# Projekat CarVana

Naše rešenje projektnog zadatka na izabranom problemu predstavićemo pomoću CRISP-DM metodologije, s tim što će poslednji korak (Primena) izostaviti obzirom da ga nismo u situaciji da ga isporučimo i integrišemo u poslovni proces.

## Opis poslovnog problema

Carvana je američka kompanija koja se bavi online prodajom polovnih automobila, sa središtem u gradu Tempe, Arizona. Ljudi kada kupuju polovna kola, rade to kako bi uštedeli, s obzirom na veliku razliku u ceni novih i polovnih automobila. Na izbor automobila utiču brojni faktori- poput starosti i istorije auta, da li moguće utvrditi prethodne vlasnike, njegovo stanje.

Glavno pitanje je da li je odluka o kupovini nekog automobila bila dobra ili ne. Pozabavićemo se ovim problemom, na osnovu niza različitih faktora koje je Carvana stavila na raspolaganje, poput prosečne cene na aukciji, modela, starosti... Za to nam je dostupno nešto manje od sedam hiljada transakcija.

Pitanje je možemo li brže i preciznije da procenimo auto pre nego što ga drugi dileri kupe što će uticati na povećanje konkurentnosti kompanije, kao i na smanjenje relativnih troškova lose kupovine.

Carvana auto prodaje po nižim cenama od drugih dilera što verovatno znači da se oslanjaju na troškovnu prednost ekonomije obima i da veći akcenat stavljaju na broju prodatih vozila nego na maržu koju su ostvarili.

## Opis podataka

U podacima koji su nam dati ima ukupno 34 atributa tj kolona, neki od njih su model, submodel, starost vozila, poreklo vozila, boja, i slično. Atribut po kojem vršimo klasifikaciju je “IsBadBuy” odnosno da li je bila loša kupovina, 0 označava da nije tj da je bila dobra kupovina dok 1 predstavlja lošu.

Takođe, uočavamo da je dosta atributa sakriveno, pre svega u koloni “Model” i “SubModel” i da će biti neophodno neke izdvojiti.

Još jedan od izazova je nebalansiranost podata. Tek nešto oko 10% transakcija pripada negativnoj klasi. Naredni problem koji uočavamo je veliki broj nedostajućih vrednosti za atribute PRIMEUNIT- da li bi vozilo imalo veću potražnji pri uobičajenoj kupovini, AUCGUART- nivo garancije koji se dobija kupovinom auta na aukciji i VNST-država u kojoj je auto kupljen. Iako su procentualno nedostajuće vrednosti drže u opsegu manjim od 10% što je neka gornja granica, svakako predstavlja izazov.

## Priprema podataka

Izbačeni atributi VehYear, BYRNO, RefId, WheelType, PurchDate (sačuvana godina kupovine).

WheelTypeID pretvoren u celobrojnu vrednost umesto vrednosti u pokretnom zarezu. Veličina kola (atr. Size) je grupisana u veliku (Large, Crossover, Van, Large Truck, Large SUV, Specialty), srednju (Medium SUV, Medium) i malu (Compact, Small SUV, Sports, Small Truck). Sve boje koje se pojavljuju u manje od 500 slučajeva grupisane su vrednost Retke. Proizvođači automobila (atr. Make) kojih je manje od sto stavljeni su grupaciju Egzotični. Iz kolone SubModel izvučen je broj vrata (od dva do pet). Iz kolone Model izvučen je atribut WheelDrive (2WD,4WD, AWD,FWD). Iz nje je takođe izvučena informacija o konfiguraciji motora (I4, I5, IR, V6, V8, V10, V12, H4, H6), kao i informacija o načinu ubacivanja goriva u motor (EFI, SFI, S, MP, MPI). Informacija o zapremini motora nalazila se u kolonama SubModel i Model.

Kreirani su atributi koji su odnosi cena u maloprodaji i na aukciji (kako prosečna cena, tako i „čista“, tekuća prosečna na aukciji i tekuća prosečna „čista“ cena). Takođe su kreirani atributi odnosa prosečne cene i čiste cene kako na aukciji, tako i novog modela. Napravljeni su i atributi kao odnosi prethodnih i sadašnjih cena na aukciji ili maloprodaji kako novog vozila tako i prosečna cena. Nakon toga su izbačeni atributi jer smo iz njih izvukli sve potrebne informacije (Model, Submodel, MMRAcquisitionRetailAveragePrice, MMRAcquisitionAuctionAveragePrice, MMRAcquisitonRetailCleanPrice, MRAcquisitionAuctionCleanPrice, MMRCurrentRetailAveragePrice, MMRCurrentAuctionAveragePrice, MMRCurrentRetailCleanPrice, MMRCurrentAuctionCleanPrice).

Vršeno je dummy enkodiranje kategoričkih promenljivih, ali pre toga je naravno, izvršeno popunjavanje nedostajućih vrednosti najčešće pojavljivanom kod kategoričkih, a prosečnom vrednošću kod numeričkih (probano je i sa medijanom, ali su modeli davali lošije rezultate). Podaci su se, nakon podele na trening i test, skalirali pomoću MinMaxScaler-a.

Pušteno je nekoliko tipova modela: modeli bez ikakvih kreiranih atributa, osnovni model (izbačeni isti atributi kao što je navedeno gore, dodata prva četiri atributa koji su odnosi cena u maloprodaji i na aukciji, model sa svim atributima i model sa nekoliko atributa (u odnosu na osnovni model dodati atributi sa promenjenom veličinom vozila, bojom i proizvođačem). Model na osnovu kojeg se kasnije vršila selekcija atributa i optimizacija parametara je tzv. osnovni model (iako je model bez ikakvih atributa dao nešto bolje rezultate, da bismo iskoristili deo kreiranih atributa ipak nastavljamo sa osnovnim modelom).

## Selekcija atributa i interpretacija dobijenih rezultata

U okviru ovog rada sprovedene su tri metode selekcije atributa: *filter,* *wrapper, embedded.* Kao limit za broj selektovanih atributa je postavljen broj 55 (isprobano je nekoliko vrednosti u opsegu 35-90 i modeli su dali najbolje rezultate za odabrani broj). U Tableli 1 vidimo rezultate kreiranih modela pomoću selektovanih atributa odgovarajućim metodama. Kako se nijedan drugi parametar u modelu nije menjao osim ulaznih podataka, zaključujemo da je najbolja metoda za ovaj problema *wrapper metoda,* iako je dužina izvršavanja bila značajno duža u poređenju sa preostale dve método.Stoga će se naredna optimizacija modela vršiti pomoću njenih selektovanih atributa.

Tabela Rezultati metoda selekcije atributa

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Accuracy | Recall | Precision | F1 | AUC | Threshold |
| Decision Tree WM | 0.875 | 0.077 | 0.520 | 0.133 | 0.733 | 0.171 |
| Random Forest WM | 0.871 | 0.003 | 0.400 | 0.006 | 0.715 | 0.175 |
| Random Forest MIC | 0.871 | 0.002 | 0.100 | 0.003 | 0.702 | 0.155 |
| Decision Tree MIC | 0.875 | 0.074 | 0.528 | 0.128 | 0.696 | 0.160 |
| Logistic Regression WM | 0.872 | 0.002 | 0.200 | 0.003 | 0.658 | 0.134 |
| Logistic Regression EM | 0.872 | 0.000 | 0.000 | 0.000 | 0.653 | 0.136 |
| Logistic Regression MIC | 0.872 | 0.000 | 0.000 | 0.000 | 0.640 | 0.134 |
| Random Forest EM | 0.870 | 0.003 | 0.133 | 0.006 | 0.637 | 0.140 |
| Naive Bayes MIC | 0.680 | 0.349 | 0.158 | 0.217 | 0.629 | 0.067 |
| Decision Tree EM | 0.868 | 0.016 | 0.283 | 0.030 | 0.624 | 0.212 |
| Naive Bayes WM | 0.691 | 0.426 | 0.188 | 0.260 | 0.592 | 0.259 |
| Naive Bayes EM | 0.255 | 0.900 | 0.137 | 0.237 | 0.562 | 0.999 |

## Razvoj modela

Nakon pripreme podataka krenuli smo u razvoj algoritama. Prvenstveno smo se odlučili za Decision Tree Classifier, Logistic Regression, Naive Bayes (Gaussian) i Random Forest Classifier.

Nakon prvih dobijenih rezultata posle dodavanja novih atributa i upoređivanja sa ranije dobijenim rezultatima, shvatili smo da smo malo “preterali” sa novim atributima i da su bolje rezultate dali modeli sa osnovnim podacima, zato smo se odlučili da ipak ne iskoristimo sve selektovane atribute, već samo neke - preuređena veličina, boja, modeli, retail/auction cene. Ali, nažalost, ni ovo nije doprinelo poboljšanju modela stoga zaključujemo da se dodavanje atributa nije pokazalo uspešnim.

Sledeća je na red došla selekcija atributa. Tu smo koristili filter metode, metode obavijanja (wrapper metode) i ugrađenje metode (embedded metode). Nakon sprovođenja procesa selekcija pomoću svake od metoda i upoređivanja dobijenih rezultata zaključujemo da se wrapper metoda selekcija atributa pokazala najuspešnijom i u nastavku ćemo koristiti atribute dobijene pomoću ove metode selekcije.

Prvobitno smo za optimizaciju parametara izabrali recall kao meru koju želimo da optimizujemo obzirom da želimo na smanjimo broj loših kupovina, ali to nam nije dalo zadovoljavajuće rezultate jer su se optimizacijom nje drastično pogoršale ostale metrike a čak ni sam recall nije dao velike vrednosti. Zbog toga smo se ipak odlučili da koristimo auc\_score kao meru optimizacije. koristili smo Randomized Search za Logističku regresiju i Grid Search za ostale modele, dobijeni rezultati su bili značajno bolji nego pri prethodnoj optimizaciji, ali i dalje su najbolje rezultate davali Baseline modeli.

Naposletku, na red su došli ansambl algoritmi, koristili smo Voting, AdaBoost, Stacking, Gradient Boosting, XGBoost i LightGBM i oni su konačno dali bolje rezultate od Baseline modela i značajno poboljšali naše rezultate. Konkretno najbolje su se pokazali LightGBM i XGBoost, a konkretne vrednosti možete pogledati u tabeli u narednom delu.

## Evaluacija procesa

Korak evaluacije procesa smo neprestalno radili pri razvoju modela i gledali da li smo uspeli da poboljšamo i razmišljali šta bismo mogli sledeće da uradimo. Na kraju, naš finalni rezultat za 10 najboljih modela je dat u tabeli 1.

Tabela Finalni rezultati za 10 najboljih modela

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Accuracy | Recall | Precision | F1 | AUC | Threshold |
| LightGBM CV | 0.888 | 0.175 | 0.802 | 0.287 | 0.770 | 0.149 |
| XGBoost Classifier CV | 0.888 | 0.175 | 0.802 | 0.287 | 0.770 | 0.149 |
| Gradient Boosting Classifier CV | 0.886 | 0.164 | 0.772 | 0.263 | 0.759 | 0.149 |
| Voting Classifier CV | 0.876 | 0.044 | 0.775 | 0.083 | 0.744 | 0.132 |
| Decision Tree WM | 0.875 | 0.077 | 0.520 | 0.133 | 0.733 | 0.171 |
| Decision Tree CV Baseline | 0.875 | 0.074 | 0.528 | 0.128 | 0.733 | 0.171 |
| Decision Tree RA | 0.875 | 0.067 | 0.385 | 0.112 | 0.729 | 0.151 |
| Decision Tree BP | 0.875 | 0.069 | 0.552 | 0.118 | 0.727 | 0.171 |
| Random Forest RA | 0.871 | 0.002 | 0.200 | 0.003 | 0.722 | 0.165 |
| Stacking Classifier CV | 0.870 | 0.121 | 0.438 | 0.188 | 0.717 | 0.130 |

Kao što se moglo i očekivati, ansambl algoritmi su se najbolje pokazali, konkretno Light Gradient Boosting Machine (LightGBM) i XGBoost su dali identične rezultate i oni su ubedljivo najbolji kada su u pitanju mere AUC, Accuracy i Precision.

Trenutno smo zadovoljni dobijenim rezultatima i nećemo više unapređivati modele.

## Zaključak

Korišćeni model bi bio *LGBMClassifier*, ili neki od Boosting algoritama, s obzirom da je bez ikakvog većeg podešavanja parametara dobijen najveći AUC skor, uz kratko vreme izvršavanja, u poređenju sa modelima koji su bili pokušani optimizovani pomoću npr. *GridSearch-a.* Bio bi korišćen samo nad atributima koji su otkriveni kao značajni, ili nad unapređenim skupom podataka sa kvalitetnije kreiranim atributima (koje smo ograničili na 55 od inicijalnih 192). Demonstrirali smo da *LGBMClassifier* daje AUC skor od 0.77 sa granicom odlučivanja od 0.149, držeći F1 na 28.7%, tačnost na 88.8%, preciznost na 80.2%, a odziv na 17.5% (što ipak nije zavidan rezultat, ali bolji od gotovo svih kreiranih modela).

U nastavku daljeg rada više pažnje bi se trebalo posvetiti pripremi podataka, tj. kreiranju atributa koji će biti korišćeni za dalju izradu modela. U trenutnoj situaciji, problem se nalazi u činjenici da kreirani atributi imaju veliki broj nedostajućih vrednosti (i preko 50%). Postavlja se pitanje kako na pametan način iskoristiti izvučene informacije, koje iz svakodnevnog života prepoznajemo kao faktore koji se uzimaju u razmatranje prilikom nabavke automobila. Apropo toga, mogu se iskoristiti i naprednije metode popunjavanja nedostajućih vrednosti, korišćenjem npr. K najbližih suseda ili MICE (eng. *Multiple Imputation by Chained Equations*). To bi potencijalno dovelo da inicijalni modeli daju bolje rezultate, pa bi možda i otklonio neuspeh isprobanih metoda analize glavnih komponenti (*IncrementalPCA, PCA, LDA, KernelPCA* koji ni optimizacijom hiperparametara nisu uspeli da poprave rezultate osnovnog modela), samim tim što bismo imali još nekoliko značajnih atributa. Takođe se može raditi na uređenju koda većom stopom njegove iskoristivosti i preglednosti, korišćenju *pipeline-ova* i može se pokušati i drugačiji pristup, na primer korišćenje neuronskih mreže.

## Reference

1. *2WD vs AWD vs 4WD: A guide to the differences and which is best for you*. (2015, Jul). Consumerreports. <https://www.consumerreports.org/cro/2012/12/2wd-awd-or-4wd-how-much-traction-do-you-need/index.htm> (datum pristupa 27.5.2021)
2. Bruzek, J. (2016, Septembar 1). *What Are The Different Types of Fuel Injection?* Cars. <https://www.cars.com/articles/what-are-the-different-types-of-fuel-injection-1420690418419/> (datum pristupa 27.5.2021)
3. *Ensemble methods*. (2020). Scikit-Learn. (2021, Jun 19). <https://scikit-learn.org/stable/modules/ensemble.html> (datum pristupa 19.6.2021)
4. *Lightgbm Documentation*. (2021). Lightgbm.Readthedocs.Io. ([https://lightgbm.readthedocs.io/en/latest/pythonapi/lightgbm.LGBMClassifier.htm](https://lightgbm.readthedocs.io/en/latest/pythonapi/lightgbm.LGBMClassifier.html)l (datum pristupa 19.6.2021)
5. Python Package Introduction. (2020). Xgboost.Readthedocs.Io. <https://xgboost.readthedocs.io/en/latest/python/python_intro.html> (datum pristupa 19.6.2021)
6. *Vehicle Types, Car Body Styles Explained*. (2019, Avgust 1). Caranddriver. <https://www.caranddriver.com/shopping-advice/g26100588/car-types/> (datum pristupa 27.5.2021)
7. Youngs, J. (2012, Avgust 10). *The Differences Between I-4, I-6, V-6 & V-8 Engines*. Jdpower. <https://www.jdpower.com/cars/shopping-guides/differences-between-i-4-i-6-v-6-v-8-engines> (datum pristupa 27.5.2021)