SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 73

Uporaba metoda strojnog učenja za analizu podataka iz turističkog sektora

Mateo Elez

Zagreb, lipanj 2022.

Zahvlaljujem se mentoru prof. dr. sc. Damiru Pintaru na stručnoj pomoći i razumijevanju tijekom izrade završnog rada.

SADRŽAJ

1. Uvod

Tema ovog završnog rada je analiza podataka turističkog sektora. Taj se sektor sve više razvija stoga u ovom tehnološki naprednom razdoblju ta se znanja mogu primijeniti i u hotelijerstvu. Podatkovna znanost i analiza podataka postali su jako popularna tema primjenjiva u mnogim područjima. Hotelijerstvo, uz kvalitetnu analizu podataka i primjenu strojnog učenja u predikcijama, može lučiti velike benefite. Uvidom u rezervacije smještaja, te zahtjeve klijenta, hoteli i drugi ponuđaći turističkih usluga mogu donijeti poslovne odluke koje mogu pridonijeti boljem iskustvu klijenta te rezultirati većim profitima.

Na početku ovog rada opisano je strojno učenje, kao glavni "alat" za predviđanja koja radimo, te su opisani modeli koje koristimo. Odlučio sam se na 3 modela čije ću rezlutate usporediti. To su logistička regresija, naivni Bayes te stablo odluke. Nakon par riječi općenito o strojnom učenju, opisana su 3 modela. Nakon toga slijedi praktični rad, napisan u Pythonu. Taj se dio sastoji od prikupljanja podataka, eksploratorne analize te izrade preditktivnih modela.

2. Strojno učenje

Strojno učenje grana je umjetne inteligencije koja se bavi oblikovanjem algoritama koji svoju učinkovitost poboljšavaju na temelju empirijskih podataka. Dosege strojnog učenja izučava teorija računalnog učenja. Zašto se baš kaže da je upravo to dio jednih od najuzbudljivijih područja računarske znanosti? Uglavnom je to jer se koristi u razne svrhe. Samo nekoliko od njih bile bi raspoznavanje uzoraka i dubinske analize podataka, robotika, računalni vid, bioinformatika i računalna lingvistika. Računala su u prošlosti mogla raditi samo ono za što su bila programirana, zato je tu strojno učenje koje im omogućava učenje na način da stroj prikuplja znanje bazirano na prošlom iskustvu. Zahvaljujući tome, stroj je sada u mogućnosti samostalno poboljšavati vlastiti rad umjesto da mu se konstantno ažurira softver. Najbolje možemo shvatiti što to strojno učenje uopće radi na primjeru programa koji ima sposobnost razlikovanja sadržaja dviju slika. Da mu se prvo nekoliko slika jednoga pa drugoga i on će pronaći „način“ (uzorke) na koji će ih zapamtiti. Na temelju toga, idući put kada mu damo neke slike, moći će razlikovati sadržaj slike.

Zanimljivo je, a i očekivano, da smo svakodnevno u doticaju s njime, a ponajviše sa nama dobro poznatim društvenim mrežama. Jedan od najbanalnijih primjera bio bi Facebook koji prikazuje objave ovisno o interesima i prošlom ponašanju pojedinca na društvenoj mreži. Neizostavan, bio bi i Google koji može zahvaliti strojnom učenju radi poboljšanja preciznosti rezultata pretraživajna. Predviđanje životnog vijeka, organiziranje pacijentovih podataka pa čak i dijagnoza bolesti samo su neke od primjena strojnog učenja s kojima eksperimentira zdrastvena industrija. Doprijelo je i do tako zvanog biznis svijeta koji pomoću njega poboljšava korisničku službu.

2.1. Pristupi strojnom učenju

Postoje različiti pristupi strojnom učenju stoga ćemo u ovom odjeljku objasniti svaki od njih.

2.1.1. Nadzirano učenje

Nadzirano učenje (engl. *supervised learning*) je pristup strojnom učenju gdje na temelju ulaznih varijabli predviđamo rezultat i uspoređujemo s izlaznom varijablom koja je također poznata. Na temelju usporedbe dobivene predikcije te zadane izlazne vrijednosti na velikom skupu podataka možemo zaključiti koliko je naš model dobar. Na primjer, dobili smo ulaz, recimo fotografiju prometnog znaka, a zadatak nam je predvidjeti točan izlaz ili oznaku, na primjer koji je prometni znak prikazan na slici (ograničenje brzine, znak za zaustavljanje itd.). U najjednostavnijim slučajevima odgovori imaju oblik „da" ili „ne" (nazivamo ih problemi binarne klasifikacije). Osim što tom metodom algoritam možemo naučiti da predviđa točne oznake u problemu klasifikacije, nadzirano učenje možemo primjenjivati i u situacijama u kojima je predviđeni ishod broj. Primjeri su predviđanje broja osoba koje će otvoriti Googleov oglas na temelju njegova sadržaja i podataka o prethodnom ponašanju korisnika na internetu, predviđanje broja prometnih nesreća na temelju uvjeta na cesti i ograničenja brzine ili predviđanje prodajne cijene nekretnine na temelju njezine lokacije, veličine i stanja.

2.1.2. Nenadzirano učenje

Nenadzirano učenje (engl. *unsupervised learning*) ne uključuje ciljani rezultat što znači da sustav ne pruža nikakvu obuku. Koristi algoritme strojnog učenja koji izvlače zaključke o neoznačenim podacima. Zadatak je otkriti strukturu podataka: na primjer, grupirati slične elemente u klastere ili svesti podatke na mali broj važnih dimenzija. Nenadziranim učenjem može se smatrati i vizualizacija podataka. Glavni cilj učenja bez nadzora je pretraživanje entiteta kao što su skupine i klasteri te smanjenje dimenzionalnosti i procjena gustoće. Primjenom uobičajenih metoda nenadziranog učenja nastoji se utvrditi određena „struktura" podataka. To se može postići, na primjer, vizualizacijom: slični se elementi postavljaju jedan blizu drugoga, a različiti elementi dalje jedan od drugoga. Isto tako, to može podrazumijevati grupiranje, pri čemu se podaci upotrebljavaju za utvrđivanje grupa ili klastera elemenata koji su međusobno slični, ali se razlikuju od podataka iz drugih klastera.

2.1.3. Podržano učenje

Podržano učenje predstavlja dio strojnog ucenja koji se bavi optimizacijom ponašanja. Kod podržanog/ojačanog učenja algoritam uči pomoću mehanizma povratnih informacija i prošlih iskustava. Za razliku od nadziranog i nenadziranog ucenja, kod podržanog

učenja razmatramo interakciju agenta i okoline u kojoj se agent nalazi (2). Agent na temelju informacija iz okoline napravi akciju te za to dobije nagradu ako je akcija dobra, odnosno kaznu ako nije. Uglavnom se primjenjuje u situacijama u kojima određeni sustav umjetne inteligencije, kao što je autonomni automobil, mora raditi u nekom okruženju i u kojima su povratne informacije o prikladnim i neprikladnim odabirima dostupne sa zakašnjenjem. Primjenjuje se i u igrama u kojima se o ishodu odlučuje tek na kraju igre.

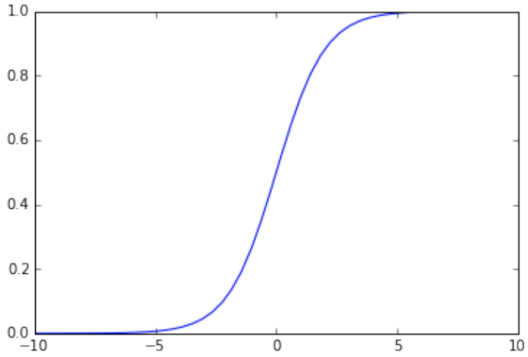
3. Metode strojnog učenja

U ovom odjeljku bit će objašnjene 3 metode strojnog učenja koje se koriste u izradi modela u ovom završnom radu, a to su logistička regresija, naivni Bayes te stablo odluke.

3.1. Logistička regresija

Logistička regresija (engl. *logistic regression*) je inačica linearne regresije u kojoj je zavisna varijabla isključivo dihotomna, tj. može poprimiti binarne vrijednosti 0 ili 1. Binarne vrijednosti, pridružene nominalnoj varijabli, označavaju pojavu nekog događaja ili prisutnost nekog atributa. Ideja je da upotrijebimo neku aktivacijsku funkciju, ali ne funkciju praga, kao kod perceptrona. Želimo "glatku funkciju praga" tako da je funkcija derivabilna na cijeloj vlastitoj domeni. Funkcija koja je idealna za to jest sigmoidna funkcija. Nju definiramo na slijedeći način:

$$\sigma(\alpha) = \frac{1}{1 + \exp(-\alpha)} \tag{3.1}$$



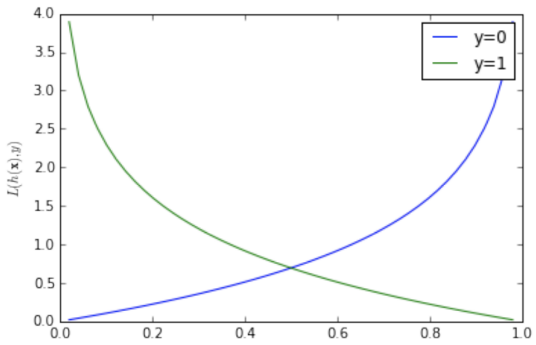
Slika 3.1: Graf sigmoidalne funkcije (7)

Sigmoida nam je važna zbog tri karakteristike. Vijednosti sažima na (otvoreni)

interval (0, 1), oblikom je slična funkciji praga, dakle davat će vrijednost blizu 1 primjerima iz jedne klase, a vrijednost blizu 0 primjerima iz druge klase i ništa manje bitno funkcija je derivabilna, što nam je važno za optimizaciju. Iz toga definiramo model logističke regresije:

$$h(x; w) = \sigma(w^T x) = \frac{1}{1 + \exp(w^T x)} \tag{3.2}$$

Za klasifikaciju nam treba gubitak koji je što sličniji gubitku 0-1, ali da je derivabilan. To bi bila funkcija koja daje relativno velik gubitak za primjere koji su pogrešno klasificirani i što manji gubitak za primjere koji su ispravno klasificirani i za koje je vjerojatnost P(y=1|x) što bliže jedinici. Taj gubitak definiramo pomoću kvadratnog gubitka. Iz toga slijedi da je minimizacija empirijske pogreške jednaka maksimizaciji (logaritma) vjerojatnosti oznaka. (7) Funkcije pogreške logističke regresije koju želimo dobiti izvest ćemo tako da krenemo od vjerojatnosti podataka. Napisat ćemo ju kao funkciju parametara i maksimizirati logiritam te funkcije po parametrima modela. Na kraju tog postupka dobit ćemo funkciju empirijske pogreške logističke regresije. Ta se funkcija pogreške naziva pogreška unakrsne entropije.



Slika 3.2: Graf funckije gubitka (7)

Funkcija gubitka unakrsne entropije (engl. *cross-entropy loss*) je funkcija dviju varijabli. Na ovom primjeru jedna je krivulja za y= 0, a druga za y = 1. Vidimo da što je gubitak bliže nuli to je h(x) bliži y, te što je gubitak veći to je h(x) dalje od y. Gubitka nema osim kada je primjer savršeno točno klasificiran stoga zaključujemo da uvijek postoji neki gubitak.

Bitno je primijetiti da s obzirom da su pogreška i gubitak povezani, može se krenuti od pogreške i izvesti gubitak, ali može se i obrnuto. U slučaju logističke regresije, ima više smisla krenuti od funkcije pogreške i potom doći do funkcije gubitka.

3.2. Naivni Bayesov klasifikator

Nakon što smo objasnili logističku regresiju, radimo još jedan probabilistički model strojnog učenja. Bayesov klasifikator nejjednostavniji je probabilistički model. Konkretno, obajsnit ćemo Gaussov Bayesov klasifikator jer ga koristimo u našem radu. On se koristi za kontinuirane značajke, a takvi su i naši podaci.

Glavno pravilo na kojem se temelji ovaj model jest Bayesovo pravilo.

$$P(y|x) = \frac{P(x,y)}{P(X)} = \frac{P(x|y)P(y)}{P(x)} \tag{3.3}$$

Praktična stvar kod Bayesovog pravila je abduktivno zaključivanje. To znači da možemo obrnuti smjer razmišljanja. U našem slučaju iz P(x|y) možemo zaključiti o vjerojatnosti P(y|x). U strojnom učenju to je dobro svojstvo jer omogućuje zaključivanje od posljedice prema uzorku.

Nakon definicije pravila, definirat ćemo glavni "dio" modela, a to je Bayesov klasifikator. On primjenjuje Bayesov teorem za računanje vjerojatnosti oznake y za zadani ulazni primjer x:

$$P(y|x) = \frac{p(x,y)}{p(X)} = \frac{p(x|y)P(y)}{p(x)} \tag{3.4}$$

Ova je formula slična gornjoj za pravilo, ali ovdje se pojavljuju veliko i malo p, gdje je veliko P vjerojatnost, a malo p je funkcija vjerojatnosti. Aposteriorna vjerojatnost oznake(engl. posterior): P(y|x) oznacava kolika je vjerojatnost da primjer x pripada klasi y. Njenu gustoću nazivamo izglednost klase (engl. class likelihood). Vjerojatnost klase neovisno o primjerima P(y) naziva se apriorna vjerojatnost klase (engl. class prior).

Odgovor na pitanje “koliko je vjerojatno da vidim x iz bilo koje klase” dat će nam gustoća vjerojatnosti p(x) koja predstavlja gustoću vjerojatnosti primjera neovisno o klasi. Konačno, dolazimo do modela Bayesovog klasifikatora.

$$h_i(x;w) = P(y=j|x) = \frac{p(x|y)P(y)}{\sum_{y'} p(x|y')P(y')} \tag{3.5}$$

Ukoliko želimo odrediti oznaku primjera, onda ćemo ga klasificirati u klasu čija je vjerojatnost najveća. U slučaju maksimum aposteriori hipoteza (MAP), model definiramo kao:

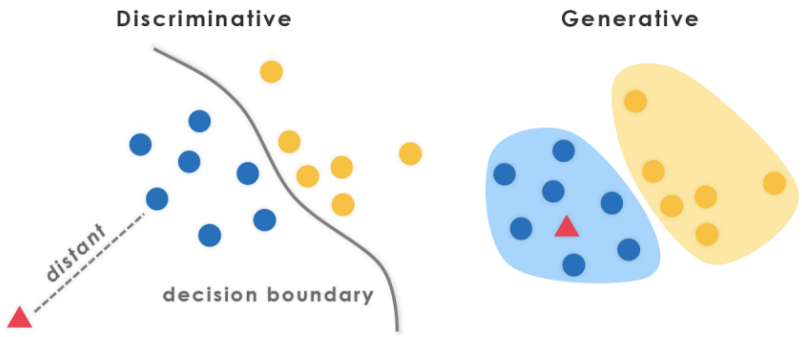
$$h(x;w) = \arg\max_{y} p(x|y)P(y) \tag{3.6}$$

Bayesov model je parametarski model. Model pretpostavlja da se primjeri x i oznake y pokoravaju nekoj teorijskoj vjerojatnosnoj distribuciji. Broj parametara tih distribucija ne ovisi o broju primjera, pa zato ni broj parametara cjelokupnog modela ne ovisi o broju primjera.

Bayesov je model generativan, za razliku od logističkog koji je diskriminativan. Sad ćemo vidjeti koja je razlika. Generativni modeli modeliraju zajedničku vjerojatnost (odnosno zajedničku gustoću vjerojatnosti) p(x,y). (4) Na temelju te vjerojatnosti može se vrlo jednostavno, primjenom Bayesovog pravila, izračunati aposteriorna vjerojatnost P(y|x), tj. vjerojatnost da primjer x pripada klasi y.

Takav pristup nazivamo generativnim jer modelira postupak generiranja (nastanka) podataka. Generativna priča objašnjava stohastički postupak(način na koji nastaju primjeri) generiranja podataka. Generativna priča također može se upotrijebiti za generiranje sintetičkih podataka. Jednom kada je model naučen, možemo slijediti generativnu priču kako bismo uzorkovali primjere iz zajedničke distribucije.

S druge strane postoje diskriminativni modeli koji izravno modeliraju aposteriornu vjerojatnost P(y|x), dok kod generativnih modela ju modeliramo indirektno, preko zajedničke vjerojatnosti! Logistička regresija je probabilistički diskriminativni model. Ona, kao i većina diskriminativnih modela, nisu probabilistički, ne modeliraju aposteriornu vjerojatnost, već direktno modeliraju granicu izmedu klasa. Razliku između dva modela najbolje prikazuje sljedeća slika:

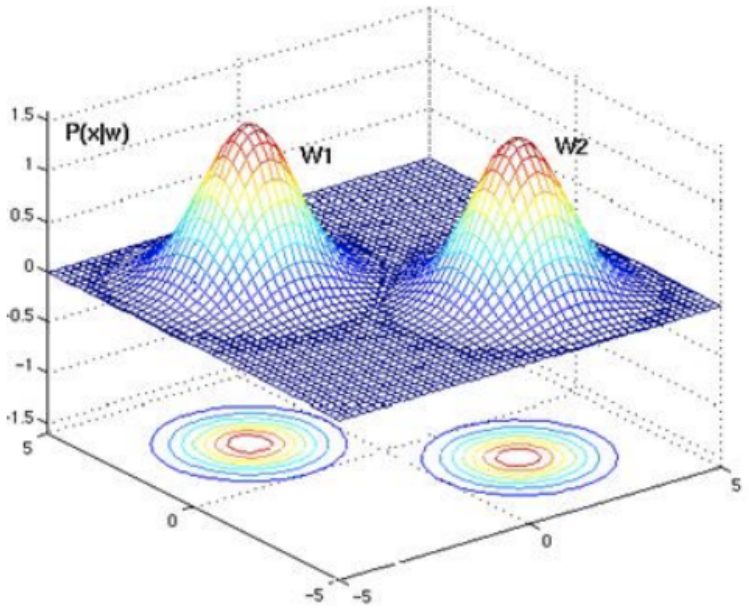


Slika 3.3: Razlika generativnog i diskriminativnog modela (4)

Možemo primjetiti da je prikazan problem binarne klasifikacije. Na lijevoj slici prikazan je rezultat dobiven diskriminativnim modelom, a desnoj rezultat dobiven generativnim modelom.

Bayesov klasifikator za kontinuirane značajke zove se Gaussov Bayesov klasifikator. Kod tog modela primjer x predstavljen je kao vektor brojeva, tj. značajke su

numeričke, što znači da ćemo izglednosti klasa P(x|y) modelirati Gaussovom distribucijom. Za dvije značajke odnosno dvodimenzijski ulazni prostor to izgleda kao na sljedečoj slici:



Slika 3.4: Gaussov Bayesov klasifikator (4)

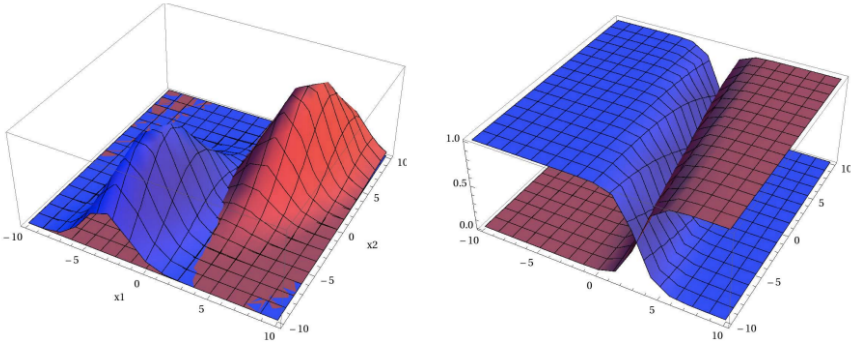
Izglednost svake klase modeliramo kao jednu zasebnu multivarijatnu Gaussovu distribuciju. Vektor μ ima najveće gustoću vjerojatnosti, on je prototipni primjer te klase. Svi primjeri koji pripadaju toj klasi trebali bi biti jednaki njemu, no zbog šuma to nije moguće. Zaključujemo da Gaussova distribucija ovdje modelira odstupanje od ideala uslijed šuma.

Naivan Bayesov klasifikator

Kod diskretnog klasifikatora, značajke su diskretne, što znači da su varijable x kategoričke (multinulijeve) varijable, tj. Beronoullijeve ukoliko se radi o samo dvije moguće vrijednosti. Razlika u odnosu na Gaussov Bayesov klasifikator je da p(x|y) tretiramo kao kategoričku razdiobu, čije su vrijednosti sve moguće kombinacije pojedinačnih kategoričkih varijabli. Parametre naivnog Bayesovog klasifikatora možemo procijeniti pomoću MLE ili MAP. (5)

Zovemo ga naivnim zbog pretpostavke o uvjetnoj nezvisnosti značajki unutar klase. Pretpostavka nekada može biti i prenaivna pa znamo korisiti polunaivan Bayesov klasifikator.

Logistička regresija zapravo je kontinuirani Bayesov klasifikator, ili, obrnuto, a kontinuirani Bayesov klasifikator zapravo je poopćeni linearni model. Pogledajmo sliku dvodimenzionalnog prostora koja nam prikazuje da istu granicu možemo dobiti logističkom regresijom ili Gaussovim Bayesovim klasifikatorom ako radimo binarnu klasifikaciju.



Slika 3.5: Logistička regresija i Bayesov klasifikator (5)

Lijeva slika prikazuje zajedničku gustoću vjerojatnosti, p(x, y), za dvije klase (plava i crvena), dok desna slika prikazuje aposteriornu vjerojatnost, p(y|x), za iste te dvije klase. Ovime smo povezali logističku regresiju (diskriminativan model koji izravno modelira aposteriornu vjerojatnost):

$$P(y = 1|x) = \sigma(w^T x) \tag{3.7}$$

i Gaussov Bayesov klasifikator (njoj odgovarajući generativni model koji tu istu vjerojatnost modelira neizravno:

$$P(y = 1|x) = \frac{p(x|y = 1)P(y = 1)}{p(x|y = 1)P(y = 1) + p(x|y = 0)P(y = 0)} \tag{3.8}$$

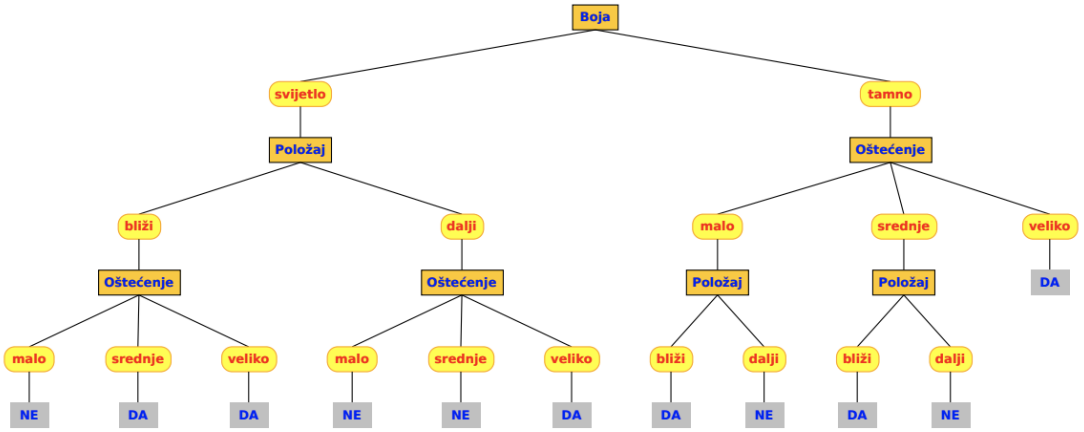
U strojnom učenju ima više takvih generativno-diskriminativnih parova modela. U kratkim crtama, Gaussov Bayesov klasifikator sa dijeljenom kovarijacijskom granicom daje istu granicu kao i logistička regresija, no ima više parametara.

3.3. Stablo odluke

Stabla odluke su formalizam koji omogućava rješavanje klasifikacijskih te aproksimacijskih zadataka temeljeći se na slijedu ispitivanja vrijednosti atributa primjerka.(3)

Korijen stabla je čvor u kojem se ispituje vrijednost određenog atributa. Čvor ima grana koliko vrijednosti taj atribut može poprimiti. Na kraju se dolazi do podstabla koji predstavlja vrijednost ciljnog atributa.

Za donošenje ispravne odluke ne moramo razmotriti vrijednosti svih atributa primjerka koji klasificiramo.Prilikom izgradnje stabala odluke htjet cemo da izgrađeno stablo ima određene karakteristike. Želimo da u većini slučajeva primjercima pridjeljuje vrijednosti ciljnog atributa kako je specificirano u primjercima za učenje. Ne želimo da to tako bude u svim slučajevima jer želimo stabla koja dobro generaliziraju, pa ako za neke primjerke utvrdimo da su stršeće vrijednosti, neće nas smetati što će konstruirano stablo njima pridijeljivati drugačiju vrijednost ciljnog atributa od one koja je zapisana u skupu primjeraka za učenje.



Slika 3.6: Primjer stabla odluke (3)

Klasifikatorska stabla mogu se graditi na različite načine. Jedan od njih je algoritam ID3. On se temelji na pojmovima entropije i informacijske dobiti. Entropija je mjera neuređenosti skupa. Entropiju skupa S definiramo s obzirom na vjerojatnosti da slučajno izvučeni primjerak pripada pojedinom razredu. Mjera je definirana kao negativna suma vjerojatnosti da primjerak pripada određenom razredu pomnoženoj dualnim logaritmom iste. Ako primjerci mogu pripadati jednom od K razreda, entropija skupa definira se izrazom:

$$Entropija(S) = \sum_{i \in S} p(c) \cdot \log_2 (p(c)) \tag{3.9}$$

Ideja algoritma ID3 jest u svakom koraku razmotriti kolika je korist ako se razmatrani skup podataka podijeli po svakom od atributa. Zatim pohlepno napravi podjelu skupa primjeraka po tom atributu, stvori čvor s oznakom odabranog atributa te rekurzivno gradi podstabla nad napravljenim podskupovima. Mjera kvalitete podjele koju koristi ID3 je informacijska dobit. Ona nam govori koliko smo uspjeli smanjiti neuređenost podjelom skupa u manje podskupove, a jednaka je entropiji početnog skupa umanjenoj za entropije napravljenih podskupova skalirane omjerom veličine podskupa i početnog skupa.

4. Praktični rad

Nakon što smo objasnili teorijsku pozadinu strojnog učenja, vidjet ćemo što smo radili u praktičnom dijelu rada. Kao što je spomenuto na početku, radi se o analizi turističkog sektora, točnije hotelijerstva. Cilj je bio napraviti što bolju i detaljniju analizu kako bi mogli donijeti neke poslovne odluke koje će s jedne strane poboljšati iskustvo klijenta u hotelima, a s druge strane pridonijeti većem profitu hotelima.

Praktični dio sastoji se od 3 dijela: prikupljanje podataka, eksploratorna analiza te strojno učenje.

Podatke o rezervacijama hotela uzeli smo s javne platforme kaggle.com. Privatne podatke koji se ne smiju spominjati zbog GDPR-a smo uklonili.

Programski dio napisan je cjelokupno u programskom jeziku Python u Jupyter notebook-u. Za upravljanje podacima korišten je alat Pandas. Eksploratornu analizu radio sam pomoću Seaborn biblioteke, dok sam modele strojnog učenja radio pomoću Scikit-learn.

Pandas je najkorištenija biblioteka u Pythonu za čišćenje podataka i upravljanje njima. Ona je tipa korištena za čitanje našeg skupa podataka sa web stranice kaggle i čišćenje nepotrebnih podataka iz skupa.

Seaborn je Python biblioteka za vizualizaciju podataka temeljena na matplotlib biblioteci. Koristi se za crtanje informativnih grafova, plotova, histograma i sl.

Scikit-learn najkorišteniji je alat za strojno učenje u programskom jeziku Python. Pruža optimizirane implementacije raznih modela za strojno učenje. Služi za rješavanje problema poput klasifikacije, regresije, grupiranja i smanjenja dimenzionalnosti.

Ta su tri alata koja koristimo međusobno kompatibilna i dobro povezana.

4.1. Prikupljanje podataka

Kao što smo rekli, podaci su dobiveni s kaggle web stranice. Dobili smo ih u .csv obliku. Pandas biblioteka ima naredbu readcsv za čitanje takvog formata podataka stoga smo nju koristili za to. Podaci su dobiveni od aplikacije za rezervaciju hotela

u razdoblju od 3 godine (2015 - 2017). Podaci su stvarni, zbog čega ćemo ukloniti osobne podatke poput imena, email arese, broja telefona... zbog poštivanja GDPR-a.

Nakon učitavanja podataka dobivamo skup od oko 120 tisuća redaka, te 36 varijabli. Podatke smo spremili u varijablu hotelBookingData.

Nakon učitavanja podataka, slijedi eksploratorna analiza.

4.2. Eksploratorna analiza

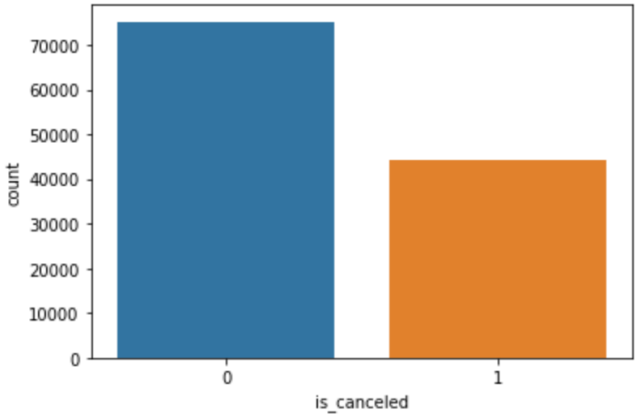
Kvalitetnom ekploratornom analizom možemo doći do značajnih zaključaka, te je ona neizostavan dio svake analize podataka.

Odmah na početku analize, samim pogledom na skup podataka u tablici, možemo zaključiti da su nam neke varijable bespotrebne. Takve ćemo izbaciti. Prvo izbacujemo sve osobne podatke zbog GDPR-a. Stupac 'market-segment' uklonjen je jer imamo 'distribution-channel' koji govori istu stvar, samo što razlikuje TO (Operator) i TA (Agent), što nam nije bitno. Stupac 'agent' je uklonjen jer nam nije potrebna informacija o ID-u turističke agnecije ili operatora. Stupac 'company' je uklonjen jer nam nije potrebna informacija o ID-u tvrtke, samo informacija da osoba ide preko tvrtke, što imamo u drugom stupcu.

Nakon izbacivanja tih i još nekih varijabli, sveli smo naš skup podataka na 23 varijable.

Sljedeće što moramo napraviti je upravljanje null vrijednostima. U ovom skupu podataka samo dva su stupca imala null vrijednosti, a to sam riješio popunjavanjem s "Unknown" na to mjesto.

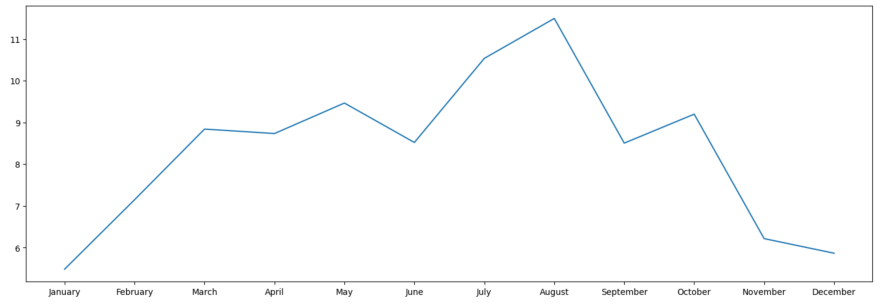
Nakon kratkog pregleda, čišćenja i uređivanja podataka, slijedi vizualizacija. Postoje univarijantna i multivarijanta analiza. U ovom primjeru samo je jedna univarijantna vizualizacija za prikaz razlike otkazanih i neotkazanih rezervacija. To nam je jedna od najbitnijih informacija jer na temelju toga možemo zaključiti tko je i zašto otkazao rezervaciju te to pokušati spriječiti u budućnosti.



Slika 4.1: Univarijantna analiza varijable `is-canceled`

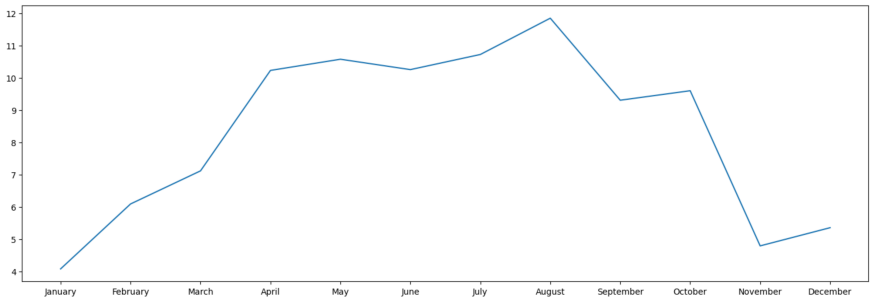
Vidimo da ima više neotkazanih rezervacija, što je i očekivano. Ali, razlika nije velika. Tijekom vizualizacije, promatrat ćemo što se događa s otkazanima, a što s neotkazanim rezervacijama stoga ćemo podijeliti skup podataka s obzirom na to. Osim toga, skup podataka možemo podijeliti i s obzirom na to je li hotel resort ili gradski, pa ćemo i to napraviti.

Hotelijerstvo je poznato po tome da se većina posla odvija tijekom sezone. Vidjet ćemo za naš skup podataka je li to tako. Na grafu ćemo vidjeti aktivnosti hotela u pojedinom mjesecu.



Slika 4.2: Prikaz rezervacija koje nisu otkazane po mjesecima

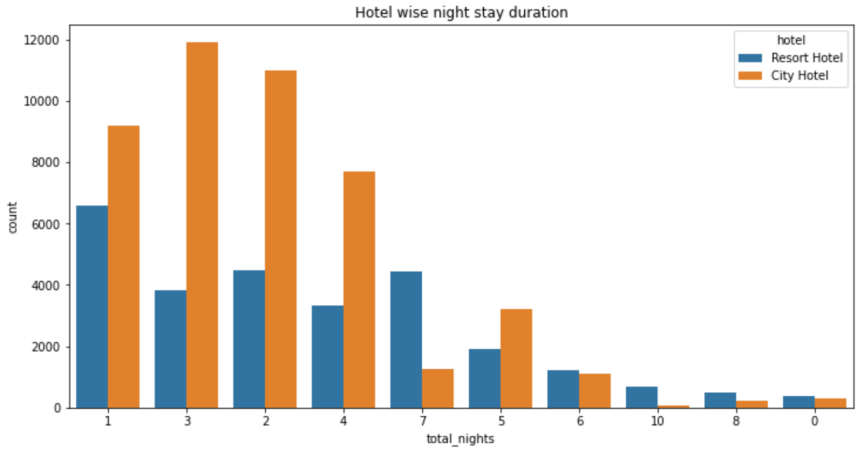
Očekivano, najviše rezervacija hotela je u ljetnoj sezoni (od 6. do 9. mjeseca), a najmanje preko zimskog razdoblja. Sad ćemo vidjeti taj raspored, ali sa skupom podataka otkazanih rezervacija, da bi mogli usporediti.



Slika 4.3: Prikaz otkazanih rezervacija po mjesecima

Možemo primjetiti da je broj otkazivanja veći u mjesecima od 4. do 9., kada je sezona, što nam govori da puno ljudi rezervira smještaj za ljeto i onda otkažu rezervaciju.

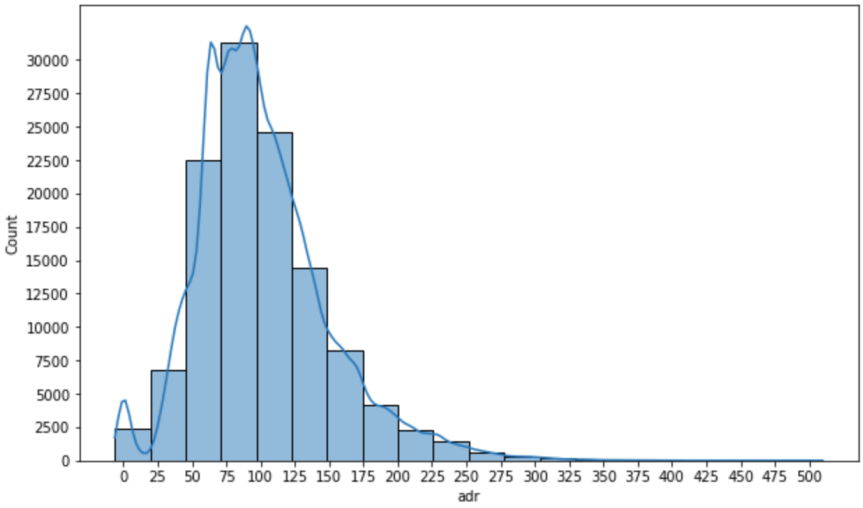
Sljedeće zanimljivo nam je broj noćenja u hotelu. Vidjet ćemo razliku u broju noćenja u resortima i gradskim hotelima. Očekuje se da u resortu bude više noćenja, dok se u gradskim hotelima očekuje manje. Također nas zanima broj noćenja otkazanih i neotkazanih rezervacija.



Slika 4.4: Broj noćenja

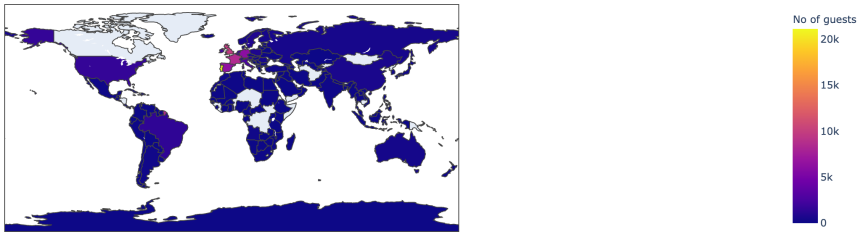
Iz priloženog vidimo da smo dobili očekivano, tj. da je za manje noćenja uvijek "pobijedio" gradski hotel, a za više noćenja resort.

Kao što smo rekli na početku, hotelima je cilj profitirati na kraju dana. Da bi mogli doći do toga kako profitirati, pogledat ćemo za početak kako se kreću cijene noćenja.



Slika 4.5: Cijene noćenja

Nakon toga pogledat ćemo i raspored rezervacija po državama da dobijemo prostorni dojam i vidimo možemo li nešto korisno iz toga dobiti. I tu ćemo također pogledati odvojeno za rezervacije koje su otkazane i one koje nisu.



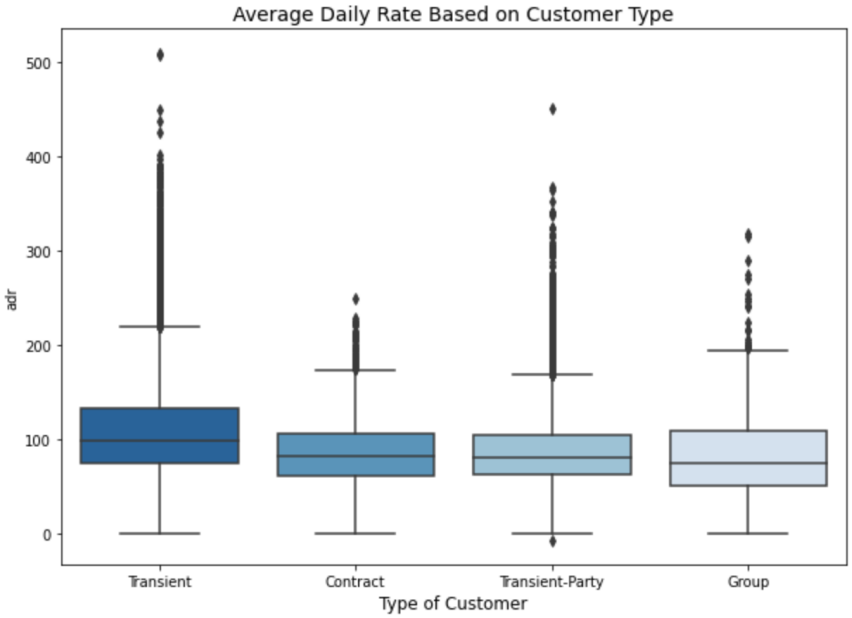
Slika 4.6: Prikaz geografske karte rezervacija koje nisu otkazane

Slika 4.7: Prikaz geografske karte otkazanih rezervacija hotela

U ovom primjeru vidimo da PRT (Portugal) ima čudno raspodijeljene podatke, tj. da je više ljudi otkazalo rezervaciju, i to 6 tisuća ljudi više.

Analiza s obzirom na tip posjetitelja

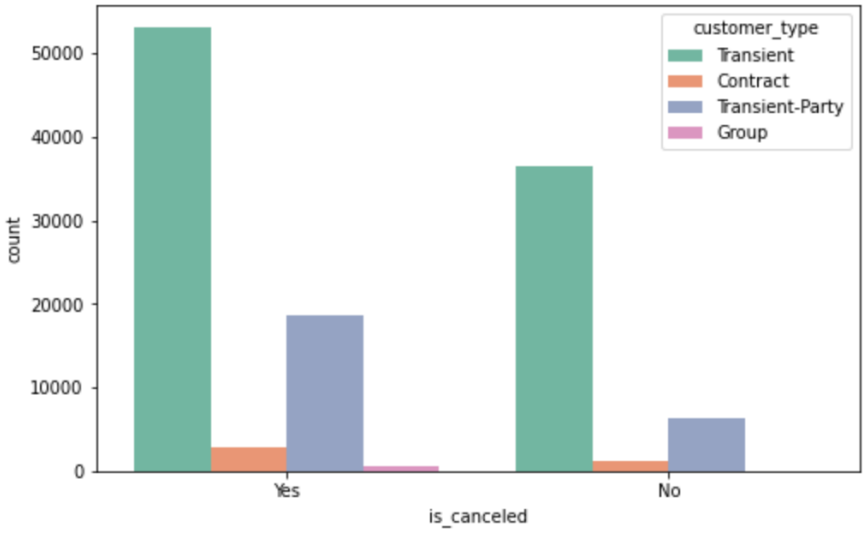
Sad ćemo pogledati kako se kreću otkazivanja rezervacija s obzirom na tip posjetitelja, pa vidjeti što iz toga možemo zaključiti. Tip posjetitelja može biti grupa, posjetitelj preko ugovora, te prijelazni (kada nije vezano ni za kakvu grupu niti ugovor). Za početak ćemo vidjeti kakve su cijene u ovisnosti o tipu posjetitelja.



Slika 4.8: Prikaz prosječne cijene noćenja s obzirom na tip posjetitelja

Možemo primjetiti da je najveća prosječna cijena kada se radi o pojedincu koji nije vezan za grupu ili za ugovor, dok su ostali slučajevi oko iste cijene. Tu možemo zaključiti da je bolje rezervirati hotel pojedincima bez ugovora, nego grupama, ako je moguće birati.

Sada gledamo u kakvom su odnosu tip posjetitelja i otkazivanje smještaja.



Slika 4.9: Prikaz broja otkazivanja rezervacija s obzirom na tip posjetitelja

Možemo vidjeti da grupne rezervacije nisu nikada bile otkazane. Sada bismo mogli donijeti odluku da se ipak prihvaćaju grupe prije pojedinaca bez ugovora zbog sigurnosti, što se kosi s prethodno donešenim zaključkom.

Također vidimo da je manje otkazivanja za rezervacije s ugovorom i za 'transient-party', koji predstavlja rezervaciju koja je prolazna, ali je povezana s nekom drugom rezervacijom.

Sljedeće gledamo odnos tipa posjetitelja i tipa rezerviranog hotela:

Slika 4.10: Odnos broja otkazivanja s obzirom na tip posjetitelja

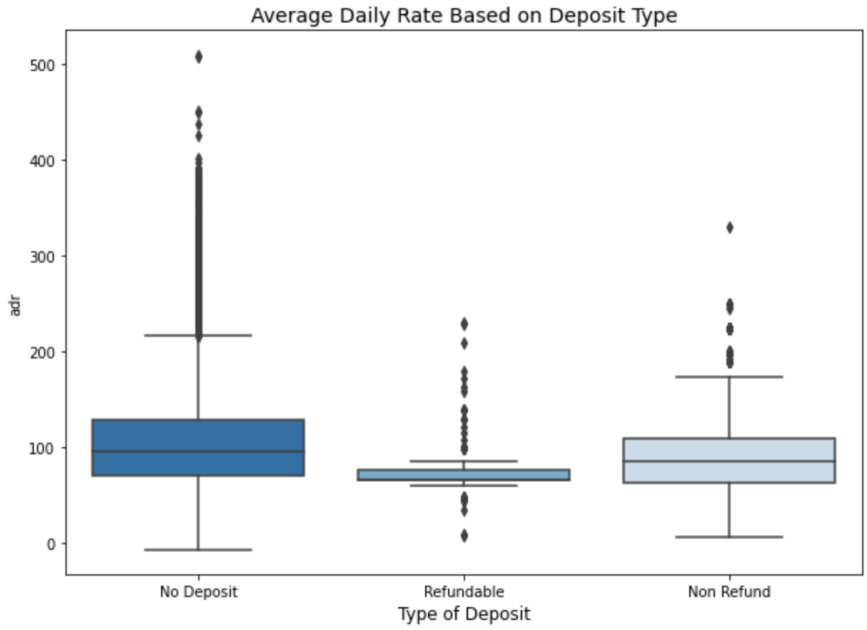
Dobiveni rezultati su donekle očekivani. Nema nekih velikih razlika u ovisnosti o tipu hotela.

Analiza s obzirom na vrstu depozita

Kad smo vidjeli za vrstu posjetitelja, radimo sličnu stvar za vrstu depozita. Postoje vrste:

- 'No Deposit' - bez depozita
- 'Refundable' - s depozitom koji je moguće dobiti nazad
- 'Non Refund' - s bespovratnim depozitom

Prvo gledamo odnos cijena noćenja i vrste depozita:

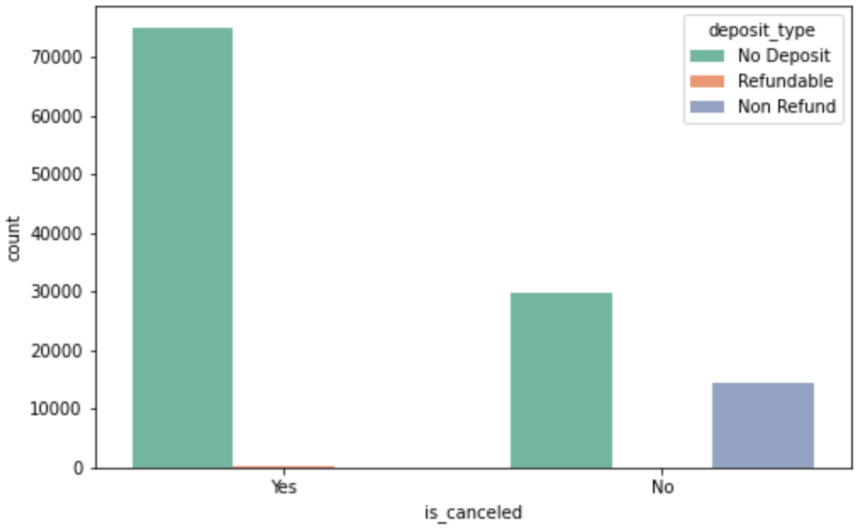


Slika 4.11: Prikaz odnosa cijene noćenja s obzirom na vrstu depozita

Prosječne cijene noćenja kod svih su vrsta depozita slične. Ono što možemo primjetiti sa slike jest da su klijenti spremni najviše platiti kada se ne traži depozit. Iz toga možemo zaključiti da klijenti ne vole kada trebaju dati novce prije dolaska u hotel, ali to je nešto što hotelima i drugim turističkim objektima daje sigurnost.

Klijent neće rezervirati ako nije siguran, a postoji bespovratan depozit koji iznosi mali postotak cijene rezervacije, a pružatelju usluga to pomaže zbog rasporeda noćenja i sigurnosti.

Odnos otkazivanja rezervacije i vrste depozita:



Slika 4.12: Prikaz broja otkazivanja rezervacija s obzirom na vrstu depozita

Kao što je i očekivano, rezervacije koje su Non-rufundable nisu otkazivane uglavnom. Da bi donijeli pametan zaključak, treba vidjeti je li pružatelju usluga bitnije da im se ne otkazuju rezervacije i sigurnost popunjenosti ili veća zarada.

Sljedeće gledamo odnos vrste depozita i vrste hotela:

Slika 4.13: Prikaz odnosa vrste depozita i vrste hotela

Vidimo da je puno više 'non-refundable' hotela zapravo gradskog tipa.

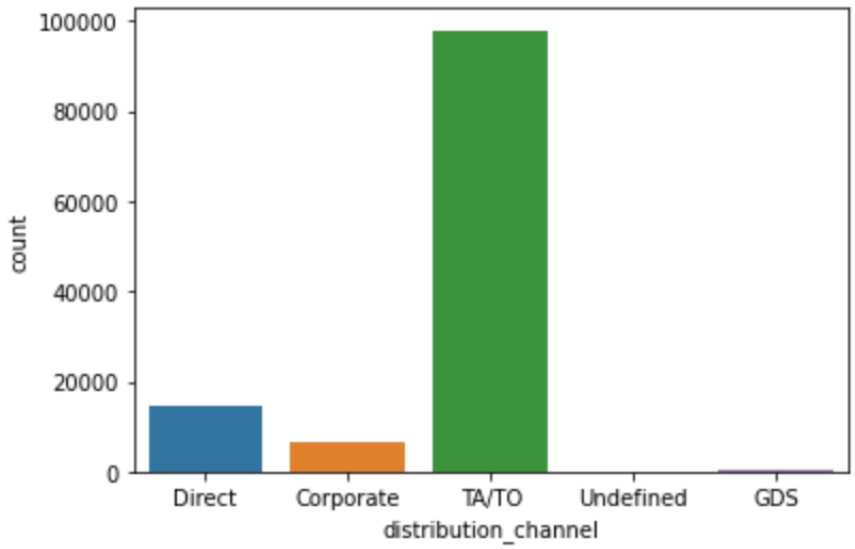
Analiza s obzirom na način rezerviranja

Sada gledamo podatke za distribucijski kanal, tj. način primanja rezervacija. Mogući načini za to su:
- direktni
- korporativni
- putničke agencije
- GDS -> Global Distribution System

Jedan od načina rezervacije je direktno. To može biti putem web stranice, ili putem direktnog poziva. To je najčešći i najlakši način rezervacije za event planere, putniče agente te starije ljude koji su se tako navikli. Zgodno je jer se odmah mogu postaviti sva pitanja koja ih zanimaju, a ne treba čekati nekoga za odgovor.

Putničke agnecije poput Bookinga i Airbnb-a su danas najčešće, kao što je vidljivo u donjem grafu.

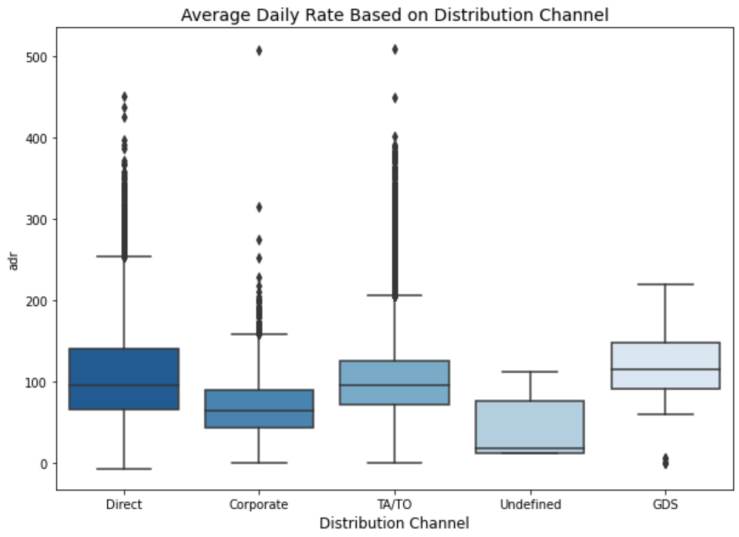
GDS je način rezervacije koji putnički agenti najviše koriste.



Slika 4.14: Prikaz broja otkazivanja rezervacija s obzirom na način rezervacije

Očekivano, najviše je rezervacija preko danas najpoznatijih turističkih operatora, a nakon toga slijedi direktno rezerviranje smještaja.

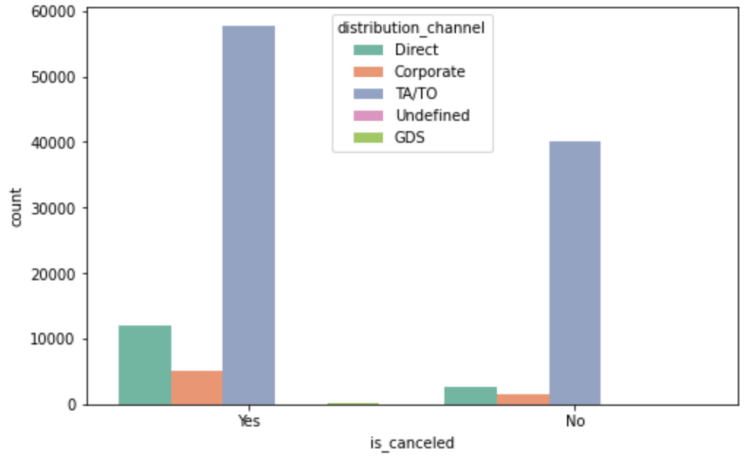
Cijena noćenja u odnosu na način širenja:



Slika 4.15: Prikaz cijene noćenja u odnosu na način rezervacije

Najskuplje sobe su one koje dolaze preko GDS-a, što nam govori da su putnički agenti oni koji idu u prosjeku u najskuplje. Za korporativni način rezervacije vidimo da je cijena noćenja najmanja.

Sada gledamo odnos načina rezervacije i otkazivanja:

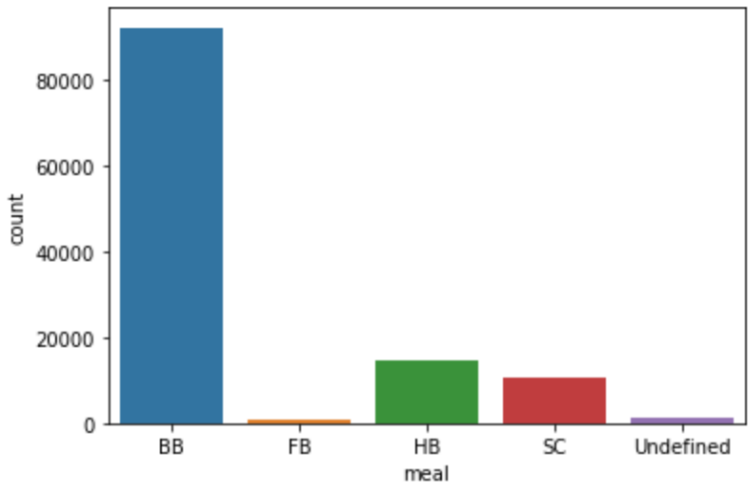


Slika 4.16: Prikaz broja otkazivanja rezervacija s obzirom na način rezervacije

Nismo dobili ništa neočekivano. Vidimo da su direktne naruđbe dobra praksa za hotele jer je najmanje otkazivanja rezervacija koje su dobivene direktno. Također, agenti koji rezerviraju pomoću GDS-a, nemaju praksu otkazivati rezervaciju.

Analiza s obzirom na paket sobe

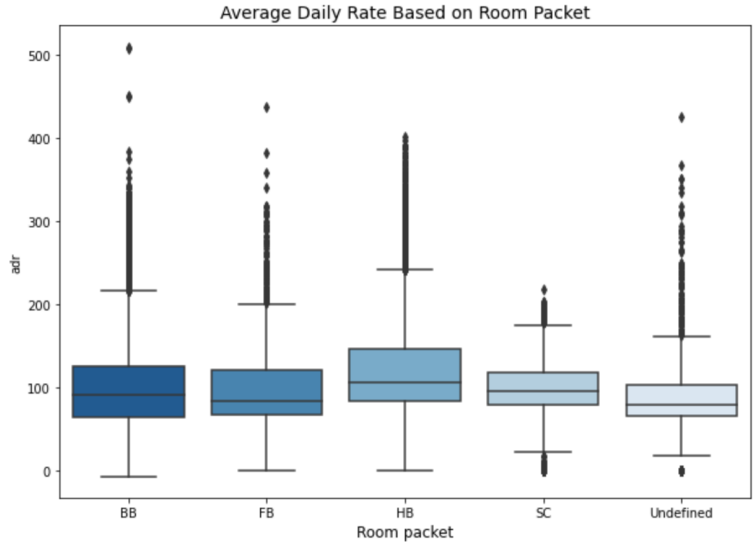
Sada gledamo razlike u ovisnosti o vrsti paketa (Room Only / Breakfast and Bed...)
- BB predstavlja paket s uključenim doručkom
- FB predstavlja paket s uključenim doručkom, ručkom i večerom
- HB predstavlja paket s uključenim doručkom i večerom
- SC predstavlja paket bez hrane



Slika 4.17: Prikaz broja smještaja u ovisnosti o vrsti paketa

Vidimo da je najčešća ponuda BB (’Bed and Breakfast’), a jako rijetka ponuda ’Full Board’.

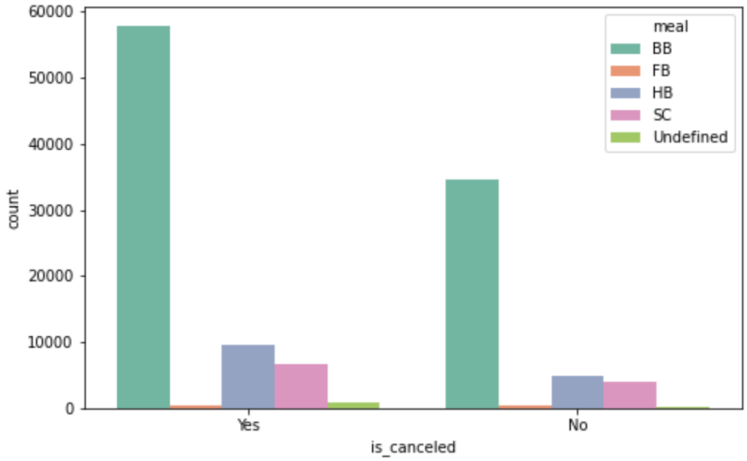
Sada gledamo cijene u ovisnosti o vrsti paketa sobe.



Slika 4.18: Prikaz cijene u ovisnosti o vrsti paketa sobe

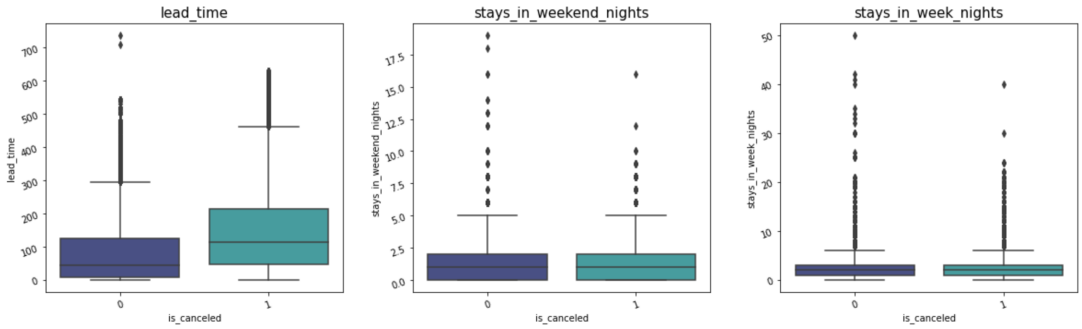
Vidimo da su prosječne cijene oko slične razine s obzirom na paket sobe. Najveće su cijene sobe s paketon ručka i večere.

Za kraj gledamo broj otkazivanja rezervacija u ovisnosti o vrsti paketa sobe.



Slika 4.19: Prikaz broja otkazivanja rezervacija s obzirom na vrstu paketa sobe

Za kraj ove ekploratorne analize ćemo vidjeti kako se mijenja otkazivanje u ovisnosti o ostalim varijablama koje nismo analizirali još. Gledamo vrijeme proteklo od rezervacije do dolaska, te koliko je noći provedeno tijekom vikenda/tijekom tjedna.



Slika 4.20: Prikaz broja otkazivanja rezervacija s obzirom na 3 navedene varijable

Što se tiče vremena proteklog od rezervacije do dolaska, vidimo da ono igra veliku ulogu. Rezervacije koje su rezervirane puno vremena prije dolaska imaju puno više otkazivanja. Iz toga možemo zaključiti da ne treba biti siguran u dolazak ako se netko rezervira puno mjeseci prije.

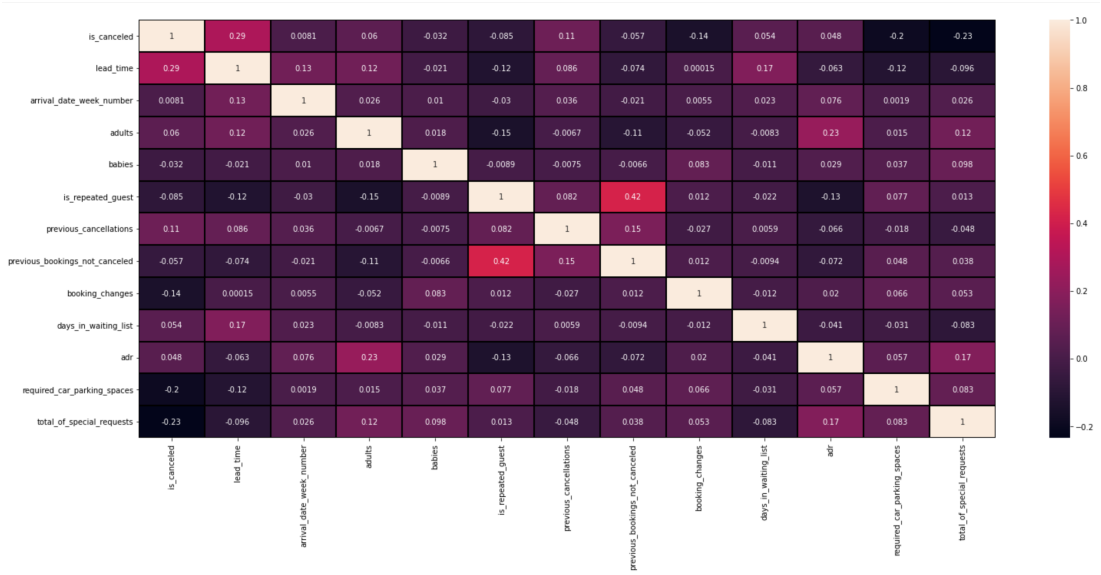
Dobra ideja za to je uvesti neki manji postotak plaćanja otkazivanja, što će natjerati ljude da ne rezerviraju smještaj bezveze, ili 'za svaki slučaj'.

Za druge dvije varijable koje nam govore o noćima preko tjedna / vikenda vidimo da nema nikakve razlike. Tu možemo zaključiti da nam one ne igraju ulogu, te ćemo ih zbog preglednosti maknuti iz skupa podataka.

4.3. Modeli strojnog učenja

Nakon odrađene eksploratorne analize te usporedbe velikog broja varijable kroz grafove i histograme, imajući na umu ono što smo analizirali i saznali prethodno, krećemo na izradu prediktinvih modela. Koristimo na početku navedeno i detaljno opisano strojno učenje. Tri metode koje koristimo su logistička regresija, naivni Bayesov klasifikator te stablo odluke koji su prethodno također detaljno opisani. Sada ćemo pogledati izradu modela, treniranje, te testiranje.

Metode koje koristimo pretpostavljaju nezavisnost podataka, ali ako su podaci malo zavisni ne smeta im previše. Zbog tog razloga ćemo vidjeti korelaciju među svim varijablama te maknuti one koje nam se čine da "jako" koreliraju.



Slika 4.21: Tablica korelacija

Podatke smo sredili tako da smo uklonili jednu od varijabli koje jako koreliraju. Nakon toga krećemo na izradu prediktivnih modela.

4.3.1. Model logističke regresije

Za početak radimo model logističke regresije. Ona prima samo numeričke varijable stoga moramo sve kategoričke pretvoriti u numeričke. To radimo mapiranjem svih

kategoričkih varijabli.

Za korištenje pojedinih metoda strojnog učenja moramo napraviti standardizaciju podataka. To radimo pomoću klase StandardScaler koji smo uzeli iz sklearn.preprocessing.

Standardiazacija podataka se radi zbog toga da svedemo sve podatke na zajedničku ljestvicu, te da imamo iste razlike u rasponu vrijednosti. U izradi prediktivnih modela koristimo udaljenosti među vrijednostima, stoga je ovaj korak svođenja podataka na zajedničku ljestvicu neizostavan korak pripreme podataka za izradu modela strojnog učenja. Osim toga, treniranje modela s takvim (standardiziranim) podacima je brže u praksi. Za takav proces koriste se dvije različite tehnike u praksi:

-> Normalizacija

-> Standardizacija

Normalizacija je tehnika min-max skaliranja, gdje se uzimaju minimalna i maksimalna vrijednost te se nova točka traži na ovakav način:

$$X_{new} = \frac{(X - X_{min})}{(X_{max} - X_{min})} \tag{4.1}$$

Normalizacija je korisnija kada nema outlier-a.

Standardizacija je tehnika oduzimanja od srednje vrijednosti i dijeljenja sa standardnom devijacijom, a nova točka računa se ovako:

$$X_{new} = \frac{(X - mean)}{Std} \tag{4.2}$$

Standardizacija je češća u primjeni jer nije osjetljiva na outlier-e, a oni su u praksi česta pojava

Nakon standardizacije podataka, možemo krenuti s izradom prediktivnih modela. Za početak ćemo podijeliti naš skup podataka na one za treniranje i testiranje. To se najčešće radi tako da uzmemo 70% za treniranje, 30% za testiranje, što možemo vidjeti u sljedećem kodu:

```
X = hotelBookingData.loc[:, hotelBookingData.columns != 'is_canceled'].copy()
y = hotelBookingData.loc[:, hotelBookingData.columns == 'is_canceled'].copy()

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=0)
```

Listing 4.1: Podjela podataka na skup za treniranje i skup za testiranje

Gledamo je li nam otprilike isti postotak ljudi yes/no subscribed u treniranom i testnom modelu, kako nebi doslo do overfittinga (značenje - "izrada analize koja preblizu ili u potpunosti odgovara određenom skupu podataka, pa stoga možda neće uspjeti uklopiti dodatne podatke ili pouzdano predvidjeti buduća opažanja".)

```
Trenirani model:
0    0.631723
1    0.368277
Name: is_canceled, dtype: float64
Testni model:
0    0.624592
1    0.375408
Name: is_canceled, dtype: float64
```

Slika 4.22: Količina treniranih i testiranih primjera

Zaključak je da su otprilike isto raspoređeni, tako da nastavljamo dalje.

Provodimo treniranje skupa podataka:

```
from sklearn.linear_model import LogisticRegression

classifier= LogisticRegression()
classifier.fit(X_train, y_train)
```

Listing 4.2: Treniranje modela logističke regresije

Na gornjem kodu vidimo kako se trenira model. Definiramo varijablu koju izjednačimo s funkcijom iz `sklearn.linear_model`. Nakon toga nad tom varijablom pozivamo fit funkciju kojoj šaljemo x i y podatke za treniranje.

Nakon toga slijedi predviđanje modela pomoću testnih podataka i usporedba s pravom izlaznom vrijednošću. Predviđanje radimo na ovaj način:

```
y_pred_Logistic = classifier.predict(X_test)
```

Listing 4.3: Testiranje modela logističke regresije

4.3.2. Model naivnog Bayesovog klasifikatora

Slično radimo i kod ostalih modela, pa tako i kod naivnog Bayesovog klasifikatora. Na početku podatke opet dijelimo na one za treniranje i za testiranje, u istom omjeru, na isti način kao prethodno.

Treniranje modela naravno nije isto jer se ne radi o istom modelu. U ovom slučaju treniranje izgleda ovako:

```
from sklearn.naive_bayes import GaussianNB

gnb = GaussianNB()
y_pred_Bayes = gnb.fit(X_train, y_train)
```

Listing 4.4: Treniranje modela naivnog Bayesovog klasifikatora

Za treniranje modela u ovom smo slučaju koristili `GaussianNB()` metodu iz `sklearn.naive_bayes`.

Za testiranje kao i prethodno koristimo metodu predict kojoj dajemo skup podataka iz dijela za testiranje.

4.3.3. Model stabla odluke

I za kraj, naš zadnji prediktivni model je stablo odluke. Priprema podataka i podjela podataka na one za treniranje i testiranje je također ista kao i prethodno.

Pogledat ćemo koju smo ovdje metodu koristili za treniranje podataka:

```
from sklearn import tree

clf = tree.DecisionTreeClassifier()
clf = clf.fit(X_train, y_train)
```

Listing 4.5: Treniranje modela stabla odluke

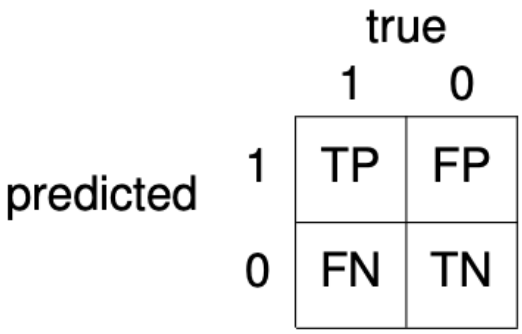
U ovom slučaju koristimo metodu `DecisionTreeClassifier()`

U sva tri slučaja smo ispisivali rezultate koje ćemo pogledati u sljedećem odjeljku. Ispisivali smo rezultat i klasifikacijsku tablicu, te matricu zabune.

4.4. Rezultati modela

Prije nego vidimo rezultate naših modela, pogledat ćemo pomoću čega uspoređujemo modele i gledamo koji je bolji. Postoji više kriterija za uspoređivanje rezultata modela, a mi ćemo pogledati točnost, preciznost te odziv.

Te ćemo pojmove objasniti pomoću matrice zabune.



Slika 4.23: Matrica zabune(1)

Matrica se sastoji od četiri kategorije:

– TP - true positive

– FP - false positive

– FN - false negative

– TN - true negative

Točnost (engl. *accuracy*) je udio točno klasificiranih primjera u skupu svih primjera(6):

$$\textit{točnost} = \frac{TP + TN}{TP + FP + FN + TN} \tag{4.3}$$

Preciznost pokazuje podatak o udjelu stvarno pozitivnih klasifikacija u svim pozitivnim klasifikacijama (6) i može se zapisati kao:

$$\textit{preciznost} = \frac{TP}{TP + FP} \tag{4.4}$$

Odziv je definiran kao udio stvarno pozitivnih klasifikacija u svim pozitivnim oznakama (6) odnosno:

$$\textit{odziv} = \frac{TP}{TP + FN} \tag{4.5}$$

Kao što je spomenuto radio sam 3 metode strojnog učenja, pa ćemo sada prikazati i usporediti rezultate za modele u kojima smo za regresore uzeli sve ulazne varijable.

```
Score for logistic regression model is:  0.790434709774688
Classification report table:
              precision    recall  f1-score   support

           0       0.78      0.93      0.85     22371
           1       0.82      0.56      0.67     13446

    accuracy                           0.79     35817
   macro avg       0.80      0.74      0.76     35817
weighted avg       0.80      0.79      0.78     35817
```

Slika 4.24: Rezultat modela logističke regresije

Dobiveni je rezultat 79% što je jako dobar rezultat u poslovnom svijetu gdje ima mnogo outliera.

```
Score for Naive Bayes model is:  0.5488176005807298
Classification report table:
              precision    recall  f1-score   support

           0       0.86      0.33      0.48     22371
           1       0.45      0.91      0.60     13446

    accuracy                           0.55     35817
   macro avg       0.66      0.62      0.54     35817
weighted avg       0.71      0.55      0.53     35817
```

Slika 4.25: Rezultat modela naivnog Bayesa

Za drugi model imamo malo lošiji rezultat nego prethodni, a to je 54.88% što također nije toliko loše. U tablici po preciznosti možemo primjetiti da je problem krivog pretpostavljanja otkazivanja kada zapravo rezervaicija ne bude otkazana. Mogući razlog zbog kojeg je logistička regresija dala značajno bolji rezultat od naivnog Bayesa u svakom kriteriju jest to što naivni Bayesov klasifikator očekuje nezavisne varijable, a naše varijable nisu skroz nezavisne, što se može primijetiti u gornjoj tablici korelacija.

Sada gledamo rezultat modela stabla odluke:

```
Score for Decision Tree model is:  0.8118212022224084
Classification report table:
              precision    recall  f1-score   support

           0       0.85      0.85      0.85     22371
           1       0.75      0.75      0.75     13446

    accuracy                           0.81     35817
   macro avg       0.80      0.80      0.80     35817
weighted avg       0.81      0.81      0.81     35817
```

Slika 4.26: Rezultat modela odluke stabla

Za ovaj model imamo najbolje rješenje, 81.18%. Stablo odluke radi dobro kad je skup podataka velik, što je u našem slučaju istina.

Iz svega navedenog možemo zaključiti da je najtočniji i najpreciznij model stabla odluke, ali ne puno bolji od modela logističke regresije. Za model naivnog Bayesa možemo reći da zaostaje za dvama ostalima. Za kraj ću koristeći stablo odluke, napraviti model koji na temelju puno manje varijabli može predvidjeti rezultat sa sličnom točnošću i preciznošću.

Promatrajući cijelu vizualizaciju napravljeno prethodno i na temelju tih saznanja odlučio sam se za ulazne varijable koristiti samo dvije: 'lead-time' te 'arrival-date-week-number'.

Rezultat koji sam dobio je jako dobar:

```
Score for decission tree model is:  0.7499511405198649
Classification report table:
              precision    recall  f1-score   support

           0       0.76      0.88      0.82     22371
           1       0.73      0.53      0.61     13446

    accuracy                           0.75     35817
   macro avg       0.74      0.71      0.71     35817
weighted avg       0.75      0.75      0.74     35817
```

Slika 4.27: Rezultat drugog modela odluke stabla

Rezultat je 75%, što je samo 6 posto manje nego kada smo koristili 18 ulaznih varijabli.

Nakon svih ispitanih modela možemo zaključiti da su varijable 'lead-time' i 'arrival-date-week-number' najrelevantnije kod predviđanja otkazivanja rezervacije. Iz toga se

da zaključiti da je manja vjerojatnost otkazivanja ako je rezervacija napravljena manje dana prije potencijalnog dolaska.

5. Zaključak

Turistički sektor je vrlo rasprostranjeno područje koje konstantno napreduje, a uz analizu podataka i strojno učenje, taj napredak može biti još brži. Cilj ovog završnog rada bio je napraviti detaljnu analizu i izraditi prediktivne modele za predikciju hoće li klijent otkazati rezervaciju ili ne.

Dobiveni skup podataka smo čistili i preslagivali kako bi pomoću alata za vizualizaciju prikazali što korisniji prikaz. Kroz eksploratornu analizu mogli smo vidjeti odnose između varijabli i na temelju toga reći pružatelju usluga što mu je pametnije raditi i kada. Eksploratorna je analiza također pomogla poslije u izradi modela pri odabiru varijabli koje ćemo koristiti kao ulzane u našim modelima.

Strojno učenje je nastalo jer ono rješava neke probleme koje računalo do sada nije znalo riješiti. Jedan od takvih problema je i predviđanje otkazivanja rezervacije u hotelima. Kroz ovaj rad testirali smo 3 metode i zaključili da je najbolje rezultate dao model stabla odluke. Po kriterijima, malo lošiji ali vrlo blizu, bila je logistička regresija. Na žalost, naivni Bayesov klaisifikator nije dao najbolje reziultate predviđanja, što možemo "opravdati" time što nam varijable nisu nezavisne kao što taj model očekuje. Na kraju smo uspjeli model stabla odluke s 18 ulaznih varjabli svesti na samo dvije ulazne varijable, a da se točnost promijeni samo sa 81% na 75%. Vlasnicima hotela ta informacija u budućnosti može dobro doći.

LITERATURA

[1] Andro Merćep Tomislav Đuričić. *Predavanje Vrednovanje modela iz predmet Strojno učenje na FER-u 2015*, 2015. URL https://www.fer.unizg.hr/_download/repository/SU-2015-Vrednovanje_modela.pdf.

[2] Marko Čupić. *Skripta Uvod u strojno učenje*. URL http://java.zemris.fer.hr/nastava/ui/ml/ml-20220430.pdf.

[3] Marko Čupić. *Odlomak Stablo odluke iz skripte Uvod u strojno učenje*, 2021./2022. URL http://java.zemris.fer.hr/nastava/ui/ml/ml-20220430.pdf.

[4] Jan Šnajder. *Predavanje Bayesov klasifikator iz predmet Strojno učenje 1 na FER-u*, 2021.. URL https://www.fer.unizg.hr/_download/repository/SU1-2021-P15-BayesovKlasifikator.pdf.

[5] Jan Šnajder. *Predavanje Bayesov klasifikator 2 iz predmet Strojno učenje 1 na FER-u*, 2021.. URL https://www.fer.unizg.hr/_download/repository/SU1-2021-P16-BayesovKlasifikator2.pdf.

[6] Jan Šnajder. *Predavanje Vrednovanje modela iz predmet Strojno učenje 1 na FER-u 2021*, 2021.. URL https://www.fer.unizg.hr/_download/repository/SU1-2021-P21-VrednovanjeModela.pdf.

[7] Jan Šnajder. *Predavanje Logistička regresija 1 iz predeta Strojno učenje 1*, 2021.. URL https://www.fer.unizg.hr/_download/repository/SU1-2021-P06-LogistickaRegresija.pdf.

,

Uporaba metoda strojnog učenja za analizu podataka iz turističkog sektora

Sažetak

Turistički sektor može polučiti velike benefite od kvalitetne primjene analize podataka. Kvalitetan uvid u dinamiku rezervacija smještaja, potražnju i potrebe klijenata može ponuđaču turističkih usluga dati mogućnost donošenja poslovnih odluka koje će s jedne strane dodatno podići kvalitetu ponude i zadovoljstvo klijenata, a s druge rezultirati većim profitima. Zadatak ovog rada jest prikupiti podatkovni skup povezan uz turistički sektor, provesti eksploratornu analizu nad tim podacima i razviti prediktivni model za odabrane zavisne varijable. Konačno rješenje potrebno je realizirati u obliku programske skripte koje će implementirati razvijene modele i evaluirati njihovu učinkovitost uz komparativnu analizu korištenih metoda.

Ključne riječi: eksploratorna analiza, analiza podataka, strojno učenje, logistička regresija, naivni Bayesov klasifikator, stablo odluke

Analyzing Data from the Tourist Sector Domain Using Machine Learning Methods

Abstract

The tourism sector can benefit from the quality application of data analysis. Quality insight into the dynamics of accommodation reservations, demand and customer needs can give the tourism service provider the opportunity to make business decisions that will on the one hand, further raise the quality of the offer and customer satisfaction, and on the other hand, result higher profits. The task of this paper is to collect a data set related to the tourism sector, to implement exploratory analysis of these data and develop a predictive model for selected dependent variables. The final solution needs to be implemented in the form of a program script that will implement the developed models and evaluate their effectiveness with comparative analysis of used methods.

Keywords: exploratory analysis, data analysis, machine learning, logistic regression, naive Bayes classifier, decision tree