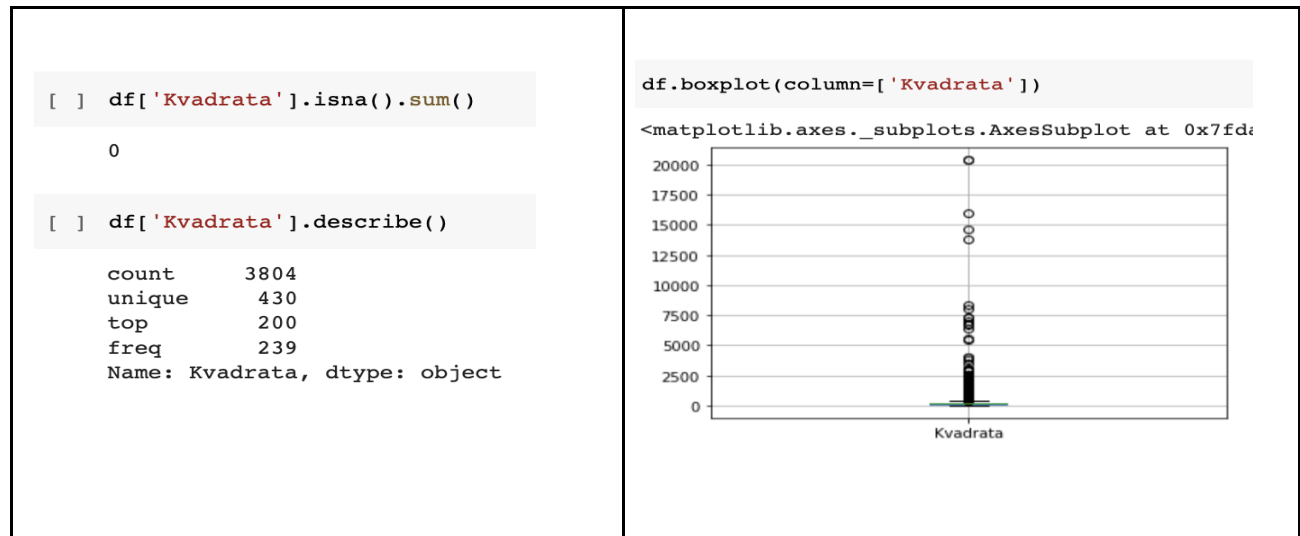


Tabela 3: Prikaz statistike i boxplota varijable Kvadratura

Godina izgradnje

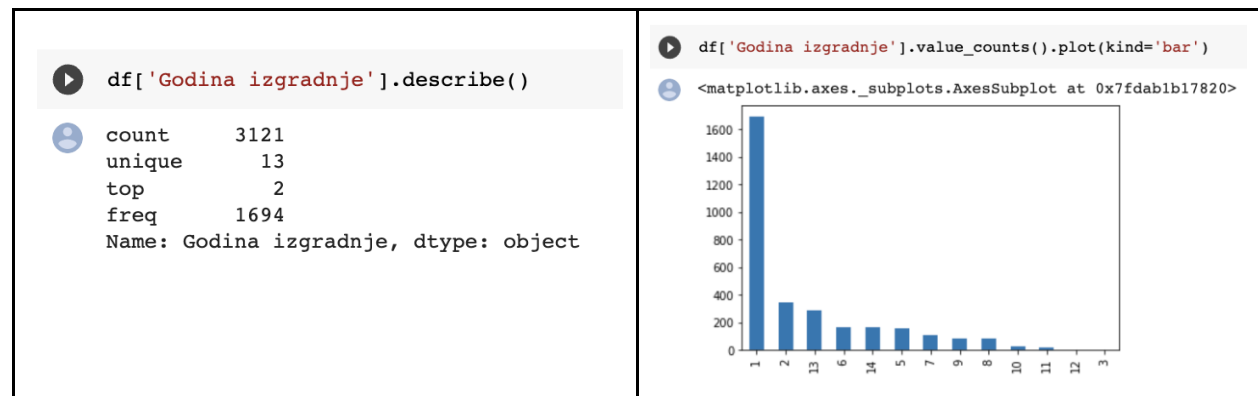


Tabela 4: Prikaz statistike i barplota varijable Godina izgradnje

Analiza numeričkih varijabli sa binarnim podacima

Analiza numeričkih varijabli, koje za podatke sadrže samo 0 ili 1, što predstavlja ima/nema je prikazana tabelarno za svaku varijablu. Analiza se sadržavala od prikaza osnovnih analitičkih i statističkih podataka dobivenih primjenom metode *describe*, i grafičkog prikaza raspodjele vrijednosti.

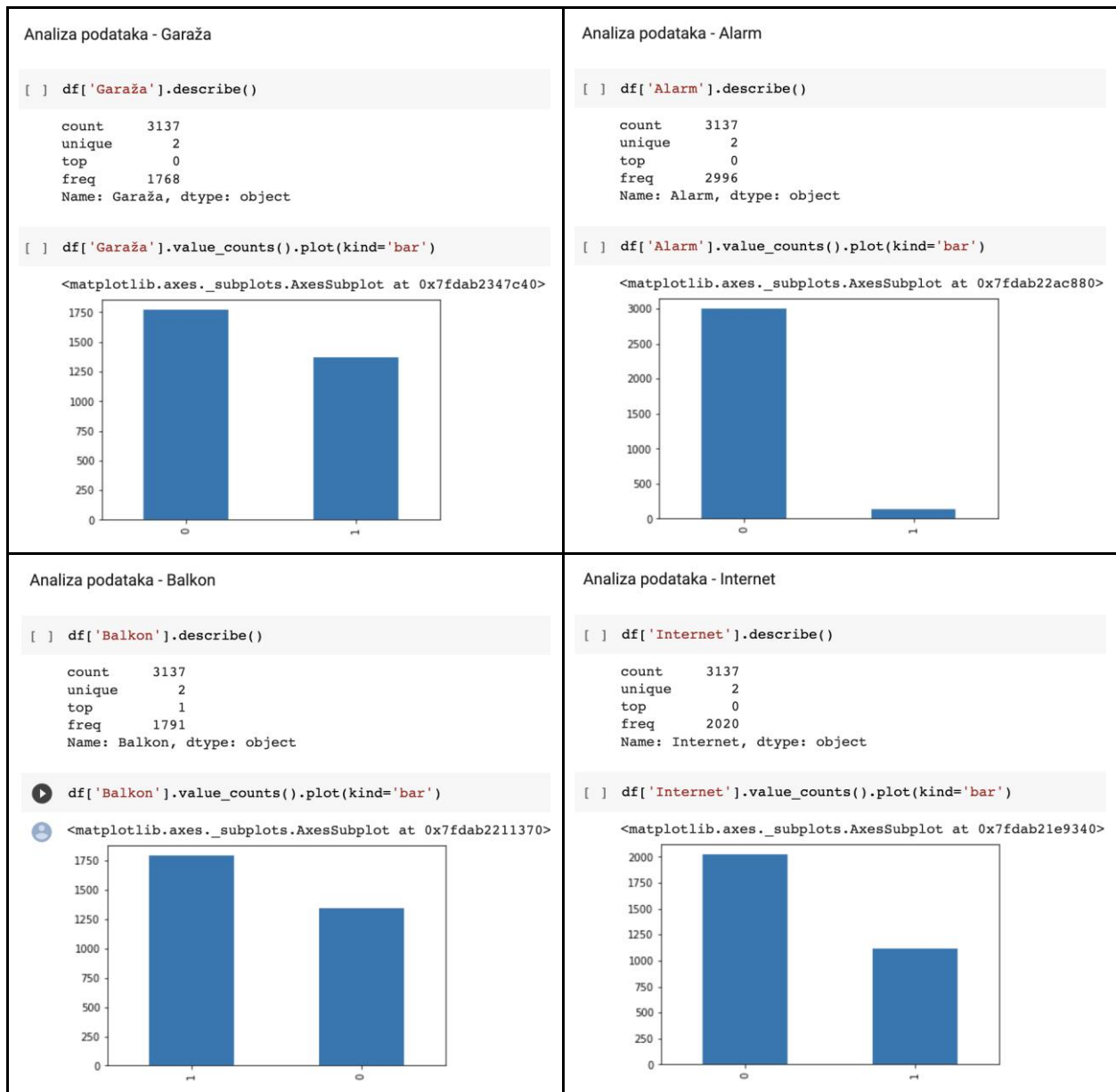


Tabela 5: Analiza podataka za varijable: Garaža, Alarm, Balkon i Internet

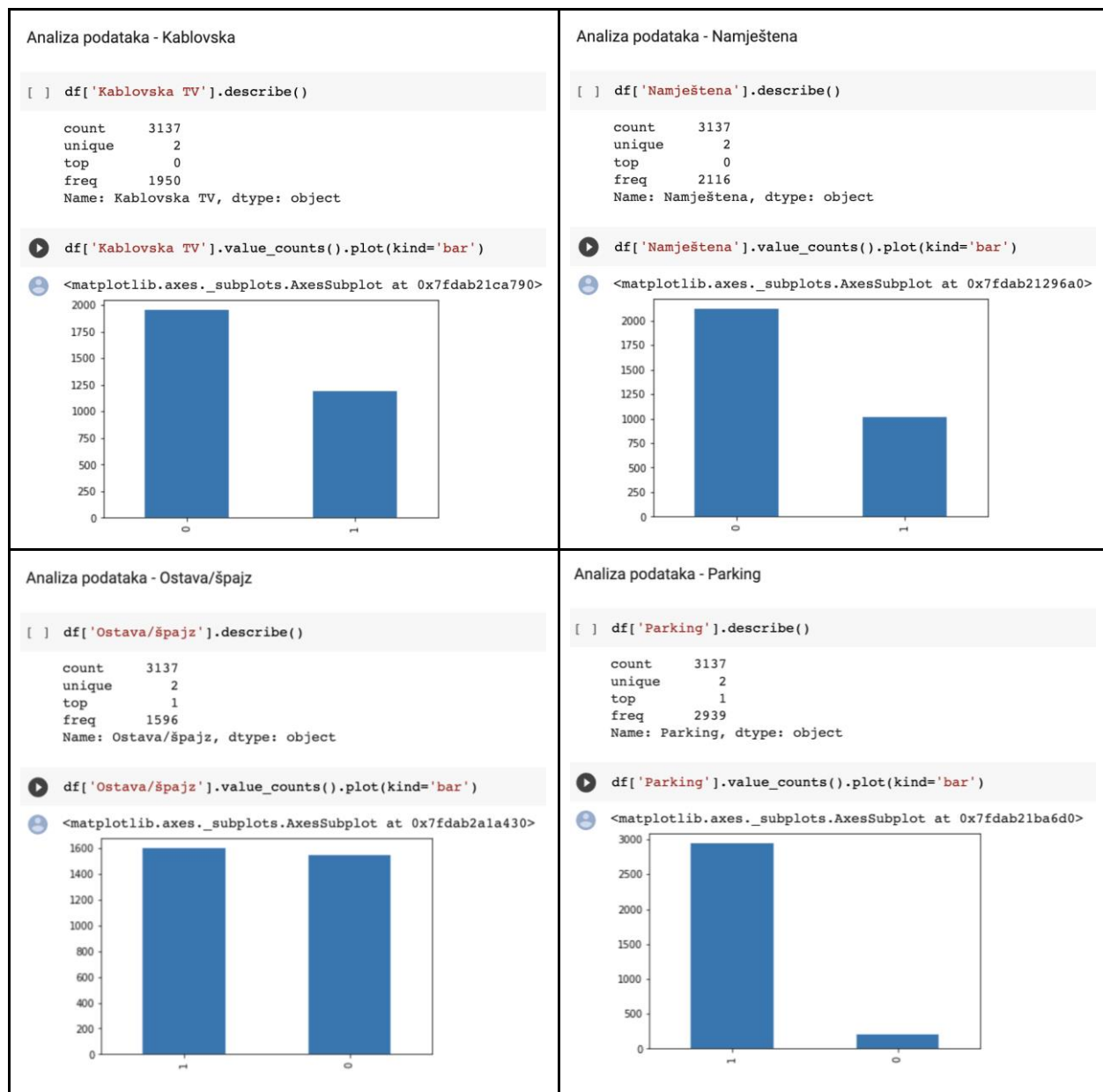


Tabela 6: Analiza podataka za varijable: Kablovska, Namještena, Ostava i Parking

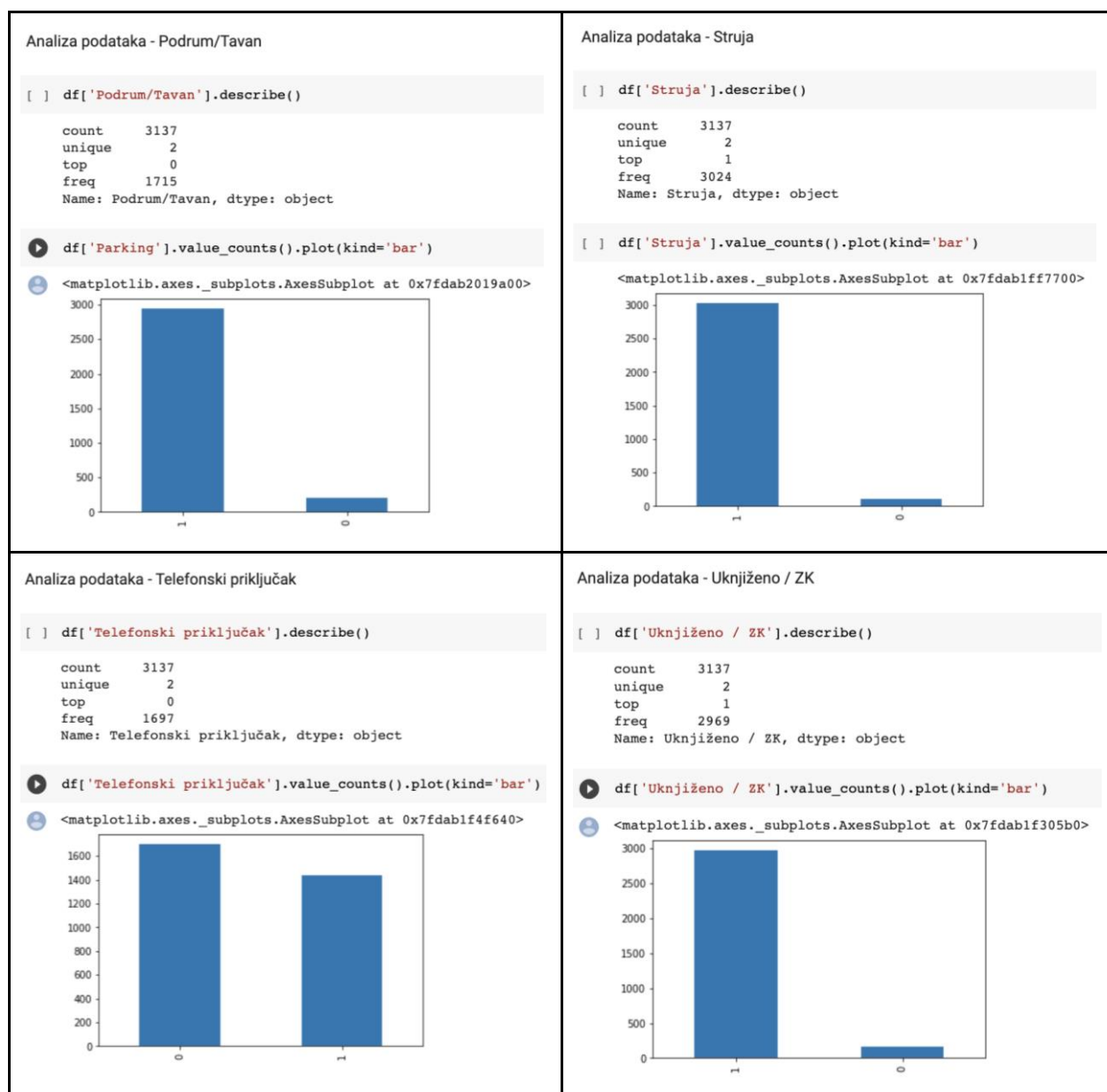


Tabela 7: Analiza podataka za varijable: Podrum, Struja, Priključak, Uknjiženo

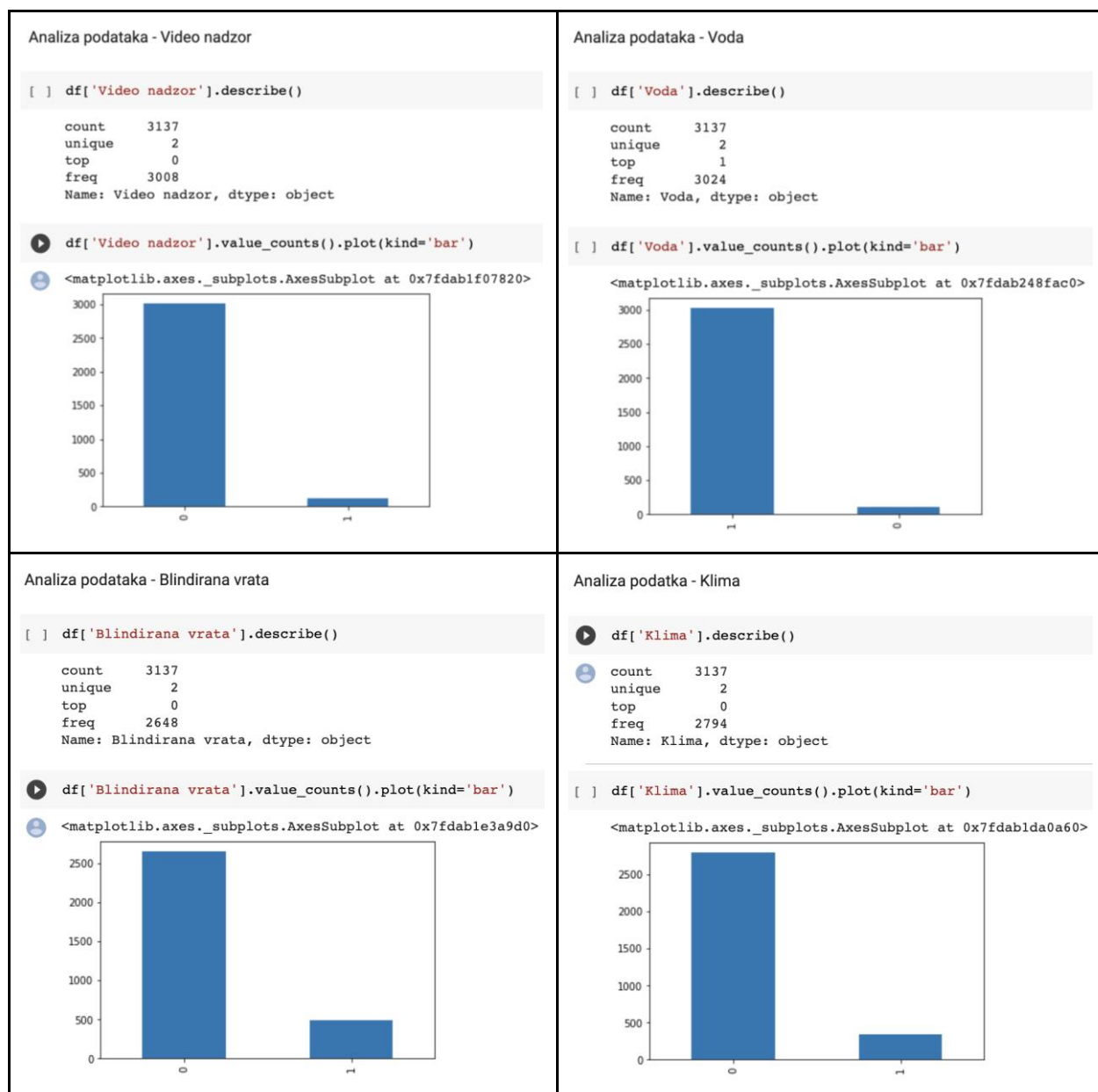


Tabela 8: Analiza podataka za varijable: Video nadzor, Voda, Blindirana vrata i Klima

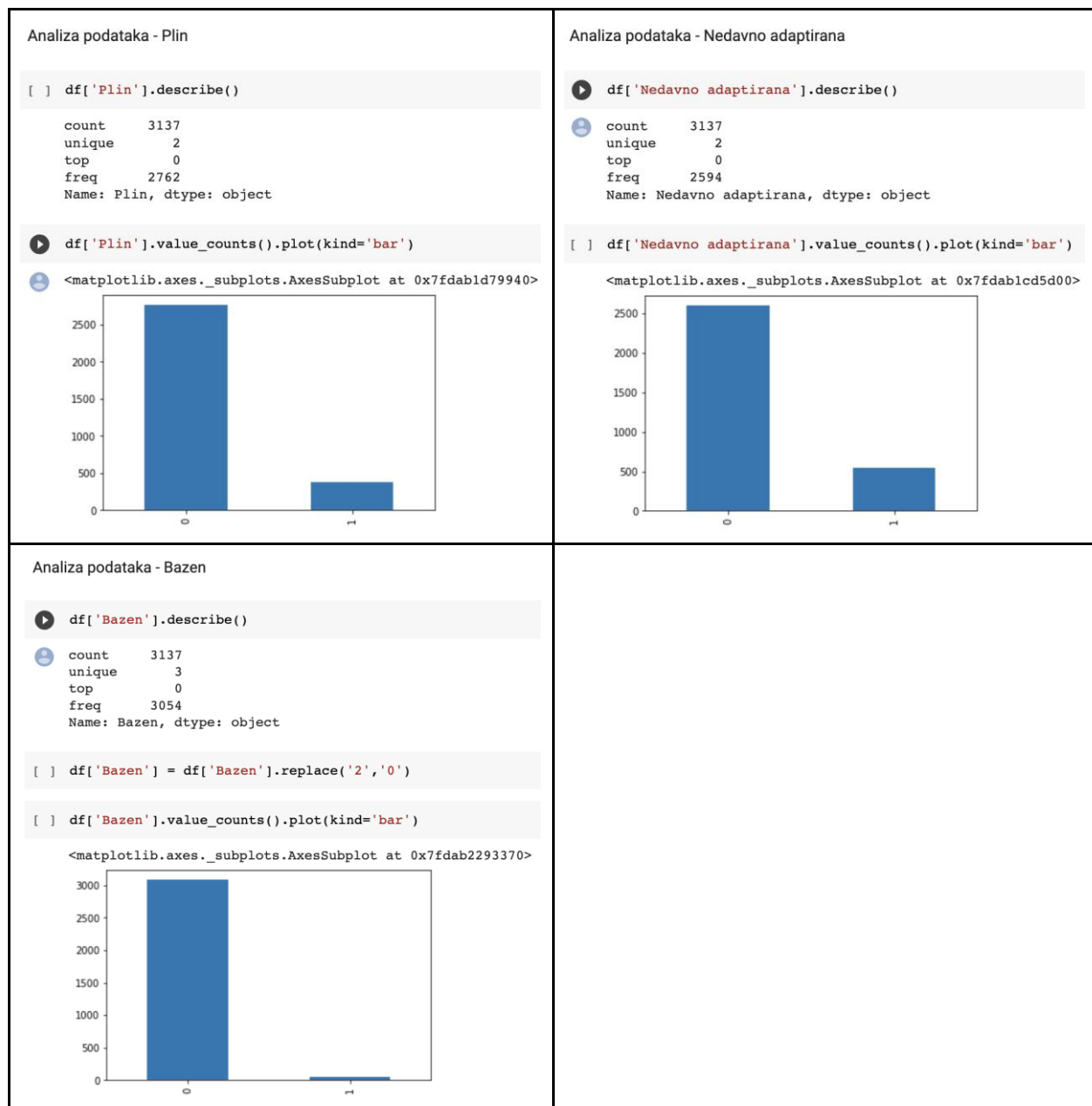


Tabela 9: Analiza podataka za varijable: Plin, Nedavno adaptirana i Bazen

Analiza kategoričkih vrijednosti

Analiza kategoričkih podataka je rađena tako da se najprije prebace u numeričke vrijednosti, i onda prikaže njihova statistika i barplotovi.

Vrsta grijanja

Prikaz transformacije kategoričkih podataka u numeričke u varijabli *Vrsta grijanja* je prikazan na slici dolje.

```
df['Vrsta grijanja'] = df['Vrsta grijanja'].replace('Plin','Centralno (Plin)')
df['Vrsta grijanja'] = df['Vrsta grijanja'].replace('NA','Drva')
df['Vrsta grijanja'] = df['Vrsta grijanja'].replace('Centralno
(Kotlovnica)','1')
df['Vrsta grijanja'] = df['Vrsta grijanja'].replace('Centralno (Plin)','2')
df['Vrsta grijanja'] = df['Vrsta grijanja'].replace('Struja','3')
df['Vrsta grijanja'] = df['Vrsta grijanja'].replace('Drva','4')
df['Vrsta grijanja'] = df['Vrsta grijanja'].replace('Ostalo','5')
```

Nakon pretvaranja kategoričkih u numeričke podatke, slijedi isti pristup analizi kao i ranije, prikazan ispod.

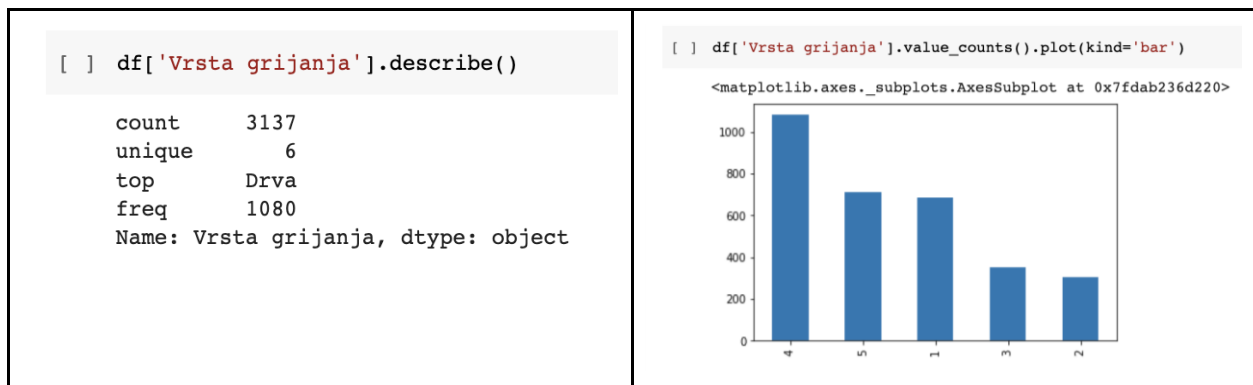


Tabela 10: Prikaz analize podataka varijable *Vrsta grijanja*

Kanalizacija

Prikaz transformacije kategoričkih podataka u numeričke u varijabli *Kanalizacija* je prikazan na slici dolje.

```
df['Kanalizacija'] = df['Kanalizacija'].replace('Standardna','1')
df['Kanalizacija'] = df['Kanalizacija'].replace('Septička jama','2')
```

```
df['Kanalizacija'] = df['Kanalizacija'].replace('Nema', '3')
```

Nakon pretvaranja kategoričkih u numeričke podatke, slijedi isti pristup analizi kao i ranije, prikazan ispod.

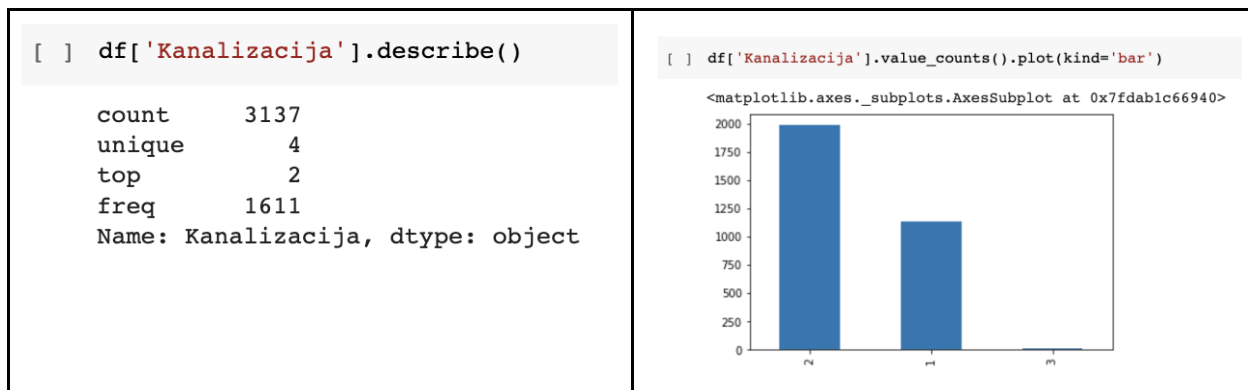


Tabela 11: Prikaz analize podataka varijable kanalizacija

Vrsta poda

Prikaz transformacije kategoričkih podataka u numeričke u varijabli *Vrsta poda* je prikazan na slici dolje.

```
df['Vrsta poda'] = df['Vrsta poda'].replace('2', 'Ostalo')
df['Vrsta poda'] = df['Vrsta poda'].replace('Parket', '1')
df['Vrsta poda'] = df['Vrsta poda'].replace('Laminat', '2')
df['Vrsta poda'] = df['Vrsta poda'].replace('Brodski', '3')
df['Vrsta poda'] = df['Vrsta poda'].replace('Beton', '4')
df['Vrsta poda'] = df['Vrsta poda'].replace('Ostalo', '5')
```

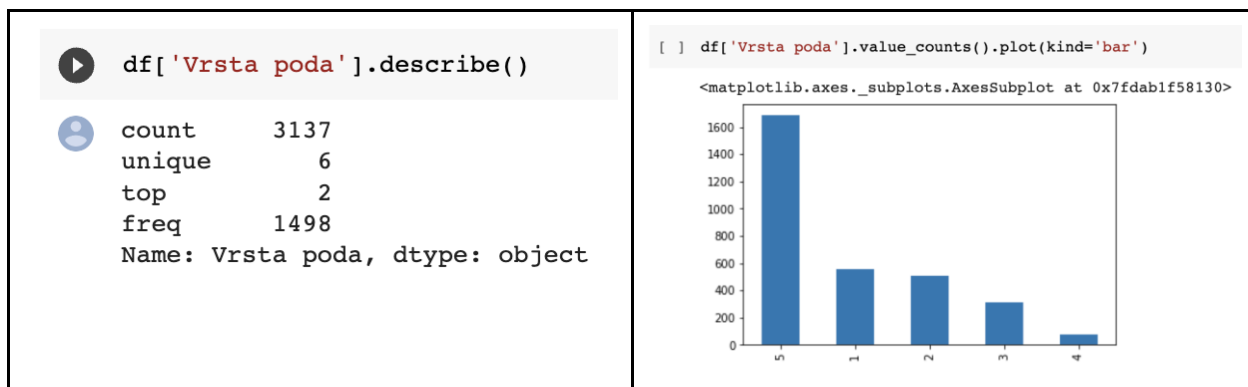


Tabela 12: Prikaz analize podataka varijable Vrsta poda

Lokacija

Za varijablu *lokacija*, nije rađena pretvorba iz kategoričkih u numeričke. Slijedi samo prikaz statistike.

<p>Analiza podataka - Lokacija</p> <pre>[] df['Lokacija'].describe()</pre> <table><tr><td>count</td><td>3137</td></tr><tr><td>unique</td><td>118</td></tr><tr><td>top</td><td>Brčko</td></tr><tr><td>freq</td><td>308</td></tr><tr><td colspan="2">Name: Lokacija, dtype: object</td></tr></table>	count	3137	unique	118	top	Brčko	freq	308	Name: Lokacija, dtype: object		<pre>[] print(df["Lokacija"].unique())</pre> <p>['Tuzla' 'Hadžići' 'Grad Mostar' 'Novo Sarajevo' 'Zenica' 'Banovići' 'Banja Luka' 'Prijedor' 'Brčko' 'Lukavac' 'Doboj' 'Iliđa' 'Konjic' 'Sarajevo - Centar' 'Zvornik' 'Stolac' 'Bijeljina' 'Visoko' 'Sarajevo, Novi Grad' 'Travnik' 'Breza' 'Živinice' 'Gradačac' 'Novi Travnik' 'Vogošća' 'Kasindo' 'Vitez' 'Šamac' 'Ilijaš' 'Trnovo' 'Žepče' 'Bihać' 'Pale' 'Foča' 'Stari Grad' 'Kiseljak' 'Cazin' 'Laktaši' 'Srbac' 'Gornji Vakuf - Uskoplje' 'Kakanj' 'Donji Vakuf' 'Usora' 'Bosanska Krupa' 'Kalesija' 'Pelagićevo' 'Čelinac' 'Srebrenik' 'Lukavica' 'Teslić' 'Odžak' 'Bugojno' 'Gradiška' 'Fojnica' 'Kreševo' 'Osmaci' 'Modriča' 'Kladanj' 'Tešanj' 'Prnjavor' 'Ljubuški' 'Derventa' 'Banjalučka, Novi Grad' 'Neum' 'Goražde' 'Olovo' 'Nevesinje' 'Velika Kladuša' 'Bosanski Petrovac' 'Busovača' 'Ravno' 'Bratunac' 'Vareš' 'Kozarska Dubica' 'Zavidovići' 'Čelić' 'Lopare' 'Višegrad' 'Gračanica' 'Ljubinje' 'Stanari' 'Široki Brijeg' 'Čapljina' 'Grude' 'Mrkonjić Grad' 'Brod' 'Trebinje' 'Donji Žabar' 'Kneževog Rogatica' 'Sapna' 'Sokolac' 'Doboj-Jug' 'Ključ' 'Orašje' 'Petrovo' 'Jablanica' 'Mačelj' 'Šipovo' 'Sanski Most' 'Jajce' 'Kupres' 'Milići' 'Livno' 'Istočni Drvar' 'Vlasenica' 'Glamoč' 'Srebrenica' 'Gacko' 'Turska' 'Novi Grad' 'Kotor Varoš' 'Ribnik' 'Drvar' 'Bileća']</p>
count	3137										
unique	118										
top	Brčko										
freq	308										
Name: Lokacija, dtype: object											

Tabela 13: Prikaz analize podataka varijable *lokacija*

Primarna orijentacija

Za varijablu *Primarna orijentacija*, također nije rađena pretvorba iz kategoričkih u numeričke. Slijedi samo prikaz statistike i barplota.

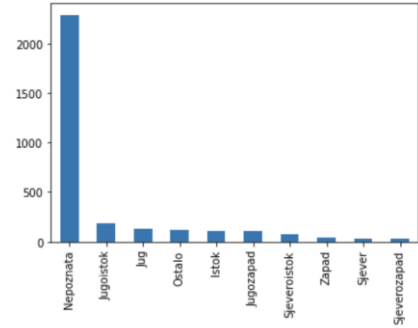
<pre>df['Primarna orijentacija'].describe()</pre> <table><tr><td>count</td><td>3121</td></tr><tr><td>unique</td><td>10</td></tr><tr><td>top</td><td>2</td></tr><tr><td>freq</td><td>2291</td></tr><tr><td colspan="2">Name: Primarna orijentacija, dtype: object</td></tr></table>	count	3121	unique	10	top	2	freq	2291	Name: Primarna orijentacija, dtype: object		<pre>[] df['Primarna orijentacija'].value_counts().plot(kind='bar')</pre> <p><matplotlib.axes._subplots.AxesSubplot at 0x7fdab1a9e910></p>  <table><tr><td>Nepoznata</td><td>2291</td></tr><tr><td>Jugistok</td><td>2</td></tr><tr><td>Jug</td><td>2</td></tr><tr><td>Ostalo</td><td>2</td></tr><tr><td>Istok</td><td>2</td></tr><tr><td>Jugozapad</td><td>2</td></tr><tr><td>Sjeveristok</td><td>2</td></tr><tr><td>Zapad</td><td>2</td></tr><tr><td>Sever</td><td>2</td></tr><tr><td>Severozapad</td><td>2</td></tr></table>	Nepoznata	2291	Jugistok	2	Jug	2	Ostalo	2	Istok	2	Jugozapad	2	Sjeveristok	2	Zapad	2	Sever	2	Severozapad	2
count	3121																														
unique	10																														
top	2																														
freq	2291																														
Name: Primarna orijentacija, dtype: object																															
Nepoznata	2291																														
Jugistok	2																														
Jug	2																														
Ostalo	2																														
Istok	2																														
Jugozapad	2																														
Sjeveristok	2																														
Zapad	2																														
Sever	2																														
Severozapad	2																														

Tabela 14: Prikaz analize podataka varijable *Primarna orijentacija*

5.5. Čišćenje podataka

Uklanjanje NaN vrijednosti smo uradili tako što smo za neke kolone jednostavno izvršili izbacivanje NaN vrijednosti iz varijable, druge smo popunili medijanom varijable ili maximumom varijable. Spomenute slučajeve vidimo na isječcima ispod.

```
df = df.fillna(df['Broj spratova'].value_counts().index[0])
```

```
df = df.fillna(df['Kvadrata'].value_counts().index[0])
```

```
df = df.fillna(df['Kanalizacija'].value_counts().index[0])
```

```
df['Vrsta grijanja'] = df['Vrsta grijanja'].replace('NA', 'Drva')
```

Uklanjanje šuma smo radili ručno, dakle pregledom svih vrijednosti za datu varijablu odredili smo određene podatke koji stvaraju šum, te ih zamijenili funkcijom *replace*.

```
df['Broj soba'] = df['Broj soba'].replace('8+', '8')
```

```
df['Kvadrata'] = df['Kvadrata'].replace('148,6', '149')
df['Kvadrata'] = df['Kvadrata'].replace('120+70+7', '197')
df['Kvadrata'] = df['Kvadrata'].replace('210 m2', '210')
df['Kvadrata'] = df['Kvadrata'].replace('300+80', '308')
df['Kvadrata'] = df['Kvadrata'].replace(' NA', 'NA')
df['Kvadrata'] = df['Kvadrata'].replace('.', 'NA')
```

```
df['Bazen'] = df['Bazen'].replace('2', '0')
```

```
df['Primarna orijentacija'] = df['Primarna
orijentacija'].replace('2', 'Nepoznata')
```

```
df['Okućnica(kvadratura)'] = df['Okućnica(kvadratura)'].replace('800m2', '800')
df['Okućnica(kvadratura)'] = df['Okućnica(kvadratura)'].replace('554m2', '554')
df['Okućnica(kvadratura)'] = df['Okućnica(kvadratura)'].replace('500m2', '500')
df['Okućnica(kvadratura)'] = df['Okućnica(kvadratura)'].replace('500 m', '500')
df['Okućnica(kvadratura)'] = df['Okućnica(kvadratura)'].replace('360m2', '360')
df['Okućnica(kvadratura)'] =
df['Okućnica(kvadratura)'].replace('1428m2', '1428')
df['Okućnica(kvadratura)'] = df['Okućnica(kvadratura)'].replace('2
```

```

duluma', '2000')
df['Okućnica(kvadratura)'] = df['Okućnica(kvadratura)'].replace('600+', '600')
df['Okućnica(kvadratura)'] = df['Okućnica(kvadratura)'].replace('10 duluma
zemlje', '10000')
df['Okućnica(kvadratura)'] = df['Okućnica(kvadratura)'].replace('50 m2', '50')
df['Okućnica(kvadratura)'] = df['Okućnica(kvadratura)'].replace('5,784', '5784')
df['Okućnica(kvadratura)'] = df['Okućnica(kvadratura)'].replace('13.3', '13')
df['Okućnica(kvadratura)'] = df['Okućnica(kvadratura)'].replace('da', 'NA')
df['Okućnica(kvadratura)'] = df['Okućnica(kvadratura)'].replace('3.3', '3300')
df['Okućnica(kvadratura)'] = df['Okućnica(kvadratura)'].replace('946,00', '946')
df['Okućnica(kvadratura)'] =
df['Okućnica(kvadratura)'].replace('1000m2', '1000')
df['Okućnica(kvadratura)'] = df['Okućnica(kvadratura)'].replace('2 000', '2000')
df['Okućnica(kvadratura)'] = df['Okućnica(kvadratura)'].replace('nema', 'NA')
df['Okućnica(kvadratura)'] = df['Okućnica(kvadratura)'].replace('NA', '0')

```

```

df['Godina izgradnje'] = df['Godina izgradnje'].replace('2', 'Nepoznata')
df['Godina izgradnje'] = df['Godina izgradnje'].replace('Nepoznata', '1')
df['Godina izgradnje'] = df['Godina izgradnje'].replace('1980-1989', '2')
df['Godina izgradnje'] = df['Godina izgradnje'].replace(' 2000-2009', '3')
df['Godina izgradnje'] = df['Godina izgradnje'].replace(' 2010+', '4')
df['Godina izgradnje'] = df['Godina izgradnje'].replace('1990-1999', '5')
df['Godina izgradnje'] = df['Godina izgradnje'].replace('1970-1979', '6')
df['Godina izgradnje'] = df['Godina izgradnje'].replace('Novogradnja', '7')
df['Godina izgradnje'] = df['Godina izgradnje'].replace('2015+', '8')
df['Godina izgradnje'] = df['Godina izgradnje'].replace('1960-1969', '9')
df['Godina izgradnje'] = df['Godina izgradnje'].replace('Prije 1950', '10')
df['Godina izgradnje'] = df['Godina izgradnje'].replace('1950-1959', '11')
df['Godina izgradnje'] = df['Godina izgradnje'].replace(' 1990-1999', '12')
df['Godina izgradnje'] = df['Godina izgradnje'].replace('2000-2009', '13')
df['Godina izgradnje'] = df['Godina izgradnje'].replace('2010+', '14')

```

Uklanjanje outliera

Nakon svakog boxplota, kada ismo uočili outlier, vršili smo izbacivanje istih, što vidimo u kodu.

```

df= df[df.Kvadrata < 300]
df= df[df.Kvadrata > 40]

```

```

df= df[df['Broj spratova'] < 4]

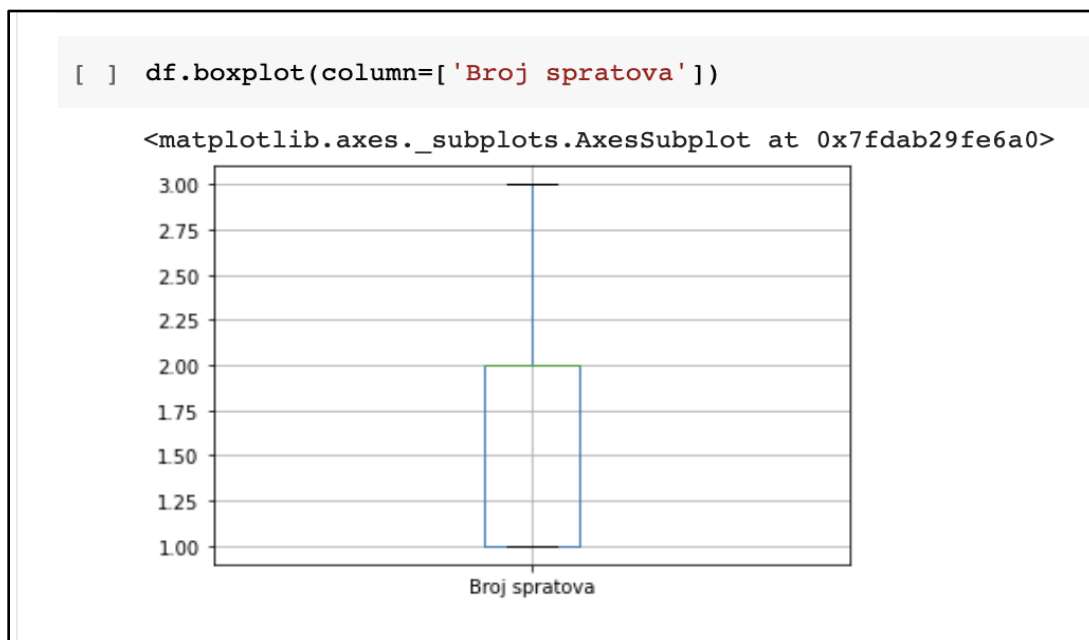
```

```

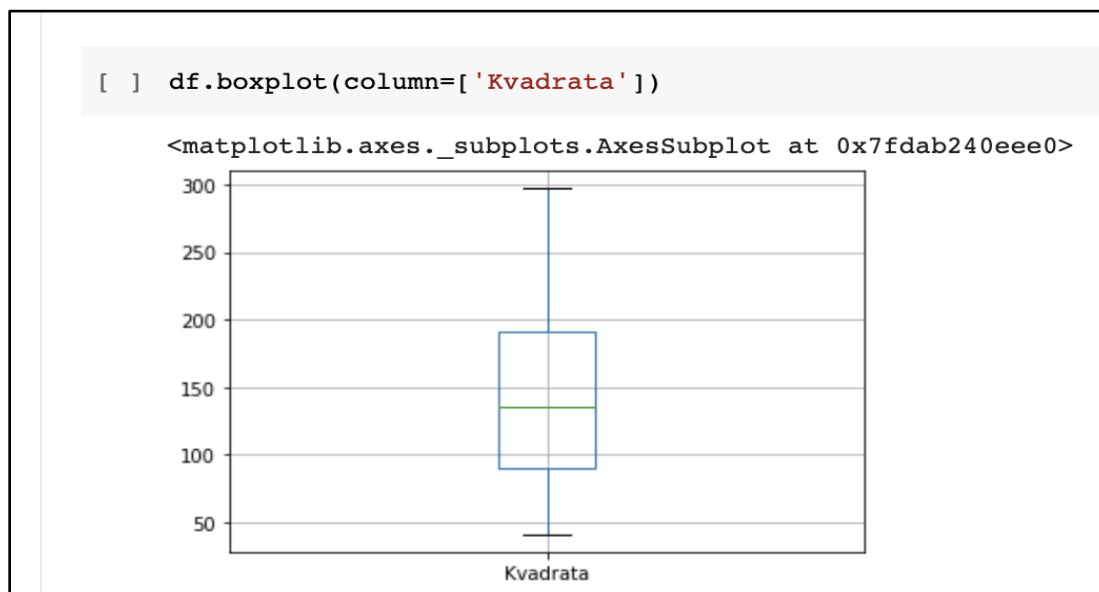
df= df[df.price < 400000]

```


Nakon prečišćavanja podataka, uklanjanja šuma i otklanjanja outliera, primjetili smo i promjene i poboljšanja na boxplotovima određenih varijabli. Promjene vidimo na slici ispod.



Slika 42 - Prikaz boxplota varijable Broj spratova



Slika 43 - Prikaz boxplota varijable Kvadrata

5.6. Transformacija podataka

Najprije smo krenuli od izjednačavanja podataka. Naime, dosta kolona u našem datasetu ima bool vrijednost, koja znači da ta nekretnina sadržava ili ne sadržava tu kolonu odnosno osobinu. Neke od tih kolona su: Alarm, Balkon, Internet, Kablovska TV, Namještena, Ostava/špajz, Parking, Tavan/podrum, Struja, Telefonski priključak, Uknjižena/ZK, Video nazor, Voda, Garaža, Blindirana vrata, Klima, Plin, Nedavno adaptirana, Iznajmljena i Bazen. Samim tim, saržaj ovih kolona je u vrijednosima 0 ili 1, osim par kolona koje sadrže vrijednosti “Da” ili “Ne”. Kako bi vrijednost svih redova u ovim kolonama bila uniformna, odlučili smo da konvertujemo “Da” vrijednosti u 1, a “Ne” vrijednosti u 0.

Ovu operaciju smo kompletirali koristeći funkciju *replace*.

```
df = df.replace('Da', '1')
df = df.replace('Ne', '0')
```

Pregled redova koji su sadržavali vrijednosti Da i Ne je prikazan na slici dolje.

J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Alarm	Balkon	Internet	Kablovska TV	Namještena	Ostava/špajz	Parking	Podrum/Tavan	Struja	Telefonski priključak	Uknjiženo / ZK	Video nadzor	Voda	Garaža
0	Da		0	0	0	Da	Da	Da	Da			0	Da
	0		0	0	0	0	Da	Da	Da	0	Da	0	Da
Da	Da		0	0	0	Da	Da	Da	Da	0	Da	Da	Da
	0	0	0	0	0	Da	Da	Da		0	Da	0	Da
	0	Da	Da	Da	Da	Da	Da	Da	Da	Da		0	Da
	0	Da	0	Da	0	0	Da	Da	Da		0	Da	0
	0	Da	Da	Da	Da	Da	Da	Da	Da	Da	Da	Da	Da
	0	Da	Da	Da	Da	Da		0	Da	Da	0	0	Da
	0	1	1	0	0	1	1	1	1	0	1	0	1
	0	1	0	0	0	0	1	1	1	0	1	0	1

Slika 44 - Prikaz primjene kategorički u numerički kao čišćenje podataka

U našem projektu smo koristili normalizaciju kao način transformacije podataka. Normalizacija je izvršena nad kolonom *Okučnica (Kvadratura)*. Razlog za to je što je raspon između pojedinih vrijednosti ove kolone velik. Zato smo min-max normalizacijom pomenute podatke postavili na raspon između 0 i 1. U nastavku je prikazan kod:

```
# copy the data
df_min_max_scaled = df.copy()

# apply normalization techniques by Column 1
column = 'Okučnica (kvadratura)'
df_min_max_scaled[column] = (df_min_max_scaled[column] -
df_min_max_scaled[column].min()) / (df_min_max_scaled[column].max() -
df_min_max_scaled[column].min())

# view normalized data
display(df_min_max_scaled)
df = df_min_max_scaled
```

	price	Lokacija	Kvadrata	Broj soba	Broj spetova	Vrsta grijanja	Okućnica (kvadratura)	Vrsta poda	Kanalizacija	Alarm	...	Voda	Garaža	Primarna orijentacija	Blindirana vrata	Klima	Plin	Godina izgradnje	Nedavno adaptirana	Iznajmljeno	Basen
1	159000.0	Tuzla	240	4	3	2	0.000014	5	2	0	...	1	0	2	0	0	0	2	0	0.0	0
2	195000.0	Hadžići	140	7	3	1	0.000014	5	2	0	...	0	0	2	0	0	0	2	0	0.0	0
3	199000.0	Grad Mostar	238	6	1	4	0.004246	5	2	0	...	1	1	2	0	0	0	2	0	0.0	0
4	350000.0	Novo Sarajevo	150	7	2	2	0.002817	1	1	1	...	1	1	Sjeveroistok	1	1	1	2	0	0.0	0
6	85000.0	Zenica	73	5	3	5	0.015683	5	2	0	...	1	1	2	0	0	0	2	0	0.0	0
...
4471	80000.0	Brčko	109	7	2	5	0.169014	2	2	0	...	1	1	2	0	0	0	2	0	2	0
4472	320000.0	Ilidža	80	3	2	1	0.013599	1	2	0	...	1	1	Ostalo	1	0	0	2000-2009	1	2	0
4473	105000.0	Pale	180	5	3	4	0.003028	2	2	0	...	1	0	Ostalo	0	0	0	2000-2009	0	2	0
4475	290000.0	Banja Luka	212	4	2	5	0.000014	5	2	0	...	1	0	2	0	0	0	2	0	2	0
4477	170000.0	Brčko	92	3	1	1	0.004049	2	2	0	...	1	0	2	0	0	0	2	0	2	0

Slika 45 - Prikaz podataka nakon normalizacije

5.7. Smanjenje podataka

Budući da se naš dataset sastoji od oko 4000 redova, nije bilo potrebe za nikakvim dodatnim izbacivanjem redova, osim izbačenih NaN vrijednosti tokom čišćenja podataka, jer bismo izgubili korisne podatke.

Smanjenje podataka smo izvršili izbacivanjem dvije kolone: *Iznajmljeno* i *url* s obzirom da one nisu relevantne za naš sistem za podršku u procjeni cijene nekretnine za kupovinu.

```
df.drop(['Iznajmljeno'], axis=1)
```

5.8. Razdvajanje skupa podataka na skup za treniranje, testiranje i validaciju

Za potrebe primjene sva tri modela: linearne regresije, random foresta i neuronske mreže, koristili smo podjelu našeg dataseta na trening i testni koristeći omjer 80/20, a validacijski podskup smo kreirali samo kod modela neuronske mreže, i on je iznosio 10%. Način podjele trening i testnog skupa

vidimo

u

kodu

ispod.

Podjela za linearnu regresiju i random forest.

```
from sklearn import linear_model
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn import metrics

x_train1, x_test1, y_train1, y_test1 =
train_test_split(df, df.copy().pop('price'), test_size = 0.2, random_state = 0)
```

Podjela za neuralne mreže:

```
#KerasRegressor for regression problem
estimator = KerasRegressor(build_fn=baseline_model, epochs=120, batch_size=64,
```

```
verbose=2, validation_split=0.1)
```

```
#cross validation
```

```
train, test, train_label, test_label = train_test_split(X,y,test_size=0.2,  
random_state = seed)
```

6. Reference

- [1] - Visit Limsombunchai, 2004, House Price Prediction: Hedonic Price Model vs. Artificial Neural Network Commerce Division, Lincoln University, Canterbury 8150, New Zealand.
https://researcharchive.lincoln.ac.nz/bitstream/handle/10182/5198/House_%20price_%20prediction.pdf?sequence=1&isAllowed=y
- [2] - A. Varma, A. Sarma, S. Doshi, R. Nair, 2018, House Price Prediction Using Machine Learning and Neural Networks, 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT)
<https://ieeexplore.ieee.org/abstract/document/8473231>
- [3] - M. Jain, H. Rajput, N. Garg, P. Chawla, 2020, Prediction of House Pricing using Machine Learning with Python, 2020 International Conference on Electronics and Sustainable Communication Systems (ICESC)
<https://ieeexplore.ieee.org/abstract/document/9155839>
- [4] - S. Pan, J. Zhong, 2019, House Pricing Prediction, University of Rochester, USA
https://www.researchgate.net/profile/Shuaidong-Pan/project/Kaggle-Competition-House-Prices-Advanced-Regression-Techniques/attachment/5c6c9e683843b0544e66ff67/AS:728176909107200@1550622312234/download/Project_Report.pdf
- [5] - J. Kalliola, J. Kapočiūtė-Dzikienė, R. Damaševičius, 2021, Neural network hyperparameter optimization for prediction of real estate prices in Helsinki
<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8064234/>
- [6] - Feng Wang, Yang Zou, Haoyu Zhang, Haodong Shi, 2019, House Price Prediction Approach based on Deep Learning and ARIMA Model, College of Information Science and Engineering Henan University of Technology <https://scihub.se/10.1109/ICCSNT47585.2019.8962443>