

Metode strojnog i dubokog učenja za predikciju brisanje rezervacija

Aldo Ferlatti

Ko-mentor: dr. sc. Nikola Tanković; Mentor: doc. dr. sc. Darko Etinger

Sveučilište Jurja Dobrile u Puli, Fakultet informatike,
Pula, Croatia

{aferlatt, ntankov, detinger}@unipu.hr
<https://fipu.unipu.hr/fipu/en>

Kolovoz 2019

Sažetak

Metode strojnog učenja su sveprisutne neovisno o vrsti industrije za koju se primjenjuje. U ovom završnom radu se analizira proces gradnje klasifikacijskoga modela: metode analiziranja podataka i gradnje data-seta; metode strojnog i dubokog učenja za gradnju modela te njegovu optimizaciju. Zbog izvornih podataka, analiza se temelji na hotelijerskoj industriji ali metode su primjenjive i u širem području.

Prema usporedbi algoritama, odabralo se XGBoosting i DNN kao algoritmi za testiranja te za optimizaciju istih se koristi metoda mrežnog pretraživanja i Bayesianova optimizacija.

Abstract

Machine learning methods are present independently of the type of industry they are applied to. In this final thesis it is analyzed the process of building a classification model: methods used for dataset building and analysis; methods of machine learning and deep learning for model creation and its optimization. Because of the nature of the source data, analysis are made for the hospitality industry but applicable on others too.

Based on algorithms comparison, XGBoosting and DNN algorithms are chosen for the tests. Grid search and Bayesian optimization are the methods for the model optimization.

1 Uvod

Strojno učenje je tehnologija koja je sve prisutnija u današnjem svijetu: predviđanje vremena, burzovnih cijena, klasifikacija kupca, itd. Strojno učenje je primjenjivo u svim industrijama iz kojih se mogu formatizirati izvorni podaci. Na raspolaganju za izradu ovog završnog rada su podaci iz hotelijerske industrije te je kao ciljni zadatak predvidjeti hoće li rezervacija biti izbrisana. Kod upravljanja prihoda, brisanje rezervacija ima veliki utjecaj na krajnju zaradu stoga posjedovanje te informacije unaprijed, omogućava poduzimanje postupaka u sprječavanju ili vođenju istih na adekvatan način. Takvi postupci mogu dovesti na sveukupni porast prihoda što je krajnji cilj bilo kojeg poduzeća.

Ispitivanje provedeno 2011. godine od Kimesa [1], pokazalo je da 24.6% ispitanika (od kojih 78.4% pripadaju hotelijerskoj industriji) misli da će u slijedećih 5 godina tehnologija imati veliku ulogu kod upravljanja prihoda a 17.8% su rekli da će prognoziranje i analitičke metode također imati veliki utjecaj.

U slijedećim poglavljima se obrađuju metode strojnog učenja i, suvremenija pod-kategorija, dubokog učenja: od prikupljanja podataka i njihovog čišćenja do krajnjeg rezultata prognoziranja, sve popraćeno sa kodom.

2 Pregled literature

Upravljanje prihodima je izvorno razvijeno 1966. godine u avionskoj industriji, tek je kasnije uvedena u drugim industrijama poput hotelijerstva, ugostiteljstva, kockarnicama itd. Značajan broj radova je izvedeno na temi predviđanja potražnje, međutim samo nekoliko njih ([2, 3, 4, 5, 6, 7, 8]) se koncentriraju specifično na metodologiji ovog završnog rada [3]. Kao što se može primijetiti u tablici 1, svi radovi koriste fokusirane podatke nad pojedinim hotelima i svi dostupni podaci su ispod 300 000 zapisa, sa iznimkom rada Koolea, Hopmana i Leeuwena [5] koji imaju bazu podataka veću od milijun zapisa. Međutim izvor nisu hoteli nego ugostiteljske nekretnine sa kapacitetom soba ne većom od dva. Korišteni podaci su PNR vrste (eng. Passenger Name Record): izraz koji potječe iz avionske industrije te je kasnije preuzet u hotelijersku industriju kao definicija podataka rezervacija; PNR podaci obuhvaćaju informacije o korisniku, tko će putovati ili prespavati prema rezervaciji, pojedinosti usluge, cijene i slično [9].

Korištene metode predviđanja variraju: korištena metoda ovisi o vrsti i veličini podataka na raspolaganju, pa tako i metode variraju od strojnog učenja do dubokog učenja. Iz tablice 1 se primjećuje da najčešći korišteni model je stablo odlučivanja i njegove varijacije: Boosted decision tree, XGBoost, Random forest. U ovom završnom radu koristiti će se model XGBoost-a, te opravdanja i razlozi odabira tog modela se obrazlažu u slijedećem poglavlju.

Tablica 1: Metode i rezultati pregledane literature

Lit.	Metoda	God.	Dataset	Br. hotela	Rezanje podataka	Točnost (min)
[7]	Boosted decision tree	2017	73K	4	Da	0.879
[3]	XGBoost	2017	N/A	2	Da	0.84
[2]	BPN GRNN	2013	N/A	N/A	N/A	0.808
[4]	N/A	2018	233K	9	Da	0.92
[6]	C4.4 RndForest SVM KLR	2009	240K	1	Ne	N/A
[5]	RndForest	2018	1.27M	7 non hotels	Da	0.89
[8]	XGBoost	2019	100K	8	Da	0.777

Pošto su podaci podijeljeni po hotelima, odnosno nisu jednoobrazni za bilo koji hotel, takvi su i rezultati istraživanja: dobivena preciznost i točnost modela vrijedi samo za taj specifičan hotel.

Baza podatak na raspolaganju za ovaj završni rad ima 661 857 zapisa od 26 hotela koji se razlikuju po veličini i kvaliteti: u model ulaze svi podaci, ne razdvojeni po hotelima, što ujednačuje rezultat za sve hotele. Morales i Wang [6] predlažu dva modela podataka: sezonski prosjek i PNR podaci. Iako je prvi jako popularan u praksi, PNR podaci su dokazali da donose bolje rezultate. Podaci na raspolaganju su kombinacija tih dva modela: količina osobnih podataka gosta je svedena na minimum, ostavljeni su samo podaci za koje se misli da mogu utjecati na završni rezultat kao što je država porijekla, prisutnost djece itd.

Također treba pripaziti na način rezervacije. Nove tehnologije dovode do stvaranja novih usluga: u ovom slučaju razvile su se online putničke agencije (eng. Online Travel Agencies, OTAs) koje značajno olakšavaju proces rezerviranja te brisanje iste. Falk i Vieru [4] su dokazali da rezervacije napravljene preko online agencija imaju veću stopu brisanja nego ostale rezervacije.

3 Metoda strojnog učenja

Ovo predstavlja klasifikacijski problem, stoga se koristi nadzirani klasifikacijski algoritam. Strojevi podržanog vektora (eng. Support Vector Machines), Stablo odlučivanja (eng. Decision Trees), Logistička regresija (eng. Logistic Regression) itd. su svi poznati algoritmi za klasifikaciju. Kako bi se odabrao najprikladniji algoritam koji će dati najbolje moguće rezultate, napravila se funkcija za usporedbu modela. Usporedba se napravila na uzorku od 5000 nasumičnih zapisa i provjerena sa 10-strukom unakrsnom validacijom.

U tablici 2 su prikazani F1-rezultati usporedbe u padajućem redoslijedu: F1-rezultat predstavlja odnos između preciznosti i odaziva, te kao takav se smatra dobrim pokazateljem kvalitete modela. Tablica 2 pokazuje da algoritam XG-Boost ima najbolji rezultat, te je korišten u modeliranju modela za predikciju brisanja rezervacija.

Tablica 2: Rezultati usporedbe algoritma

Model	F1-rezultat
SVC-linear	0.578 ± 0.137
SVC-rbf	0.678 ± 0.001
Decision tree	0.692 ± 0.016
Logistic regression	0.697 ± 0.016
Gradient boosting	0.708 ± 0.008
XGBoost	0.768 ± 0.012

Povećanje gradijenta (eng. Gradient Boosting) radi na način da završno predviđanje sastavi od puno slabijih modela predviđanja, te se na svakoj interakciji novi slabi klasifikator nadoda na prijašnji model na način da ispravlja grešku. Ekstremno povećanje gradijenta (eng. Extreme Gradient Boosting, XGBoosting) radi na sličan način ali rezultat je točniji jer kontrolira pre-treniranje te je procesorski efikasniji pošto koristi algoritam za paralelizaciju stabla [10]. Kao i kod svakog algoritma temeljenom na stablu odlučivanja, tako i kod XGBoostinga najveći je problem odrediti strukturu stabla: postoji puno kombinacija stabala te pronaći optimalnu može zahtijevati veliku procesorsku snagu [11]. U tu svrhu, XGBoosting koristi pohlepan algoritam uveden od Chena i Guestrina [12], "Osnovno točni pohlepni algoritam" (eng. "Basic exact greedy algorithm"): prvo sortira podatke prema vrijednostima atributa a zatim posjećuje vrijednosti kako bi sakupio gradijentnu statistiku za ocijeniti strukturu prema jednadžbi 1.

$$G = \frac{1}{2} \left[\frac{(\sum_{i \in I_L} g_i)^2}{\sum_{i \in I_L} h_i + \lambda} + \frac{(\sum_{i \in I_R} g_i)^2}{\sum_{i \in I_R} h_i + \lambda} - \frac{(\sum_{i \in I} g_i)^2}{\sum_{i \in I} h_i + \lambda} \right] - \gamma \quad (1)$$

Gdje g_i i h_i predstavljaju prvi i drugi redoslijed gradijenta na funkciji gubitka; I_L i I_R označavaju skupove uzoraka lijeve i desne grane stabla; λ je konstanta te γ je parametar kompleksnosti. Algoritam se zaustavlja kada $G < 0$, te najveći G označuje optimalno grananje na čvoru [11].

Za pisanje algoritama i modeliranje XGBoostinga koristila se knjižnica Scikit-learn: prikladna je za nadzirane i ne nadzirane probleme srednjih veličina, raspolaze sa jednostavnim sučeljem sa svrhom dovođenja prednosti strojnog učenja ljudima koji ne raspolazu takvim predznanjem, odnosno za amatere [13].

4 Metoda dubokog učenja

Duboko učenje je pod-polje strojnog učenja koji ima kompleksniju strukturu i temelji se na umjetnim neuronskim mrežama: mreže su sastavljene od više nelinearnih skrivenih razina, gdje rezultat svake razine predstavlja ulaz slijedećoj razini [14]. Neuronska mreža je paralelna, distribuirana struktura obrade informacija koja obrađuje elemente međusobno povezanim jednosmjernim signalnim kanalima [15]. Dubinsko učenje se u većini slučajeva koristi za procesiranje podataka na ljudskoj razini, kao na primjer prepoznavanje slika i govora. Međutim može se aplicirati i za jednodimenzionalne ulazne podatke kao što je u ovom slučaju: ulaz u mrežu je jednodimenzionalni vektor koji predstavlja jedan zapis. Zbog komplicirane strukture mreža potrebna je veća procesorska snaga za treniranje modela. Dodatni problem predstavlja definiranje parametra mreža: teško je precizno definirati prikladnu dubinu mreže i ostale parametre kao što su stopa učenja ili broj ciklusa. Kao takve, dubinske neuronske mreže su dobar kandidat za hiper-parametarsko pretraživanje koje će se razmotriti u 6. poglavlju.

Kao i u slučaju algoritma strojnog učenja, uspoređuju se tri klasifikacijski nadzirani algoritmi dubinskog učenja te provjereni sa 10-strukom unakrsnom validacijom nad uzorkom od 5000 zapisa: gusta neuronska mreža (eng. Dense Neuron Network, DNN), rekurzivna neuronska mreža (eng. Recursive Neuron Network, RNN) te konvolucijska neuronska mreža (eng. Convolutional Neuron Network, CNN).

Prema tablici 3 DNN daje najbolji rezultat te kao takav se koristi za slijedeća testiranja u sklopu dubinskog učenja.

Tablica 3: Usporedba dubinskih modela

Model	accuracy
DNN	0.696 ± 0.0005
RNN	0.632 ± 0.132
CNN	0.688 ± 0.021

DNN predstavlja jednu od jednostavnijih mreža iz skupine algoritama dubinskog učenja: sastavljene su od niza elemenata, tzv. neuroni, koji uzimaju ulaz i težinski faktor nad konekcijom koja varira kako bi se smanjio rezultat određene funkcije gubitka [16]. Pohrana ulaznih podataka se razvija slijedno, odnosno širenje u naprijed se izvršava nivo po nivo, bez preskakanja. Osobnost DNNa je što je svaki neuron spojen sa svakim neuronom slijedeće skrivene razine, što znači da sa svakom dodatnom razinom struktura i vrijeme treniranja postaju zahtjevniji. Jedan od izazova korištenja neuronskih mreža je što se smatraju algoritmima crne kutije: iako se poznaje način rada neurona, veliki broj

konekcija i razina predstavlja problem za interpretirati unutarnji rad istih [16].

$$F(x) = \max(0, x) = \begin{cases} 0, & x < 0 \\ x, & x \geq 0 \end{cases} \quad (2)$$

U tablici 4 je prikazana struktura, odnosno skrivene razine, neuronske mreže: sve razine su guste vrste; veličina razina se postepeno smanjuje prema izlaznoj razini koja ima 7 izlaznih neurona, odnosno odgovara sa mogućim brojem kategorija; aktivacijske funkcije su 'relu' (funkcija 2), osim izlazne razine koja ima aktivacijsku funkciju 'softmax' za određivanje predviđene kategorije.

Tablica 4: Struktura DNN modela

Vrsta sloja	Veličina (broj neurona)	Aktivacijska funkcija
Dense	512	relu
Dense	256	relu
Dense	128	relu
Dense	128	relu
Dense	128	relu
Dense	32	relu
Dense	7	softmax

5 Analiza podataka

U kreiranju predikcijskih modela, velik utjecaj ima vrsta podataka na raspolaganju, njihova čistoća i odabir prikladnih atributa. Kod modeliranja, veliki dio vremena se provodi upravo na tom procesu: čišćenju i odabiru podataka. Za ovaj završni rad na raspolaganju je dataset od anonimnih rezervacija od 26 hotela kroz razdoblje od tri godine (2016, 2017, 2018); od kuda proizlaze 661 857 rezervacija.

Za svrhu analize podataka se koristila Python knjižnica Pandas: alat za statističku analizu podataka dizajnirana kao zamjena za R verziju za manipulaciju podataka; sa temeljnom strukturom "Podatkovnog okvira" (eng. DataFrame), knjižnica nadopunjuje ostatak znanstvenih Python knjižnica, čineći ju dobrim kandidatom i za veće baze podataka [17].

5.1 Inženjerstvo atributa

Dobro koncipirani atributi neke pute mogu efikasnije obuhvatiti važnost informacije nego izvorni atributi [18].

Dataset ima puno tekstualnih podataka, što za procesiranje klasifikacijskoga modela predstavlja problem. Kako bi se normalizirali tekstualni podaci, primijenjeno je one-hot kodiranje, specifično na slijedećim atributima: COUNTRY, CHANNEL te STATUS RESERVATION.

Vremenski atributi kao što je VRIJEME KREIRANJA REZERVACIJE ne daje predvidljivu vrijednost: datum, iako brojčana vrijednost, nema informacijsku vrijednost za model. Za vremenske attribute potrebno je izvest važnije vremenske attribute koje promatrane kao cjelinu opisuju početni vremenski atribut: DAY OF THE WEEK (dan u tjednu), YEAR (godina), DAYS TO CANCELLATION (dani to brisanja), DAYS TO CHECKIN (dani do prijave).

$$x_{sin} = \sin\left(\frac{2\pi x}{\max(x)}\right) \quad (3)$$

$$x_{cos} = \cos\left(\frac{2\pi x}{\max(x)}\right) \quad (4)$$

Neki od tih atributa imaju cikličku prirodu i kao takvi moraju se tretirati prikladno: najveća vrijednost se nalazi odmah pokraj najmanje vrijednosti. To se postiglo koristeći \sin (eq. 3) i \cos (eq. 4) funkcije [19]. Funkcije 3 i 4 pretvore vremenski podatak u koordinate kruga, koji točno prikazuje ciklički podatak.

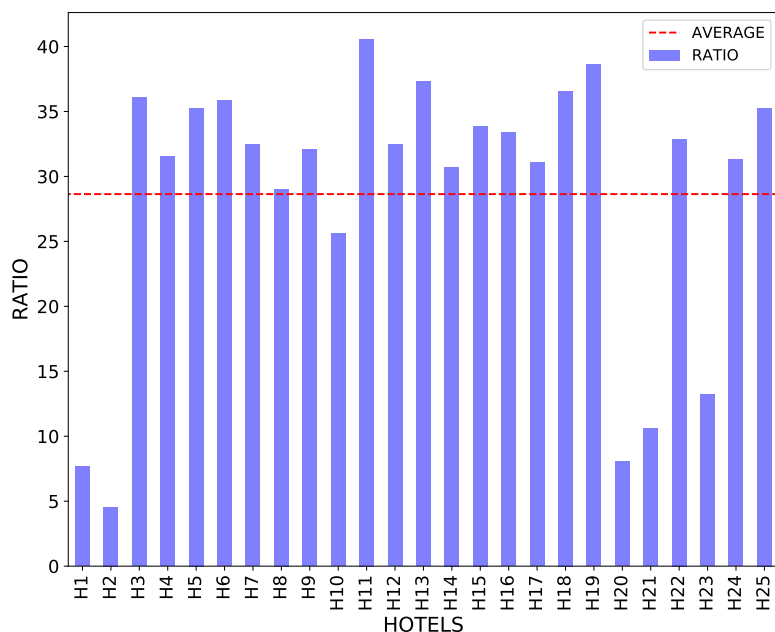
Tablica 5: Odabrani atributi

Naziv	Raspon (max - min)	Opis
YEAR	2016 - 2018	Year when reservation was first created
NUMBER OF DAYS	1 - 640	Booked days
COUNTRY	0 - 162	Costumer home country
ROOM TYPE	0 - 75	Type of the room
DEPOSIT	0 - 143663	Amount of the deposit
ROOM NUMBER	1 - 450	Number of rooms booked
CHILDREN	0/1	Indicates if children are present
PERSONS	1 - 90	Number of persons
NIGHTS	0 - 3948	Number of booked nights
SIN/COS WEEKDAY CREATED	$(-0.866) - 0.866 / (-1) - 1$	Day of the week when reservation was created
SIN/COS WEEK CREATED	$(-0.9995) - 0.9995 / (-1) - 1$	Week of the year when reservation was created
SIN/COS MONTH CREATED	$(-1) - 1 / (-1) - 1$	Month of the year when reservation was created
SIN/COS WEEKDAY CONFIRMED	$(-0.866) - 0.866 / (-1) - 1$	Day of the week when reservation was confirmed
SIN/COS WEEK CONFIRMED	$(-0.9995) - 0.9995 / (-1) - 1$	Week of the year when reservation was confirmed
SIN/COS MONTH CONFIRMED	$(-1) - 1 / (-1) - 1$	Month of the year when reservation was confirmed
SIN/COS WEEKDAY CHECK IN	$(-0.866) - 0.866 / (-1) - 1$	Day of the week of check in date
SIN/COS WEEK CHECK IN	$(-0.9995) - 0.9995 / (-1) - 1$	Week of the year of check in date
SIN/COS MONTH CHECK IN	$(-1) - 1 / (-1) - 1$	Month of the year of check in date
SIN/COS WEEKDAY CHECK OUT	$(-0.866) - 0.866 / (-1) - 1$	Day of the week of check out date
SIN/COS WEEK CHECK OUT	$(-0.9995) - 0.9995 / (-1) - 1$	Week of the year of check out date
SIN/COS MONTH CHECK OUT	$(-1) - 1 / (-1) - 1$	Month of the year of check out date
CHANNEL	0 - 8	Method of reservation
RESERVATION STATUS	0 - 10	Reservation status
DAYS TO CHECK IN	0 - 1224	Number of days between created reservation date and check in date

5.2 Prikaz podataka

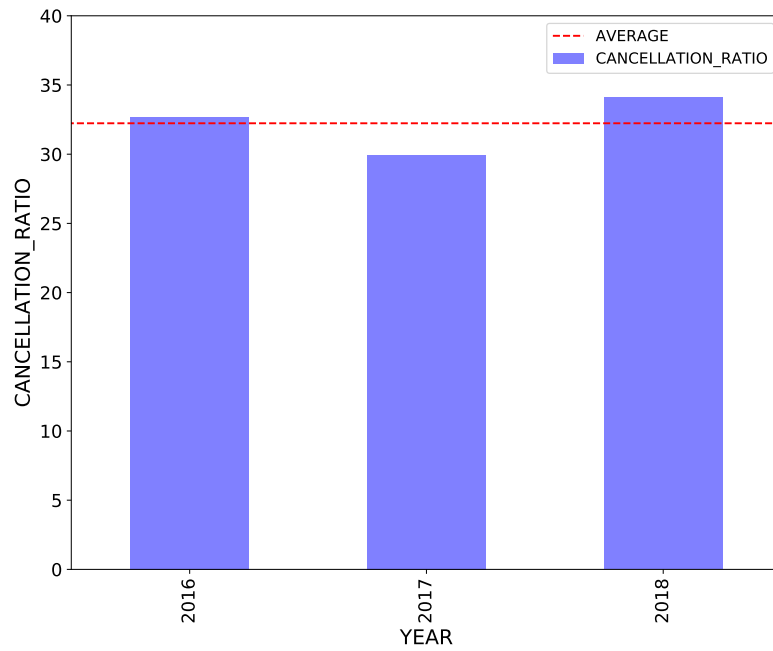
Kako bi se odabrali relevantni atributi koji ulaze u model, potrebno je shvatiti podatke na raspolaganju, njihovo značenje i ponašanje. Ovo poglavlje analizira kretanje podataka te grafički prikaz istih.

Na slici 1 su prikazane stope brisanja za svaki hotel: pojedinačni hotel ima različitu stopu brisanja, koja se kreće od minimuma 4.54% do maksimuma 40.58%. Međutim, sveukupni prosjek je od 28.64% što je u skladu sa prijašnjim radovima.



Slika 1: Stopa brisanja rezervacija po hotelu

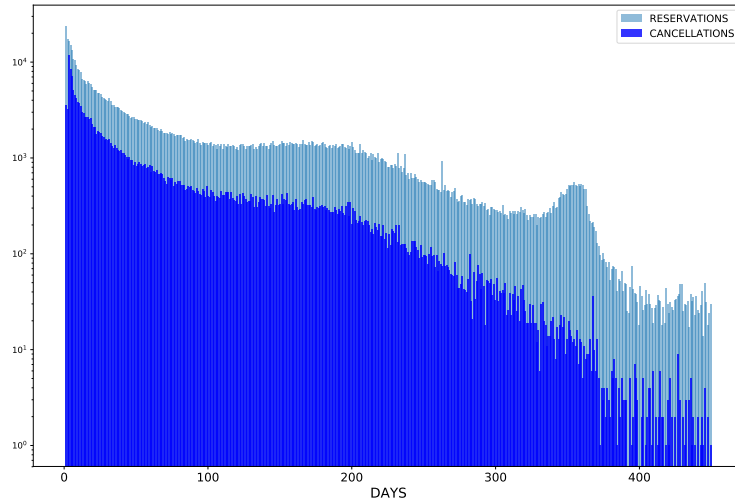
Falk i Vieru [4] su dokazali da povećana uporaba online agencija također dovodi do povećane stope brisanja rezervacija. Prateći taj zaključak, zbog unaprjeđenja tehnologije i pristupa online agencijama, kroz godine bi stopa brisanja rezervacija rasla. Međutim u slučaju podataka na raspolaganju, kao što se može primijetiti na slici 2, stopa brisanja je stabilna kroz sve tri godine, sa prosjekom od 32.23% i standardnom devijacijom od 2.11%.



Slika 2: Stopa brisanja rezervacija po godini

Iz slike 3 (prikazana sa logaritamskom skalom) se primjećuje da ima očekivani eksponencijalni rast u broju rezervacija sa manjim rasponom u broju dana prije prijave u hotel. Međutim, ima i neočekivani rast u rezervacijama skoro godinu dana unaprijed. Objašnjenje za taj neočekivani rast stoji u činjenici da prosječno 65% tih rezervacija je za sezonsko razdoblje (uzimajući u obzir da je sezonsko razdoblje između 01.05. i 30.09.) sa prosječno 6.6% brisanih rezervacija. Ostalih 35% neočekivanih rezervacija su za razdoblje izvan sezone, rezultirajući sa prosječnom stopom brisanja od 29.9%.

Broj brisanja rezervacija je u skladu sa eksponencijalnim rastom rezervacija: to dokazuje da rezervacija ima veću vjerojatnost da se izbriše kako se datum prijave približava.



Slika 3: Odnos obrisanih podataka između datuma rezervacije i broj dana brisanja prije prijave u hotel (check-in)

6 Optimizacija i testiranja

6.1 Testiranje dataseta

Nakon provođenja postupka inženjerstva atributa, dobio se dataset sa tri vrste ciljnog atributa. Kako bi se odabrao najprikladniji provelo se testiranje nad svim triju datasetima. Ciljni atributi se razlikuju po slijedećim aspektima:

- Binarni izlaz: dvije izlazne klase gdje jedinica označava *brisano* a nula *ne obrisano*.
- Kategorički izlaz prve vrste **CAT CH**: izlaz je podijeljen u 8 kategorija temeljene na broj dana između stvaranje rezervacije i prijave u hotel.
- Kategorički izlaz druge vrste **CAT RES**: izlaz je podijeljen u 7 kategorija temeljene na broj dana između kreiranja rezervacije i datuma brisanja rezervacije (gdje nulta kategorija označava ne obrisane rezervacije).

Testiranje se provelo na dva različita seta parametra. Pošto su izlazi kategoričke vrste, potrebne su i agregacijske metode kako bi se doveo rezultat na binarinu predikciju.

Set parametra su:

- Set parametra **A**: stopa učenja (eng. learning rate) 0.2; veličina stabla

(eng. number of estimators) 200; maksimalna dubina (eng. maximum depth) 2.

- Set parametra **B**: stopa učenja (eng. learning rate) 0.01; veličina stabla (eng. number of estimators) 1000; maksimalna dubina (eng. maximum depth) 4.

te agregacijske metode:

- Agregacijska metoda **50%**: prag od 50%, gdje rezultati ispod praga znače da rezervacija *nije obrisana*.
- Agregacijska metoda **binarna**: nulta kategorija označava da rezervacija *nije obrisana*, dok sve druge označavaju da je rezervacija *obrisana*.

Iz tablice 6 se vidi da dataset sa kategoričkim izlazima, temeljeni na broj dana između kreiranja rezervacije i datuma brisanja iste (CAT RES), skupa sa binarnom agregacijskom metodom (binary) te set parametra sa nižim vrijednostima (A), daju najbolji rezultat. Kao takav, taj dataset se koristio za sva slijedeća treniranja i testiranja.

Tablica 6: Testiranje dataseta

Parametri	Ciljani izlaz	Agregacija	F1-score
A	binary	None	0.77
B	binary	None	0.78
A	CAT CH	50%	0.75
A	CAT CH	binary	0.74
A	CAT RES	50%	0.72
A	CAT RES	binary	0.83
B	CAT CH	50%	0.77
B	CAT CH	binary	0.74
B	CAT RES	50%	0.72
B	CAT RES	binary	0.66

6.2 Optimizacija hiper-parametra

Algoritmi strojnog učenja i dubokog učenja su definirani po parametrima i hiper-parametrima. Parametri se mogu definirati direktno iz početne strukture algo-

ritma, ali hiper-parametri su višeg nivoa te se moraju odrediti i optimizirati prije početka treniranja jer mogu drastično utjecati na efikasnost algoritma [11]. Postoje razni algoritmi za traženje optimalnih vrijednosti hiper-parametra; za algoritam XGBoostinga se koristila izravna metoda pretrage: mrežno pretraživanje (eng. grid search). Metoda prolazi kroz sve kombinacije predefiniranih vektora mogućih vrijednosti, te kombinacija sa najboljim rezultatom se smatra optimalnim rješenjem. Negativna strana ove metode je što brzina izvođenja ovisi o veličini dataseta i algoritma nad kojim se aplicira.

Za algoritam XGBoostinga se odabralo tri hiper-parametra za koje se smatra da utječu na kvalitetu modela i brzinu treniranja istog: stopa učenja (eng. learning rate), broj procjenitelja (eng. number of estimators) i dubina stabla (eng. tree depth). Stopa učenja je vrijednost doprinosa funkciji gubitka nakon svake iteracije; broj procjenitelja predstavlja broj stabala u strukturi, odnosno veličina završnog stabla. Za veće količine podataka se preporuča imati manji broj procjenitelja pošto veći broj ulazi u problem pre-treniranja [11]. Prema tome, korelacija između veličine stabla i dubine stabla je definirana prema sljedećoj formuli:

$$J \leq 2^D \quad (5)$$

Gdje J označava veličinu stabla te D maksimalnu dubinu stabla [11].

Mrežno pretraživanje se izvelo nad raznim veličinama podataka te validirano sa 10-strukom unakrsnom metodom. U tablici 7 su prikazane provjerene vrijednosti hiper-parametra: veza između veličine stabla i dubine stabla je u skladu sa jednadžbom (5).

Tablica 7: Grid search - distribucije

Hiper-parametar	Vrijednosti
Learning rate	[0.05, 0.1, 0.15]
Number of estimators	[100, 200, 300]
Maximum tree depth	[3, 5, 7, 9]

Tablica 8 prikazuje najbolje rezultate pretrage: veličina dataseta sa korištenim hiper-parametrima za dobiveni rezultat. Rezultati pokazuju da kombinacija hiper-parametra [Stopa učenja, veličina stabla, dubina stabla]=[0.15, 300, 9] daje najbolji rezultat.

Tablica 8: Grid search - rezultati

Dio dataset	Stopa učenja	Veličina stabla	Dubina stabla	F1-score
1%	0.15	300	7	0.787
10%	0.15	300	9	0.892
20%	0.15	300	9	0.909
40%	0.15	300	9	0.922
60%	0.15	300	9	0.928
80%	0.15	300	9	0.932
100%	0.15	300	9	0.935

Druga metoda za optimizaciju hiper-parametra je Bayesianova optimizacijska metoda. Ova metoda se koristila za optimizaciju modela dubinskog učenja DNN. Za razliku od mrežne metode, Bayesianova metoda radi na način da vrijednosti parametra imaju Gaussianovu razdiobu, te se kreće po razdiobi dok ne pronađe optimalnu kombinaciju [20].

Kao što ime sugerira, Bayesianova optimizacijska metoda se temelji na Bayesovom teoremu vjerojatnosti: vjerojatnost modela (M) prema danim dokazima (E) je proporcionalno vjerojatnosti E prema M pomnoženoj sa 'prior' vjerojatnosti od M [21]:

$$P(M|E) \propto P(E|M)P(M) \quad (6)$$

Kod optimizacijske primjene, 'prior' označava vjerovanje o mogućim vrijednostima za hiper-parametre. Bayesianova metoda je različita od ostalih metoda jer napravi model vjerojatnosti od ulazne funkcije, te koristi taj model kako bi odredila gdje se slijedeće pomaknuti na razdiobi vrijednosti hiper-parametra [20].

Kod mrežne metode je potrebno odrediti vektore vrijednosti kako bi algoritam odradio sve kombinacije, dok kod Bayesianove metode je potrebno odrediti samo minimum i maksimum vrijednosti za svaki hiper-parametar.

Za DNN model se odabralo četiri hiper-parametra za optimizaciju: stopa učenja (eng. Learning rate), stopa izbačaja podataka (eng. Dropout rate), broj razdoblja (eng. Epochs) te veličina hrpe (eng. Batch size). U tablici 9 su prikazane odabrane granice razdiobe za optimizaciju.

Tablica 9: Bayesianova optimizacija - početne vrijednosti

Hiper-parametar	Vrijednosti
Learning rate	(1e-9, 1e-4)
Dropout rate	(0.1, 0.3)
Epochs	(5, 30)
Batch size	(60, 120)

Kod Bayesianove optimizacije dva parametra su značajna: broj optimizacija, odnosno koliki broj puta će se algoritam pomicati po modelu vjerojatnosti te broj nasumičnih istraživanja unutar ponuđenih granica. U svrhu ovog testa, oba parametra su postavljena na 15. Test je odrađen na 20% dataseta, od kojih 20% je odvojeno za validaciju.

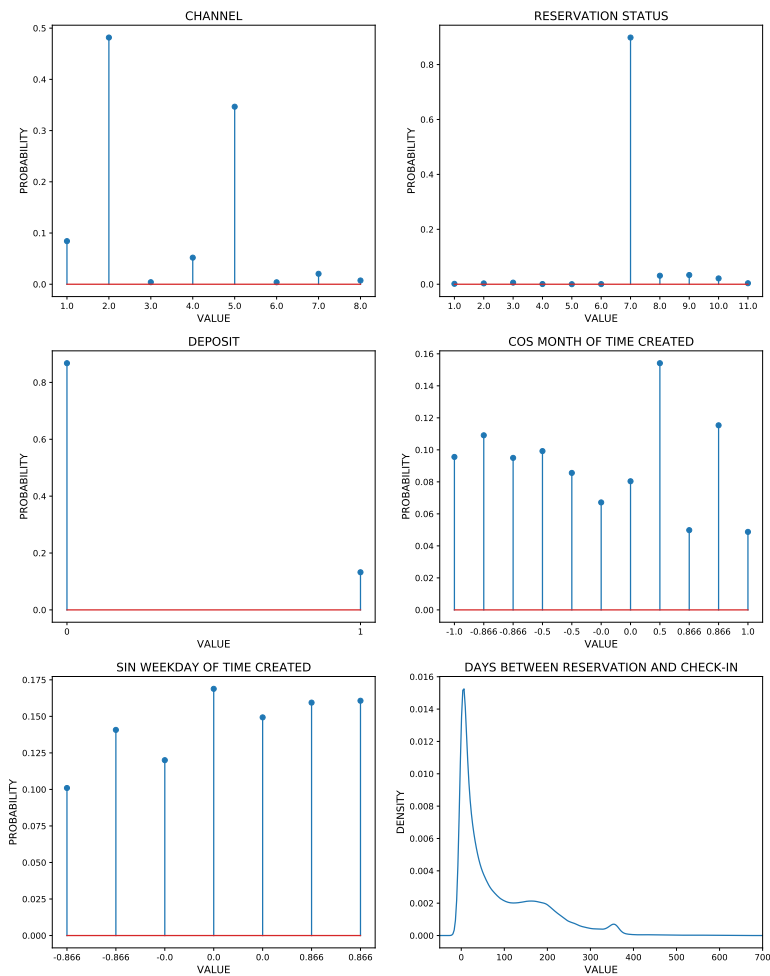
Tablica 10: Bayesianova optimizacija - optimizirane vrijednosti

Hiper-parametar	Vrijednosti
Learning rate	1e-8
Dropout rate	0.14
Epochs	6
Batch size	80
Accuracy	0.7531

6.3 Važnost atributa

Kod treniranje modela, atributi imaju različite utjecaje na rezultat: jedan atribut može više utjecati na način da njegova promjena u vrijednosti odlučuje u krajnjem rezultatu. Algoritam XGBoost dopušta izvući vrijednost važnosti svakog atributa. Na slici 4 je prikazana distribucija vjerojatnosti od šest atributa koji najviše utječu na model. Važnost atributa pomaže u donošenju odluka u stvarnim okolnostima: atributi u modelu predstavljaju opis jednog događaja; prema tome ako jedan atribut ima veći utjecaj na rezultat modela, također može imati veći utjecaj na ishod događaja u stvarnim okolnostima.

Kao što je dokazao Leeuwen [5], atribut kanala je među najvažnijima u modelu: to pokazuje da odabir kanala za rezervaciju ima veći utjecaj na ishod brisanja rezervacije. Također imaju veliku važnost vremenski atributi: vrijeme kreiranja rezervacije i broj dana između rezervacije i datuma prijave. To pokazuje da osim kanala, veliki utjecaj ima i odabir vremena rezervacije.



Slika 4: Distribucija vjerojatnosti od šest najvažnijih atributa

7 Zaključak

U ovom završnom radu se koncentriralo na postignuće visoke preciznosti u prognoziranju brisanja rezervacija koristeći razne metode strojnog i dubokog učenja. Algoritam XGBoost je dao bolje rezultate u odnosu na gustu neuronsku mrežu (DNN), prvenstveno radi nedostatka procesorske snage za optimizaciju i treniranje kompleksnije mreže. Unatoč tome, XGBoost je dobar odabir za klasiifikacijske probleme, zbog svoje brzine treniranja, dotreniravanja i mogućnosti paralelizacije procesa.

Bayesianova metoda optimizacije je zbog svoje osobnosti pretraživanja temeljenom na modelu vjerojatnosti bolja od mrežne metode koja ovisi o vektorskim ulazima odabranih sa strane stvaratelja modela i optimizacije. U ovom radu, odabir atributa izveo se na vizualan i analitički način; buduće bi izvedbe obuhvatile naprednije načine selekcije atributa: Challita, Khalil i Beausery [22] predlažu metodu strojnog i dubinskog učenja, temeljenoj na težinskim faktorima koja efikasno odabere najvažnije attribute kao ulaz modela.

Literatura

- [1] S. E. Kimes, “The future of hotel revenue management,” *Journal of Revenue and Pricing Management*, vol. 10, no. 1, pp. 62–72, 2011.
- [2] H.-c. Huang, A. Y. Chang, and C.-c. Ho, “Using Artificial Neural Networks to Establish a Customer-cancellation Prediction Model,” no. 1, pp. 178–180, 2013.
- [3] N. Antonio, A. De Almeida, and L. Nunes, “Predicting hotel bookings cancellation with a machine learning classification model,” *Proceedings - 16th IEEE International Conference on Machine Learning and Applications, ICMLA 2017*, vol. 2018-Janua, pp. 1049–1054, 2017.
- [4] M. Falk and M. Vieru, “Modelling the cancellation behaviour of hotel guests,” *International Journal of Contemporary Hospitality Management*, vol. 30, no. 10, pp. 3100–3116, 2018.
- [5] R. van Leeuwen, G. Koole, and D. Hopman, “Cancellation Predictor for Revenue Management applied in the hospitality industry,” 2018.
- [6] D. R. Morales and J. Wang, “Passenger Name Record Data Mining Based Cancellation Forecasting for Revenue Management,” *Innovative Applications of OR*, vol. 202, pp. 554–562, 2008.
- [7] N. Antonio, A. De Almeida, and L. Nunes, “Predicting hotel booking cancellations to decrease uncertainty and increase revenue,” *Tourism & Management Studies*, vol. 13, no. 2, pp. 25–39, 2017.
- [8] N. Antonio, A. De Almeida, and L. Nunes, “Big Data in Hotel Revenue Management: Exploring Cancellation Drivers to Gain Insights Into Booking Cancellation Behavior,” *Cornell Hospitality Quarterly*, 2019.
- [9] J. Sokel, R. Liew, and M. J. Alford, “System and methods for synchronizing passenger name record data,” 2002.
- [10] A. Gupta, K. Gusain, and B. Popli, “Verifying the Value and Veracity of eXtreme Gradient Boosted Decision Trees on a Variety of Datasets,” 2015.
- [11] Y. Xia, C. Liu, and Y. Li, “A boosted decision tree approach using Bayesian hyper-parameter optimization for credit scoring,” *Expert Systems with Applications*, 2017.
- [12] T. Chen and C. Guestrin, “XGBoost: A Scalable Tree Boosting System,” *The Journal of the Association of Physicians of India*, pp. 785–794, 1994.
- [13] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Tihirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, “Scikit-learn: Machine Learning in Python,” *Psychological Science*, pp. 1682–1690, 2011.

- [14] D. Yu and D. Li, “Deep Learning and Its Applications to Signal and Information Processing,” *IEEE Signal Processing Magazine*, pp. 145–150, 2011.
- [15] R. Hecht-Nielsen, “Theory of the Backpropagation Neural Network,” *Neural Networks for Perception*, pp. 65–93, 1992.
- [16] P. Farré, A. Heurteau, O. Cuvier, and E. Emberly, “Dense neural networks for predicting chromatin conformation,” *BMC Bioinformatics*, 2018.
- [17] W. McKinney, “pandas: a Foundational Python Library for Data Analysis and Statistics,” *Python for High Performance and Scientific Computing*, 2011.
- [18] J. Howbert, *Introduction to Machine Learning*. University of Washington Bothell, 2012.
- [19] D. Chakraborty and H. Elzarka, “Advanced machine learning techniques for building performance simulation: a comparative analysis,” *Journal of Building Performance Simulation*, pp. 193–207, 2018.
- [20] J. Snoek, H. Larochelle, and P. R. Adams, “Practical Bayesian Optimization of Machine Learning Algorithms,” *The Lancet Public Health*, 2017.
- [21] E. Brochu, M. V. Cora, and N. de Freitas, “A Tutorial on Bayesian Optimization of Expensive Cost Functions, with Application to Active User Modeling and Hierarchical Reinforcement Learning,” 2010.
- [22] N. Challita, M. Khalil, and P. Beausery, “New feature selection method based on neural network and machine learning,” *2016 IEEE International Multidisciplinary Conference on Engineering Technology, IMCET 2016*, pp. 81–85, 2016.