Predicting Auto Auction Risks: Identifying 'Kicks' to Optimize Dealership Inventory

Final Report for Introduction of Machine Learning in 2023 Fall

Bi-Lian Chen †
Business Analytics, M.S. Lally School of Management.
Rensselaer Polytechnic Institute
Troy, NY
chenb10@rpi.edu

ABSTRACT

This project endeavors to empower automotive dealers in predicting potential risks associated with their vehicle investments by leveraging machine learning models. Focused on essential vehicle data such as make, year, and mileage, the objective is to offer valuable insights for informed decision-making. Preprocessing techniques involve meticulous separation of input variables and labels, as well as the application of KNNImputer for numerical features and One-Hot encoding for categorical features.

The class label "IsBadBuy" serves as a crucial metric, identifying whether a purchased vehicle was an avoidable investment. Various notebooks employ distinct feature and model approaches, each contributing to the overall understanding of predictive accuracy. Critical features like VehYear, Make, and VehOdo are scrutinized for their impact on determining the quality of used cars.

Benchmarking against other Kaggle solutions reveals diverse approaches, with an emphasis on data cleaning, feature engineering, and a blend of machine learning models. The data preprocessing phase involves grouping and splitting data, ensuring it is optimized for subsequent analysis. The subsequent data visualizations and analyses delve into IsBadBuy rates for different makes, the distribution of vehicles sold each year, and the IsBadBuy rate for cars manufactured in different years.

The modeling phase introduces Decision Tree, Random Forest, and XGBClassifier models. Performance metrics such as ROC curves and AUC values are scrutinized, offering insights into the models' discriminative abilities. Notable feature importance analyses uncover influential factors like WheelTypeID, VNZIP1, VehOdo, and VehBCost. The Cross-Correlation Function (CCF) is employed to assess the relevance of different independent variables, revealing intriguing relationships between vehicle age and IsBadBuy rates.

In summary, this project provides a comprehensive exploration of predictive modeling for used car investments, combining preprocessing techniques, model selection, and feature importance analyses to enhance decision-making for automotive dealers.

KEYWORDS

Predictive Modeling, Machine Learning, Used Car Investments, Decision Tree, Random Forest, XGBClassifier, IsBadBuy, Feature Importance, Preprocessing Techniques, Data Visualization, AUC (Area Under the Curve), ROC Curve, Cross-Correlation Function (CCF), Vehicle Data, Kaggle Solutions, Data Cleaning

1 Executive Summary

This comprehensive project is designed to empower dealers with the capability to forecast potential risks associated with vehicle investments. By meticulously analyzing fundamental vehicle data, including make, manufacturing year, and mileage, the project aims to offer dealers invaluable insights. Through this predictive analysis, dealers can optimize their decision-making processes, leading to more prudent investment choices and ultimately contributing to improved business outcomes in the competitive automotive industry.

You can find the dataset and all the Kaggle solution is below link: https://www.kaggle.com/competitions/DontGetKicked

1.1 Preprocessing Techniques

a. Split X(input variables) and y(target, label)
b. Feature type separation into numerical_features and categorical_features, but we only use numerical_features.
   (a) Use KNNImputer for numerical_features.
   (b) Max - min Normalization (we use this here).
   (c) Remove the variables whose std = 0, i.e., uninformative_variables.

1.2 Class Label

In this project, we have one binary class label, which is shown in Table 1 in below. The label is 'IsBadBuy', Identifies if the kicked vehicle was an avoidable purchase.

	Yes	No
IsBadBuy	1	0

Table 1: Class Label and Definition

1.3 Critical Features

There are 38 features and more than 100,000 data points in total are considered in this case, we presented 16 important features in Table 2 below.

	Feature Description
VehYear	The manufacturer's year of the vehicle
Make	Vehicle Manufacturer
MMRAcquisitionAuctionAveragePrice	Acquisition price for this vehicle in average condition at time of purchase
MMRAcquisitionAuctionCleanPrice	Acquisition price for this vehicle in the above Average condition at time of purchase
MMRAcquisitionRetailAveragePrice	Acquisition price for this vehicle in the retail market in average condition at time of purchase
MMRAcquisitonRetailCleanPrice	Acquisition price for this vehicle in the retail market in above average condition at time of purchase
MMRCurrentAuctionAveragePrice	Acquisition price for this vehicle in average condition as of current day
MMRCurrentAuctionCleanPrice	Acquisition price for this vehicle in the above condition as of current day
MMRCurrentRetailAveragePrice	Acquisition price for this vehicle in the retail market in average condition as of current day
VehOdo	The vehicles odometer reading
PurchDate	The Date the vehicle was Purchased at Auction
VehBCost	Acquisition cost paid for the vehicle at time of purchase
Model	Vehicle Model
VehicleAge	The Years elapsed since the manufacturer's year
VNZIP	Zipcode where the car was purchased
WheelTypeID	The type id of the vehicle wheel

Table 2: Critical Features and Definition

2 Benchmarking of Other Solutions

2.1 Analyze Three Other Solutions

Identify 3 other Kaggle solutions completed by others. The solution should include a score on the Kaggle prediction task. You can find it by selecting on the project and then clicking on the link to Kernels. Summarize the features, modeling approach, and performance in Table 3.

Notebook Name	Feature Approach	Model Approach	Train/Test Performance
DontGetKicked_Feature_selection Score: 0.15973	Use all Features available in this dataset.Only remove features withvhigh percentage of missing values.	XGBoost Classifier	x_Train: accuracy = 0.75. x_vaild: accuracy = 0.74. x_test: accuracy = 0.74.
Don't get kicked! Score: 0.25909	Use one-hot on categorial features.Also reclassify some numerical features, for example, binary coded variable urban-rural based on zip codes.	XGBoost Classifier	The author did not share performance score
Predict if a car purchased at auction is a lemon Score: 0.24367	Use all the data except these four types which are less likely to contribute: 1. unique_id 2. with_many_categories 3. Redundant 4. high_correlation	Logistic Regression, K-Nearest Neighbors, Decision Tree, Random Forest	logistic regression test accurary:89.56. K-Nearest Neighbors test accurary: 87.48. Decision Tree test accurary: 82.17. Random Forest test accurary: 89.85.

Table 3: Summary of Three Other Solutions

2.2 Observations of Three Other Solutions

For first solution 'DontGetKicked_Feature_selection', The author of this project mainly focused on data cleaning. First of all, the author separates high v/s low cardinality Numeric and Categorical features. Second, perform different missing value treatments for Numerical Features and Categorical Features. After all the understanding and cleaning data, the model becomes more reliable with better input data.

As for the second solution 'Don't get kicked! ', it can be clearly seen that his project is more focused on feature engineering. The fundamental approach to feature engineering in this case involves transforming the provided data in a manner that enhances the model's ability to differentiate between the various target categories. Each newly created feature was systematically assessed to determine its impact on the model and its structural relationship.

The last solution 'Predict if a car purchased at auction is a lemon' does an outstanding job that the author trying to approach the best performance by different machine learning models, such as Logistic Regression, K-Nearest Neighbors, Decision Tree, and Random Forest. In this case, the author can successfully choose the most suitable model for this question and get a great performance from its prediction.

3 Data description and Initial Processing

3.1 Data Preprocessing and Cleaning

In this part, I conducted a two-step data preparation process. Firstly, I loaded data from local Excel files and organized it by calculating the mean value of 'MMRAcquisitionAuctionAveragePrice' within specific groups defined by columns such as 'Make,' 'Model,' and others. These results were stored in new columns.

Secondly, I split the data into input variables (X) and target labels (y), checking their shapes for understanding. Further, I separated numerical features by replacing None with KNNImputer, applying max-min normalization, and removing uninformative variables. Categorical features underwent One-Hot Encoding. The comprehensive approach aimed to optimize the dataset for subsequent analysis or modeling.

3.1.1 Grouping Data

1. Calculate the mean value of the 'MMRAcquisitionAuctionAveragePrice' feature within specific groups defined by various columns in the 'data' DataFrame, such as 'Make,' 'Model,' 'Trim,' 'SubModel,' 'Color,' and 'Transmission.'
2. Stored results in new columns with names like 'mean_MMRAcquisitionAuctionAveragePrice_Make,' 'mean_MMRAcquisitionAuctionAveragePrice_Model,' etc. These new columns contain the mean values of 'MMRAcquisitionAuctionAveragePrice' for each respective group.

3.1.2 Split Data

1. Split X(input variables) and y(target, label):
   (1.) Check X, y data shape for better understandings of data.
2. Feature type separation into numerical_features and categorical_features.
   (1.) For numerical_features:
      a. Replace None in "numerical_features" with KNNImputer.
      b. There are 2 kinds of normalization method:
         i. mean - std Normalization
         ii. max - min Normalization (use this here)
      c. Remove the variables whose std = 0, aka uninformative_variables.
      d. Split trainning dataset and valid dataset.
   (2.) For categorical_features:
      a. Use One hot encoding
         i. Creating a one hot encode array.
         ii. Process y_train and y_val with one hot encoding.

3.2 Data Visualizations and Basic Analysis

3.2.1 IsBadBuyRate for Different Makes

Fig.1 indicates the percentage of each make vehicle that has how many vehicles sold is considered a "Bad buy", as the ratio closer to 1, the more "Bad buy" ratio these makes tend to have.

It shows that some makes of vehicles are having better chance to be considered "Bad buys" than others. For example, as long as we get a Plymouth vehicle, we will have almost 50% chance to make a bad buy. Other makes of vehicles that are more likely to be considered "Bad buys" include Mini, Infiniti, Acura, and Lexus.

On the other hand, some makes of vehicles are less likely to be considered "Bad buys". For example, Toyota and Honda are known for producing reliable and affordable vehicles. Other makes of vehicles that are less likely to be considered "Bad buys" include Dodge, Chevrolet, and Isuzu.

However, there is one thing that should be pointed out in this visualization, which is some of the makes only have small sample size here, but we cannot tell in this image. Thus, we need to generate the other visualization to cover this information.

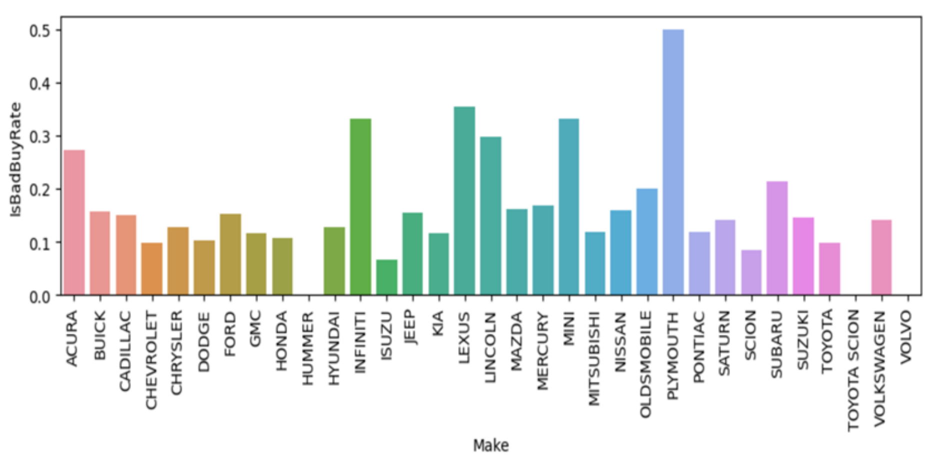

Figure 1: IsBadBuyRate for each make

In fig.2, it indicates the specific counts of each Make vehicle that has how many vehicles sold is considered a "Bad buy" or not. In previous visualization, we observed that every Plymouth will have almost 50% chance to be a bad buy. Also Mini, Infiniti, Acura, and Lexus are likely to be a bad buy with more than 30% chance. However, we can clearly find out that this conclusion may not be referable because these models have such a small data size. On the other hand, we can make sure those makes which have over 1k data size are more considerable to be an useful feature when we are working on this data and predicting.

Overall, the image provides some useful information about which makes of vehicles are more likely to be considered "Bad buys". However, it is important to note that this is just a general list, and there may be some exceptions.

On the other hand, Fig.3 shows the number of vehicles sold each year, labeled as "VehYear", versus whether the vehicle is considered a "Good buy" or "Bad buy". The data is shown for the years 2001 to 2010. By this picture, we can understand the trend of good buy/bad buy and the total vehicles sold in this period

The overall trend is that as there are more vehicles sold each year, and the percentage of vehicles that are considered a "Bad buy" is roughly decreasing or remain the same based on this visualization.

The largest amount in the number of "Bad buys" occurred between 2004 and 2006 but it seems to be simply increasing with the total number of used car manufacturers keep growing in these years.

However, we cannot know the actual ratio in which year is the best year that we will want our used car is made. We will need another visualization that can tell us this important information.

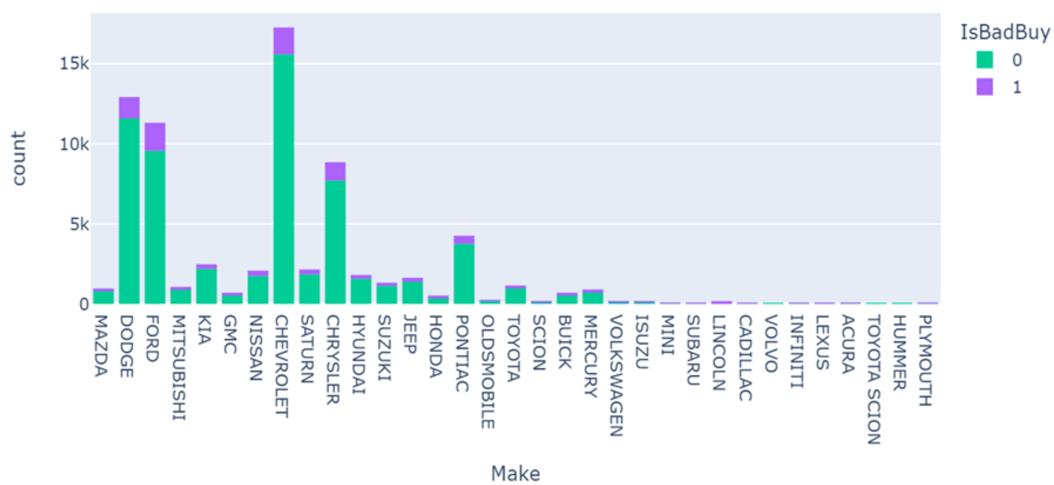

Figure 2: IsBadBuyRate counts for each make

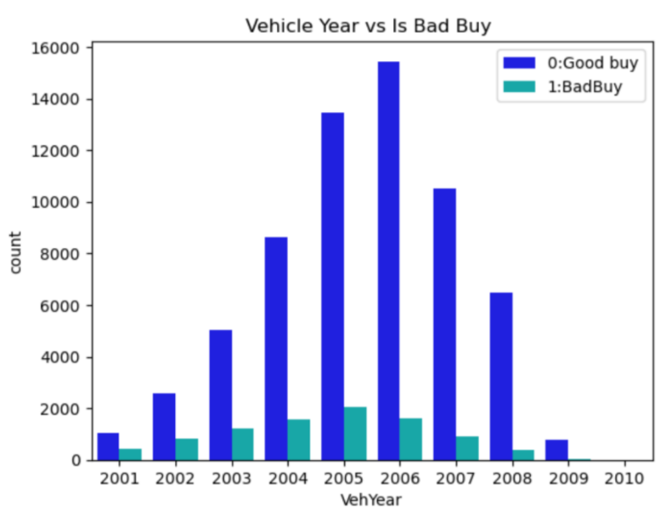

Figure 3: Distribution of vehicle years

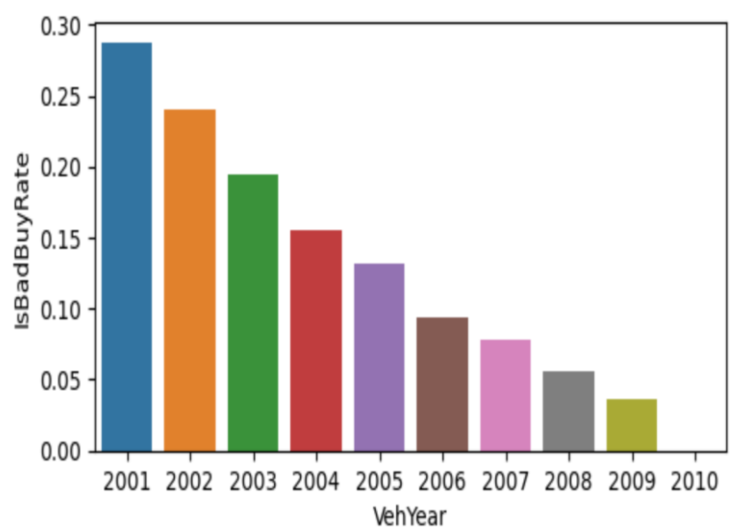

Figure 4: IsBadBuyRate for different vehicle years

Finally, we plot the IsBadBuyRate for cars manufactured in different years as it showed in Fig.4. The IsBadBuyRate is a measure of how likely a car is to be considered a "Bad buy". A higher IsBadBuyRate means that more cars from that year are "Bad buys".

It shows that the IsBadBuyRate has been decreasing in recent years. This means that a lower percentage of cars manufactured in recent years are considered to be "Bad buys".

The isBadBuyRate has decreased from about 0.28 in 2001 to under 0.05 in 2009. This means that the percentage of cars manufactured in recent years that are "Bad buys" has decreased by over 70%.

There are a few possible explanations for this trend. One possibility is that the quality of cars has been increasing in recent years. Another possibility is that consumers are becoming less demanding and are expecting less from their used vehicles. It is also possible that the definition of a "Bad buy" is changing. Or simply the newer car is likely in better condition.

4 Modeling

During this intricate and multifaceted project, a strategic trifecta of sophisticated machine learning models has been judiciously applied to address the myriad challenges at hand. These models, namely the Decision Tree, Random Forest, and the eXtreme Gradient Boosting Classifier (abbreviated as XGBClassifier, signifying "eXtreme Gradient Boosting Classifier"), have been carefully chosen.

A pivotal aspect of our exploration lies in a comprehensive analysis of the performance metrics derived from their application. This entails a meticulous examination of metrics such as the Receiver Operating Characteristic (ROC) curve, the Area Under the Curve (AUC), and an exploration into the relevance of different independent variables as elucidated by these models.

Furthermore, a discerning exploration into the relevance of different independent variables as inferred by these models will provide invaluable insights into the intricate interplay between predictor variables, elucidating their impact on the predictive outcomes generated by the models. Besides the model selections, we also perform Cross-Correlation Function (CCF) analysis to calculate the relevance of different independent variables.

4.1 Decision Trees Model

Decision Trees (DTs) represent a powerful approach in supervised learning, suitable for both classification and regression tasks. The objective is to construct a model capable of predicting the outcome of a target variable. This is achieved by extracting straightforward decision rules from the features within the dataset. Visualize a Decision Tree as a step-by-step guide, breaking down the data into segments and making predictions based on these simple rules. Essentially, a Decision Tree provides a straightforward and segmented estimation of the target variable.

In this project, we selected Decision Tree in the beginning since this is a binary decision question, which is very suitable for DTs.

4.1.1 Test Size Selection

In Fig.5, we can clearly see the trend of AUC and Test Size. Where AUC in machine learning stands for "Area Under the ROC Curve." It is a performance metric used to evaluate the ability of a binary classification model to discriminate between positive and negative examples. AUC represents the area under the receiver operating characteristic (ROC) curve, which plots the true positive rate against the false positive rate at various threshold settings. A higher AUC value (closer to 1) indicates better model performance.

Therefore, we decide to utilize test size 0.3 in our Decision Trees modeling as it will have the highest AUC 0.615 based on Fig. 5.

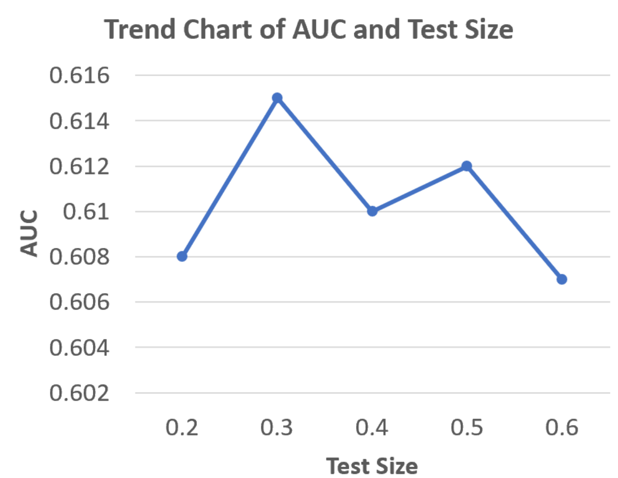

Figure 5: AUC under different Test Size in DTs Model

4.1.2 Performance Analysis in DTs Model

In the analysis depicted in Figure 5, a noteworthy observation arises from the utilization of the Decision Tree Model, revealing an associated Area Under the Curve (AUC) of 0.615. This particular AUC value, falling within the range of 0.5 to 0.7, indicates a moderate level of accuracy but does not reach a level considered highly precise. Specifically, the model's discriminative performance in distinguishing between positive and negative instances appears to be suboptimal, signaling a potential limitation in its ability to robustly capture the underlying patterns within the dataset. The discernment of a 0.615 AUC prompts a thoughtful consideration of potential enhancements or alternative model to achieve a more desirable predictive accuracy.

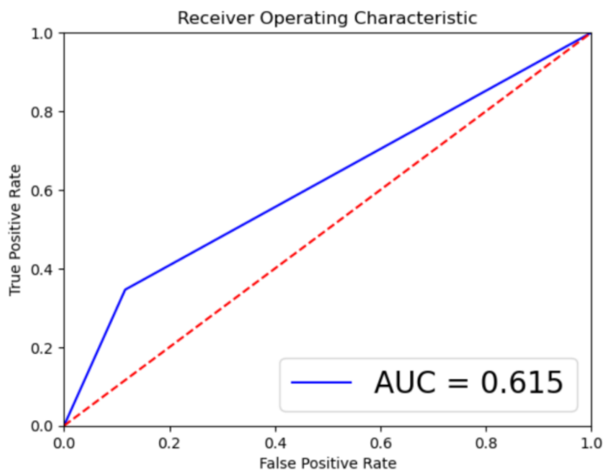

Figure 6: ROC Curve of DTs Model

4.1.3 Features Importance in DTs Model

Within the confines of this dedicated section, our primary objective is to delve into a comprehensive discussion surrounding the pivotal significance of various features encapsulated within the dataset. By meticulously scrutinizing and understanding the relative importance of each feature, we aim to glean invaluable insights that will empower us to discern and prioritize the most crucial aspects. This discernment is particularly pivotal when dealers are navigating the intricacies of auctions, striving to secure advantageous deals. Effectively identifying and highlighting the features that wield substantial influence in determining successful outcomes becomes a strategic imperative in enhancing decision-making processes within the auction context.

In the illustrative depiction presented in Figure 6, an insightful analysis of feature importance reveals that WheelTypeID emerges as the most pivotal among the dataset's attributes. This observation serves as a critical revelation, underscoring the prominence of

WheelTypeID in influencing the overall assessment of a used car's quality. The dataset elucidates the existence of merely two distinct WheelTypeIDs, denoted as 1 and 2, thereby signifying the presence of two distinct types of wheels. This revelation inherently implies that one of these WheelTypes is deemed more reliable over time, emerging as a discernible and consequential factor in the determination of the overall condition and desirability of a used car.

Concurrently, the analysis underscores the significance of the second most influential feature, VNZIP1, denoting the zip code where the car was acquired. While it might initially appear unconventional to accord such importance to a seemingly peripheral factor like zip code in the realm of used car acquisition, a deeper examination of the dataset unveils a noteworthy pattern. Remarkably, out of the expansive dataset comprising over 70,000 data points, a mere 126 zip codes recur repeatedly. This recurrence pattern strongly suggests that each zip code essentially represents a limited number of auctions or even singular auctions exclusively. In effect, this statistical insight substantiates the rationale behind assigning significance to VNZIP1. The data implies that certain auctions, designated by specific zip codes, consistently present superior sources of used cars or employ more robust quality management practices, thus justifying the recognition of zip code as a meaningful feature in the evaluation of used car acquisitions.

In the comprehensive analysis delineated in Figure 6, VehOdo emerges as the third pivotal feature, encapsulating the odometer readings of vehicles. This feature assumes prominence due to its intuitive correlation with vehicular longevity. The logical inference is rooted in the understanding that, as a car accumulates higher mileage over extended journeys, its reliability tends to diminish. VehOdo stands as a tangible metric, representing the cumulative travel undertaken by each vehicle. This recognition extends beyond a mere data point, serving as a robust indicator of wear, tear, and overall reliability.

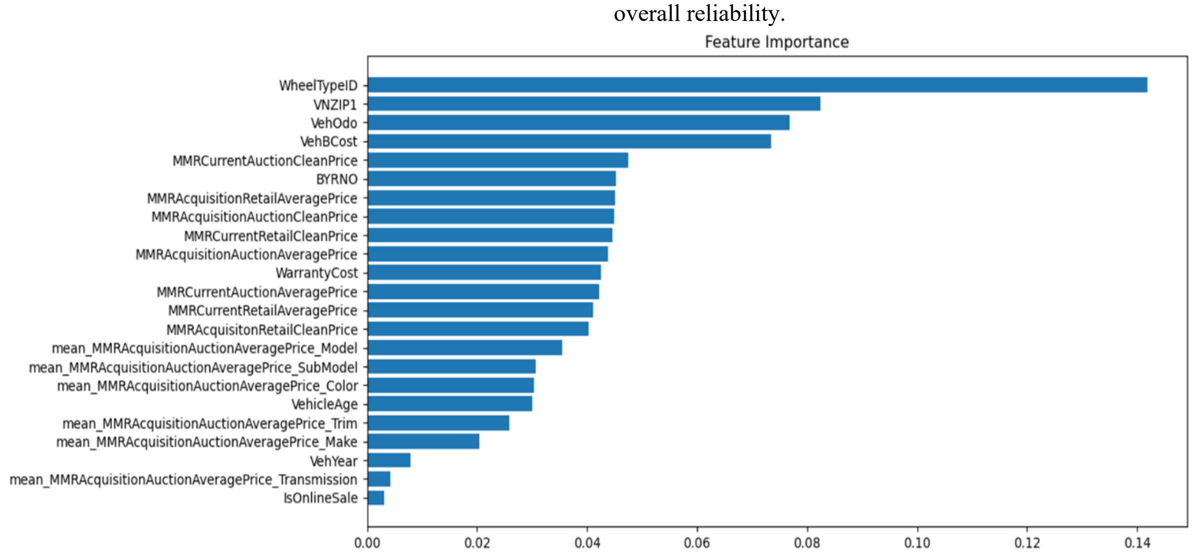

Figure 6: Feature Importance in DTs Model

4.2 Random Forests Model

Random Forests (RF), also known as Random Decision Forests, are a sophisticated ensemble learning technique used for tasks like classification and regression. Instead of relying on a single decision tree, it constructs multiple decision trees during training. In classification, the final output is determined by the majority vote of the trees, while in regression, it's the average prediction of individual trees. One key advantage is mitigating the tendency of decision trees to overfit their training data. This means Random Forests enhances generalization, making them a robust choice for various machine learning applications where accuracy and reliability are paramount.

4.2.1 Test Size Selection

In Fig.7, we find out trend of AUC is still highest when test size equal to 0.3 in Random Forests modeling. Thus we still use 0.3 as our test size in later analysis.

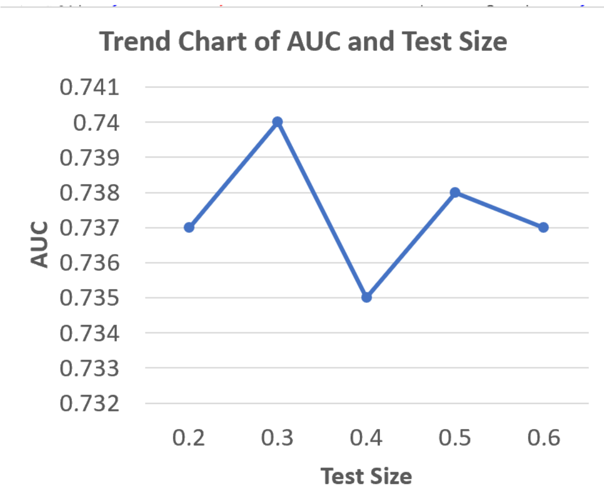

Figure 7: AUC under different Test Size in RF Model

4.2.2 Performance Analysis in RF Model

In the analysis depicted in Figure 8, a noteworthy observation arises from the utilization of the Decision Tree Model, revealing an associated Area Under the Curve (AUC) of 0.74. This constitutes a substantial advancement compared to the preceding Decision Tree models, which yielded an AUC of 0.615. It's pivotal to underscore that an AUC falling within the range of 0.7 to 0.8 is universally acknowledged as generally acceptable in the realm of classification models. The discernment of an improved AUC signifies a considerable enhancement in predictive accuracy, thereby bolstering the reliability of our prognostications within the confines of this refined model.

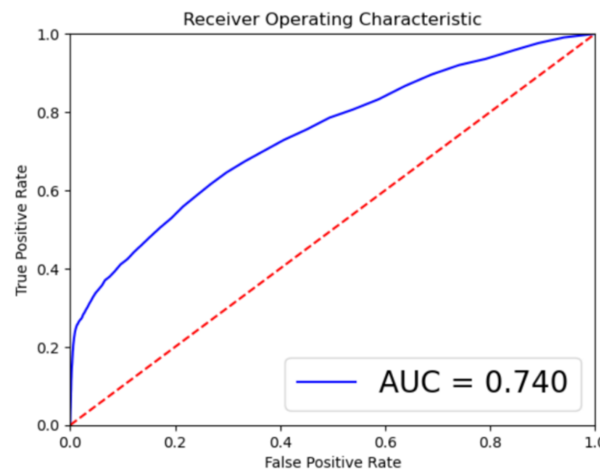

Figure 8: ROC Curve of RF Model

4.2.3 Features Importance in RF Model

In Figure 9, a perceptive scrutiny into feature importance within the Random Forest (RF) model unfolds a notable revelation—WheelTypeID and VehOdo retain their status as the two most influential features, mirroring the insights derived from

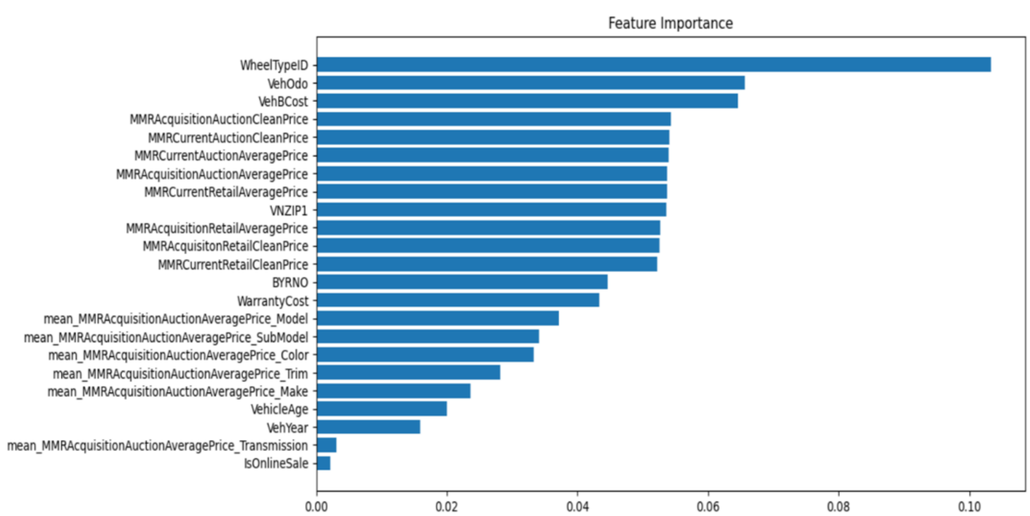

Figure 9: Feature Importance in RF Model

the previous Decision Trees (DTs) model. This correlation is expected, considering that a Random Forest model essentially comprises a conglomerate of Decision Trees. Nevertheless, there emerges a noteworthy addition to this feature hierarchy: VehBCost, recognized as the third most important feature in the RF model, previously held the position of the fourth most crucial feature in the preceding model. VehBCost, denoting the acquisition cost paid for the vehicle at the time of purchase, logically assumes significance, aligning with the intuitive premise that well-maintained cars command higher prices.

4.3 XGBClassifier Model

The XGBClassifier Model, an abbreviation for "eXtreme Gradient Boosting Classifier," represents a cutting-edge advancement in machine learning. This model, akin to the gradient boosting framework but notably more efficient, integrates both linear model solver and tree learning algorithms. The key to its remarkable speed lies in its unique ability to conduct parallel computations on a single machine.

4.3.1 Test Size Selection

In Fig.10, we find out trend of AUC is still highest when test size equal to 0.3 in XGBClassifier model. Thus we still use 0.3 as our test size in later analysis. Also, it means the best test size for all three models we selected in this case is 0.3.

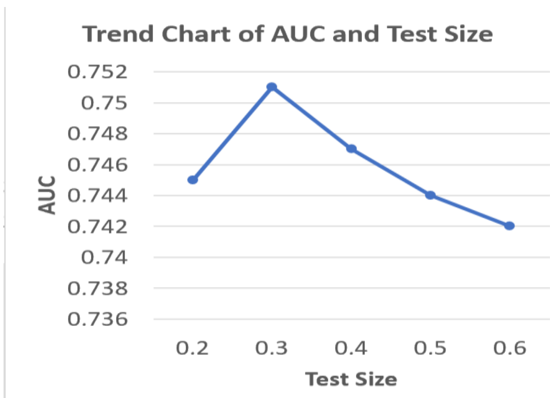

Figure 10: AUC under different Test Size in XGBClassifier

4.3.2 Performance Analysis in RF Model

As per the insights gleaned from Figure 11, it becomes apparent that the XGBClassifier Model exhibits a marginal improvement in terms of the Area Under the Curve (AUC), registering a value of 0.751. This represents a slight enhancement compared to the AUC of 0.740 observed in the Random Forest (RF) model. Notably, the AUC of 0.751 stands as the highest among the three models under consideration. This elevation in AUC signifies that the XGBClassifier Model outperforms its counterparts, asserting itself as the most accurate predictor in discerning whether each purchase is deemed a good or bad acquisition.

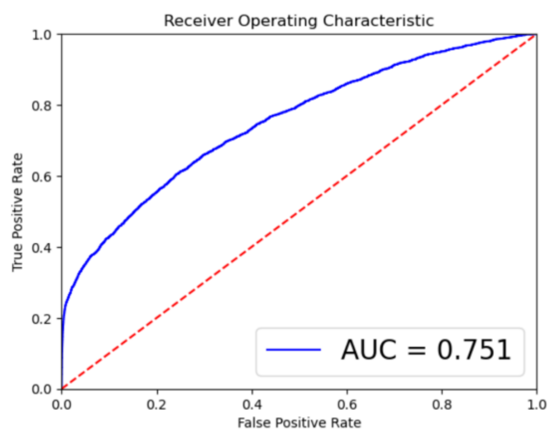

Figure 11: ROC Curve of XGBClassifier Model

4.3.3 Features Importance in XGBClassifier Model

Figure 12 vividly highlights the salience of the top three features: VehOdo, VNZIP, and VehBCost. Strikingly, these features resurface as the key determinants across Decision Trees (DTs), Random Forest (RF), and the eXtreme Gradient Boosting Classifier (XGBClassifier) models, consistently occupying the pinnacle of importance. Notably, VehOdo, capturing the odometer reading, emerges as a pivotal commonality in all three models, underscoring its paramount significance. Its consistent prominence, especially in the most accurate XGBClassifier model, reinforces its role as a robust predictor.

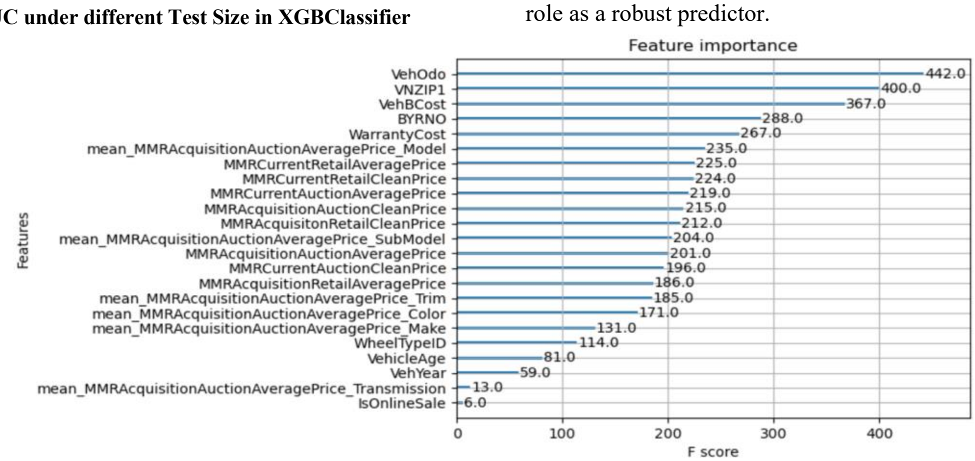

Figure 12: Feature Importance in XGBClassifier Model

4.4 Comparison of Result in Three Models

In this section, we want to compare the result of three different models. As we present in Table 4, it is clear that XGB Classifier has the best accuracy in it's predictions. Other noteworthy information is that the first three models are highly repetitive in this table, which means that all three models agree VehOdo, VNZIP, WheelTypeID, and VehBCost are highly useful to refer if this car purchase is a bad buy or not.

	1st important feature	2nd important feature	3rd important feature	AUC
Decision Trees	WheelTypeID	VNZIP1	Veh0do	0.615
Random Forest	WheelTypeID	Veh0do	VehBCost	0.740
XGB Classifier	Veh0do	VNZIP1	VehBCost	0.751

Table 4: Comparison of Three Models

4.5 Cross-Correlation Function (CCF)

In this section, we want to find out and explain the relevance of different independent variables. In order to achieve this goal, we used the Cross-Correlation Function (CCF) is a mathematical tool used to measure the similarity between two signals or datasets as they vary together over time. It quantifies the degree of correlation or similarity between corresponding values of the two signals at different time points. In simpler terms, CCF helps to identify patterns or relationships between two sets of data by examining how they move in relation to each other. A high cross-correlation indicates a strong similarity or correlation, while a low value suggests a weaker relationship.

In Fig. 13, the x axis from left to right is in the same order as the y axis from up to down. Since our label is 'IsBadBuy', we mainly discuss the other features' cross-correlation with this label. As we can see in the Figure 13, the most related features are Vehicle year and Vehicle Age, which are represent to the same thing. Interestingly, Vehicle Age is never the first 3 important features to consider according to our three models. Therefore, we know that we cannot based on this CCF to filter features out before we do the modeling in this case, otherwise it will likely affect the accuracy of our model.

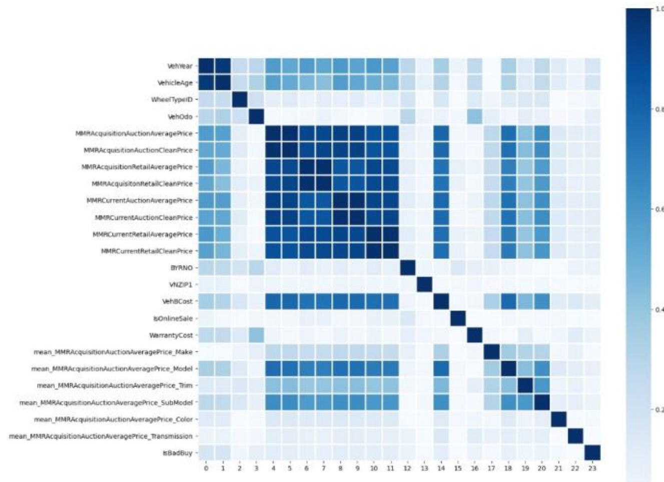

Figure 13: Cross-correlation Distribution

5 Novelty

The novelty in my project is that I also tried to figure out which features are more important to help us to predict if a purchase is a bad buy but not just build a model and produce a yes-or-no answer. The reason for this is that I do think it will be informative and useful for everyone who wants to buy a used car but not only for dealerships.

6 Appendix

As required, well-commented codes already submitted to LMS system.

In this section, we want to mention the drawback of our solutions and how can we handle it.

In the beginning, we did the Cross-Correlation Function (CCF) first before we built three models. We were trying to use CCF to filter more important features since we have 38 features in this project. However, after we filter out half of the less important features based on CCF result and try to fit the rest of features in models, it is much less accurate then we consider everything in it. So we decide still to use all of the features and add-up the analysis of feature importance in every single model. As you can see the result in Section 4, the result of CCF and the models' important features is quite different, which is the reason why we cannot filter out our features based on the result of CCF.

REFERENCES

[1] https://www.kaggle.com/competitions/DontGetKicked
[2] https://www.kaggle.com/code/shabeenashafiulla/dontgetkicked-feature-selection
[3] https://www.kaggle.com/code/mahajo/don-t-get-kicked
[4] https://www.kaggle.com/code/dingli/predict-if-a-car-purchased-at-auction-is-a-lemon