EKONOMSKI FAKULTET U SUBOTICI
Univerzitet u Novom Sadu

PRIMENA PYTHON PROGRAMSKOG JEZIKA U REGRESIONOJ ANALIZI TRŽIŠTA NEKRETNINA

ZAVRŠNI RAD

Mentor:

Prof. dr Boris Radovanov

Student:

Ljubiša Danilov

D012/13

Novi Sad, 2022. godina.

Univerzitet u Novom Sadu, Ekonomski fakultet u Subotici

KLjUČNA DOKUMENTACIJSKA INFORMACIJA

Autor, AU:		Ljubiša Danilov
Mentor, MN:		Prof. dr Boris Radovanov
Naslov rada, NR:		Primena Python programskog jezika u regresionoj analizi cena nekretnina
Jezik publikacije, JP:		Srpski
Zemlja publikovanja, ZP:		Srbija
Uže geografsko područje, UGP:		Vojvodina
Godina, GO:		2022.
Mesto i adresa, MA:		Segedinski put 9-11, Subotica
Fizički opis rada, FO: (poglavlja/strana/citata/tabela/slika/grafika/priloga)		Poglavlja: 4 Strana: 41 Citata: 20 Tabela: 2 Slika: 11
Naučna oblast, NO:		Ekonomija
Naučna disciplina, ND:		Ekonometrija
Predmetna odrednica/Ključne reči, PO:		tržiste nekretnina, Python, regresiona analiza, linearna regresija, XGB model
Čuva se, ČU:		Biblioteka Ekonomskog fakulteta u Subotici
Važna napomena, VN:		
Izvod, IZ		Uvod 1. Tržište nekretnina 2. Primena programskih jezika u ekonometrijskoj analizi 3. Metodologija regresione analize 4. Rezultati primene analize na tržištu nekretnina Zaključak Literatura
Datum prihvatanja teme, DP:		2.9.2022.
Datum odbrane, DO:		7.12.2022.
Članovi komisije, KO:	Predsednik, član:	Prof. dr Dragan Stojić
	Mentor, član:	Prof. dr Boris Radovanov

SADRŽAJ

.

UVOD

Nekretnine su imovina koja se kupuje i od strane investitora, ali i od strane fizičkih lica, te zbog toga ovo tržište izaziva interesovanje i praćeno je od strane velikog broja ljudi. U uslovima ekonomske nestabilnosti izazvane pandemijom virusa *Covid-19*, cene nekretnina zabeležile su povećan rast, što može ukazivati na to da investitori smatraju da je tržište nekretnina sigurno za ulaganje kada većina tržišta beleže pad, te bi se nagli skok cena nekretnina mogao povezati i sa preusmeravanjem investicija na tržište nekretnina. Zbog svega navedenog veoma je važno analizirati tržište nekretnina, odnosno faktore koji utiču na kretanje cena, odrediti način na koji utiču na cene, kao i njihovu značajnost pri formiranju cena nekretnina na tržištu.

U ovom radu biće analizirano tržište nekretnina, odnosno kratak istorijat ovog tržišta, kretanje cena, kao i faktori koji utiču na cene. Nakon toga biće sagledana primena programskih jezika u ekonometrijskoj analizi, odnosno prednosti *Python* programskog jezika. Na kraju rada, biće analizirano tržište nekretnina primenom regresione analize u *Python* programskom jeziku. Regresiona analiza obuhvata veliki broj modela kao što su linearna regresija, polinomska regresija, *Ridge* regresija, *Lasso* regresija itd. Od tradicionalnih regresionih modela biće primenjena linearna regresija, a pored nje biće primenjeni i napredniji oblici regresije, odnosno *Decision Tree*, *Random Forest*, *Extra Trees* i *Extreme Gradient Boosting* algoritmi za regresionu analizu. Nakon analize biće izvršena komparacija moći predviđanja ispitivanih modela.

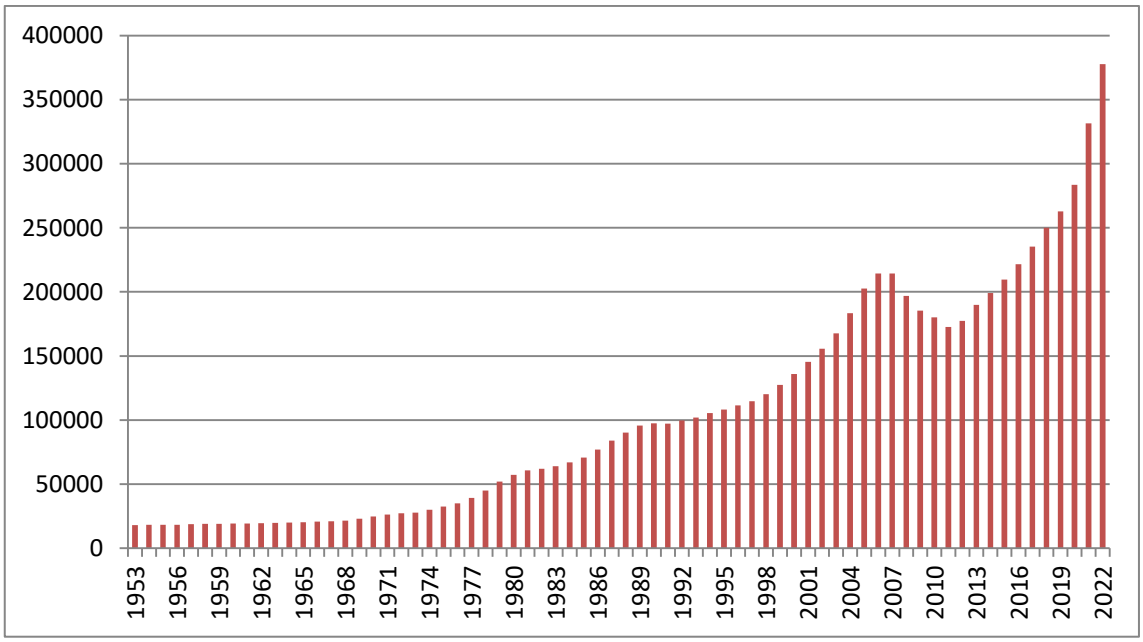
Obzirom da se za statističke analize u većini slučajeva koriste statistički softverski paketi jer imaju korisnički interfejs i jednostavniji su za upotrebu u poređenju sa programskim jezicima, ciljevi ovog rada su prikazivanje prednosti i mogućnosti *Python* programskog jezika pri obradi i analizi podataka, kao i poređenje performansi, odnosno moći predviđanja različitih regresionih modela.

1. Tržište nekretnina

Tržište nekretnina ne može da postoji bez koncepta vlasništva nad nekretninama. Potpisivanjem prvog engleskog ustavnog dokumenta *Magna Carta* 1215. godine Engleska uspostavlja koncept vlasništva nad nekretninama što predstavlja i početak postojanja tržišta nekretnina. Definicija vlasništva nad nekretninama u to vreme bila je sledeća: „Osoba poseduje imovinu, on je kralj njegovog dvorca, ona je kraljica svoje zemlje“ (Solis-Nekretnine, 2017).

Od perioda nastanka tržišta pa do danas tržište nekretnina neprestano raste i razvija se. U 2021. godini dostiglo je vrednost od 3,69 triliona dolara na svetskom nivou, dok je u Sjedinjenim Američkim Državama vrednost ovog tržišta dostigla 699,9 milijardi dolara (Grand View Research, 2022).

U prilog tome govore i podaci o prosečnim cenama nekretnina u Sjedinjenim Američkim Državama u periodu od 1953-2022. godine (DQYDJ, 2022) prikazani na slici 1. Rast je posebno izražen u periodu posle svetskog ekonomskog kraha 2008. godine. Takođe, poslednjih nekoliko godina cene nekretnina su izrazito porasle iako je pandemija virusa *Covid-19* ostavila teške posledice na ekonomiju, što može da bude pokazatelj da u nestabilnim uslovima investitori vide tržište nekretnina kao sigurnu luku za svoj novac.



Slika 1: Prosečne cene nekretnina izražene u dolarima u Sjedinjenim Američkim Državama u periodu od 1953-2022. godine
Izvor: autor, na osnovu Historical US Home Prices: Monthly Median from 1953-2022 (dqydj.com)

Brojni su faktori koji utiču na cenu nekretnine. Faktori kao što su veličina i lokacija nekretnine su univerzalni i uvek značajni, međutim postoji još mnogo faktora koje je neophodno analizirati i koji se razlikuju u zavisnosti od države, grada, razvijenosti industrije u okolini itd. U ovom radu analizirane su cene nekretnina u Bostonu (Harrison D. & Rubinfeld D. L., 1978) koje su označene sa *medv* i izražene u hiljadama dolara, a faktori koji su uzeti u obzir su:

- *crim* – stopa kriminala po stanovniku;
- *zn* – zona za placeve veće od 2300 m^2;
- *indus* – udeo zemljišta namenjenog za industriju (izraženo u *acres* jedinici, 1 *acre cca*. 4000 m^2);
- *chas* – Charles River varijabla je tzv. *dummy variable* koja se u našoj literaturi prevodi kao veštačka promenljiva koja može da uzme vrednosti 1 i 0 (ukoliko se plac nalazi pored reke dodeljuje mu se 1, dok se 0 dodeljuje kada to nije slučaj);
- *nox* – koncentracija azotnih oksida (udeo čestica azotnih oksida u deset miliona svih drugih čestica koje se nalaze u vazduhu);
- *rm* – broj soba u stambenoj jedinici;
- *age* – proporcija stambenih jedinica izgrađenih pre 1940. godine;
- *dis* – ponderisano rastojanje od poslovnih oblasti u Bostonu;
- *rad* – pristup auto-putevima;
- *tax* – stopa poreza na nekretnine;
- *ptratio* – odnos broja učenika i učitelja;
- *black* – udeo Afričkih Amerikanaca u ukupnoj populaciji u posmatranom delu grada;
- *lstat* – procenat populacije sa nižim ekonomskim statusom;

2. Primena programskih jezika u ekonometrijskoj analizi

Ekonometrija je oblast ekonomije gde se statističke i matematičke metode koriste za analizu ekonomskih podataka (CFI Team, 2021). Termin *ekonometrija* je prvi put upotrebljen od strane poljskog ekonomiste Pawel Ciomp-a 1910. godine (Israel, 2016). Termin dobija na značaju i postaje priznat širom sveta tek nakon rada objavljenog od strane Ragnar Frisch-a i Jan Tinberg-a na temu razvoja i primene dinamičkih modela za analizu ekonomskih procesa za koji su dobili prvu Nobelovu nagradu dodeljenu u oblasti ekonomije 1969. godine.

Ekonometrija se koristi za razvoj teorija ili testiranje hipoteza u ekonomiji, kao i za predviđanje budućih trendova na osnovu postojećih istorijskih podataka. U zavisnosti od toga da li se koristi za testiranje teorijskih pretpostavki ili razvoj novih hipoteza, ekonometrija se može podeliti na (Hayes, 2022):

- teorijsku i
- primenjenu

Postoji širok spektar alata koji se koriste za sprovođenje ekonometrijske analize. Grubo, oni se mogu podeliti na:

- softverske pakete i
- programske jezike.

Softverski paketi predstavljaju kolekciju softverskih rutina i pridruženih informacija koje imaju zajednički interfejs, a čija svrha je da direktno doprinese generisanju neke vrste statističke analize, uključujući pri tome i izvršenje drugih pomoćnih zadataka, kao što je upravljanje podacima. Neki od softverskih paketa za ekonometrijsku analizu su: *Statistical Package of the Social Sciences* (*SPSS – IBM*, Sjedinjene Američke Države), *Statistical Analysis System* (*SAS – SAS Institute*, Sjedinjene Američke Države), *Stata* (*Stata Corp*, Sjedinjene Američke Države), *Regression Analysis of Time Series* (*RATS – Estima Inc*, Sjedinjene Američke Države), *Develve* (*Develve*, Sjedinjene Američke Države), *Minitab* (*Minitab LLC*, Sjedinjene Američke Države) itd. (Pat Research, 2015)

Programski jezici su jezici koji se koriste za pripremanje računarskih programa. Sa aspekta ekonometrijske analize, izbor nije toliko raznovrstan kao kod softverskih paketa, jer je dostupnih programskih jezika mnogo manje nego softvera. Za ekonometrijsku analizu mogu da se koriste: *Python, R, Scala, Java, C++, Julia, Java Script* itd. Svi prethodno navedeni programski jezici su veoma moćan alat za eknometrijsku analizu, a ubedljivo najpopularniji su *R* i *Python*.

Programski jezik *R* je napravljen za statistiku. Dizajniranje i razvoj ovog programskog jezika otpočeto je 1992. godine, a prva verzija lansirana je 1995. godine. Veoma je popularan u naučnim zajednicama jer je besplatan za preuzimanje i korišćenje, poseduje pakete za transformaciju sirovih podataka u struktuirane informacije *(data-wrangling)*, vizuelizaciju, podržava mnoštvo statističkih modela, omogućava mašinsko učenje itd. (Data Science Nerd).

Programski jezik *Python* je kreiran 1991. godine od strane holandskog programera Guido van Rossum-a i baziran je na programskom jeziku *ABC* (*Python (programming language)*, Wikipedia). Koristi se za razvoj veb aplikacija, u oblasti veštačke inteligencije, robotike, za statističku obradu podataka, itd. Dugi niz godina *R* je bio prvi izbor kada je reč o *Data Science* oblasti, međutim *Python* je u poslednjih nekoliko godina preuzeo primat i važi za najbolji alat za statističku obradu podataka (Chiluka, 2022). Prednosti *Python* programskog jezika u odnosu na ostale koji se koriste u *Data Science* oblasti su sledeće (Sharma, 2019):

- *Jednostavnost* – sintaksa programskog jezika je vrlo jednostavna i čitljiva, te je kriva učenja mnogo kraća u odnosu na ostale programske jezike, pa i *R*;
- *Biblioteke* – poseduje više od 137.000 biblioteka koje olakšavaju rad;
- *Podržava više paradigmi programiranja* – programska paradigma određuje stil programiranja, odnosno perspektivu programera u odnosu na program i njegovo izvršavanje. Na primer, u objektnom programiranju programer razmišlja o programu kao o skupu objekata, dok u proceduralnom programiranju razmišlja o redosledu naredbi. *Python* podržava funkcionalno, proceduralno i objektno orijentisano programiranje;
- *Enterprise Application Integration (EAI)* – omogućava ugrađivanje koda u aplikacije pisane u drugim programskim jezicima, te je moćan alat za integraciju aplikacija u preduzećima;
- *Jupyter Notebook* – veb aplikacija u okviru koje je moguće pisati kod i izvršiti vizuelizaciju u okviru jednog dokumenta, što je vrlo korisno za statističku obradu podataka;
- *Programerska zajednica* – radi se o zajednici koja broji veliki broj *Python* programera u okviru koje se dele iskustva, u vrlo kratkom periodu se rešavaju problemi vezani za kod, ideju kako napisati program itd.

Svaki softverski paket i programski jezik ima svoje prednosti i nedostatke. U narednoj tabeli prikazano je, generalno posmatrano, koje su to prednosti i nedostaci softverskih paketa s jedne, i programskih jezika s druge strane.

.

Tabela 1: Poređenje softverskih paketa i programskih jezika

	Softverski paketi	Programski jezici
Prednosti	▪ rasprostranjenost u industriji i na univerzitetima ▪ korisnički interfejs lak za korišćenje ▪ korisnička podrška od strane vlasnika licence za softver	▪ besplatni za preuzimanje i korišćenje ▪ velika baza ljudi koja konstantno unapređuje i kreira nove biblioteke koda koje olakšavaju analizu ▪ prilagođavanje trendovima (mogućnost kreiranja novih funkcija, modela...)
Nedostaci	▪ visoke cene licenci koje se moraju obnavljati ▪ sporo adaptiranje novim trendovima analize ▪ limitiranost ugrađenim funkcionalnostima softvera	▪ nema korisničkog interfejsa ▪ nema korisničke podrške ▪ neophodno vreme za upoznavanje sa sintaksom programskog jezika i osnovama programiranja

Izvor: autor

3. Metodologija regresione analize

Regresija je metod kojim se ispituje zavisnost između dve ili više promenljivih, odnosno pojava. Cilj regresije jeste da se odredi onaj regresioni model koji najbolje opisuje vezu između pojava i da se na osnovu tog modela ocene i predvide vrednosti zavisne promenljive *(Y)* za odabrane vrednosti nezavisnih promenljivih *(X)*. U ovom radu na osnovu regresione analize tržišta nekretnina možemo da saznamo:

- za koliko dodatna soba (*rm*) povećava cenu nekretnine (*medv*),
- za koliko stopa kriminala (*crim*) smanjuje cenu nekretnine,
- u kojoj meri i sa kolikom značajnošću odnos broja učenika i učitelja (*ptratio*) ili bilo koji drugi regresor utiče na cenu nekretnine itd.

Analiza tržišta nekretnina izvršena je pomoću više regresionih modela u okruženju *Jupyter Notebook* putem programskog jezika *Python*. Podaci su najpre učitani i pregledani radi dobijanja osnovnih informacija, zatim je izvršena vizuelizacija podataka, linearna regresija, primena naprednih regresionih metoda i komparacija regresionih modela.

3.1. Učitavanje podataka i biblioteka i pregled podataka

Uvoz odgovarajućih biblioteka gotovog koda naredbom *import* zarad brže i jednostavnije analize i učitavanje seta podataka bili su prvi koraci u analizi. Nakon toga, funkcijom *head*, dobijen je tabelarni prikaz zaglavlja i prvih pet redova podataka sa ciljem da se izvrši uvid u to da li su se podaci učitali na pravi način, da li su sve kolone neophodne u daljoj obradi podataka i da tabela ne zauzme mnogo prostora jer nije neophodno da se učita svaki podatak. Metoda *info* iskorišćena je sa ciljem da se dobije pregled podataka sa aspekta da li ima praznih ćelija u tabeli i kog su tipa podaci. Metodom *describe* dobijen je numerički sažetak uzorka po kolonama koji sadrži rezultate sledećih statistika:

- count: broj ćelija koje nisu prazne,
- mean: prosečna vrednost,
- std: standardna devijacija,
- min: minimalna, odnosno najmanja vrednost,
- 25%: 25% podataka date kolone je manje od vrednosti prikazane u koloni,
- 50%: 50% podataka date kolone je manje od vrednosti prikazane u koloni, dakle, ta vrednost predsavlja medijanu, odnosno medijalnu vrednost,

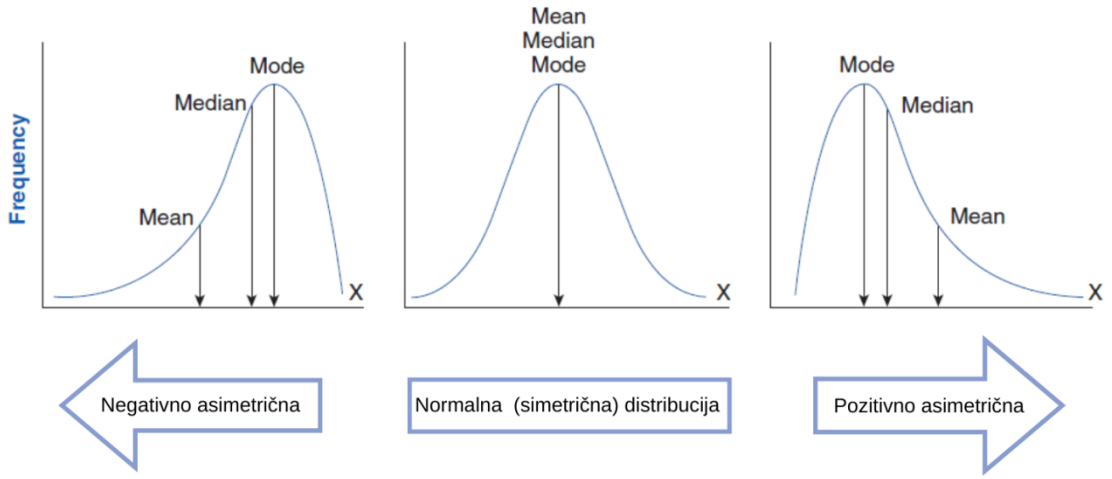
- 75%: 75% podataka date kolone je manje od vrednosti prikazane u koloni,
- max: maksimalna vrednost.

3.2. Vizuelizacija podataka

Za vizuelizaciju uzorka korišćeni su *Box-plot* dijagram, *Dist-plot* dijagram, dijagram rasturanja (disperzije) i korelaciona matrica *(Heat-map).*

Na *Box-plo*t dijagramu pravougaonikom je ilustrovan interkvartilni raspon uzorka koji predstavlja razliku između trećeg i prvog kvartila uzorka i daje sliku o disperziji uzorka, odnosno varijansi (Matematički fakultet, Univerzitet u Beogradu). Crvene tačke na dijagramu reprezentuju autlajere *(outliers)* koji predstavljaju ekstremne, često pogrešne vrednosti koje bi trebalo proveriti. U ovoj analizi umesto proveravanja svakog autlajera korišćen je veći broj pokazatelja za testiranje modela; testiranje će biti detaljnije objašnjeno u nastavku rada. Horizontalna linija unutar pravougaonika označava medijalnu vrednost.

Na *Dist-plot* dijagramu prikazana je distribucija podataka. Jedna od pretpostavki linearnog modela jeste normalna distribucija (Kiš, Čileg, Vugdelija i Sedlak, 2005). Normalna distribucija podrazumeva distribuciju koja je simetrična, zvonastog oblika, kontinuirana i pravilna. Kod normalne distribucije, tačno u centru, u temenu krivulje, nalaze se aritmetička sredina *(Mean),* medijana *(Median)* i modus *(Mode)* distribucije podataka i dele je na dva jednaka dela, gde se u levom delu distribucije nalaze vrednosti manje od proseka, a u desnom vrednosti veće od proseka (Pi Statistics). Ukoliko podaci ne prate normalnu distribuciju, oni su asimetrični ili vizuelno gledano zakrivljeni, nagnuti na jednu stranu. Kada je distribucija podataka negativno asimetrična, tada se modus, medijana, pa i aritmetička sredina nalaze u zoni većih rezultata, a rep distribucije razvučen je prema manjim vrednostima. Tada važi *Mode* > *Median* > *Mean*. Kod pozitivno asimetrične distribucije je obrnuto, tada se modus, medijana i aritmetička sredina nalaze u zoni manjih rezultata, a rep distribucije razvučen je prema većim vrednostima. Za pozitivo asimetričnu distribuciju važi *Mean* > *Median* > *Mode*. Simetrična i asimetrična distribucija predstavljene su na slici 2.



Slika 2: Simetrična i asimetrična distribucija podataka
Izvor: https://pistatistics.com/kurs/deskriptivna-statistika/lekcije/asimetricnost-i-homogenost-distribucije/

Obzirom da su pojedine promenljive u uzorku imale vrednosti na x osi koje su značajno veće nego kod ostalih promenljivih, izvršena je min-max normalizacija kako model ne bi favorizovao promenljive sa većim vrednostima, odnosno da bi tretirao sve promenljive podjednako.

Dijagram rasturanja sa regresionom linijom pruža informacije o disperziji podataka, zatim obliku, smeru i jačini veze. Oblik veze nam kazuje da li su podaci linearnog ili pak nekog drugog oblika spram njihove disperzije. Smer veze može biti pozitivan ili negativan u zavisnosti od toga da li su tačke grupisane iz donjeg levog ka gornjem desnom uglu ili od gornjeg levog ka donjem desnom uglu, respektivno. Da li je veza jaka ili slaba moguće je zaključiti na osnovu odstupanja tačaka od zamišljene prave, gde mala odstupanja reprezentuju jaku vezu, i obrnuto (Kiš, Čileg, Vugdelija i Sedlak, 2005).

Na kraju dela analize vezanog za vizuelizaciju podataka generisana je korelaciona matrica *(Heat-map)* na osnovu koje je moguće analizirati korelacionu vezu između svih varijabli.

3.3. Linearna regresija

Nakon pregleda i vizuelizacije podataka primenjena je linearna regresija metodom običnih najmanjih kvadrata pomoću biblioteke *statsmodel.formula*. Na osnovu rezultata linearne ragresije dobijen je ocenjen model i prokomentarisani su parametri modela. Statistike koje su analizirane su R^2, $\bar{R}^2$, *AIC, P-value* i *Durbin-Watson-ova statistika,* što je objašnjeno u nastavku.

Obzirom da je u linearni model uključeno trinaest regresora, radi se o višestrukoj linearnoj regresiji, čiji model opisuje sledeća jednačina (Kiš, Čileg, Vugdelija i Sedlak, 2005):

$$Y_i = b_1 + b_2X_{i2} + b_3X_{i3} + \cdots + b_kX_{ik} + u_i \quad (1)$$

gde:

- *Y* označava zavisnu promenljivu,
- *X* označava nezavisnu promenljivu,
- *b* označava parametar (koeficijent) uz nezavisnu promenljivu (regresor), pri čemu b_1 predstavlja parametar nivoa i uz njega je uvek *X = 1*, dok ostali parametri predstavljaju parametre nagiba,
- *i* u indeksu označava da se radi o prostornim podacima,
- u_i označava slučajni ili stohastički deo modela (reziduali, greške).

Koeficijent determinacije R^2 izračunat je primenom sledeće jednačine (Kiš, Čileg, Vugdelija i Sedlak, 2005):

$$R^2 = 1 - \frac{\sum e_i^2}{\sum y_i^2} \quad (2)$$

gde:

- $\sum {e_i}^2$ označava neobjašnjene varijacije modela,
- $\sum {y_i}^2$ označava varijacije zavisne promenljive

i pokazuje procenat varijacija zavisne promenljive koji je objašnjen varijacijama nezavisne promenljive, odnosno modelom i izražava se u procentima. Procenat koji fali do 100% predstavlja rezultat delovanja modelom neobuhvaćenih faktora čije je dejstvo uključeno u grešku modela.

Korigovani koeficijent determinacije $\bar{R}^2$ opisan sledećom jednačinom (Kiš, Čileg, Vugdelija i Sedlak, 2005):

$$\bar{R}^2 = 1 - \frac{n-1}{n-k} * (1 - R^2) \quad (3)$$

gde:

- *n* označava broj jedinica posmatranja,

- *k* označava broj parametara modela,
- R^2 označava koeficijent determinacije,

ukazuje na opravdanost proširenja modela dodatnim regresorima. Poredi se sa korigovanim koeficijentom determinacije sa manje/više regresora u istom modelu, pri čemu veća vrednost označava bolji model.

Akaike-ov informacioni kriterijum takođe ukazuje na opravdanost proširenja modela dodatnim regresorima, a formula za njegovo izračunavanje prikazana je u nastavku (Kiš, Čileg, Vugdelija i Sedlak, 2005):

$$AIC = -\frac{2l}{n} + \frac{2k}{n} \tag{4}$$

gde:

- *l* označava funkciju maksimalne verodostojnosti
- *n* označava broj jedinica posmatranja
- *k* označava broj parametara modela

P-value je statistička vrednost pomoću koje zaključujemo da li postoji veza između zavisne i nezavisne promenljive (regresora). Ukoliko regresor ima *P-value* veću od 0,05, to znači da on nije statistički značajan i ne treba ga uključiti u model.

Durbin-Watson-ova statistika testira da li postoji korelacija između reziduala koja bi mogla da invalidira celu regresionu analizu. Vrednost *Durbin-Watson*-ove statistike bi trebalo da bude između 1 i 3, idealno oko 2 (Goss-Sampson, 2019).

3.4. Napredne regresione metode

Linearna regresija metodom najmanjih običnih kvadrata se najčešće koristi kao školski primer regresije. Međutim, ona će retko dati najbolje rezultate, te se u praksi koriste naprednije regresione metode. U ovoj analizi su primenjene sledeće regresione metode:

- *Decision Tree Regressor,*
- *Random Forest Regressor,*
- *Extra Trees Regressor* i
- *Extreme Gradient Boosting (XGB).*

Decision Tree ili stablo odluke je algoritam koji se primenjuje na problemima klasifikacije i u regresionoj analizi. *Decision Tree Regressor* je, kao što se može iz naziva zaključiti, algoritam stabla odluke napisan za regresionu analizu. Kako bi se algoritam razumeo najpre je neophodno razumeti redom (Chelliah, 2021):

- *Root Node:* čvor koji sadrži uzorak za analizu;
- *Branch:* grananje do podčvorova ili konačnih rezultata algoritma *(leaf)*;
- *Leaf/Terminal nodes:* završni čvorovi u kojima se nalaze konačni rezultati algoritma, tj. vrednosti zavisne promenljive.

Kompletan uzorak smešten je *Root Node*-u odakle počinje proces deljenja *(Branching)*. Grananje se vrši binarno u odnosu na uslov i završava se u *Terminal node*-u koji sadrži predikciju zavisne promenljive. Dakle, konstrukcija drveta nastaje idući od makro ka mikro nivou (*top-down* pristup), odnosno od *Root node*-a ka *Terminal Node*-ovima izračunavajući prilikom svakog grananja vrednost na osnovu koje će ono biti izvršeno, odnosno vrednost koja će dati najmanji *Sum of Squares (SS)* – najmanji zbir kvadriranih reziduala (*best-split* podataka). Kada se na osnovu uslova čvor ne može dalje podeliti, on postaje list *(Leaf/Terminal node)* i u njemu je sadržana predikcija zavisne promenljive u vidu numeričke vrednosti. *Decision Tree* je sklon *overfittng*-u, koji se javlja kada model savršeno opisuje skup podataka. To znači da će performirati bez greške na trening podacima, naučiće ih praktično napamet, međutim neće zbog toga moći da obradi na kvalitetan način nove podatke. Postoje načini da se spreči *overfiting* putem korigovanja podrazumevanih vrednosti parametara koji se odnose na način razvoja stabla (*hyperparameter tuning)*, ali to neće biti analizirano u okviru rada. Problem *overfitinga* nastoje da reše modeli predstavljeni u nastavku.

Random Forest Regressor predstavlja „šumu stabala odlučivanja“, odnosno kolekciju stabala odlučivanja, pri čemu se svakom stablu prosleđuje deo uzorka. Algoritam je isti kao za *Decision Tree Regressor* što znači da se grananje stabla i dalje vrši po *best-split* principu, s tim da se trenira više stabala odjednom sa ciljem sprečavanja *overfitting*-a. Predikcija za zavisnu promenljivu u ovom slučaju jeste prosečna vrednost predikcija svih stabala odlučivanja.

Extra Trees Regressor zapravo ima jednu ključnu razliku u odnosu na *Random Forest Regressor*, a to je da se podela podataka ne vrši po *best-split* principu, već je podela nasumična.

Prethodna dva modela funkcionišu na *bagging* principu – paralelno treniranje velikog broja modela. *XGB* funkcioniše drugačije, tačnije na *boosting* principu (Chen, 2019) – treniranje velikog broja modela sekvencijalno, tako da svaki model uči na greškama prethodnog modela, pri čemu se za izračunavanje greške koristi *SS*. Ovo praktično znači da se greška modela u svakoj iteraciji smanjuje i samim tim se moć

predviđanja modela povećava. Konačna predikcija modela u pogledu vrednosti zavisne promenljive je predikcija iz poslednje iteracije.

Pre nego što se u analizi prešlo na napredne regresione metode na uzorku je ponovo primenjena linearna regresija sa razlikom da je korišćena biblioteka *sklearn* jer omogućava trening modela i izračunavanje pokazatelja koji su korišćeni za upoređivanje performansi modela. Kako bi se utvrdilo da su rezultati isti kao kod korišćenja biblioteke *statsmodels*, atributima *intercept_* i *coef_* iz klase *LinearRegression* generisani su parametar nivoa i parametri nagiba, respektivno. Sa ciljem da svi modeli na kraju budu lako uporedivi, performanse su evaluirane pomoću prethodno objašnjenih pokazatelja R^2 *i* $\bar{R}^2$, i pomoću grešaka modela izraženih sledećim pokazateljima: *Mean Absolute Error*, *Mean Squared Error* i *Root Mean Squared Error*. Greška modela ili rezidual predstavlja razliku između predviđene i stvarne vrednosti zavisne promenljive.

Mean Absolute Error (MAE) predstavlja prosečnu vrednost greške u apsolutnom iznosu, kako i sam naziv govori, a računa se kada se suma grešaka u apsolutnom iznosu podeli brojem jedinica posmatranja. Razlog zbog kojeg se koristi apsolutna mera je da se negativne i pozitivne greške ne bi međusobno isključivale. Ovaj pokazatelj je manje osetljiv na autlajere.

Mean Squared Error (MSE) ili prosečna kvadrirana greška računa se kao suma kvadriranih grešaka podeljena brojem jedinica posmatranja. Kvadriranjem grešaka se automatski rešava problem međusobnog isključivanja pozitivnih i negativnih grešaka. Pokazatelj je osetljiv na autlajere jer se greške kvadriraju, što dovodi do toga da veće vrednosti nose i veću težinu.

Root Mean Squared Error (RMSE) dobija se korenovanjem vrednosti pokazatelja *MSE*. Postoji nekoliko razloga zašto korenovati *MSE*. Jedan od razloga jeste olakšana interpretacija pokazatelja. Ukoliko korenujemo, dobijena vrednost je izražena u jedinici kao i zavisna promenljiva (npr. ako je zavisna promenljiva izražena u dolarima, vrednost *RMSE* je takođe izražena u dolarima), dok *MSE* daje kvadratnu razliku u odnosu na zavisnu promenljivu (dobijena vrednost je kvadriran broj izražen u dolarima, te je moramo korenovati kako interpretacija pokazatelja bila smislenija). Vrednost *RMSE* takođe zadržava uticaj većih vrednosti (osetljiva je na autlajere), što se jasno vidi upoređivanjem sa *MAE* koji je uvek manji, osim u ekstremnom slučaju kada je vrednost *MSE* vrednost manja od 0. Pitanje koje se samo nameće jeste zašto su *MSE* i *RMSE* pokazatelji korišćeni kada su osetljivi na autlajere, tj. zašto nije korišćen samo *MAE* pokazatelj? U situcijama kada duplo veća greška ima više nego duplo veće posledice, u tom slučaju veće greške je potrebno i više penalizirati, odnosno dodeliti im veći značaj. Jedan od najlakših načina da se to postigne je kvadriranje grešaka. U analizi cena nekretnina koja sledi, jedan od regresora jeste stopa kriminala *(crim)* – duplo veća

stopa kriminala može da ima više nego duplo veće posledice i samim tim i više nego duplo veći uticaj na cenu nekretnina, zbog čega je bilo važno uključiti i pokazatelje koji većim vrednostima reziduala daju veću težinu.

Koraci analize primenjeni za sve ispitivane modele bili su sledeći:

- podela uzorka na trening i test uzorak;
- trening modela na trening uzorku;
- vizuelizacija *feature importance* pokazatelja – odražava važnost svakog od regresora pri konstruisanju modela;
- predviđanje zavisne promenljive na trening podacima i sagledavanje performansi putem pokazatelja – R^2, $\bar{R}^2$, *MAE, MSE, RMSE;*
- predviđanje zavisne promenljive na test podacima i sagledavanje performansi putem pokazatelja – R^2, $\bar{R}^2$, *MAE, MSE, RMSE.*

Na kraju analize izvršena je komparacija moći predviđanja ispitivanih regresionih modela putem poređenja prethodno navedenih pokazatelja i detaljnije je prokomentarisan model sa najboljim performansama. Za komparaciju su korišćeni rezultati modela na test podacima, odnosno podacima koji nisu prikazani modelima prilikom njihovog treninga.

4. Rezultati primene analize na tržištu nekretnina

U ovom radu analizirane su performanse različitih regresionih modela putem *Python* programskog jezika. Iskorišćeni su podaci o tržištu nekretnina u Bostonu pre svega zbog velikog broja dostupnih varijabli koje utiču na cenu, kao i zbog veličine samog uzorka jer što je broj jedinica posmatranja veći, više podataka imamo na raspolaganju za trening modela.

4.1. Učitavanje podataka i biblioteka i pregled podataka

Biblioteke koje su uvezene naredbom *import* u programski jezik *Python* radi analize podataka su:

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
import warnings
warnings.filterwarnings("ignore")
from statsmodels.formula.api import ols
from sklearn import metrics
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor
from sklearn.ensemble import ExtraTreesRegressor
import xgboost as xgb
```

Funkcijom *read_csv* iz biblioteke *pandas* učitani su podaci u varijablu *df*, dok je funkcijom *head* koja je pozvana nad varijablom *df* dobijeno zaglavlje i prvih pet redova uzorka:

```
df = pd.read_csv("Boston.csv")
df.head()
```

	Unnamed: 0	crim	zn	indus	chas	nox	rm	age	dis	rad	\
0	1	0.00632	18.0	2.31	0	0.538	6.575	65.2	4.0900	1	
1	2	0.02731	0.0	7.07	0	0.469	6.421	78.9	4.9671	2	
2	3	0.02729	0.0	7.07	0	0.469	7.185	61.1	4.9671	2	
3	4	0.03237	0.0	2.18	0	0.458	6.998	45.8	6.0622	3	
4	5	0.06905	0.0	2.18	0	0.458	7.147	54.2	6.0622	3	

	tax	ptratio	black	lstat	medv
0	296	15.3	396.90	4.98	24.0
1	242	17.8	396.90	9.14	21.6
2	242	17.8	392.83	4.03	34.7
3	222	18.7	394.63	2.94	33.4
4	222	18.7	396.90	5.33	36.2

Funkcija *head* je korisna jer ne znamo koliko redova ima fajl sa podacima, a za pregled kolona i podataka suvišno je učitavati sve podatke i usporavati program. Nakon pregleda podataka, metodom *drop* izbačena je kolona *Unnamed: 0* jer ne predstavlja ni zavisnu promenljivu ni regresor, zatim je ponovo učitano prvih pet redova podataka radi provere da li je kolona zaista uklonjena:

```
df.drop(columns=["Unnamed: 0"], axis=0, inplace=True)
df.head()
```

```
      crim    zn  indus  chas    nox     rm   age     dis  rad  tax  ptratio  \
0  0.00632  18.0   2.31     0  0.538  6.575  65.2  4.0900    1  296     15.3
1  0.02731   0.0   7.07     0  0.469  6.421  78.9  4.9671    2  242     17.8
2  0.02729   0.0   7.07     0  0.469  7.185  61.1  4.9671    2  242     17.8
3  0.03237   0.0   2.18     0  0.458  6.998  45.8  6.0622    3  222     18.7
4  0.06905   0.0   2.18     0  0.458  7.147  54.2  6.0622    3  222     18.7

    black  lstat  medv
0  396.90   4.98  24.0
1  396.90   9.14  21.6
2  392.83   4.03  34.7
3  394.63   2.94  33.4
4  396.90   5.33  36.2
```

Metodom *info* dobijene su informacije o podacima sa ciljem utvrđivanja veličine uzorka, da li ima praznih polja u podacima i kojeg su tipa podaci:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 506 entries, 0 to 505
Data columns (total 12 columns):
 #   Column   Non-Null Count  Dtype
---  ------   --------------  -----
 0   crim     506 non-null    float64
 1   zn       506 non-null    float64
 2   chas     506 non-null    int64
 3   nox      506 non-null    float64
 4   rm       506 non-null    float64
 5   dis      506 non-null    float64
 6   rad      506 non-null    int64
 7   tax      506 non-null    float64
 8   ptratio  506 non-null    float64
 9   black    506 non-null    float64
 10  lstat    506 non-null    float64
 11  medv     506 non-null    float64
dtypes: float64(10), int64(2)
memory usage: 47.6 KB
```

RangeIndex pruža informaciju o broju jedinica posmatranja. Kolona *Column* pokazuje zaglavlje uzorka. *Non-null Count* kolona nam pokazuje koliko ima polja koja nisu prazna po koloni, a oznake *int64* i *float64* u okviru kolone *Dtype* ukazuju da se radi o celim i decimalnim brojevima, respektivno.

Metodom *describe* generisane su prethodno opisane statističke informacije:

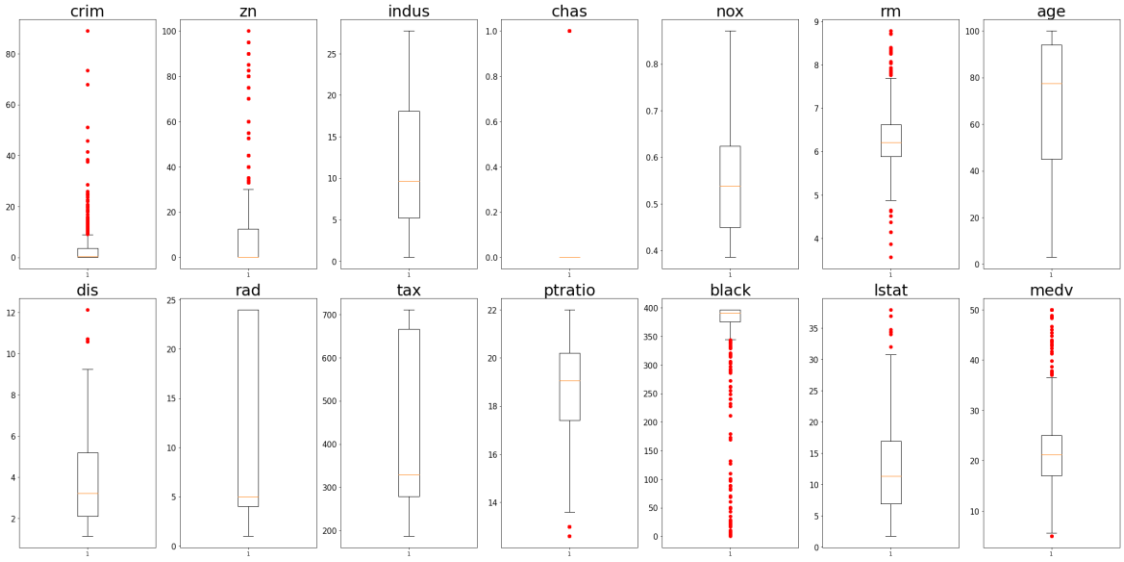
```
df.describe()
```

```
             crim          zn       indus        chas         nox          rm  \
count  506.000000  506.000000  506.000000  506.000000  506.000000  506.000000
mean     3.613524   11.363636   11.136779    0.069170    0.554695    6.284634
std      8.601545   23.322453    6.860353    0.253994    0.115878    0.702617
min      0.006320    0.000000    0.460000    0.000000    0.385000    3.561000
25%      0.082045    0.000000    5.190000    0.000000    0.449000    5.885500
50%      0.256510    0.000000    9.690000    0.000000    0.538000    6.208500
75%      3.677083   12.500000   18.100000    0.000000    0.624000    6.623500
max     88.976200  100.000000   27.740000    1.000000    0.871000    8.780000

              age         dis         rad         tax     ptratio       black  \
count  506.000000  506.000000  506.000000  506.000000  506.000000  506.000000
mean    68.574901    3.795043    9.549407  408.237154   18.455534  356.674032
std     28.148861    2.105710    8.707259  168.537116    2.164946   91.294864
min      2.900000    1.129600    1.000000  187.000000   12.600000    0.320000
25%     45.025000    2.100175    4.000000  279.000000   17.400000  375.377500
50%     77.500000    3.207450    5.000000  330.000000   19.050000  391.440000
75%     94.075000    5.188425   24.000000  666.000000   20.200000  396.225000
max    100.000000   12.126500   24.000000  711.000000   22.000000  396.900000

            lstat        medv
count  506.000000  506.000000
mean    12.653063   22.532806
std      7.141062    9.197104
min      1.730000    5.000000
25%      6.950000   17.025000
50%     11.360000   21.200000
75%     16.955000   25.000000
max     37.970000   50.000000
```

4.2. Vizuelizacija podataka

Pomoću biblioteke *matplotlib* generisana je matrica *Box-plot* dijagrama, a ključna metoda za to je funkcija *boxplot*. Crveni kružići na dijagramu predstavljaju autlajere, pravougaonik predstavlja interkvartilni raspon podataka, a narandžasta linija reprezentuje medijalnu vrednost (slika 3).

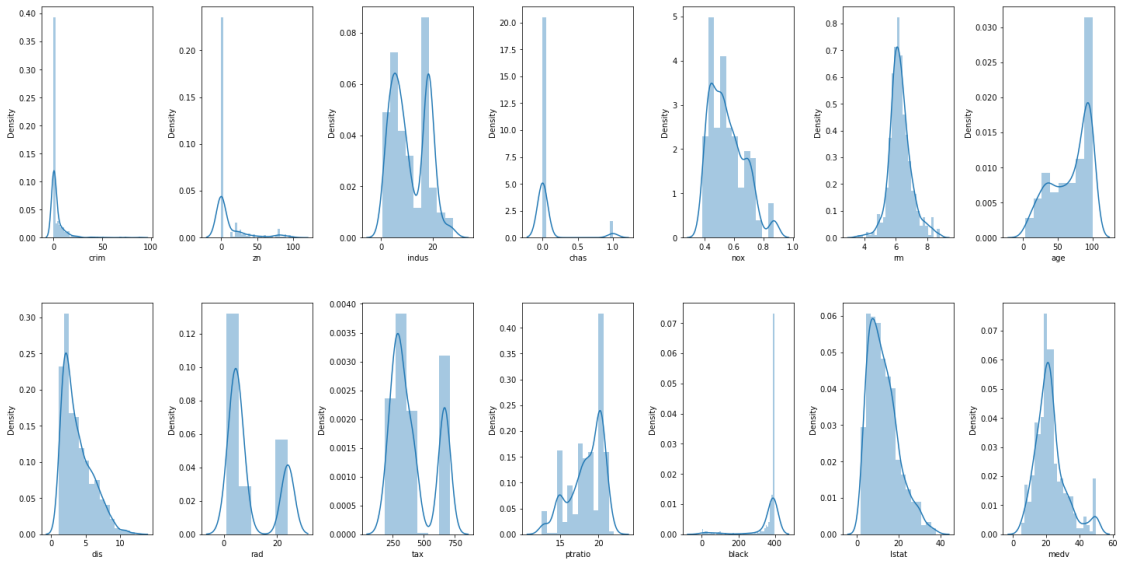
```
red_circle = dict(markerfacecolor="red", marker="o", markeredgecolor="red")

fig, axes = plt.subplots(nrows=2, ncols=7, figsize=(30, 15))
for i, ax in enumerate(axes.flat):
    ax.boxplot(df.iloc[:,i], flierprops=red_circle)
    df.iloc[:,i]
    ax.set_title(df.columns[i], fontsize=30)
    ax.tick_params(axis="y", labelsize=15)

plt.tight_layout()
```



Slika 3: *Box-plot* prikaz analiziranih podataka o tržištu nekretnina
Izvor: autor

Pomoću biblioteka *matplotlib* i *seaborn* generisana je matrica *Dist-plot* dijagrama, odnosno generisana je vizuelizacija distribucije podataka za regresore i zavisnu promenljivu (slika 4).

```
fig, axes = plt.subplots(nrows=2, ncols=7, figsize=(20,10))
index = 0
ax = axes.flatten()
for col, value in df.items():
    sns.distplot(value, ax=ax[index])
    index+=1
plt.tight_layout(pad=0.5, w_pad=0.7, h_pad=5.0)
```



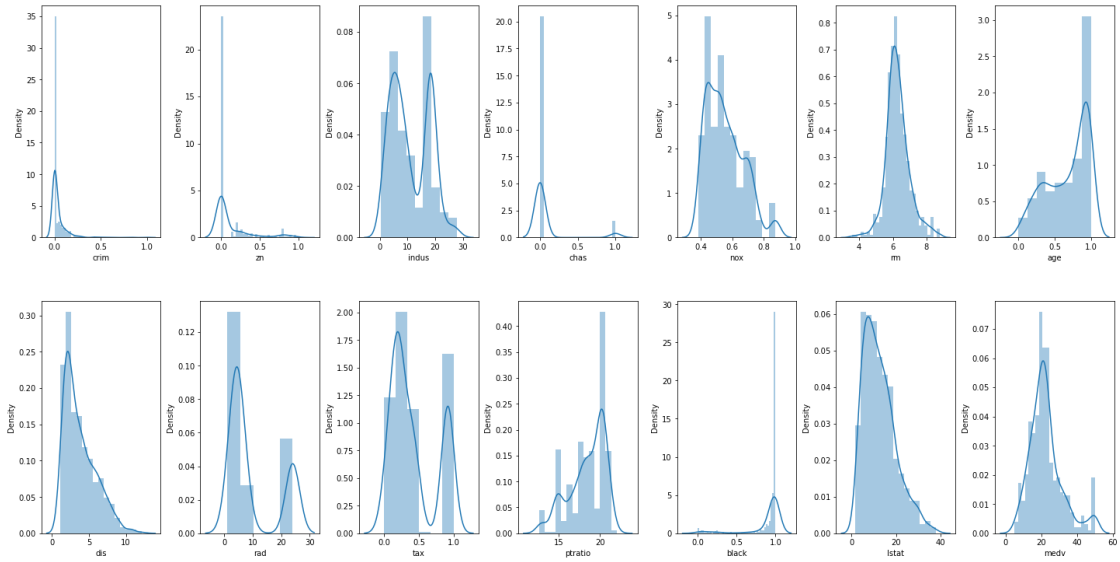
Slika 4: Dist-plot prikaz analiziranih podataka o tržištu nekretnina
Izvor: autor

Veoma je važno da svi regresori budu podjednako uvaženi prilikom modelovanja. Kako regresori sa većim vrednostima ne bi bili protumačeni kao važniji, izvršena je *min-max* normalizacija za promenljive *crim, zn, age, tax black*:

```
cols = ["crim", "zn", "age", "tax", "black"]
for col in cols:
    minimum = min(df[col])
    maximum = max(df[col])
    df[col] = (df[col] - minimum) / (maximum - minimum)
```

Dist-plot nakon *min-max* normalizacije (slika 5):

```
fig, axes = plt.subplots(nrows=2, ncols=7, figsize=(20,10))
index = 0
ax = axes.flatten()

for col, value in df.items():
    sns.distplot(value, ax=ax[index])
    index+=1

plt.tight_layout(pad=0.5, w_pad=0.7, h_pad=5.0)
```



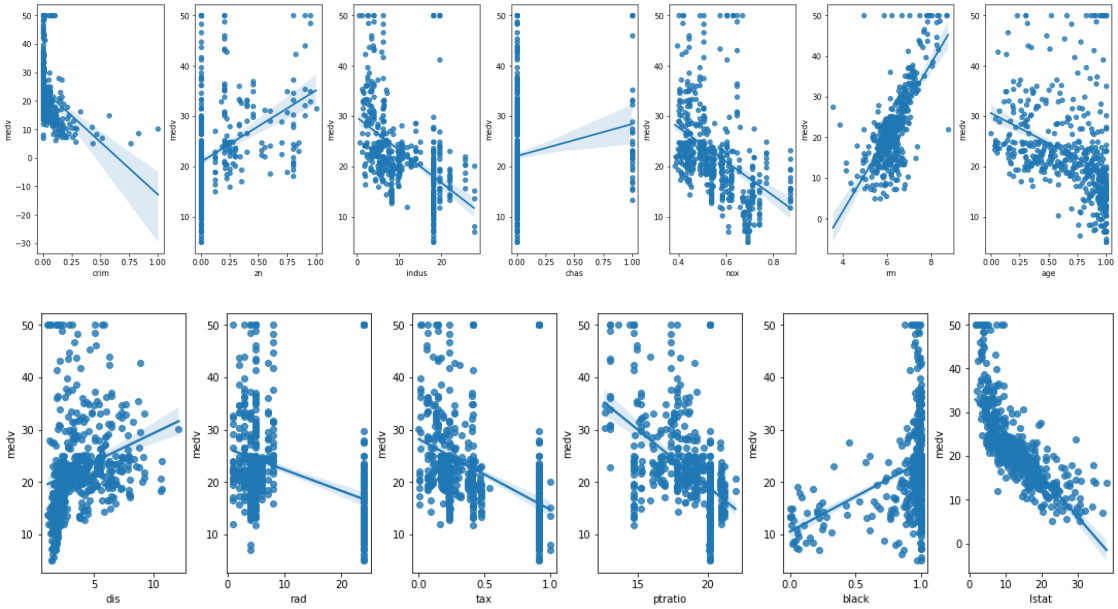
Slika 5: *Dist-plot* prikaz analiziranih podataka o tržištu nekretnina nakon min-max normalizacije
Izvor: autor

Pomoću biblioteka *matplotlib* i *seaborn* generisan je dijagram rasturanja sa regresionom linijom (slika 6):

```
column_names0 = ['crim', 'zn', 'indus', 'chas', 'nox', 'rm', 'age']
x0 = df.drop(["medv"], axis = 1)
y0 = df["medv"]

fig, axes = plt.subplots(ncols=7, nrows=1, figsize=(20, 5))
index = 0
ax = axes.flatten()
```

```
for i, k in enumerate(column_names0):
    sns.regplot(y=y0, x=x0[k], ax=axes[i])
plt.tight_layout(pad=0.4, w_pad=0.5, h_pad=5.0)
```

```
column_names1 = ['dis', 'rad', 'tax', 'ptratio', 'black', 'lstat']
x1 = df.drop(["medv"], axis = 1)
y1 = df["medv"]

fig, axes = plt.subplots(ncols=6, nrows=1, figsize=(15, 4))
index = 0
ax = axes.flatten()
for i, k in enumerate(column_names1):
    sns.regplot(y=y1, x=x1[k], ax=axes[i])
plt.tight_layout(pad=0.3, w_pad=0.4, h_pad=3.0)
```



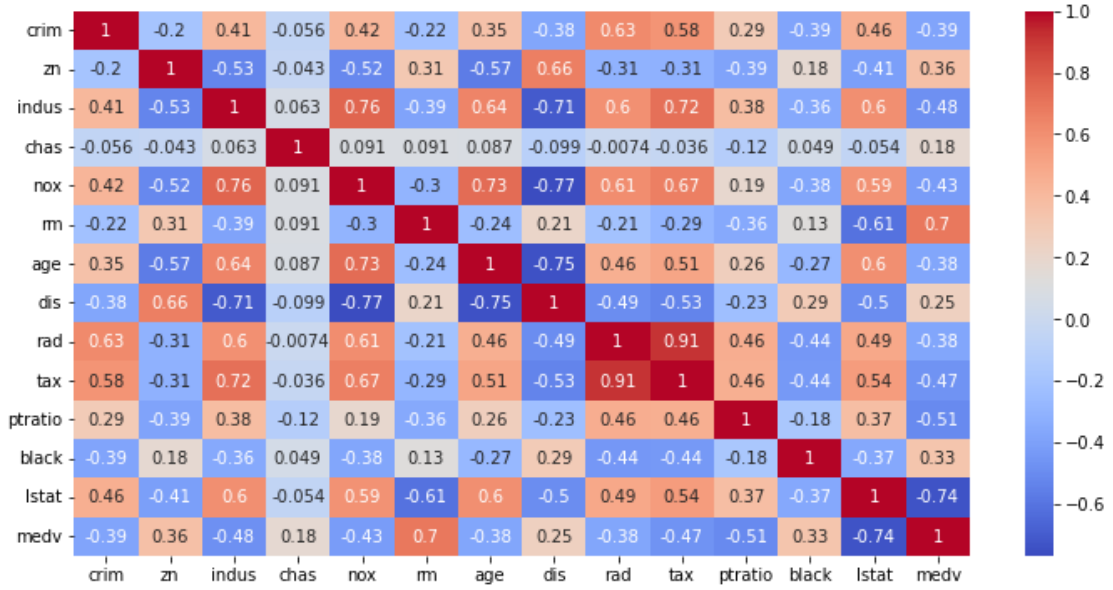
Slika 6: Prikaz analiziranih podataka o tržištu nekretnina u okviru dijagrama rasturanja sa regresionom linijom
Izvor: autor

Plave tačke na dijagramu reprezentuju disperziju podataka, dok plava linija predstavlja regresionu liniju (slika 6). Promenljiva *rm* je primer za pozitivan smer, linearan oblik i jaku vezu zbog malog odstupanja tačaka od prave, dok je promenljiva *lstat* primer za negativan smer, linearan oblik i jaku vezu jer postoje mala odstupanja tačaka od prave.

Iz biblioteke *seaborn* pozvana je funkcija *heatmap* i na taj način kreirana je korelaciona matrica (slika 7):

```
plt.figure(figsize=(12,6))
sns.heatmap(df.corr(), annot = True, cmap = "coolwarm")
```

```
<AxesSubplot:>
```



Slika 7: Korelaciona matrica analiziranih podataka o tržištu nekretnina
Izvor: autor

Na osnovu korelacione matrice vizuelno se može utvrditi korelacija između promenljivih: nijanse crvene boje predstavljaju pozitivnu korelaciju s tim da je svetlijom nijansom reprezentovana slabija, a tamnijom jača korelacija (slika 7). Negativna korelacija predstavljena je nijansama plave boje istom logikom. Vrednost iznad 0,7 ukazuje na jaku korelaciju. U posmatranom slučaju samo promenljiva *lstat* ima jaku negativnu korelaciju, dok jedino *rm* ima jaku pozitivnu korelaciju sa *medv*.

4.3. Linearna regresija

Pomoću biblioteke *statsmodels* izvršena je linearna regresija metodom običnih najmanjih kvadrata *(OLS – Ordinary Least Squares)*:

```
mod0 = ols('medv ~ crim + zn + indus + chas + nox + rm + age + dis + rad + ta
x + ptratio + black + lstat', data = df).fit()

print(mod0.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                   medv   R-squared:                       0.741
Model:                            OLS   Adj. R-squared:                  0.734
Method:                 Least Squares   F-statistic:                     108.1
Date:                Thu, 29 Sep 2022   Prob (F-statistic):          6.72e-135
Time:                        16:49:43   Log-Likelihood:                -1498.8
No. Observations:                 506   AIC:                             3026.
Df Residuals:                     492   BIC:                             3085.
Df Model:                          13
Covariance Type:            nonrobust
==============================================================================
```

```
                coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     34.1572      5.078      6.727      0.000      24.180      44.134
crim          -9.6098      2.924     -3.287      0.001     -15.355      -3.865
zn             4.6420      1.373      3.382      0.001       1.945       7.339
indus          0.0206      0.061      0.334      0.738      -0.100       0.141
chas           2.6867      0.862      3.118      0.002       0.994       4.380
nox          -17.7666      3.820     -4.651      0.000     -25.272     -10.262
rm             3.8099      0.418      9.116      0.000       2.989       4.631
age            0.0672      1.283      0.052      0.958      -2.453       2.587
dis           -1.4756      0.199     -7.398      0.000      -1.867      -1.084
rad            0.3060      0.066      4.613      0.000       0.176       0.436
tax           -6.4633      1.971     -3.280      0.001     -10.335      -2.592
ptratio       -0.9527      0.131     -7.283      0.000      -1.210      -0.696
black          3.6928      1.065      3.467      0.001       1.600       5.786
lstat         -0.5248      0.051    -10.347      0.000      -0.624      -0.425
==============================================================================
Omnibus:                      178.041   Durbin-Watson:                   1.078
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              783.126
Skew:                           1.521   Prob(JB):                    8.84e-171
Kurtosis:                       8.281   Cond. No.                         787.
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
```

Koeficijent determinacije R^2 ima vrednost 0,741 ili 74,1%, što predstavlja objašnjeni deo modela. Korigovani koeficijent determinacije $\bar{R}^2$ ima vrednost 0,734 ili 73,4%. *Akaike*-ov informacioni kriterijum *AIC* iznosi 3026. Na osnovu *P-value* pokazatelja utvrđeno je da regresori *indus* i *age* nisu statistički značajni jer je njihova vrednost iznad 0,05, što znači da nemaju veliki uticaj na formiranje cena nekretnina u posmatranom uzorku. *Durbin-Watson*-ova statistika iznosi 1,078 što znači da nema štetne auto korelacije.

U cilju unapređenja modela usledilo je izbacivanje iz uzorka regresora koji nisu statistički značajni:

```
df.drop(columns=["indus", "age"], axis=1, inplace=True)
df.head()
```

```
       crim    zn  chas    nox     rm     dis  rad       tax  ptratio  \
0  0.000000  0.18     0  0.538  6.575  4.0900    1  0.208015     15.3
1  0.000236  0.00     0  0.469  6.421  4.9671    2  0.104962     17.8
2  0.000236  0.00     0  0.469  7.185  4.9671    2  0.104962     17.8
3  0.000293  0.00     0  0.458  6.998  6.0622    3  0.066794     18.7
4  0.000705  0.00     0  0.458  7.147  6.0622    3  0.066794     18.7

      black  lstat  medv
0  1.000000   4.98  24.0
1  1.000000   9.14  21.6
2  0.989737   4.03  34.7
3  0.994276   2.94  33.4
4  1.000000   5.33  36.2
```

Nakon izbacivanja regresora *indus* i *age* izvršeno je ponovno generisanje rezultata linearne regresije metodom običnih najmanjih kvadrata:

```
mod1 = ols('medv ~ crim + zn + chas + nox + rm + dis + rad + tax + ptratio +
black + lstat', data = df).fit()

print(mod1.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                   medv   R-squared:                       0.741
Model:                            OLS   Adj. R-squared:                  0.735
Method:                 Least Squares   F-statistic:                     128.2
Date:                Thu, 29 Sep 2022   Prob (F-statistic):          5.54e-137
Time:                        16:52:18   Log-Likelihood:                -1498.9
No. Observations:                 506   AIC:                             3022.
Df Residuals:                     494   BIC:                             3072.
Df Model:                          11
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     34.1410      5.048      6.763      0.000      24.222      44.060
crim          -9.6455      2.916     -3.307      0.001     -15.376      -3.915
zn             4.5845      1.352      3.390      0.001       1.928       7.241
chas           2.7187      0.854      3.183      0.002       1.040       4.397
nox          -17.3760      3.535     -4.915      0.000     -24.322     -10.430
rm             3.8016      0.406      9.356      0.000       3.003       4.600
dis           -1.4927      0.186     -8.037      0.000      -1.858      -1.128
rad            0.2996      0.063      4.726      0.000       0.175       0.424
tax           -6.1717      1.767     -3.493      0.001      -9.644      -2.700
ptratio       -0.9465      0.129     -7.334      0.000      -1.200      -0.693
black          3.6846      1.060      3.475      0.001       1.601       5.768
lstat         -0.5226      0.047    -11.019      0.000      -0.616      -0.429
==============================================================================
Omnibus:                      178.430   Durbin-Watson:                   1.078
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              787.785
Skew:                           1.523   Prob(JB):                     8.60e-172
Kurtosis:                       8.300   Cond. No.                         702.
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
```

Na osnovu novih rezultata korigovani koeficijent determinacije se povećao sa 0,734 na 0,735, dok se *AIC* smanjio za 4, što su indikatori da je izbacivanje regresora koji nisu statistički značajni bilo opravdano (Bevans, 2020).

Intercept u *coef* koloni unutar *OLS* tabele predstavlja parametar/koeficijent nivoa, što znači da broj 34,1410 u tabeli označava da cena nekretnina iznosi 34.141$ ako su regresori jednaki nuli. Ostali koeficijenti u *coef* koloni predstavljaju koeficijente regresora koji se redom tumače na sledeći način:

- *crim*: koeficijent -9,6455 pokazuje da se pri povećanju stope kriminala od 1% vrednost nekretnine smanjuje za 9.645,5$ pod uslovom da nema promene ostalih regresora;
- *zn*: pozitivna vrednost koeficijenta pokazuje da povećana proporcija zona za placeve veće od 2300 m^2 ima pozitivnu vezu sa cenom nekretnina;
- *chas*: koeficijent pokazuje razliku u vrednosti nekretnine u zavisnosti od toga da li plac izlazi ili ne izlazi na reku;
- *nox*: negativna vrednost koeficijenta pokazuje da povećanje koncentracije udela čestica azotnih oksida u vazduhu utiče na pad vrednosti nekretnina;
- *rm*: pozitivna vrednost koeficijenta pokazuje da povećanje broja soba ima pozitivnu vezu sa cenom nekretnina;
- *dis*: negativna vrednost koeficijenta pokazuje da veća ponderisana razdaljina od poslovnih oblasti utiče na smanjenje vrednosti nekretnina;
- *rad*: pozitivna vrednost koeficijenta pokazuje pozitivnu vezu između pristupa auto-putu i vrednosti nekretnina;
- *tax*: koeficijent -6,1717 pokazuje da se pri povećanju poreza za 1% cena nekretnine smanjuje za 6.171,7$ pod uslovom da nema promene ostalih regresora;
- *ptratio*: negativna vrednost koeficijenta pokazuje da povećanje odnosa učenik/učitelj ima negativnu vezu sa cenom nekretnina;
- *black*: koeficijent 3,6846 pokazuje da se pri povećanju broja Afričkih Amerikanaca za 1% cena nekretnina povećava za 3.684,6$ pod uslovom da nema promene ostalih regresora;
- *lstat*: koeficijent -0,5225 pokazuje da se pri povećanju populacije nižeg ekonomskog statusa za 1% cene nekretnina smanjuju za 522,5$ pod uslovom da nema promene ostalih regresora.

Kako bi se primenila linearna regresija pomoću *sklearn* biblioteke koja omogućava trening modela, izvršeno je smeštanje vrednosti regresora u varijablu *x* i zavisne promenljive u varijablu *y*:

```
x = df.drop(["medv"], axis = 1)
y = df["medv"]
```

Klasa *LinearRegression* iz biblioteke *sklearn* smeštena je u promenljivu *lm0* i pozivanjem metode *fit* kojoj su kao argumenti prosleđeni *x* (nezavisne promenljive) i *y* (zavisna promenljiva) primenjena je linearna regresija:

```
lm0 = LinearRegression()
lm0.fit(x, y)
```

```
LinearRegression()
```

Vrednosti parametra nivoa i parametara nagiba identični su kao u *OLS* rezultatima. Parametar nivoa dobijen je pomoću atributa *intercept_* koji pripada klasi *LinearRegression*:

```
lm0.intercept_
```

```
34.14095186434517
```

Parametri nagiba, odnosno koeficijenti regresora dobijeni su pomoću atributa *coef_* koji takođe pripada klasi *LinearRegression*:

```
coefficients = pd.DataFrame([x.columns, lm0.coef_]).T
coefficients = coefficients.rename(columns = {0: "Regressors", 1: "Coefficien
ts"})
coefficients
```

```
  Regressors Coefficients
0       crim    -9.645522
1         zn     4.584493
2       chas     2.718716
3        nox   -17.376023
4         rm     3.801579
5        dis    -1.492711
6        rad     0.299608
7        tax    -6.171658
8    ptratio    -0.946525
9      black     3.684563
10     lstat    -0.522553
```

Nakon što je utvrđeno da su parametri dobijeni korišćenjem *statsmodels* i *sklearn* biblioteka isti, izvršena je podela uzorka na trening i test uzorak pomoću *train_test_split* funkcije iz biblioteke *sklearn* radi dalje analize:

```
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.3, ra
ndom_state = 42)
```

Kreiranje i trening modela za predviđanje izvršeno je smeštanjem klase *LinearRegression* u promenljivu *lm1* nad kojom je pozvana metoda *fit* kojoj su kao argumenti prosleđeni *x_train* i *y_train*:

```
lm1 = LinearRegression()
lm1.fit(x_train, y_train)
```

```
LinearRegression()
```

Vrednost parametra nivoa iz trening uzorka različita je od vrednosti iz originalnog uzorka jer je izračunata na manjem uzorku.

```
lm1.intercept_
```

```
30.319226145440005
```

Vrednosti parametara nagiba, odnosno koeficijenti regresora za trening uzorak takođe se razlikuju od koeficijenata dobijenih primenom linearne regresije na ceo uzorak.

```
coefficients = pd.DataFrame([x.columns, lm1.coef_]).T
coefficients = coefficients.rename(columns = {0: "Regressors", 1: "Coefficien
ts"})
coefficients
```

```
  Regressors Coefficients
0       crim   -11.983225
1         zn     3.553453
2       chas     3.167579
3        nox     -15.3744
4         rm     3.950963
5        dis    -1.363138
6        rad     0.232298
7        tax    -3.904776
8    ptratio    -0.907145
9      black     4.602622
10     lstat    -0.556242
```

Predviđanje trening podataka izvršeno je pomoću metode *predict* kojoj je kao argument prosleđen *x_train*, dok je izračunavanje pokazatelja za ocenu kvaliteta modela izvršeno pretežno pomoću modula *metrics* iz biblioteke *sklearn*:

```
y_pred = lm1.predict(x_train)
```

```
r2_train = metrics.r2_score(y_train, y_pred)
print("R^2: ", r2_train)
print("Adjusted R^2: ", 1 - (1 - r2_train) * (len(y_train) - 1) / (len(y_trai
n) - x_train.shape[1]-1))
print("MAE: ", metrics.mean_absolute_error(y_train, y_pred))
print("MSE: ", metrics.mean_squared_error(y_train, y_pred))
print("RMSE: ", np.sqrt(metrics.mean_squared_error(y_train, y_pred)))
```

```
R^2:  0.742828888425931
Adjusted R^2:  0.7345573029659462
MAE:  3.36569175910762
MSE:  22.604448172809246
RMSE:  4.754413546675262
```

Evaluacija modela na test podacima izvršena je pomoću metode *predict* kojoj je kao argument prosleđen *x_test*:

```
y_test_pred = lm1.predict(x_test)
```

```
r2_test = metrics.r2_score(y_test, y_test_pred)
print("R^2: ", r2_test)
print("Adjusted R^2: ", 1 - (1 - r2_test) * (len(y_test) - 1) / (len(y_test)
- x_test.shape[1]-1))
print("MAE: ", metrics.mean_absolute_error(y_test, y_test_pred))
print("MSE: ", metrics.mean_squared_error(y_test, y_test_pred))
print("RMSE: ", np.sqrt(metrics.mean_squared_error(y_test, y_test_pred)))
```

```
R^2:  0.714903934890736
Adjusted R^2:  0.6925035297750082
MAE:  3.113787289886374
MSE:  21.243390345509425
RMSE:  4.6090552552024615
```

4.4. Decision Tree Regressor

Kreiranje i trening modela za predviđanje izvršeno je smeštanjem klase *DecisionTreeRegressor* u promenljivu *dtr* nad kojom je pozvana metoda *fit* kojoj su kao argumenti prosleđeni *x_train* i *y_train*:

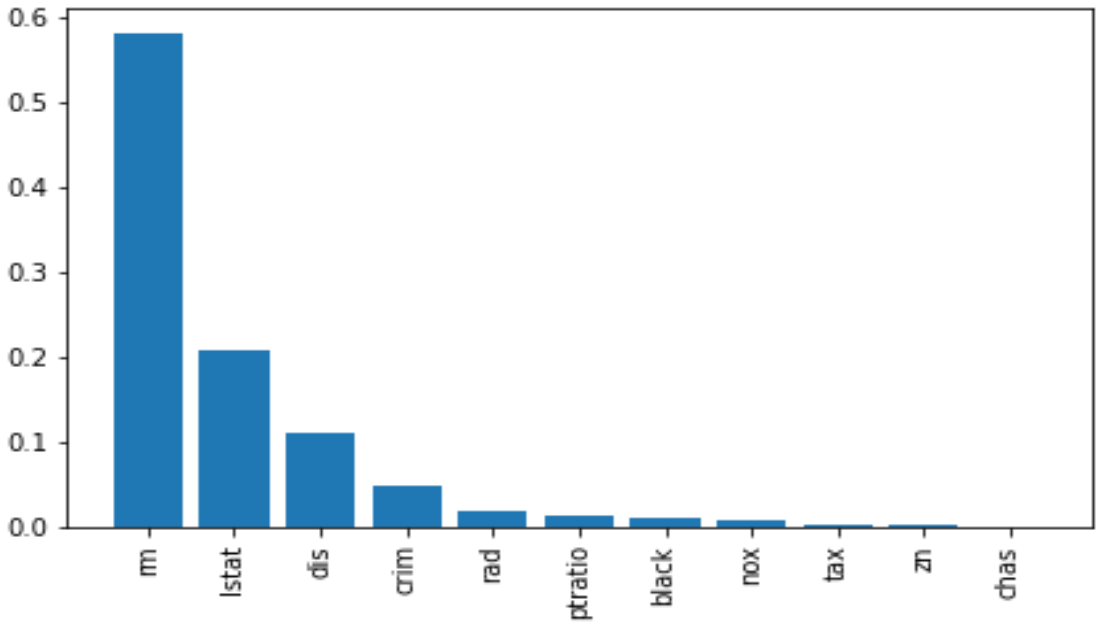
```
dtr = DecisionTreeRegressor()
dtr.fit(x_train, y_train)
```

```
DecisionTreeRegressor()
```

Generisanje grafičkog prikaza značajnosti svakog regresora pri konstruisanju modela *(feature importance)* izvršeno je pomoću atributa *feature_importances_* iz klase *DecisionTreeRegressor* (slika 8):

```
feature_importances_dtr = dtr.feature_importances_
sorted_indices = np.argsort(feature_importances_dtr)[::-1]
plt.bar(range(x_train.shape[1]), feature_importances_dtr[sorted_indices])
plt.xticks(range(x_train.shape[1]), x_train.columns[sorted_indices], rotation
=90)
```

```
plt.tight_layout()
```



Slika 8: *Feature Importance – Decision Tree Regressor*
Izvor: autor

Predviđanje trening podataka izvršeno je pomoću metode *predict* kojoj je kao argument prosleđen *x_train*, zatim je izvršeno izračunavanje pokazatelja za ocenu kvaliteta modela:

```
y_pred = dtr.predict(x_train)
```

```
r2_train = metrics.r2_score(y_train, y_pred)
print("R^2: ", r2_train)
print("Adjusted R^2: ", 1 - (1 - r2_train) * (len(y_train) - 1) / (len(y_trai
n) - x_train.shape[1]-1))
print("MAE: ", metrics.mean_absolute_error(y_train, y_pred))
print("MSE: ", metrics.mean_squared_error(y_train, y_pred))
print("RMSE: ", np.sqrt(metrics.mean_squared_error(y_train, y_pred)))
```

```
R^2:  1.0
Adjusted R^2:  1.0
MAE:  0.0
MSE:  0.0
RMSE:  0.0
```

Evaluacija modela na test podacima izvršena je pomoću metode *predict* kojoj je kao argument prosleđen *x_test*:

```
y_test_pred = dtr.predict(x_test)
```

```
r2_test = metrics.r2_score(y_test, y_test_pred)
print("R^2: ", r2_test)
print("Adjusted R^2: ", 1 - (1 - r2_test) * (len(y_test) - 1) / (len(y_test)
- x_test.shape[1]-1))
print("MAE: ", metrics.mean_absolute_error(y_test, y_test_pred))
print("MSE: ", metrics.mean_squared_error(y_test, y_test_pred))
print("RMSE: ", np.sqrt(metrics.mean_squared_error(y_test, y_test_pred)))
```

```
R^2:  0.838173977301021
Adjusted R^2:  0.8254590755175297
MAE:  2.585526315789474
MSE:  12.058157894736842
RMSE:  3.4724858379461883
```

4.5. Random Forest Regressor

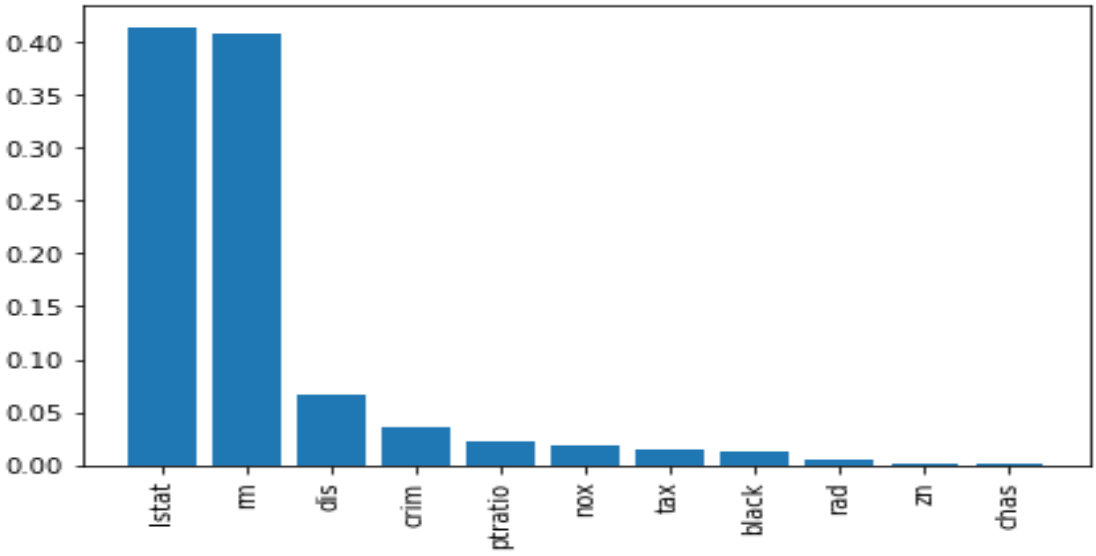
Kreiranje i trening modela za predviđanje izvršeno je smeštanjem klase *RandomForestRegressor* u promenljivu *rfr* nad kojom je pozvana metoda *fit* kojoj su kao argumenti prosleđeni *x_train* i *y_train*:

```
rfr = RandomForestRegressor()
rfr.fit(x_train, y_train)
```

```
RandomForestRegressor()
```

Generisanje grafičkog prikaza značajnosti svakog regresora pri konstruisanju modela *(feature importance)* izvršeno je pomoću atributa *feature_importances_* iz klase *RandomForestRegressor* (slika 9):

```
feature_importances_rfr = rfr.feature_importances_
sorted_indices = np.argsort(feature_importances_rfr)[::-1]
plt.bar(range(x_train.shape[1]), feature_importances_rfr[sorted_indices])
plt.xticks(range(x_train.shape[1]), x_train.columns[sorted_indices], rotation
=90)
plt.tight_layout()
```



Slika 9: *Feature Importance – Random Forest Regressor*
Izvor: autor

Predviđanje trening podataka izvršeno je pomoću metode *predict* kojoj je kao argument prosleđen *x_train*, zatim je izvršeno izračunavanje pokazatelja za ocenu kvaliteta modela:

```
y_pred = rfr.predict(x_train)

r2_train = metrics.r2_score(y_train, y_pred)
print("R^2: ", r2_train)
print("Adjusted R^2: ", 1 - (1 - r2_train) * (len(y_train) - 1) / (len(y_trai
n) - x_train.shape[1]-1))
print("MAE: ", metrics.mean_absolute_error(y_train, y_pred))
print("MSE: ", metrics.mean_squared_error(y_train, y_pred))
print("RMSE: ", np.sqrt(metrics.mean_squared_error(y_train, y_pred)))
```

```
R^2:  0.9771717404370671
Adjusted R^2:  0.976437498170423
MAE:  0.9315734463276821
MSE:  2.0065247881355903
RMSE:  1.4165185449317599
```

Evaluacija modela na test podacima izvršena je pomoću metode *predict* kojoj je kao argument prosleđen *x_test*:

```
y_test_pred = rfr.predict(x_test)

r2_test = metrics.r2_score(y_test, y_test_pred)
print("R^2: ", r2_test)
print("Adjusted R^2: ", 1 - (1 - r2_test) * (len(y_test) - 1) / (len(y_test)
- x_test.shape[1]-1))
print("MAE: ", metrics.mean_absolute_error(y_test, y_test_pred))
print("MSE: ", metrics.mean_squared_error(y_test, y_test_pred))
print("RMSE: ", np.sqrt(metrics.mean_squared_error(y_test, y_test_pred)))
```

```
R^2:  0.8667356898725066
Adjusted R^2:  0.8562649226482035
MAE:  2.139499999999999
MSE:  9.929936276315786
RMSE:  3.1511801402515514
```

4.6. Extra Trees Regressor

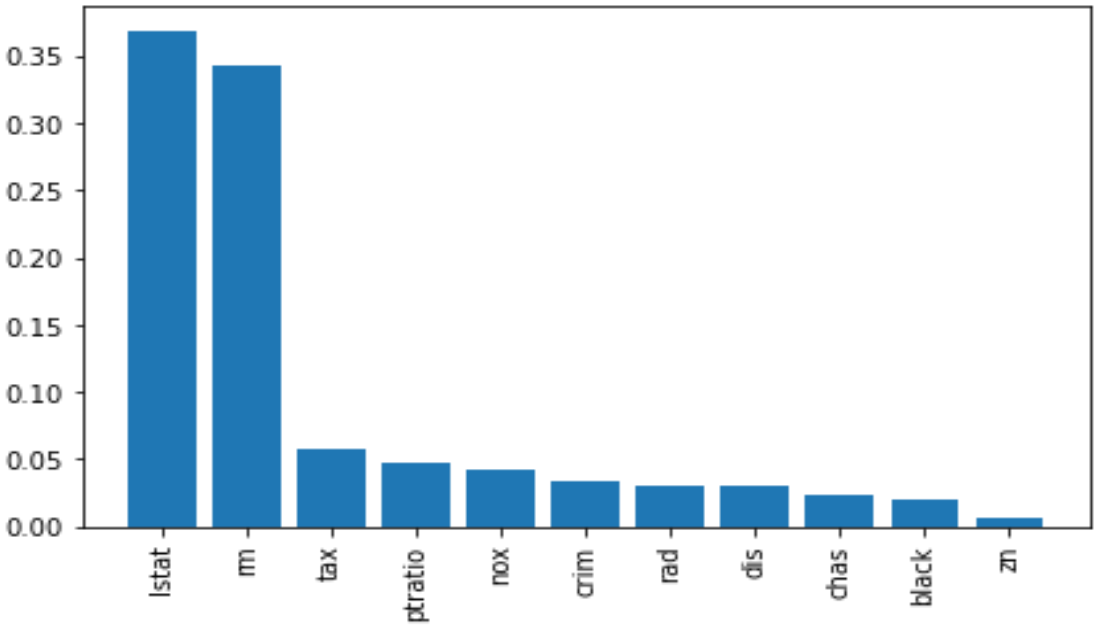
Kreiranje i trening modela za predviđanje izvršeno je smeštanjem klase *ExtraTreesRegressor* u promenljivu *etr* nad kojom je pozvana metoda *fit* kojoj su kao argumenti prosleđeni *x_train* i *y_train*:

```
etr = ExtraTreesRegressor()
etr.fit(x_train, y_train)
```

```
ExtraTreesRegressor()
```

Generisanje grafičkog prikaza značajnosti svakog regresora pri konstruisanju modela *(feature importance)* izvršeno je pomoću atributa *feature_importances_* iz klase *ExtraTreesRegressor* (slika 10):

```
feature_importances_etr = etr.feature_importances_
sorted_indices = np.argsort(feature_importances_etr)[::-1]
plt.bar(range(x_train.shape[1]), feature_importances_etr[sorted_indices])
plt.xticks(range(x_train.shape[1]), x_train.columns[sorted_indices], rotation
=90)
plt.tight_layout()
```



Slika 10: *Feature Importance – Extra Trees Regressor*
Izvor: autor

Predviđanje trening podataka izvršeno je pomoću metode *predict* kojoj je kao argument prosleđen *x_train*, zatim je izvršeno izračunavanje pokazatelja za ocenu kvaliteta modela:

```
y_pred = etr.predict(x_train)
```

```
r2_train = metrics.r2_score(y_train, y_pred)
print("R^2: ", r2_train)
print("Adjusted R^2: ", 1 - (1 - r2_train) * (len(y_train) - 1) / (len(y_trai
n) - x_train.shape[1]-1))
print("MAE: ", metrics.mean_absolute_error(y_train, y_pred))
print("MSE: ", metrics.mean_squared_error(y_train, y_pred))
print("RMSE: ", np.sqrt(metrics.mean_squared_error(y_train, y_pred)))
```

```
R^2:  0.9999999874017251
Adjusted R^2:  0.9999999869965175
MAE:  7.909604522244233e-05
MSE:  1.1073446327679771e-06
RMSE:  0.0010523044392037777
```

Evaluacija modela na test podacima izvršena je pomoću metode *predict* kojoj je kao argument prosleđen *x_test*:

```
y_test_pred = etr.predict(x_test)
```

```
r2_test = metrics.r2_score(y_test, y_test_pred)
print("R^2: ", r2_test)
print("Adjusted R^2: ", 1 - (1 - r2_test) * (len(y_test) - 1) / (len(y_test)
- x_test.shape[1]-1))
print("MAE: ", metrics.mean_absolute_error(y_test, y_test_pred))
```

```
print("MSE: ", metrics.mean_squared_error(y_test, y_test_pred))
print("RMSE: ", np.sqrt(metrics.mean_squared_error(y_test, y_test_pred)))
```

```
R^2:  0.8644196230402899
Adjusted R^2:  0.8537668791363127
MAE:  2.005243421052631
MSE:  10.102513585526316
RMSE:  3.178445152197268
```

4.7. XGB Regressor

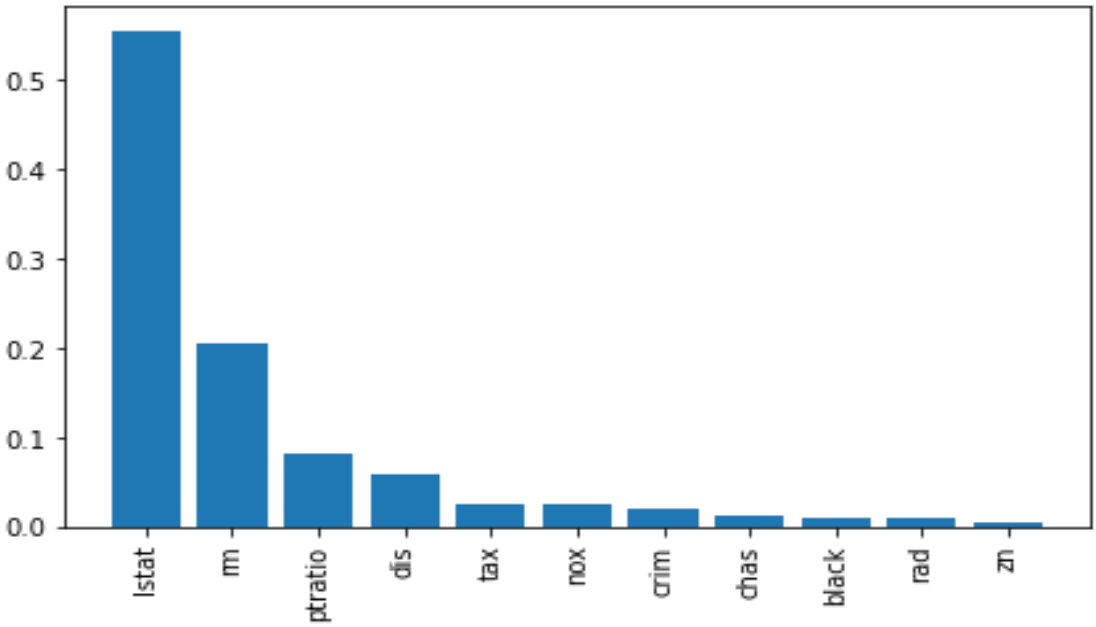
Obzirom da je biblioteka *xgboost* importovana u varijablu *xgb* kreiranje i trening modela za predviđanje izvršeno je pozivanjem klase *XGBRegressor* iz biblioteke importovane kao *xgb (xgb.XGBRegressor)*, zatim je sve smešteno u novu promenljivu *xgb* nad kojom je pozvana metoda *fit* kojoj su kao argumenti prosleđeni *x_train* i *y_train*:

```
xgb = xgb.XGBRegressor()
xgb.fit(x_train, y_train)
```

```
XGBRegressor(base_score=0.5, booster='gbtree', colsample_bylevel=1,
             colsample_bynode=1, colsample_bytree=1, enable_categorical=False,
             gamma=0, gpu_id=-1, importance_type=None,
             interaction_constraints='', learning_rate=0.300000012,
             max_delta_step=0, max_depth=6, min_child_weight=1, missing=nan,
             monotone_constraints='()', n_estimators=100, n_jobs=2,
             num_parallel_tree=1, predictor='auto', random_state=0, reg_alpha=0,
             reg_lambda=1, scale_pos_weight=1, subsample=1, tree_method='exact',
             validate_parameters=1, verbosity=None)
```

Generisanje grafičkog prikaza značajnosti svakog regresora pri konstruisanju modela *(feature importance)* izvršeno je pomoću atributa *feature_importances_* iz klase *XGBRegressor* (slika 11):

```
feature_importances_xgb = xgb.feature_importances_
sorted_indices = np.argsort(feature_importances_xgb)[::-1]
plt.bar(range(x_train.shape[1]), feature_importances_xgb[sorted_indices])
plt.xticks(range(x_train.shape[1]), x_train.columns[sorted_indices], rotation
=90)
plt.tight_layout()
```



Slika 11: *Feature Importance – Extreme Gradient Boosting Regressor*
Izvor: autor

Predviđanje trening podataka izvršeno je pomoću metode *predict* kojoj je kao argument prosleđen *x_train*, zatim je izvršeno izračunavanje pokazatelja za ocenu kvaliteta modela:

```
y_pred = xgb.predict(x_train)

r2_train = metrics.r2_score(y_train, y_pred)
print("R^2: ", r2_train)
print("Adjusted R^2: ", 1 - (1 - r2_train) * (len(y_train) - 1) / (len(y_trai
n) - x_train.shape[1]-1))
print("MAE: ", metrics.mean_absolute_error(y_train, y_pred))
print("MSE: ", metrics.mean_squared_error(y_train, y_pred))
# print("RMSE: ", np.sqrt(metrics.mean_squared_error(y_train, y_pred))
```

Razlog zbog kojeg je linija koda za izračunavanje *RMSE* je pretvorena u komentar (linija koda koja počinje sa #) je taj što je vrednost toliko niska da je program prijavljivao grešku jer nije mogao da izvrši izračunavanje. Obzirom da je prema formuli neophodno vrednost pokazatelja *MSE* 0,00048404827632196476 korenovati, vrednost *RMSE* pokazatelja je približno 0.

```
R^2:  0.9999944929762352
Adjusted R^2:  0.9999943158497399
MAE:  0.015250870602278987
MSE:  0.00048404827632196476
```

Evaluacija modela na test podacima izvršena je pomoću metode *predict* kojoj je kao argument prosleđen *x_test*:

```
y_test_pred = xgb.predict(x_test)

r2_test = metrics.r2_score(y_test, y_test_pred)
print("R^2: ", r2_test)
print("Adjusted R^2: ", 1 - (1 - r2_test) * (len(y_test) - 1) / (len(y_test)
- x_test.shape[1]-1))
print("MAE: ", metrics.mean_absolute_error(y_test, y_test_pred))
print("MSE: ", metrics.mean_squared_error(y_test, y_test_pred))
print("RMSE: ", np.sqrt(metrics.mean_squared_error(y_test, y_test_pred)))
```

```
R^2:  0.8876769112532812
Adjusted R^2:  0.8788515257088961
MAE:  2.0556964748784115
MSE:  8.369541046262274
RMSE:  2.893015908401175
```

4.8. Komparacija regresionih modela

U tabeli su prikazane performanse prethodno analiziranih modela na test uzorku, odnosno uzorku podataka koji su sakriveni od modela prilikom treniranja kako bi se sagledale performanse na nepoznatim podacima i utvrdila moć predviđanja modela.

Tabela 2: Komparacija regresionih modela

Performanse modela	Linearna regresija	Decision Tree	Random Forest	Extra Trees Regressor	XGB
R^2	0,7149	0,8382	0,8667	0,8644	0,8877
$\bar{R}^2$	0,6925	0,8255	0,8563	0,8538	0,8789
MAE	3,1138	2,5855	2,1395	2,0052	2,0557
MSE	21,2434	12,0582	9,9299	10,1025	8,3695
RMSE	4,6091	3,4725	3,1512	3,1784	2,8930

Izvor: autor

Na osnovu tabelarnog prikaza evidentno je da linearni model ne može na najbolji način da opiše skup podataka, samim tim ima vidno lošije rezultate od svih ostalih modela. *Decision Tree* je sklon *overfiting*-u što je uticalo na nešto lošije rezultate u odnosu na *Random Forest* i *Extra Trees Regressor* koji funkcionišu na *bagging* principu. Najbolje rezultate pokazao je *XGB* koji funkcioniše na *boosting* principu.

Kod *XGB* modela vrednost R^2 je bila 0,8877 znači da je modelom objašnjeno 88,77% ukupnih varijacija modela dok je preostalih 11,33% rezultat modelom neobuhvaćenih faktora čije je dejstvo uključeno u grešku modela. *MAE* označava da

greška između stvarnih i predviđenih cena nekretnina iznosi 2.055,7$, dok *RMSE* ukazuje na grešku od 2.893$.

Prilikom konstruisanja *XGB* modela najveći doprinos za predviđanje cene nekretnina imali su regresori *lstat*, *rm* i *ptratio*, respektivno, s tim da je značaj procentualnog udela stanovnika sa nižim ekonomskim statusom *(lstat)* bio dominantan. Najmanji doprinos konstruisanju modela dali su regresori *black*, *rad* i *zn*, respektivno. Značaj svakog regresora *XGB* modela grafički je predstavljen na slici 11.

Zaključak

Programski jezik *Python* ima veoma široku namenu, a svakako je analiza i obrada podataka jedna od najčešćih primena ovog programskog jezika. Kod ekonometrijske analize podataka *Python* pruža mogućnost korišćenja mnoštva biblioteka koda u kojima su smešteni unapred isprogramirani algoritmi statističkih modela koje je moguće menjati, modifikovati, nadograđivati u skladu sa potrebama obrade i analize podataka na relativno jednostavan način, što predstavlja jednu od njegovih najvećih prednosti. Takođe, omogućava i primenu veštačke inteligencije u analizi podataka, što omogućava rezultate analize koji su drugačiji i kvalitetniji od rezultata dobijenih primenom tradicionalnih statističkih modela, uz pretpostavku da je izabran odgovarajući model i da su podaci adekvatno obrađeni.

Nakon kratkog pregleda istorijata i trendova na tržištu nekretnina i sagledavanja upotrebe programskih jezika u ekonometrijskoj analizi sa posebnim osvrtom na programski jezik *Python*, izvršena je regresiona analiza tržišta nekretnina i za to su korišćeni: linearna regresija, *Decision Tree* model, *Extra Trees Regressor* model *Random Forest* model i *Extreme Gradient Boosting (XGB)* model.

Najlošije rezultate sa aspekta koeficijenta determinacije i reziduala svakog od analiziranih modela pokazala je linearna regresija, što je bilo i očekivano jer je u praksi manje verovatno da linearni model može da opiše distribuciju podataka na najbolji mogući način. Sa druge strane, najbolje rezultate u regresionoj analizi tržišta nekretnina, odnosno najveću moć predviđanja cena nekretnina, pokazao je *XGB* model. Primenom ovog modela postignuta je objašnjenost modela od 88,77%, dok greška modela prilikom predviđanja cene varira približno u rangu od 2000 do 3000 dolara u zavisnosti od korišćene metrike.

Obzirom da su prilikom analize podataka korišćene podrazumevane vrednosti parametara koje model uzima u obzir, rezultate analize moguće je dodatno unaprediti korigovanjem vrednosti tih parametara u cilju otključavanja maksimalnog potencijala *XGB* modela, što bi moglo da predstavlja potencijalni nastavak istraživanja iz okvira ovog rada. Takođe, moguće je kreirati sličan uzorak i sprovesti analizu za tržište Srbije ili određeni grad u Srbiji. Koraci analize bi u tom slučaju mogli biti isti, a kako je uzorak koji je korišćen u ovom radu iskrojen za grad Boston i njegovo okruženje, najpre bi bilo neophodno izvršiti reviziju faktora koji utiču na cenu, odnosno zadržati one koji su univerzalni *(crim, rad, tax, ptratio, age itd.)*, izostaviti one koji su specifični za Boston *(zn, indus, chas i dis)* i uvrstiti faktore koji su specifični za oblast za koju bi analiza bila sprovedena.

Literatura

Bevans, R. (2020, 3 26). *Akaike Information Criterion | When & How to Use It (Example)*. Preuzeto sa Scribbr: https://www.scribbr.com/statistics/akaike-information-criterion/

CFI Team (2021, Maj 11). *Econometrics*. Preuzeto sa Corporate Finance Institute: https://corporatefinanceinstitute.com/resources/economics/econometrics/

Chelliah, I. (2021, Maj 7). *Bagging Decision Trees — Clearly Explained*. Preuzeto sa Towards Data Science: https://towardsdatascience.com/bagging-decision-trees-clearly-explained-57d4d19ed2d3

Chen, L. (2019, 1 2). *Basic Ensemble Learning (Random Forest, AdaBoost, Gradient Boosting)- Step by Step Explained*. Preuzeto sa Towards Data Science: Basic Ensemble Learning (Random Forest, AdaBoost, Gradient Boosting)- Step by Step Explained

Chiluka, V. (2022, November 3). *Why is Python the language of choice for data scientists?* Preuzeto sa Tutorials Point: https://www.tutorialspoint.com/why-is-python-the-language-of-choice-for-data-scientists

Why Data Scientists Use R: The Pros, Cons & Alternatives. Preuzeto sa Data Science Nerd: https://datasciencenerd.com/why-data-scientists-use-r/

Goss-Sampson, M. A. (2019). *STATISTICAL ANALYSIS IN JASP: A GUIDE FOR STUDENTS (2nd Edition JASP v0.10.2 July 2019).* Greenwich.

Harrison D., R. D. (1978). Hedonic prices and the demand for clean air. *Journal of Environmental Economics and Management, 5*(1), 81-102.

Hayes, A. (2022, May 12). *Econometrics: Definition, Models, and Methods*. Preuzeto sa Investopedia: https://www.investopedia.com/terms/e/econometrics.asp

Israel, K.-F. (2016, September 1). *Pawel Ciompa and the Meaning of Econometrics: A Comparison of Two Concepts.* Preuzeto sa Papers SSRN: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3288520#:~:text=Abstract%3A%20Pawel%20Ciompa%E2%80%99s%20conception%20of%20econometrics%20is%20compared,prior%20to%20Frisch%2C%20who%20used%20the%20French%20equivalent

Kiš, T., Čileg, M., Vugdelija, D., Sedlak, O. (2005). *Kvantititativni metodi u ekonomiji.* Subotica: Ekonomski fakultet u Subotici.

Matematički fakultet u Beogradu (n.d.). *Matematički fakultet, Univerzitet u Beogradu.* Preuzeto sa Matematički fakultet, Univerzitet u Beogradu: http://www.matf.bg.ac.rs/p/files/42-VISB1cas.pdf

Pat Research (2015, July 8). *Top 48 Statistical Software*. Preuzeto sa Pat Research: https://www.predictiveanalyticstoday.com/top-statistical-software/

Asimetričnost i homogenost distribucije. Preuzeto sa Pi Statistics: https://pistatistics.com/kurs/deskriptivna-statistika/lekcije/asimetricnost-i-homogenost-distribucije/

Pojam normalne distribucije i testiranje normalnosti distribucije. Preuzeto sa Pi Statistics: https://pistatistics.com/kurs/statistika-varijable-podaci/lekcije/normalna-distribucija/

Historical US Home Prices: Monthly Median from 1953-2022 (2022, Avgust). Preuzeto sa DQYDJ: https://dqydj.com/historical-home-prices/

Grand View Research (n.d.). *Real Estate Market Size & Trends Report, 2022-2030*. Preuzeto sa Grand View Research: https://www.grandviewresearch.com/industry-analysis/real-estate-market

Sharma, R. (2019, July 25). *7 Advantages of using Python for Data Science*. Preuzeto sa upGrad: https://www.upgrad.com/blog/advantages-of-using-python-for-data-science/

Solis-Nekretnine (2017, Oktobar 15). *Istorijat nekretnina*. Preuzeto sa SOLIS-NEKRETNINE: https://www.solis-nekretnine.com/blognekretnine/Istorijat-nekretnina-13.html

Python (programming language). Preuzeto sa Wikipedia: https://en.wikipedia.org/wiki/Python_(programming_language)