Nazif Azizi, Rafael Piloto, Gustavo Soto-Reyes, Albert Terc

Dr. Pouriyeh

CS 334

1 May 2023

Project Limes

Abstract

One of the current challenges in the used car business is making sure that the cars being sold are of good quality. Utilizing a Kaggle dataset provided by the car rental company Carvana, we apply various machine learning methods to predict whether a used car is a bad buy. The dataset contains 34 categorical and numerical features that describe different aspects of the car, and more than 70,000 observations. First, we perform feature selection via a correlation matrix and select the ones most correlated with the label IsBadBuy through feature engineering, like the odometer reading and the vehicle age. For preprocessing, we first encode the remaining categorical features into numerical ones for compatibility with Scikit Learn models through one-hot encoding, then we remove missing values to maintain the integrity of the data, lastly we upsample the minority class—where the car is in fact a bad buy—to correct imbalancing issues. We then run various supervised models on our data, such as decision trees, logistic regression, KNN, naive bayes—as well as some ensemble methods like random forest and XGBoost. Our best performing non-ensemble model was the decision tree model, which had an accuracy and AUC of 0.98 and 0.95 respectively. However, XGBoost performed even better, with an accuracy and AUC of 0.99. Our XGBoost and Random Forest models greatly outperform previous works, which we infer is due to removing missing values instead of filling them with statistical

aggregations. In this way, we remove excess noise, which allows the models to more easily recognize patterns in the dataset.

Introduction

The auto dealership industry faces many challenges, one of which is the risk associated with purchasing used cars at auto auctions. These purchases can result in the acquisition of cars with underlying problems that render them unsellable or undesirable to customers, commonly referred to as “kicks” within the auto community. Kicked cars may have tampered odometers, unresolved mechanical issues, difficulties in obtaining the car title, or other unforeseen problems. These cars not only burden dealers with additional costs in transportation and repairs, but also lead to market losses in reselling the car. In this research paper, we aim to develop an effective predictive model to identify cars with a high risk of being kicks, therefore enabling auto dealerships to minimize potential losses and improve their inventory selection for customers. By leveraging machine learning methods, we seek to tackle the challenge of predicting whether the car purchased at auction is a kick which means it is a bad buy, and ultimately providing invaluable insight to dealerships and enhancing their decision-making process.

Related Work

We have identified two similar research papers that look at the same Kaggle dataset. The most recent one, done by Reza Karimi and Zelalem Gero at Emory University in 2017, performed a lot of similar steps as us, like numerical encoding of categorical features, and upsampling the minority class. However, they included extra steps in their preprocessing phase. They employed feature engineering by extracting additional features out of the car ‘model’ and ‘submodel’ features. Moreover, they transformed the price features into relative prices instead of absolute ones (Karimi 4). They ended up not removing any features in feature selection,

including the additional ones they engineered. Also, they used an imputer for missing values, using mean for numerical features and mode for categorical features. For their models, they chose Random Forest, Multilayer Perceptrons, XGBoost, and GDBoost, and recorded the accuracy, recall, F2 score, and AUC. Their models performed well, with accuracies and AUC scores in the 0.80s. They then utilized ensembling techniques to boost their models' performance, like ensemble Random Forest, and got slightly better results, with accuracies around 0.91 for all models and AUCs around 0.88.

The second paper we identified was one from Stanford University students Albert Ho, Robert Romano, and Xin Alice Wu from 2012. They also made similar observations about the nature of the data, but their preprocessing steps also differed from the previous paper in significant ways. For instance, although they imputed mean for numerical features, they instead created new values for missing categorical data. From here they experimented with preliminary models like Support Vector Machines and Logistic Regression before realizing they needed to preprocess the data further. In doing so, they performed feature normalization and balanced the dataset via upsampling. These two techniques improved the performance of the two preliminary models but left them curious about how other models and ensembles would perform. For this, they used Weka and the models AdaBoost, LogitBoost, and various ensembles that included weaker models like Naive Bayes. For all their models, they ran them on both the unbalanced and balanced versions of the dataset. In general, the balancing of the dataset did not help much with the AUC, and in some cases made it worse. Their best performing model was LogitBoost with an accuracy of 0.90 and an AUC of 0.758 for the unbalanced dataset.

Dataset Description

VehYear	VehicleAge	VehOdo	MMRAcquisitionA...	MMRAcquisitionA...
Year the car was produced	Age of the car	How far the car has driven in km	Price of the car when is was bought at auction, average	Price of the car when it was bought at auction, before fees
2k – 2.01k	0 – 9	5.37k – 116k	0 – 35.7k	0 – 36.9k

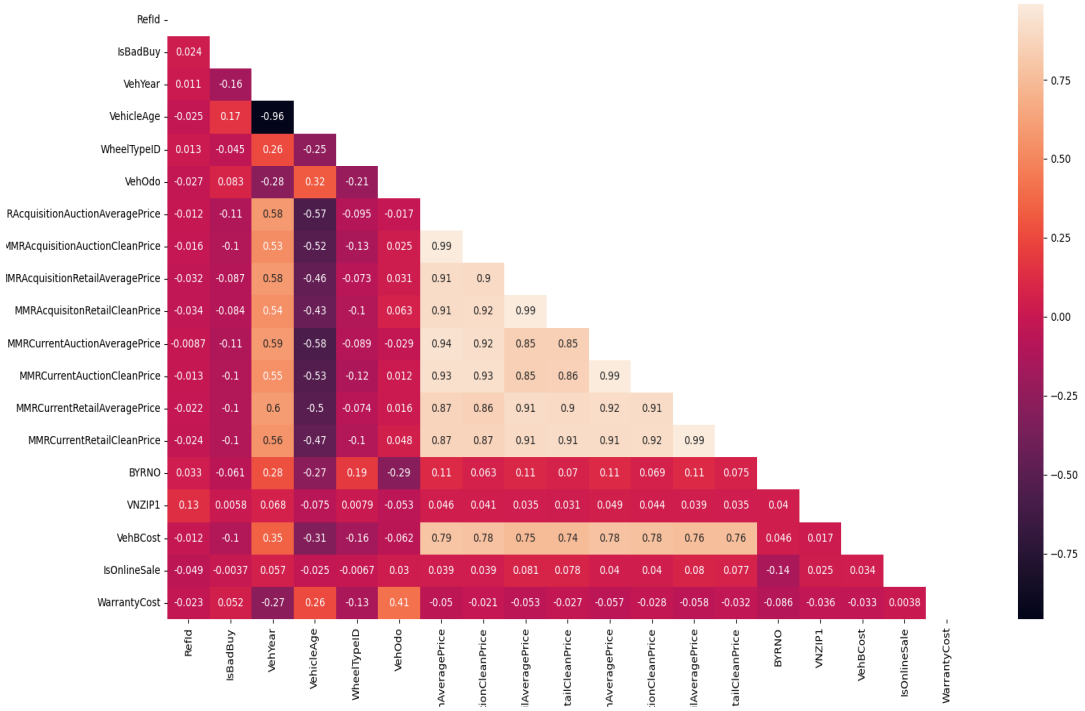
MMRCurrentAuct...	MMRCurrentRetai...	MMRCurrentRetai...	VehBCost	WarrantyCost
Current price of the car at auction, before fees	Current price of the car at a retail store, average	Current price of the car at a retail store, before fees	B price of the car	Cost of car warranty
0 – 36.9k	0.00 - 1954.00 Count: 510; 0 – 39.1k	0 – 40.3k	1.4k – 35.9k	462 – 7.5k

Auction		Make		Model		Trim		SubModel	
Location of auction		Producer of the car		Model of the car		Trim level of the car		Submodel of the car	
MANHEIM	58%	CHEVROLET	25%	'PT CRUISER'	3%	Bas	20%	'4D SEDAN'	20%
OTHER	23%	DODGE	18%	IMPALA	3%	LS	15%	'4D SEDAN LS'	7%
Other (12679)	19%	Other (38291)	57%	Other (63094)	94%	Other (44171)	66%	Other (49307)	73%

Class
Class of the car
0 – 1

We utilized a Kaggle dataset provided by Carvana, a car rental company, to develop a machine learning model that predicts if a used car is a bad buy. This dataset includes more than 70,000 observations and 34 categorical and numerical features that describe the vehicle. Since the objective was to develop a binary classification model, we are using a feature with binary values as our target variable. This variable is "IsBadBuy" where a bad buy is 1 and a good buy is 0.

Preprocessing and Feature Selection



We applied various techniques for processing and feature selection. We first implemented feature selection using Pearson correlation, which allowed us to determine the numerical features that were most correlated with the target variable. By setting a correlation threshold of 0.1, we dropped the columns with a correlation of less than 0.1, and we were also able to eliminate

redundant features that were highly correlated with each other (for example, a lot of the price features and cost were highly correlated with each other).

We dropped features that did not make sense to include in our model and were left with 20 relevant features. Among these features, seven are categorical, and thirteen are numerical. The categorical features provide discrete information about the location of the auction where the vehicle was sold, the age, make, model, submodel, and type of transmission. The numerical features provide continuous information about the vehicle's auction and retail prices.

We then proceeded with feature processing, where we applied imputing to replace null values in numerical features with the mean of those features, and for categorical features, we replaced null values with the most frequent. To convert the categorical columns into numerical values, we utilized one-hot encoding, which created new columns from the values in our categorical features. We then added all of our transformations to a ColumnTransformer, to apply transforms to various datasets. However, we noticed that this approach resulted in an unbalanced dataset with low AUC scores. To handle this, we decided to drop all null values and randomly duplicate records in the minority class, which resulted in a more balanced dataset.

Our techniques helped us to identify the most relevant features to produce the most accurate results for our machine learning models.

Models and Evaluation

To evaluate our model, we will be using accuracy and AUC. We find that these metrics are reasonable given that our data is now balanced after pre-processing. We evaluate the following models: KNN, Linear Regression, Logistic Regression, Naives Bayes, Decision Trees, Random Forest, and XGBoost. We used all of these methods because they are great models for supervised learning. We utilized KNN, which classifies a given prediction by observing the

majority label of K-neighbors near the prediction. Linear and Logistic regression work by tuning coefficients using MSE and then classify kicks if the result is greater/lower than a threshold. These models were good benchmarks in our initial unbalanced dataset because a lot of values were filled, but proved to be less accurate after balancing our data. Naive Bayes was an exploratory model we decided to include and it classifies results by observing the probability that given all the features, the prediction is a kick or good buy. This model was a little tricky to work with given our dataset, and we were only able to get good results with only numerical data, which still proved to be a weak model. Lastly, our decision tree, random forest, and XGBoost models produced some of the best results. Decision trees function by selecting features and splitting on them to come to a classification. A random forest is an ensemble of decision trees which collectively make a prediction. Lastly, XGBoost provides parallel tree boosting to random forests and provided the best results for our balanced dataset.

	Our Work		**Karimi (2017)**		**Ho (2012)**	
Model	**Accuracy**	**AUC**	**Accuracy**	**AUC**	**Accuracy**	**AUC**
KNN	0.98	0.98	-	-	-	-
Linear Regression	.82	.82	-	-	-	-
Logistic Regression	.85	.84	-	-	.83	.71
Decision Tree	.98	.95	-	-	-	-
Random Forest	.95	.98	.89	.85	-	-

	Our Work		**Karimi (2017)**		**Ho (2012)**	
Naive Bayes	.63	.68	-	-	.89	.75
XGBoost	.99	.99	.85	.79	-	-

For KNN, we found after cross validation that K=1 yielded best results. For decision trees, we found that 10-K Fold Cross Validation yielded the best results and Random Forest with one hundred estimators and depth of eight yielded an improvement over the AUC, but slight decrease in accuracy. Lastly, our best model, XGBoost, was able to yield 99% accuracy with 99% AUC on our balanced dataset with a max depth of five and two hundred estimators.

We observe that after preprocessing, training, and cross validation, we were able to exceed the performance of other relevant works by appropriately addressing the data imbalance issue. All other works worked to address the imbalanced dataset, however, all tackled it by upsampling. The issue with upsampling on our dataset is that there are a large number of columns with missing values. As such, when pre-processing, a lot of these columns are resolved with either the mean, median, or mode. With there being so many of these rows that are being filled, the data is ultimately being filled with constants. This means that models like linear or logistic regression, KNN, or Naives Bayes would perform better and overall show a higher AUC, but we are fitting to the aggregates rather than defining features of what makes a purchase a kick. As such, when we came to this conclusion, removing rows with missing values meant that our models were fitting to real data rather than aggregates. Our models were picking up on features that are relevant to the label rather than aggregates that introduce noise. It is for that reason that we observe significantly higher accuracy and AUC for our models.

Conclusions

In conclusion, we found that working with our dataset during preprocessing was the most important step in our analysis. Our initial analysis without addressing the balance issue was able to yield comparable results to other works but a significantly lower AUC. Though we tried upsampling like other works, we did not feel it was an appropriate method given how many columns were missing per sample. It was through these insights that we hypothesized that removing the artificial noise we were introducing allowed our model to hone in on critical features and thus far exceeding performance compared to any other model. XGBoost provided the best results in our analysis, though there is still more work to be done. Our model relies on all features being present for a given sample which is not always a feasible task at auctions. Further data analysis can be performed to more cleverly address missing values during upsampling. If it is possible to do so without adding artificial noise, the model would be able to more accurately and robustly predict lemons despite not having the full picture.

Contributions

- Albert: Worked on Naive Bayes model and compared initial results to other works. For the paper, wrote the abstract and discussion of related works.
- Rafael: Worked on feature selection, the decision tree model, random forest model, and XGBoost model. For the paper, wrote Models/Evaluation & Conclusion.
- Nazif: Worked on the KNN model and found references/other works. For the paper, wrote the introduction and references.
- Gustavo: Worked on preprocessing steps and linear & logistic regression models. For the paper, wrote the data overview and data preprocessing.

References

Darden Business School University of Virginia, *Carvana: IsBadBuy*

https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3614450

STANFORD UNIVERSITY, CS229 - MACHINE LEARNING, *Don't Get Kicked - Machine Learning Predictions for Car Buying*

https://cs229.stanford.edu/proj2012/HoRomanoWu-KickedCarPrediction.pdf

Reza Karimi, Zelalem Gero, *Don't Get Kicked: Predict if a Car Purchased at Auction is Lemon,*

https://0xreza.com/papers/dontgetkicked.pdf

Dataset and GitHub

https://www.kaggle.com/datasets/ulrikthygepedersen/car-kick

https://github.com/RafaelPiloto10/limes