Извештај за пројекат из предмета Машинско учење

Опис и разумевање проблема

Пројекат смо радили на скупу података *Carvana*, који садржи податке о возилима америчке компаније која се бави препродајом половних возила. Главни циљ пројекта је идентификација возила који имају проблеме који спречавају њихову препродају – такозваних ***Kick (bad buy)*** возила.

Изазов задатка је био препознати атрибуте који су кључни за решавање проблема и алгоритме који ће дати најбоље резултате, али и поставити одговарајуће хиперпараметре.

Решавање овог проблема највише ће значити компанији која ће применити решења у даљем приступу купцима у зависности од тога да ли је аутомобил *Kick* или не.

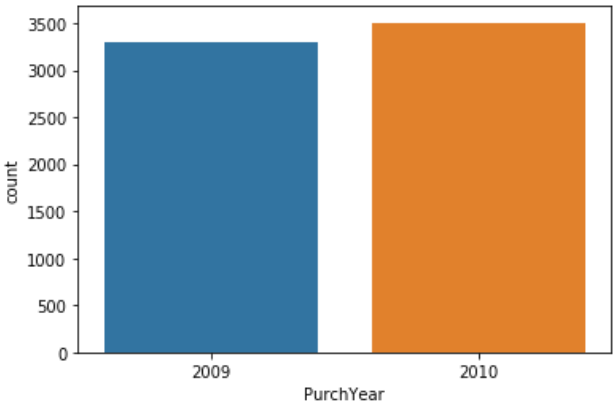
Опис, разумевање и припрема података

Carvana скуп података садржи 6798 опсервација и 34 атрибута, од којих је један атрибут излазни ***IsBadBuy*** и даје нам информацију о томе да ли је куповина возила, односно његово укључивање у даљу продају, била добар потез.

Пре даљег рада, идентификовали смо недостајуће вредности, који је било код 18 атрибута. Атрибуте који су имали превише недостајућих вредности (***Auction, PRIMEUNIT, AUCGUART, BYRNO, VNST***) смо искључили из даље анализе, а остале смо попунили одговарајућим вредностима – оне код нумеричких атрибута смо попунили медијаном осталих вредности датог атрибута, а недостајуће вредности категоричког типа смо попунили најфреквентнијом вредношћу.

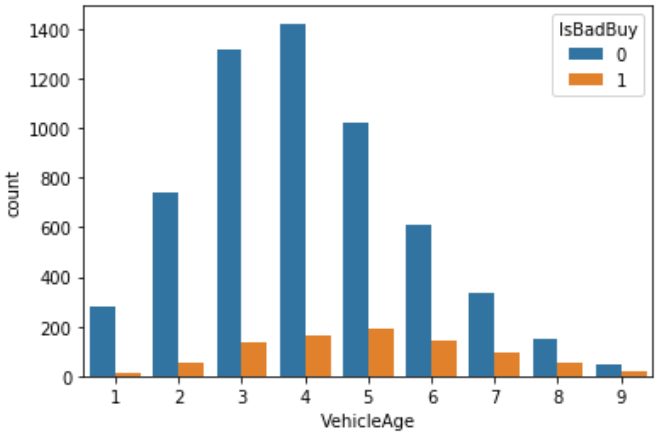
Након тога, анализирали смо сваки од атрибута појединачно.

- Кад је у питању атрибут ***PurchDate*** – датум када је возило постављено на аукцију, сматрали смо да је корисније извући само годину, а полазни атрибут изоставити из даље анализе.

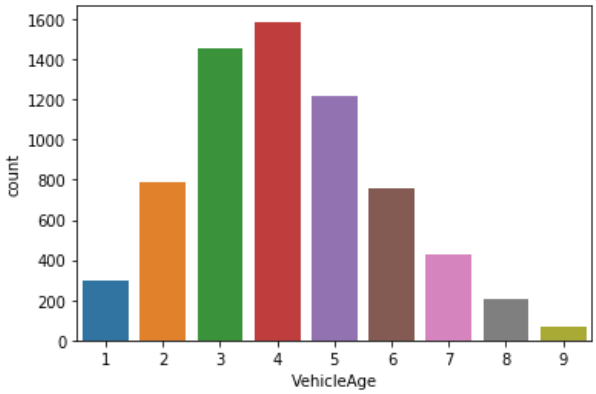


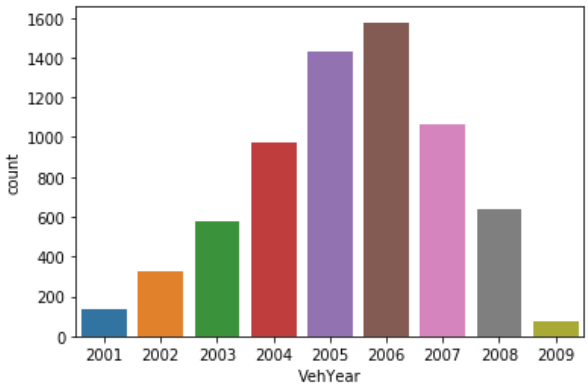
- Из атрибута ***Make*** видимо да скуп података садржи 32 марке возила од којих је четири најчешће
- Атрибут ***Model*** се односи на модел возила, али садржи више информација од самог имена модела – назив модела, погон и број цилиндара на мотору, што је резултовало великим бројем категорија, чак 632. Због тога смо поделили атрибут на три различита који садрже сваку од наведених карактеристика, тако да почетни атрибут садржи само назив модела. Тиме смо број категорија смањили на 185.

- Атрибут ***VicheAge*** – старост возила се показао као јако битан за предвиђање излазне варијабле, што се може видети са графикона:

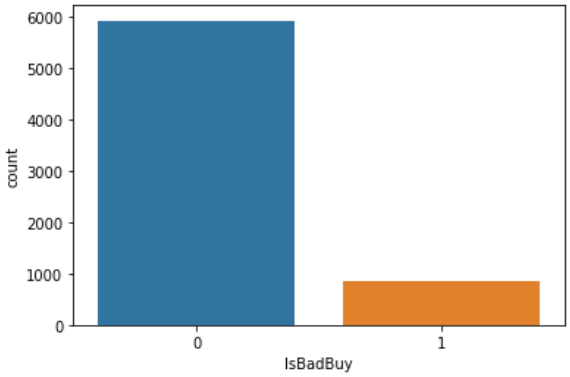


- Одређене атрибуте смо изоставили из анализе јер представљају шифре и не носе податке битне за анализу: ***RefId, WheelTypeID, VNZIP1***.
- Атрибуте ***VehYear*** смо искључили из даље анализе због високе корелације са атрибутом ***VehicleAge*.**





- Из истог разлога смо искључили и атрибуте ***Nationality* и *TopThreeAmericanName*** – због корелације са атрубтом ***Make***
- Због великог броја јединствених вредности које би могле довести до претренирања модела, изоставили смо из анализе и атрибуте ***SubModel* и *Trim***.
- Од категоричких атрибута ***Make, Model, Color, Size* и *DriveType*** смо креирали dummies вредности, како бисмо спречили да се вредност категоричких атрибута мери њиховом нумеричком вредношћу.
- Излазна варијабла нам говори да постоји значајан дисбаланс између броја добрих и лоших куповина, у корист добрих, што је за препродавца позитивно:



Тренирање алгоритама и интерпретација добијених резултата

Приликом валидације модела је користили смо десетоструку унакрсну валидацију.

Када смо одлучивали за коју меру евалуације да се одлучимо, разматрали смо цену грешке. Дошли смо до закључка да нас мање кошта *FalsePozitive* тип грешке од *FalsePozitive* типа, јер нам је већи трошак да купимо возило за које смо лажно мислили да није *Kick*, него да прескочимо куповину возила за које смо погрешно закључили да јесте. То нас је навело да помислимо да је најбоља мера евалуације за наш проблем *Recall* (одзив), али пошто нисмо знали реалну цену коштања сваке од грешака, за меру евалуације смо изабрали AUC јер је свеобухватнија у односу на остале мере – узима у обзир више различитих вредности за границу одлучивања.

Класификација

Алгоритми класификације који су коришћени у овом раду су:

- Наивни Бајес
- К најближих суседа
- Стабла одлучивања
- Логистичка регресија
- Алгоритми ансамбли:
  - Voting
  - Stacking
  - Bagging
  - Random forest
  - Gradient boosting

Model	CV
Naive Bayes	63.24
K neighbors	54.66
Decision tree	53.26
Logistic regression	62.10
Voting	63.04
Stacking	60.83
Bagging NB	63.33
Random forest	64.38
Gradient boosting	67.09

Резултати свих алгоритама су прилично уједначени и варијансе сваког модела су релативно мале. Из даље анализе биће избачени модели К најближих суседа, Стабло одлучивања и *Stackig* јер су показали најлошије резултате, а њиховим избацивањем смо и убрзали анализу.

Селекција атрибута

Приликом селекције атрибута, користили смо следеће методе:

- Филтер метода – *mutual_info_classif*
- Обавијајућа метода – *Wrapper*
- Уграђена метода – *Embedded*

Резултати се након селекције нису значајно побољшали, али смо успели да знатно смањимо број атрибута – са 266 на 50-90.

Model	CV	Filter	Wrapper	Embedded
Naive Bayes	63.24	62.40	63.32	60.74
Logistic regression	62.10	60.86	62.09	61.96
Voting	63.04	62.78	63.46	61.66
Bagging NB	63.33	62.56	63.50	60.78
Random forest	64.38	62.82	64.44	59.66
Gradient boosting	67.09	65.77	67.07	61.73

Из табеле се види да обавијајућа метода – *Wrapper* даје најбоље резултате, тако да даљи рад настављамо са селекцијом атрибута коју је изабрала *Wrapper* метода.

Оптимизација параметара

Приликом оптимизације параметара коришћена је неисцрпна претрага – non-exhaustive search. Оптимизацију смо вршили за четири алгоритма:

- Логистичка регресија – за коју смо мењали параметре *penalty* и *C*
  - Оптимални параметри:
    - C = 1.94
    - penalty = l1
  - Најбоља AUC вредност = 67.36%
- Bagging – за коју смо мењали параметре *n_estimators, bootstrap, max_samples* и *max_features*
  - Оптимални параметри:
    - n estimators = 100
    - bootstrap = 20
    - max samples = 0.4233
    - max features = 0.9589
  - Најбоља AUC вредност = 64.85%
- RandomForest – за коју смо мењали параметре *n_estimators, max_depth, min_samples_leaf* и *criterion*

    - Оптимални параметри:
        - n estimators = 120
        - max depth = 15
        - min samples leaf = 7
        - criterion = entropy
    - Најбоља AUC вредност = 67.43%
- *GradientBoosting* – за коју смо мењали параметре *loss, n_estimators* и *learning_rate*
    - Оптимални параметри:
        - loss = deviance
        - learning rate = 0.079
        - n estimators = 100
    - Најбоља AUC вредност = 67.36%

Закључак

Пошто су резултати за модел GradientBoostingClassifier били константо добри, у практичној примени бисмо користили њега. За нову инстанцу возила измерили бисмо вредности атрибута који су улазни за наш алгоритам, и применили бисмо модел са хиперпараметрима које смо одредили као оптималне и навели у раду.

У даљем раду би дефинитивно било добро за меру евалуације узети Recall, или евентуално, на основу цене коштања направити нашу меру евалуације. С обзиром на то да нам конкретне цене нису биле познате, то смо изоставили из рада, јер нисмо хтели да нагађамо и на тај начин можда не решимо реалан проблем.