Melbourne Airbnb Price Prediction

Jana Živković 28/2021

Uroš Paunković 191/2021

1 Uvodni deo

1.1 Opis tržišta kratkoročnog izdavanja nekretnina

Kratkoročno izdavanje nekretnina podrazumeva iznajmljjivanje stambenih objekata kao što su sobe, apartmani ili kuće, koji se izdaju na kraći vremenski period (od jedne noći do par nedelja). Ovakvo tržište ima veliki uticaj na ekonomiju i turizam. Pored očiglednih pogodnosti za turiste – kao što su veći izbor u odnosu na tradicionalne hotele, fleksibilni raspon cena – kratkoročni najam nekretnina ima i šire ekonomske implikacije. Prema izveštaju iz 2023. godine, više od 55% kratkoročnih smeštaja u Evropskoj uniji nalazi se van velikih gradova [2], čime se doprinosi ravnomernijoj regionalnoj distribuciji turističke potrošnje i koristi. Međutim porast popularnosti kratkoročnog izdavanja dovodi do smanjenja ponude dugoročnih stanova, što potencijalno utiče na rast cena nekretnina i zakupnina, naručito u urbanim sredinama.

1.2 Specifičnosti Airbnb-a u Melbourne-u

Airbnb tržište u Melbourne-u karakterišu brojne specifičnosti koje ga izdvajaju u odnosu na druge gradove, kako u Australiji, tako i globalno. Kao jedan od najvažnijih urbanih centara u zemlji, Melbourne je poznat po raznolikosti svoje populacije, bogatom kulturnom životu i velikom broju međunarodnih turista, što direktno utiče na ponudu i potražnju kratkoročnog smeštaja.

Visoka koncentracija ponude u centralnim delovima grada
Najveći broj Airbnb oglasa u Melbourne-u koncentrisan je centralnim četvrtima kao što su Melbourne CBD, Southbank, Carlton i Fitzroy [4]. Ova područja pored turista velikim delom privlače i poslovne putnike zbog blizine glavnih atrakcija, univerziteta, kulturnih centara i glavnih poslovnih zona. Veliki broj stanova u ovim delovima grada funkcionišu isključivo kao kratkoročni najam, što izaziva zabrinutost lokalnih zajednica u vezi sa dostupnošću stanova u svrhu dugoročnog stanovanja.

Regulative i odnos prema kratkoročnom izdavanju
Vlada države Viktorija i grad Melbourne su u više navrata donosili propise kojima su nastojali da regulišu tržište kratkoročnog izdavanja. Na primer, uvedena su ograničenja koja vlasnicima dozvoljavaju da izdaju svoje prostore najviše 180 dana godišnje ukoliko se ne radi o njihovoj adresi stanovanja [1]. Takođe, postoji obaveza registracije i saradnje sa telima koja se bave bezbednošću i komunalnim redom. Sprovođenje ovih propisa u praksi je izazovno, posebno kada se oglasi plasiraju putem više platformi i pod različitim imenima [3].

Uticaj na tržište nekretnina
Rast popularnosti Airbnb-a u Melbourne-u imao je veliki uticaj na lokalno tržište nekretnina. U određenim periodima primećen je trend da vlasnici preusmeravaju stanove sa dugoročnog najma na kratkoročni, što dodatno utiče na već postojeći nedostatak pristupačnih stanova za lokalno stanovništvo [3]. Pored toga, sve više se ulaže u kupovinu stanova namenjenih isključivo za kratkoročno izdavanje, posebno u novijim stambenim kompleksima.

Sezonalnost i događaji
Melbourne je domaćin mnogih međunarodnih događaja poput Austalian Open-a, Formule 1, Grand Prix-a i Melbourne International Comedy festivala. Tokom ovih perioda, potražnja za Airbnb smeštajima naglo raste [8], što omogućava vlasnicima da povećaju cene i samim tim ostvare veću zaradu od izdavanja. Ova sezonalnost čini ovo tržište dinamičnim, ali isto toliko i izuzetno nepredvidivim za korisnike koji traže smeštaj van ovih turističkih sezona.

Profili korisnika i ponude
Airbnb ponuda u Melboure-u je izuzetno raznolika: od pristupačnih soba u deljenim stanovima do tzv. *boutique* apartmana u delovima grada od istorijskog značaja i luksuznih penthausa sa pogledom na zaliv [5]. Tokom godine, postoji znatan broj domaćih turista i ljudi koji preferiraju kućnu atmosferu naspram hotelskog smeštaja.

2 Baza podataka

Baza podataka "Melbourne Airbnb Open Data"pruža detaljne i sažete informacije o aktivnostima Airbnb smeštaja u Melburnu, Australiji. Nalazi se na linku Melbourne Airbnb Open Data na Kaggle. Ovaj folder baza podataka sadrži podatke prikupljene 7. decembra 2018. godine i u njemu postoji nekoliko .csv fajlova, a u daljem radu koristićemo fajl `cleansed_listing_dec18.csv`.

2.1 Problemi

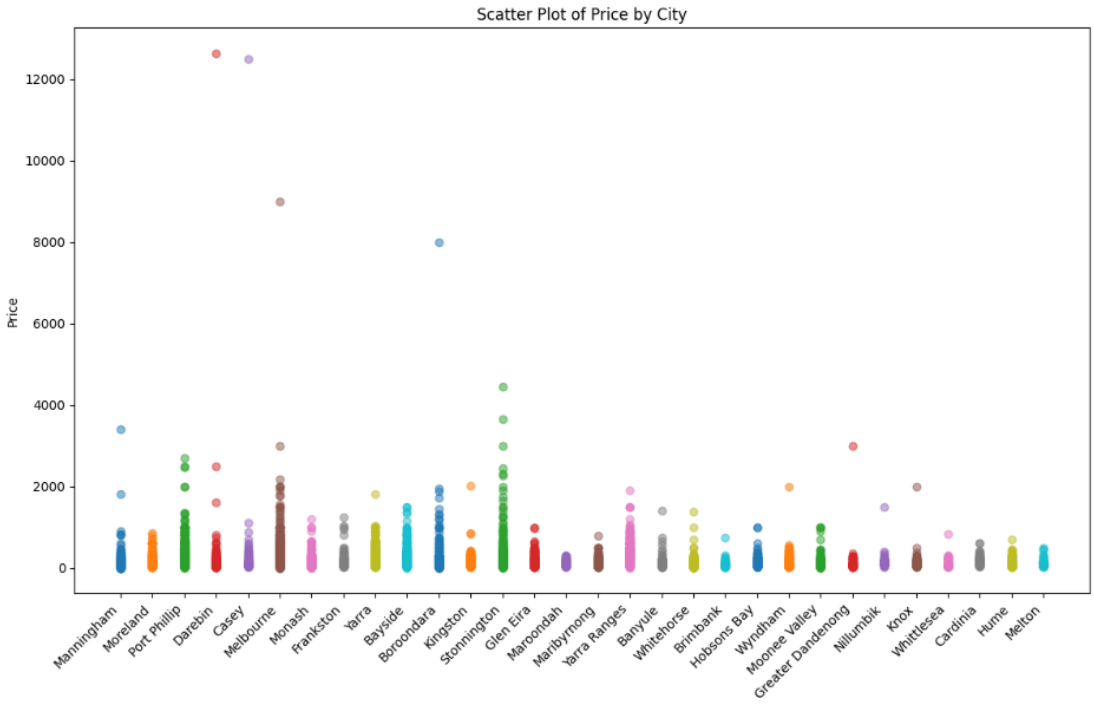
Baza podataka sadrži 84 kolone i 22.895 redova. Uklonjene su sve kolone koje imaju više od 20% nedostajućih podataka, kao i kolone sa specifičnim identifikacionim vrednostima poput *id, host_url* i slično. Nakon uklanjanja ovih kolona, ostalo je 220 redova sa nedostajućim vrednostima, koji su takođe uklonjeni.

Pored navedenih nedostataka u bazi, uočene su i određene ekstremne vrednosti. Uklonjeni su smeštaji čije su cene veće od 1.500$ ili manje od 10$, kao i smeštaji kod kojih je broj ležajeva jednak nuli.

Nakon svih ovih koraka, baza sadrži 43 kolone i 22.498 redova.

2.2 Feature engineering

Feature engineering predstavlja proces stvaranja novih atributa, transformacije postojećih kao i selekcije relevantnih atributa, a sve sa ciljem poboljšanja performansi modela mašinskog učenja. Ovaj korak je ključan jer kvalitet, reprezentativnost i informativnost atributa direktno utiču na tačnost i robusnost (otpornost na promene) modela.
U kontekstu Melbourne Airbnb skupa podataka, ovaj korak omogućava bolje razumevanje podataka koji utiču na cenu kratkoročnog izdavanja smeštaja.
Neke faze ovog procesa su već obrađene u delu Problemi ovog rada: nedosledne i nedostajuće vrednosti, kao



Slika 1: Scatter plot - Cene po gradovima (naseljima)

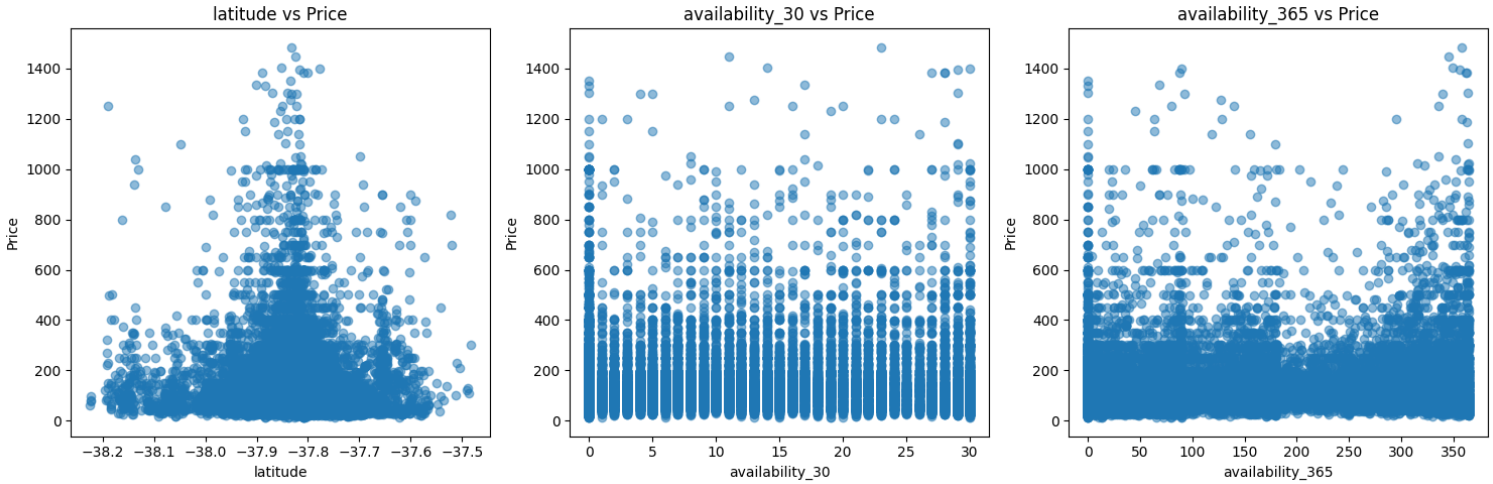
i rukovanje sa autlajerima.

Transformacija numeričkih i kategoričkih podataka
Na neke numeričke promenljive je poželjno primeniti *log-transformaciju* kako bi se smanjila asimetrija raspodele, posebno desna asimetrija. Ovime postižemo sledeće:

1. Smanjenje uticaja autlajera
2. Približili smo se normalnoj raspodeli što može pomoći modelu
3. Olakšavamo modelu da efikasnije nauči odnose između prediktora i ciljne promenljive

Kako naša baza ima objektivno nemali broj kolona, koristićemo sledeći kriterijum: Računaćemo meru asimetrije, poznatiju kao *skewness*. Ukoliko je $|skewness| < 0.5$ podaci su gotovo simetrični i nije potrebna nikakva transformacija. Ukoliko je $|skewness| \geq 1$, tada je asimetrija podataka jaka i poželjna je transformacija. Ukoliko je $1 > |skewness| \geq 0.5$ imamo umerenu asimetriju ali transformacija nije uvek neophodna. U tom slučaju možemo gledati zavisnost između ciljne promenljive (*price*) i trenutne promenljive (slika 2) i na osnovu grafika zaključiti da li je transformacija opravdana ili ne. Transformacije radimo zbog kasnijeg treniranja linearnih modela, konkretno ridge i lasso regresije, jer one podrazumevaju linearnu zavisnost između prediktora i ciljne promenljive. Ukoliko podaci nisu sređeni na ovaj način a postoji realna potreba za time, onda model može biti neefikasan i loše generalizovati i možemo imati uočljivo nesrazmeran uticaj podataka na ocene koeficijenata modela. U ovom radu biće izgrađeni i RF i XGB modeli koji ne zahtevaju ovakve transformacije jer su to modeli zasnovani na stablima, ali ne smeta jer su ti modeli sami po sebi prilično robusni.
Nakon sprovedenog postupka, uočavamo 3 numeričke promenljive čiji je *skew* u opsegu $[0.5, 1)$:

Vidimo da je *latitude* skoncentrisan u vrlo uskom intervalu. Ona predstavlja geografsku koordinatu i nema prirodnu skalu koja zavisi od magnitude kao recimo cena, površina, broj noćenja itd. Transformacija ove

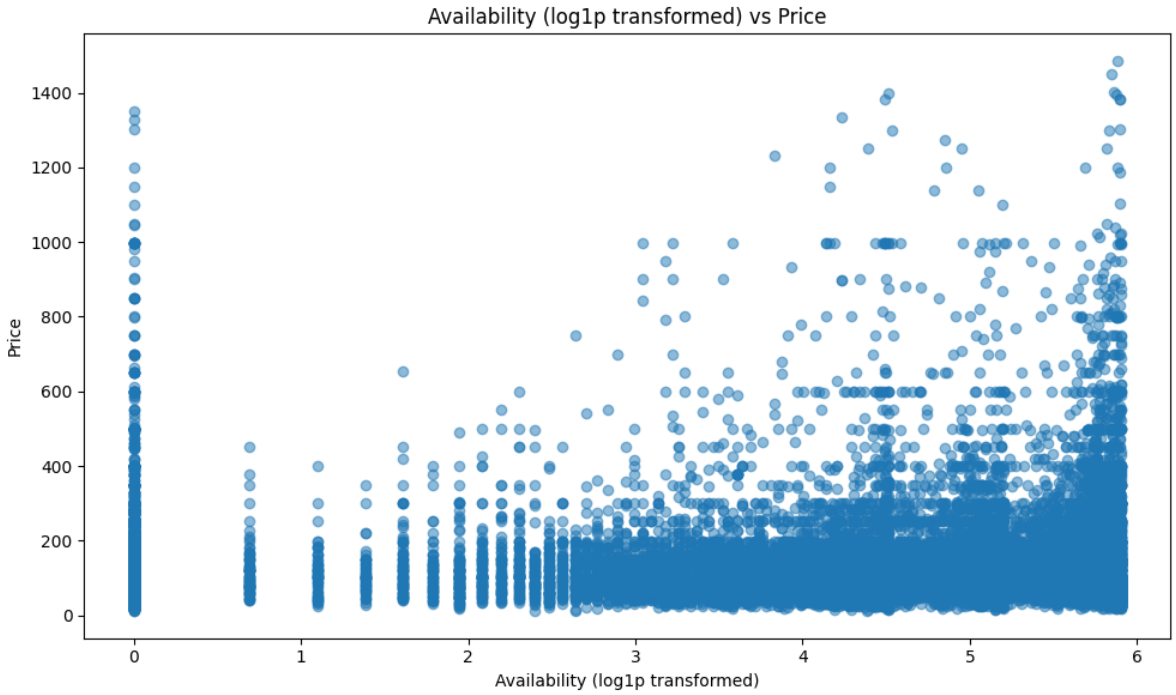


Slika 2: Scatter plots - features vs price

promenljive logaritmom (*log1p*) ne bi imala fizičko kao ni interpretativno značenje, tako da je nećemo transformisati.
Napomena: Ovo ne znači da je primena transformacije pogrešna, samo neće ništa doprineti ako je odradimo. Sa druge strane, *availability30* ima diskretne vrednosti sa malim rasponom (od 0 do 30) pa ne očekujemo korist od log-transformacije jer bi ona samo spljoštila podatke; pritom ovako gledano nema jasne zavisnosti između ove promenljive i targeta a uz to su podaci i *stapled* (mnogo tačaka na istoj vertikali). Za *availability365* vidimo da ima širok opseg (od 0 do 365) diskretnih vrednosti. Vidimo da ima mnogo malih vrednosti i da se raspon širi ka desnoj strani, odnosno uočljiva je desna asimetrija. Ovde bi log-transformacija pomogla jer bi sabila velike vrednosti i rastegnula male, što će doprineti tome da odnos sa cenom postane glađi.
Nakon transformacije *availability365* imamo sliku 3.

$$skew = |skewness| = \left| \frac{1}{(n-1)(n-2)} \sum_{i=1}^{n} \left(\frac{x_i - \bar{x}}{\sqrt{\tilde{s}}} \right)^3 \right|$$

Što se tiče kategoričkih promenljivih, njih smo kodirali tako što smo napravili jednu sveobuhvatnu funkciju. Ta funkcija izdvaja promenljive tipa *category* i *object*. Ukoliko promenljiva nije ordinalna i ima manje od 10 jedinstvenih vrednosti, radi *one-hot encoding*; ukoliko jeste ordinalna, radi *ordinal encoding* sa prosleđenim redosledom i ukoliko nije ordinalna i ima vise od 10 jedinstvenih vrednosti, radi *target encoding* tako što koristi srednju vrednost ciljne promenljive. Ovo su tri osnovna načina kodiranja, pri čemu se One-hot encoding koristi za promenljive koje imaju mali broj kategorija i nemaju prirodni redosled, Ordinal encoding ukoliko kategorije imaju prirodni redosled i Target encoding kada kategorije nemaju prirodni redosled i ima ih dosta. Mi smo se odlučili da reper bude broj od 10 kategorija. Primena logaritamske transformacije na kodirane promenljive nema smisla jer te numeričke vrednosti nisu kvantitativne u pravom smislu i iskvario bi se ordinalni odnos. Ponekad, ukoliko imamo jako dobar razlog možemo ovo uraditi kod target encodinga, ali postoji veliki rizik jer možemo da unesemo *data leakage*, tako da ćemo taj segment preskočiti.
Na kraju sve promenljive su skalirane zbog lasso i ridge modela. XGB i RF ne zahtevaju skaliranje pošto su otporni na promene, to jedino ima smisla ukoliko se prave uz druge neke modele kojima je to pogodnost, kao što je ovde slucaj.



Slika 3: Scatter plot - feature vs price

3 Metode

3.1 Ridge regresija

Uvod i osnovna ideja

Ridge regresija predstavlja oblik regularizovane linearne regresije koji se koristi kako bi se rešio problem multikolinearnosti. Osnovna ideja je da se, pored minimizacije rezidualne sume kvadrata, dodatno penalizuje zbir kvadrata koeficijenata regresije. Time se svi koeficijenti uvlače ka nuli, ali nijedan ne postaje tačno nula.

Matematički, Ridge regresija rešava sledeći optimizacioni problem:

$$\hat{\boldsymbol{\beta}}^{\text{ridge}} = \arg\min_{\boldsymbol{\beta}} \left\{ \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \right\}, \tag{1}$$

gde je $\lambda \geq 0$ parametar regularizacije koji kontroliše intenzitet penalizacije. Alternativno, ovaj problem se može formulisati i kao minimizacija greške pod uslovom da suma kvadrata koeficijenata ne prelazi zadatu granicu t:

$$\hat{\boldsymbol{\beta}}^{\text{ridge}} = \arg\min_{\boldsymbol{\beta}} \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 \quad \text{uz uslov} \quad \sum_{j=1}^{p} \beta_j^2 \leq t. \tag{2}$$

Na ovaj način se direktno ograničava veličina koeficijenata, što doprinosi stabilnijem modelu i smanjenju disperzije procena. Budući da penalizacija nije invarijantna na skalu promenljivih, neophodno je standardizovati

sve prediktore pre primene Ridge regresije. Intercept β_0 se ne penalizuje kako bi predikcije bile konzistentne pri translaciji ciljne promenljive y.

Statistička interpretacija i svojstva

Ridge regresija se može zapisati i u matricnom obliku kao:

$$\mathrm{RSS}(\lambda) = (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^T(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) + \lambda\boldsymbol{\beta}^T\boldsymbol{\beta}, \tag{3}$$

pri čemu je rešenje dato eksplicitno kao:

$$\hat{\boldsymbol{\beta}}^{\text{ridge}} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{y}, \tag{4}$$

gde $\mathbf{I}$ označava identičnu matricu dimenzija $p \times p$. Dodavanjem pozitivne konstante λ na dijagonalu matrice $\mathbf{X}^T\mathbf{X}$ izbegava se problem singularnosti, što čini ovu metodu posebno korisnom u slučajevima kada postoji veliki broj prediktora ili jaka kolinearost.

Zanimljivo je i da Ridge regresija može biti interpretirana i iz Bajesovske perspektive.

3.2 Lasso regresija [7]

Uvod i osnovna ideja

Lasso regresija (Least Absolute Shrinkage and Selection Operator), koju je predstavio kanadski statističar i profesor Stanford univerziteta Robert Tibshirani 1996. godine, je metoda regularizacije koja istovremeno vrši redukciju koeficijenata i selekciju promenljivih. Njena svrha je da unapredi prediktivnu moć linearnog regresionog modela i poboljša njegovu interpretabilnost tako što forsira neke regresione koeficijente da postanu tačno 0.

U klasičnoj linearnoj regresiji cilj je minimizacija sume kvadrata reziduala

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j\right)^2 = \|\mathbf{y} - \mathbf{X}\beta\|_2^2$$

po parametrima modela $\beta_0, \beta_1, ..., \beta_p$. Sa $\|\cdot\|_2$ je označena $\mathbf{L}^2$ (odnosno euklidska) norma u $\mathbb{R}^n$, a sa $\mathbf{X}$ dizajn matrica, $\mathbf{y} = (y_1, ..., y_n)^T$, $\beta = (\beta_0, ..., \beta_p)^T$.

Kod Lasso regresije, glavnu ulogu igra tzv. $\mathbf{L}^1$ penalizacija regresionih koeficijenata koju predstavlja suma

$$\lambda\sum_{i=1}^{p}|\beta_i| = \lambda\|\beta^*\|_1$$

, gde je $\lambda \geq 0$ hiperparametar regularizacije koji kontroliše veličinu penalizacije, $\beta^* = (\beta_1, ..., \beta_p)$ i $\|\cdot\|_1$ $\mathbf{L}^1$ (ili tzv. Manhattan, taxicab) norma u $\mathbb{R}^p$.

U Lasso regresiji cilj je minimizovati izraz

$$\|\mathbf{y} - \mathbf{X}\beta\|_2^2 + \lambda\|\beta^*\|_1$$

po parametrima modela. Ako je $\lambda = 0$, onda se ovo svodi na optimizacioni problem u klasičnoj linearnoj regresiji, dok kada $\lambda \to \infty$, tada $\beta_i \to 0$ za sve $i \in \{1, 2, ..., p\}$. Dakle, penalizuje se $\mathbf{L}^1$ norma, tj. koeficijenti modela se *guraju* ka 0.

Za razliku od Ridge regresije koja koristi $\mathbf{L}^2$ penalizaciju, Lasso regresija zbog $\mathbf{L}^1$ penalizacije može neke koeficijente postaviti tačno na 0.
Prednosti su poboljšana generalizacija na nove podatke i to što model radi dobro i kada je broj observacija manji od broja prediktora, a mane su te što nije stabilna kada je prisutna visoka multikolinearnost i ukoliko su neki prediktori visoko korelisani zadržava jedan ili mali broj koeficijenata, za razliku od Ridge, koja zbog $\mathbf{L}^2$ penalizacije raspodeli težinu među svim korelisanim prediktorima.

Statistička interpretacija
Lasso regresija može da se interpretira i kao MAP (maximum a posteriori) prema Bajesu, ako uzmemo Laplasovu apriornu raspodelu za koeficijente modela:

$$f(\beta_i) = \frac{\lambda}{2} e^{-\lambda|\beta_i|},\ i = 1, ..., p$$

i ako stavimo recimo $f(\beta_0) \propto 1$
MAP je Bajesovski analog klasičnom metodu maksimalne verodostojnosti.

$$\hat{\beta}^{\text{MAP}} = \arg\max_{\beta} f(\beta|\mathbf{y}, \mathbf{X}) = \arg\max_{\beta} f(\mathbf{y}|\mathbf{X}, \beta) f(\beta) = \arg\max_{\beta} ln(f(\mathbf{y}|\mathbf{X}, \beta) f(\beta)) =$$

$$= \arg\min_{\beta} \{-lnf(\mathbf{y}|\mathbf{X}, \beta) - lnf(\beta)\} \propto \arg\min_{\beta} \left\{ \sum_{i=1}^{n} \left(y_i - x_i^T \beta^*\right)^2 + \lambda \sum_{i=1}^{p} |\beta_i| \right\}$$

gde je $x_i = (x_{i1}, ..., x_{ip})^T$. Ovo je motivacija za $\mathbf{L}^1$ penalizaciju.

3.3 Random Forest Regresija

Uvod i osnovna ideja

Slučajna šuma (*Random Forest*) predstavlja snažan i robusan ansambl algoritam koji koristi pristup proste agregacije (*bagging*, odnosno *bootstrap aggregation*) nad stablima odlučivanja [6]. Osnovna ideja je da se kombinuje više slabih modela – u ovom slučaju stabala odlučivanja – kako bi se dobio stabilan i precizan prediktor.

Stabla odlučivanja su poznata po visokoj disperziji, što znači da su osetljiva na promene u trening podacima. Da bi se to ublažilo, koristi se *bagging*, gde se svako stablo trenira na različitim uzorcima dobijenim uzorkovanjem sa ponavljanjem (bootstrap uzorci). Na taj način se smanjuje disperzija, dok pristrasnost ostaje slična kao kod pojedinačnog stabla.

Slučajna šuma dodatno poboljšava ovaj proces smanjenjem međusobne korelacije između stabala, čime se povećava ukupna preciznost ansambla. Ovo se postiže tako što se pri svakom grananju stabla bira nasumičan podskup od m prediktora (gde je tipično $m = \frac{p}{3}$ za regresiju), i od njih se bira optimalno razdvajanje. Na taj način se svako stablo gradi na drugačijoj strukturi, pa su manje slična međusobno, što povećava raznovrsnost i doprinosi boljim rezultatima generalizacije.

Tipične vrednosti hiperparametara uključuju $m = \frac{p}{3}$ za regresione zadatke i minimalnu veličinu lista od pet uzoraka.

Statistička interpretacija

Random Forest se može posmatrati kao nelinearna metoda ansambliranja koja balansira pristrasnost i disperziju. Agregacijom velikog broja slabih i nekorelisanih modela (stabala), dobija se snažan model sa znatno smanjenom varijancom.

Algorithm 1 RF regresija

1. Za $b = 1$ do B:
   (a) Izvrši se uzorkovanje sa ponavljanjem (bootstrap) sa ukupno Z^* od N uzoraka iz trening skupa.
   (b) Pravi se stablo odlučivanja T_b rekurzivnim grananjem sve dok se ne dostigne minimalni broj uzoraka u čvoru $n_{\min}$:
      i. Nasumično se izabere m prediktora od ukupno p.
      ii. Među tih m bira se najbolje moguće razdvajanje.
      iii. Formira se novo grananje na osnovu izabranog razdvajanja.
2. Vraća se niz stabala $\{T_b\}_{b=1}^{B}$.

Konačna procena za novi primerak x je srednja vrednost predikcija svih stabala:

$$\hat{f}_{\text{rf}}^{B}(x) = \frac{1}{B}\sum_{b=1}^{B} T_b(x)$$

Statistički gledano, svako stablo predstavlja procenu funkcije regresije $\hat{f}_b(x)$, dok konačna procena $\hat{f}_{\text{rf}}(x)$ predstavlja njihovu prosečnu vrednost. S obzirom da se stabla treniraju na različitim bootstrap uzorcima i koriste različite podskupove prediktora, disperzija procene se smanjuje proporcionalno broju stabala, pod pretpostavkom da su modeli nekorelisani:

$$\text{Var}(\hat{f}_{\text{rf}}(x)) \approx \frac{1}{B}\text{Var}(T_b(x)) \quad \text{ako su } T_b \text{ nezavisna.}$$

U praksi modeli nisu potpuno nezavisni, ali dodatna randomizacija preko izbora podskupa prediktora pri grananju doprinosi smanjenju međuzavisnosti i povećanju robusnosti.

Slučajna šuma je naročito korisna kada postoji veliki broj prediktora i moguća multikolinearnost. Nasumični izbor podskupa prediktora u svakom čvoru onemogućava dominaciju jakih prediktora u svim stablima i omogućava „glas“ i slabije korelisanim varijablama, čime se često poboljšava prediktivna moć.

3.4 XGBoost

XGBoost (E**X**treme Gradient Boosting) je algoritam mašinskog učenja kojeg odlikuju visoke performanse. Zasnovan je na tehnici gradijentnog pojačavanja (Gradient Boosting), odnosno predstavlja njegovo unapređenje. Razvio ga je Tianqi Chen i koristi se i za regresione i za klasifikacione probleme, s tim što je za klasifikaciju potrebno napraviti određene modifikacije u odnosu na ono što ćemo mi predstaviti a što je orijentisano ka regresiji.

Osnovna ideja Gradient Boosting-a

Gradijentno pojačavanje predstavlja ansambl tehniku učenja, tj. tehniku gde se više stabala odlučivanja kombinuju u moćan model. Proces učenja odvija se iterativno, gde svaki naredni model pokušava da ispravi greške prethodnih.

Neka je dat skup podataka $\{\mathbf{x}_i, y_i\}_{i=1}^{n}$, $\mathbf{x}_i = (x_{i1}, ..., x_{ip})^T$. Cilj je izgraditi model $F(\cdot)$ koji će sa ulazne vrednosti $\mathbf{x} \in \mathbb{R}^p$ da predvidi vrednost ciljne promenljive $y \in \mathbb{R}$ sa $F(\mathbf{x})$ tako da $\mathcal{L}(y, F(\mathbf{x}))$ bude što manje moguće, gde je $\mathcal{L}$ data funkcija gubitka, najčešće kvadratna, ako je u pitanju regresija, odnosno tzv. log-loss function ukoliko je u pitanju klasifikacija.

Iterativna izgradnja

K je hiperparametar koji predstavlja broj iteracija. Ako je K previše malo, postoji rizik od underfitting-a, a ako je isuviše veliko, onda od overfitting-a. Za njegovo određivanje se najčešće koristi early-stopping metoda, ali se neretko određuje i eksperimentalno pomoću gridsearching-a ili randomsearching-a.
Uzme se na početku da je

$$F_0(\mathbf{x}) = \arg\min_{\gamma} \sum_{i=1}^{n} \mathcal{L}(y_i, \gamma)$$

i onda se u svakoj od K iteracija ponavlja sledeće:
Računa se negativni gradijent funkcije gubitka u tačkama $F_{k-1}(\mathbf{x}_i)$ (on predstavlja pravac u kom treba korigovati trenutnu predikciju), tj. vrednosti

$$r_i^{(k)} = -\frac{\partial \mathcal{L}(y_i, t)}{\partial t}(F_{k-1}(\mathbf{x}_i))$$

koje se nazivaju još i pseudo-ostacima. Zatim se trenira novo stablo odlučivanja f_k koje predviđa pseudo-ostatke, a potom se određuje

$$\gamma_k = \arg\min_{\gamma} \sum_{i=1}^{n} \mathcal{L}(y_i, F_{k-1}(\mathbf{x}_i) + \gamma f_k(\mathbf{x}_i))$$

, odnosno koeficijent koji minimizuje ovaj izraz u smeru f_k. Generalno, $f_k(\mathbf{x})$ je generisana nekom funkcijom $q(\mathbf{x})$ koja predstavlja redni broj lista stabla koje se dodeljuje ulazu $\mathbf{x}$. Ona ulazu $\mathbf{x}$ dodeljuje fiksiranu predikciju na listu $q(\mathbf{x})$, tj. vrednost $\omega_{q(\mathbf{x})}$. Na kraju se ažurira model

$$F_k(\mathbf{x}) = F_{k-1}(\mathbf{x}) + \eta\gamma_k f_k(\mathbf{x})$$

, gde je $\eta \in (0, 1]$ hiperparametar koji se naziva koeficijent učenja. On govori o tome koliko jako svako novo stablo utiče na konačnu predikciju. Vrednosti ovog parametra bliske nuli označavaju da model uči sporije, ali je tada i stabilniji, tj. imamo manji rizik od prenaučenosti, dok sa druge strane vrednosti bliske jedinici označavaju da model brže uči, ali imamo veći rizik od lošije generalizacije. Najčešće se uzima vrednost u intervalu $[0.01, 0.3]$, pri čemu se kompenzacija za njegovu malu vrednost vrši tako što se uzme veće K.
Nakon ovog procesa uzima se $F(\mathbf{x}) = F_K(\mathbf{x})$.

XGBoost

Za razliku od gradient boosting-a, kod xgb algoritma 2 cilj je u svakoj iteraciji pronaći stablo f_k koje minimizuje

$$\sum_{i=1}^{n} \mathcal{L}(y_i, F_{k-1}(\mathbf{x_i}) + f_k(\mathbf{x}_i)) + \Omega(f_k)$$

, gde je $\Omega(f_k) = \varepsilon^{T_k} + \frac{1}{2}\lambda \sum_{j=1}^{T_k} \omega_j^2$ regularizacioni član koji služi da kontroliše kompleksnost stabla, odnosno da spreči da model postane prenaučen. To je glavna razlika u odnosu na gradient boosting. T_k je broj listova stabla, ω_j je težina pridružena j-tom listu (kao pseudo-ostaci kod gradient boostinga), a ε i λ su regularizacioni pozitivni hiperparametri. ε kontroliše broj listova, velike vrednosti ovog parametra penalizuju stabla sa velikim brojem listova; dok λ kontroliše težine listova, velike vrednosti ovog parametra penalizuju stabla sa velikim težinama.
Kako je glavna pretpostavka ta da je funkcija gubitka dva puta diferencijabilna po predikciji, optimalnije je koristiti Tejlorovu aproksimaciju drugog reda:

$$\sum_{i=1}^{n} \mathcal{L}(y_i, F_{k-1}(\mathbf{x_i}) + f_k(\mathbf{x}_i)) \approx \sum_{i=1}^{n} (g_i f_k(\mathbf{x}_i) + \frac{1}{2} h_i f_k^2(\mathbf{x}_i))$$

Algorithm 2 XGBoost

1. $F_0(\mathbf{x}) = \arg\min_\gamma \sum_{i=1}^{n} \mathcal{L}(y_i, \gamma)$
2. za $k = 1$ do K
   (a) Računa se
   $$g_i = \frac{\partial \mathcal{L}(y_i, t)}{\partial t}(F_{k-1}(\mathbf{x}_i))$$
   $$h_i = \frac{\partial^2 \mathcal{L}(y_i, t)}{\partial t^2}(F_{k-1}(\mathbf{x}_i))$$
   (b) Izgradi se stablo f_k koje minimizuje izraz
   $$\sum_{i=1}^{n}(g_i f_k(\mathbf{x}_i) + \frac{1}{2} h_i f_k^2(\mathbf{x}_i)) + \Omega(f_k)$$
   (c) Ažurira se model
   $$F_k(\mathbf{x}) = F_{k-1}(\mathbf{x}) + \eta f_k(\mathbf{x})$$
3. Vraća se finalni model
   $$F(\mathbf{x}) = F_K(\mathbf{x}) = F_0(\mathbf{x}) + \sum_{k=1}^{K} \eta f_k(\mathbf{x})$$

Može se pokazati da stablo koje treba izgraditi u algoritmu 2 ima predikcije na listovima, odnosno težine:

$$\omega_j = -\frac{\sum_{i \in I_j} g_i}{\sum_{i \in I_j} h_i + \lambda}$$

, gde je I_j indeksni skup, tj. idemo po svim y_i takvim da $\mathbf{x}_i$ pripada listu j.
Stablo je binarno i ima strukturu stabla odlučivanja. Dubina stabla jeste broj prelazaka od korena stabla do najdubljeg lista. Pretpostavimo da je unapred data maksimalna dubina stabla koju ćemo označiti sa maxDepth. Kada se dostigne ova dubina više se ne pokušavaju podele, odnosno čvor postaje list. Pretpostavimo da je dat i neki minimalni broj podataka koji ćemo označiti sa minSamples. Kada se dostigne ovaj broj dalje grananje se ne isplati ni statistički ni računarski. Oba ova podatka deo su hiperparametara metode.
Pseudokod za rekurzivnu izgradnju stabla dat je u nastavku:

Algorithm 3 IzgradiStablo(podaci,dubina)

if dubina $\geq$ maxDepth **or** broj podataka $\leq$ minSamples **then**
  $\omega \leftarrow -\frac{\sum g_i}{\sum h_i + \lambda}$
  return List sa težinom ω
end if
for Svaka moguća podela podataka **do**
  Uzmi levo i desno podstablo
  Izračunaj Gain
end for
Izaberi podelu sa najvećim Gain-om (maxGain)
if maxGain $> \varepsilon$ **then**
  $l \leftarrow$ IzgradiStablo($levo\_podstablo, dubina + 1$)
  $d \leftarrow$ IzgradiStablo($desno\_podstablo, dubina + 1$)
  return Čvor sa levim l i desnim d podstablom
else
  $\omega \leftarrow -\frac{\sum g_i}{\sum h_i + \lambda}$
  return List sa težinom ω
end if=0

Kriterijum podele i Gain

Prilikom građenja stabla u algoritmu 3 u svakom čvoru razmatraju se sve moguće podele na levo i desno. Da bi se odabrala optimalna podela koristi se metrika Gain definisana sa

$$\text{Gain} = \frac{1}{2}\left[\frac{G_L^2}{H_L + \lambda} + \frac{G_R^2}{H_R + \lambda} - \frac{(G_L + G_R)^2}{H_L + H_R + \lambda}\right] - \varepsilon$$

, gde su:

1. G_L zbir vrednosti g_i po levoj grani
2. G_R zbir vrednosti g_i po desnoj grani
3. H_L zbir vrednosti h_i po levoj grani
4. H_R zbir vrednosti h_i po desnoj grani

3.5 Evaluacija

Evaluacija modela sprovedena je korišćenjem 5-struke unakrsne validacije, uz prethodnu optimizaciju hiperparametara za svaki model posebno. Za sve modele korišćen je `GridSearchCV` metod iz biblioteke `scikit-learn`, pri čemu su kao metričke funkcije korišćeni R^2 koeficijent determinacije (kao primarna metrika za optimizaciju) i srednja kvadratna greška (MSE).

Optimizacija je sprovedena na sledeći način:

- **Lasso i Ridge regresija:** Optimizovani su u odnosu na hiperparametar α, uz raspon vrednosti koji uključuje i male i velike regularizacione faktore.
- **Random Forest:** Optimizacija je vršena nad brojem stabala (`n_estimators`), maksimalnom dubinom stabla (`max_depth`), brojem karakteristika pri svakoj podeli (`max_features`) i minimalnim brojem primera za podelu (`min_samples_split`).

- **XGBoost:** Optimizovani su parametri: broj estimatora (`n_estimators`), stepen učenja (`learning_rate`), maksimalna dubina stabla (`max_depth`) i udeo karakteristika koje se koriste za svako stablo (`colsample_bytree`

Za svaki model, GridSearchCV je automatski odabrao kombinaciju parametara koja maksimizuje prosečni R^2 skor dobijen tokom unakrsne validacije. Nakon identifikacije optimalnih parametara, izračunata je i odgovarajuća srednja kvadratna greška (MSE) tog modela, koja je dodatno prikazana u tabeli rezultata.

Model	**Najbolji R^2**	**MSE za najbolje param.**	**Najbolji parametri**
Lasso	0.9748	393.03	{alpha: 0.1}
Ridge	0.9748	393.27	{alpha: 1}
Random Forest	0.9154	1316.57	{max_depth: 20, max_features: 'sqrt', min_samples_split: 2, n_estimators: 200}
XGBoost	0.9773	354.31	{colsample_bytree: 1.0, learning_rate: 0.1, max_depth: 3, n_estimators: 200}

Tabela 1: Performanse modela i najbolji pronađeni hiperparametri

Primena na podacima je pokazala da je najbolje prediktivne rezultate proizveo XGBoost model što je i bilo očekivano.

Literatura

[1] City of Melbourne. *Short-stay accommodation regulation overview*. https://www.melbourne.vic.gov.au/residents/home-property/Pages/short-stay-accommodation.aspx. Accessed: 2025-04-18. 2023.

[2] Oxford Economics. "Short-term rentals generate €149B economic impact, 2.1M jobs across EU in 2023". U: (2023.). Accessed: 2025-04-07. URL: https://www.oxfordeconomics.com/resource/short-term-rentals-generate-e149b-economic-impact-2-1m-jobs-across-eu-in-2023/.

[3] Nicole Gurran i Peter Phibbs. "When Tourists Move In: How Should Urban Planners Respond to Airbnb?" U: *Journal of American Planning Association* 83.1 (2017.), str. 80–92. DOI: 10.1080/01944363.2016.1249011.

[4] Inside Airbnb. *Melbourne - Data set and analysis*. http://insideairbnb.com/melbourne. Accessed: 2025-04-18. 2024.

[5] Makarand Mody, Courtney Suess i Xinran Lehto. "The Accomodation Experiencescape: A Comparative Assessment of Hotels and Airbnb". U: *International Journal of Contemporary Hospitality Management* 29.9 (2017.), str. 2377–2404. DOI: 10.1108/IJCHM-09-2016-0501.

[6] J. Friedman T. Hastie R. Tibshirani. *The Elements of Statistical Learning - Data Mining, Inference, and Prediction*. Springer, 2008.

[7] Robert Tibshirani. "Regression shrinkage and selection via the lasso". U: *Journal of the Royal Statistical Society: Series B (Methodological)* 58.1 (1996.), str. 267–288. URL: https://www.jstor.org/stable/2346178.

[8] Georgios Zervas, Davide Prosperpio i John W. Byers. "The Rase of the Sharing Economy: Estimating the Impact of Airbnb on the Hotel Industry". U: *Journal of Marketing Research* 54.5 (2017.), str. 687–705. DOI: 10.1509/jmr.15.0204.