# 1. Opis I razumevanje problema – Uvod

Problem koji je obrađen u ovom radu, sastoji se od pomoganja automobilskim kućama koje trguju polovnim automobilima, kako bi izbegle kupovinu “kick” automobila. Kick automobili predstavljaju one automobile koji deluju kao da su u dobrom stanju, ali zapravo nisu, usled nekih problema koji nisu odmah očigledni, poput promene kilometraže na odometru.

Ukoliko automobilska kuća kupi automobil za koji se ispostavi da je kick, nakon otklanjanja svih problema koje on poseduje, ona će zapravo da izgubi novac prilikom preprodaje automobila. Ovaj problem predstavlja problem binarne klasifikacije.

# 2. Opis i razumevanje podataka

Dat je veliki broj atributa koji opisuje konkretna vozila, poput godišta, vrste motora, boje, veličine, proizvođača, modela, pod-modela... Poslednji atribut, odnosno kolona u datasetu, zove se “IsBadBuy” I označava da li se konkretan automobil ispostavio kao dobra kupovina (0) ili loša kupovina (1). Kontra-intuitivno, negativna klasa predstavlja naš željeni rezultat, a pozitivna klasa neželjeni. Postoji značajan disbalans među klasama, čak 87% automobila pripada negativnoj klasi, odnosno ispostavili su se kao dobra kupovina. Takav disbalans će uticati na naše modele, davajući im “pessimistic bias”, odnosno modeli će većinski klasifikovati automobile kao dobre kupovine. Automobilskoj kući se više isplati da propusti dobru kupovinu, nego da kupi loš automobil. To znači da je potrebno da smanjimo broj false negative predikcija. False negative predstavlja loše kupovine, koje je naš model pogrešno klasifikovao kao dobre. U cilju minimizovanja ovakve greške, trebalo bi da izaberemo model sa najvećim recallom.

# 3. Priprema podataka

Prilikom sređivanja podataka, uočen je veliki broj kategoričkih atributa, koji imaju veliki broj jedinstvenih vrednosti. Jedan od takvih atributa je “Model” koji u originalnom datasetu ima čak 632 različite vrednosti. Za ovakve attribute, izračunat je procentualni udeo svake od jedinstvenih vrednosti u koloni. One vrednosti čiji je procentualni udeo bio ispod određenog praga, su promenjene na vrednost “Other”. Time smo sačuvali one vrednosti koje se pojavljuju veliki broj puta, a one koje se pojavljuju veoma retko smo sveli na jednu vrednost, čime smo značajno smanjili broj jedinstvenih vrednosti. Atribut “SubModel” sadrži veliki broj jedinstvenih vrednosti, koje nose veoma sličnu informaciju. Kod ovog atributa, sve različite vrednosti koje sadrže substring “Sedan” su jednostavno svedene na istu vrednost. Isti postupak je primenjen I za ostale pod modele. Atribut “Color” je sadržao veliki broj različitih boja. Radi smanjenja broja jedinstvenih vrednosti, boje su grupisane u svetle, tamne, šarene i ostale. Time je broj vrednosti sa 16, smanjen na samo 4. Atribut “Size” predstavlja veličinu automobila. Iako je ovaj atribut kategorički, može se sortirati, od najmanje veličine ka najvećoj. Takav kategorički atribut se naziva ordinalni. Ovaj atribut je kvantifikovan tako što su mu jednostavno dodeljene numeričke vrednosti, gde je najmanja veličina dobila vrednost 0, a svaka sledeća u sortiranom nizu je dobila vrednost za 1 veću nego prethodni član. Atributi koji imaju samo 2 jedinstvene vrednosti, su transformisani dodelom vrednosti 0 jednoj vrednosti I 1 drugoj.

Kod numeričkih podataka nedostajuće vrednosti su popunjene medijanom odgovarajućih kolona, a kod kategoričkih podataka modusom.

Atribut “VehicleAge” je izbačen iz dataseta, jer nosi identičnu informaciju kao i “VehYear” atribut. Oni su potpuno negativno korelisani, ukoliko se VehicleAge smanji za jedan, VehYear će da se poveća za jedan. Razlog zašto je baš VehicleAge izabran za izbacivanje između ova dva atributa, je taj što se on nikada neće menjati, odnosno “otporan” je na vreme, I nosiće istu ispravnu informaciju kada god se koristi, dok VehYear ukoliko ne bude ažuriran svake godine, neće nositi ispravnu informaciju.

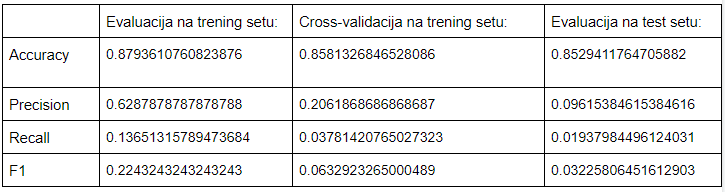
Dataset je podeljen an trening I test deo, u odnosu 70:30.

Na kraju je izvršena normalizacija, primenom MinMax metode. Objekat koji vrši normalizaciju naučen je na trening setu, a korišćen je za normalizaciju i trening i test seta, kako bi se izbegao overfit kasnije kod modela.

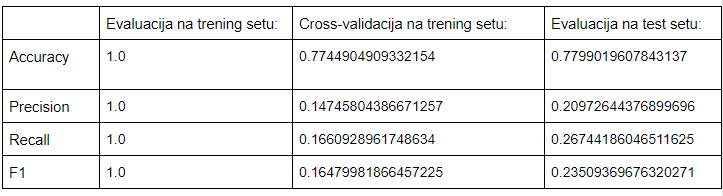
# 4. Treniranje algoritama I interpretacija rezultata

Kao što je naznačeno u uvodu, ovo predstavlja problem klasifikacije. Iz tog razloga izabrani su KNN, DecisionTree I RandomForest algoritmi za klasifikaciju. Napravljen je svaki od ovih modela, sa njihovim podrazumevanim vrednostima parametara, a zatim izvršena njihova evaluacija na trening setu, cross validacija na trening setu I evaluacija na test setu.

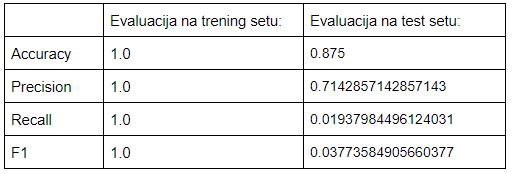
KNN:



DecisionTree:



RandomForest:

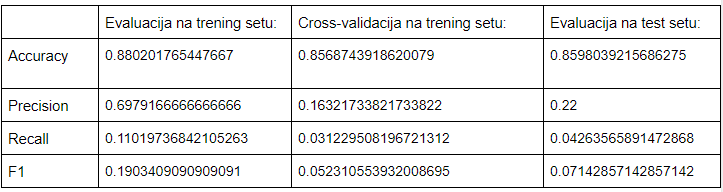


Može se primetiti da je došlo do savršene predikcije na trening setu kod DecisionTree i RandomForest algoritama, što ukazuje da su se modeli preučili. Primenom cross validacije i evaluacije na test setu, može se videti da oni ipak nisu savršeni. Svi modeli imaju dobar accuracy, ali izuzetno mali recall, što nama ne odgovara. Kao što je ranije naznačeno, nama je cilj da maksimiziramo recall. Ovakvi rezultati ukazuju na to da su modeli većinu automobila klasifikovali kao dobre kupovine, što je posledica prirode našeg dataseta, odnosno disbalansa klasa u izlaznoj varijabli.

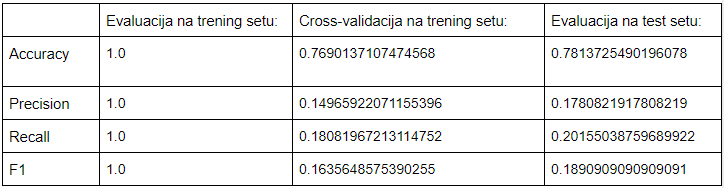
# 5. Selekcija atributa

Kako bismo pokušali da dobijemo bolje rezultate, vršimo selekciju atributa. U ovom radu je za selekciju izabrana VarianceThreshold metoda selekcije. Ova metoda će izbaciti sve one kolone u kojima je varijansa vrednosti jako mala. Za prag je izabrano 95%, što znači da će biti izbačene sve one kolone u kojima se jedna vrednost pojavljuje u 95% redova. Ovime je datasetu smanjen broj kolona na 33, sa početnih 113. Zatim je novodobijeni dataset korišćen za kreiranje novih modela, kako bismo uporedili rezultate.

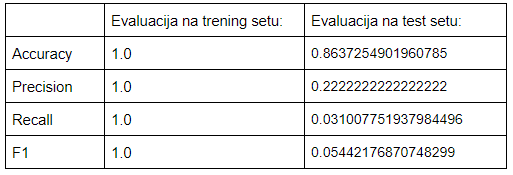
KNN:



DecisionTree:



RandomForest:



Rezultati dobijeni nakon selekcije atributa I kreiranja novih modela su lošiji nego oni pre selekcije. To ukazuje da smo izgubili neke bitne informacije I da prag ili metoda selekcije koji su primenjeni nisu odgovarajući. Zbog toga će u nastavku rada biti koriščen dataset onakav kakav je bio pre selekcije atributa.

# 6. Optimizacija parametara

Do sada su korišćeni modeli koji su uzimali svoje podrazumevane vrednosti za hiperparametre, što je očekivano donelo rezultate koji nisu veoma dobri. Usled poboljšanja modela, izvršena je optimizacija parametara za svaki od 3 modela koji su korišćeni. Korišćen je GridSearchCV algoritam za optimizaciju, kojem smo prosledili parametere koje želimo da optimizujemo, kao I skup vrednosti sa kojima će algoritam da vrši testiranje. Za ocenu parametara korišćena je recall metrika, odnosno algoritam će nam kao najbolje parameter vratiti one za koje je recall bio najveći.

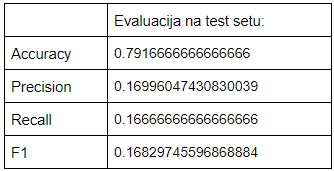
Parametri koji su optimizovani za KNN model su “n\_neighbors” i “weights”, a dobijeni rezultati su 1 i “uniform” respektivno.

Parametri koji su optimizovani za DecisionTree model su “max\_depth“, “min\_samples\_split“ I “criterion“, a dobijeni rezultati su 160, 2 I “gini“ respektivno.

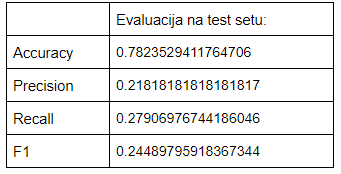
Parametri koji su optimizovani za RandomForest model su “n\_estimators“, “min\_samples\_split“ I “criterion“, a dobijeni rezultati su 10, 2 I “gini“ respektivno.

Kreirani su modeli sa svojim optimizovanim parametrima, I izvršena njihova evaluacija na test setu.

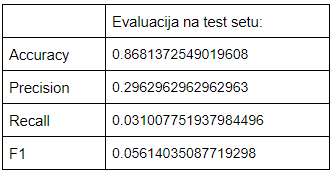
KNN:



DecisionTree:



RandomForest:



Čak I nakon optimizacije parametara, nijedan model nema visok recall. Najveći recall od ova 3 modela ima DecisionTree model, I on iznosi približno 28%, što je značajno bolje od druga dva modela. Takođe možemo uočiti da mu je I accuracy za 1% veći nego što je bio pre optimizacije.

# 7. Zaključak

Nijedan od dobijenih modela u ovom radu nema visoku vrednost recall metrike, ali najbolji među njima je DecisionTree model nakon optimizacije parametara. On ima najveći recall, koji iznosi 28%, što znači da će najmanje od svih ostalih modela predvideti automobile kao dobru kupovinu, dok on zapravo to nije, I time će smanjiti troškove automobilskim kućama usled loših kupovina.

U slučaju da sam imao više vremena I znanja, istraživao bih još različitih ansambl modela, koji bi možda bili bolji izbor za ovaj problem. Takođe bih posvetio više vremena obradi podataka, I pogotovo selekciji atributa. Sa više računarskih resursa I vremena bih optimizovao veći broj hiperparametara modela I zadao im veći opseg za testiranje. Pored GridSearchCV algoritma bih iskoristio još neki, poput RandomSearch-a, radi poređenja rezultata i daljeg testiranja.