Table of Contents

1. Business Understanding

Used car auctions have been around for decades now in the United States. An auction house hosts a used car auction and they invited buyers and sellers to come and participate in the market. While this market can be very profitable for buyers of used cars since they are often sold for below retail prices, there is the large risk that buyers face in purchasing a faulty vehicle. Commonly referred to as "kicks", these purchases can severely hurt the finances of a car dealership business, as the money spent on them ultimately becomes a wasted sunk cost.

Seemingly functional used cars that end up having no utility value – "lemons" – pose a significant risk to auto dealerships because they may be very difficulty to detect at an auction. This is especially true in cases when there is not much information available to the dealership companies at the time of the auction, or if a car is said to have passed a series of inspections before being auctioned. This information asymmetry has enabled cars with non-trivial defects, including tampered odometers and other unforeseeable mechanical issues, to be auctioned off to unknowing dealerships. These companies are then forced to deal with the steep costs of the kick, including the transportation costs, repair work costs, losses on the resale of the car, and the opportunity cost of missing out on another potentially more viable purchase option. If there is no customer willing to buy the lemon, the dealership is left stuck with unsellable inventory.

Given the high stakes involved for auto dealerships and the incentive they have to ensure every car they purchase at an auction will be sold to a customer thereafter, it would be extremely useful to find a better way to predict at the time of the auction whether or not a car is a lemon. Greater predictability will reduce the likelihood of bad, costly purchases, which on average costs Carvana $6,259 and will allow businesses to improve their inventories, reduce unforeseeable costs, and redefine the customer experience. With a new system, based on an alternative approach to purchasing decisions, used car transactions no longer have to be obscure gambles. In this case, predictive supervised data mining could be used to isolate variables and information about cars that may have a high likelihood of indicating how high the risk of a car being a lemon is.

In delving into this intriguing data-mining problem, our team selected data to be used by Carvana, a start-up business launched with the ambition of improving the auto purchase, finance and trade processes. The company seeks to utilize technological tools and new analytics and systems to change the face of the industry as a whole. Coming up with a useful predictive model that would estimate the risk of an auctioned car being a lemon would provide great value to Carvana, allowing it to maximize the value offered to its customers and enabling for the first time a trustworthy and viable online auto purchase prospect.

To approach this business problem from a data mining perspective, the goal would be to develop a relatively accurate and dependable predictive model that would estimate the probability of a car being a lemon or not. The target variable would be whether or not the car is good or bad, expressed by a probability. It is also important to note that analyzing the problem keeping cost-sensitivities in mind, we recognize the high cost of a false negative, or in other words falsely predicting that a lemon has a higher probability of being a good buy. This would result in a the dealership buying the car thinking it would sellable, incurring transportation/repair costs, and then realizing it is left with a defective car and unsellable inventory. There is also an opportunity cost associated with a false positive, in this case falsely predicting that a good car has a higher likelihood of being a lemon. In this case the dealership would refrain from purchasing a car that would have otherwise generated profit for the company after being successfully sold to a customer. This unrealized profit would be an opportunity cost for the company, but given that dealership did not pay any money to purchase the car at the auction, it would not have a big tangible impact on the company.

Finally we must recognize the important role played by domain knowledge in this specific business problem. In addition to the variables mechanically sorted through in a data set, there are many nuances to the art of predicting the quality of a car at an auction and purchasing it. Thus it is vital that any data-mining model be used in conjunction with domain knowledge, experience and expertise in order to maximize the value it can provide. Ideally, once the dealership has made its initial decision as to whether or not to purchase the car, it can utilize such a data-mining model to heighten assurance that the

decision is correct, to gauge the risk involved in the purchase, and to estimate how much therefore the purchase is worth.

2. Data Understanding

To begin our analysis of this business problem, our team obtained a data set via the Kaggle competition site. The dependent target variable in the data obtained was *isBadBuy*, which is a binary variable categorizing data instances as a bad buy or not a bad buy. Each data instance was a car that was sold at an auction, access to which was provided through the historical training set. The data set was obtained in the form of a .csv file, making it convenient to read and analyze using tools like Microsoft excel, WEKA and python. The entire set contained 72,983 instances, including 32 independent attributes. The data set did, however, contain missing values. The base rate for the target variable *isBadBuy* in the training set was (8976/72983) = 12.298%.

A list of each attribute included in the original data set and its description is included below, as obtained from Kaggle. Each data instance, or each car that was purchased at the auction, was given a unique reference id number. Apart from this there were several features related to the vehicle specifications, including the make, model, submodel, year, trim, color, transmission, nationality and size. Two variables attempted to disclose the demand level for the car, by indicating whether it was manufactured by on of the top three American manufacturers and whether it was a prime unit. Purchase date, acquisition type, kick date, and purchase zip code/state were the variables relating to information about the purchase itself. The data also included a unique identifier number for the buyer that purchased the vehicle. The variable called *AUCGUART* indicated the level of guarantee that the auction provided for the vehicle, ranging form a green light to a yellow light and a red light. For this feature, a green light indicated the car had a guaranteed value, yellow indicated potential for issues, and red indicated it was sold as is. The feature warranty cost indicated how much the customer paid for a warrantee when purchasing the vehicle. Lastly there were a series of numeric features indicating the acquisition cost of the vehicle at both the current time and the time of purchase in good or above good condition in the retail market.

In using this data set, it is important to note that each instance in the set provides information about a vehicle that the dealership actually ended up purchasing at the auction. Since it is historical data, the information is only available to us because the car was actually deemed promising enough for the dealership to have actually purchased it. Once it was purchased, details about the car itself were made available, and we got access to information about whether or not the car ended up being a lemon. Thus, there is to some extent an innate bias in our data set. For this reason it is extremely important that, as mentioned before, any data-mining model developed be used along with domain expertise in order for it to have any useful meaning. Once the dealership has made its preliminary decision about the quality of the car, the model can be used to further assist in assessing the risk inherent in the purchase.

Figure 2.1 List of Features Included in Data Set (See Appendix C)

3. Data Preparation

While data preparation is often an overlooked step in the entire data mining process, it is perhaps the most time consuming and fundamentally important as it is the building block for all other steps that follow. “Often the quality of the data mining solution rests on how well the analysts structure the problems and craft the variables” (Provost and Fawcett). With this in mind, we began our the first step of analyzing the available attributes and finding the ones that presented the most information gain and importance in solving our problem.

It is also important to note that while this section and the others that follow are written in a linear fashion, the process itself was not at all. There was constant rethinking of ideas, methods and conclusions that took place in real time, however for the sake of the reader, we shall present it as a linear process.

3.1 Attribute Selection

The first step was taking a look at the available attributes and converting numeric attributes into nominal ones using the following function in R:

Figure 3.1

```
Test_data$Attribute <- factor(Test_data$Attribute)
```

After converting many of the variables to the desired format, we then proceeded to take a look at the information gain using a ranker method in WEKA as seen in Figure 3.2. It should be noted that numeric variables were discretized on the optimal splitting points that gave the highest information gain.

Figure 3.2

Rank	Attribute	Information Gain
1	MMRCurrentRetailCleanPrice	0.1591972
2	MMRAcquisitonRetailCleanPrice	0.1525066
3	MMRCurrentRetailAveragePrice	0.1521401
4	MMRAcquisitionRetailAveragePrice	0.1475890
5	MMRCurrentAuctionCleanPrice	0.1378067
6	MMRAcquisitionAuctionCleanPrice	0.1352096
7	MMRCurrentAuctionAveragePrice	0.1276983
8	MMRAcquisitionAuctionAveragePrice	0.1272369
9	WheelTypeID	0.0637566
10	WheelType	0.0637026
11	Model	0.0319518
12	SubModel	0.0240122
13	VehicleAge	0.0196533
14	VehYear	0.0179228
15	VehBCost	0.0131958
16	VNZIP1	0.0126275
17	PurchDate	0.0117275
18	WarrantyCost	0.0092531
19	BYRNO	0.0072054
20	Trim	0.0070979
21	VehOdo	0.0056278

Figure 3.2 gives an idea of the top 21 most relevant variables when trying to predict the target variable *IsBadBuy*. Although the attribute *Model* seems to show a relatively high information gain, it does not tell the whole story. The variable *Model*, has over a thousand unique variables, making its information gain highly likely to be a case of overfitting, which is why we did not include in the following model. The same is true for *SubModel*, *Trim, VNZIP1 and BYRNO,* all of which contain an excessive amount of unique values thus representing a case of overfitting.

It is also important to realize that the market price of the car represented by the six attributes that begin with *MMR*, all take into account the Model, Trim and Submodel of

the car as well as many of the unique features[1]. However, it should also be noted that the market price of a car in absolute terms should have little predictive ability since some cars are worth more than others. With this in mind, the market price of a car is only useful when it is used in relative terms.

In terms of the information gain given by *VehBCost*, we could not include it since it represents a case data leakage[2]. The cost of the vehicle (price paid) is not known when looking to bid on a car, so it can not be used in training our model since it will not be available to us when predicting our target variable. You only know the price paid for the vehicle once the bidding has ended. For this reason it could not be used.

The variable *PurchDate* was also not included in the model for intuitive purposes. The past date of purchase of car has little, if not zero, predictive ability when looking at future cars in an auction.

By looking at the information gain given by each attribute on the target variable of *IsBadBuy*, we were able to grasp what variables to include and what variables to use in feature creation process that was to follow.

3.2 Feature Creation

Using the information gain of each variable and the domain knowledge and intuition to remove features that represented leakage, overfitting and other potential data mining obstacles, we proceed to use our understanding of the data to come up with new features that could possibly have higher information gains.

We first identified the top attributes that represented the car's market price at current day and when it was first bought. As we noted before, the stand-alone market price of the car says nothing without it being compared to a relative term. With this in mind, we constructed 4 attributes that represented the difference between the auction and retail market price of the car when it was first purchased and the auction and retail market price of the car in current day adjusted for the age of the car.

---

[1] http://www.manheim.com/help/mmr#mmraccess

[2] Data leakage occurs when data that is unattainable at the time of prediction is used as an attribute in training the model.

We also created one more variable to represent the fact that a pure odometer reading for the car says little without accounting for how old it is. To better illustrate this, think of a car that was bought and then driven form the east coast to the west coast in a year and then sold, versus a car that was driven from the east cost to the west coast in 3 years. Their odometer would read the same, however the average miles driven would be drastically different. Each of these cars should be treated differently in terms of the quality of the car, which is why we created an *AdjVehOd* that took into account this information. For this fact we created an adjusted odometer attribute that represented the average miles driven per year rather than a pure odometer reading.

Figure 3.2 (Python Script in Appendix A)

1. Difference in Aquired v. Current Average Auction Price

$$DifAquiredAvg\_CurAvgAuction = \frac{MMRAcquisitionAuctionAveragePrice - MMRCurrentAuctionAveragePrice}{VehicleAge}$$

2. Difference in Aquired v. Current Above Average Auction Price

$$DifAquiredAvg\_CurAboveAvgAuction = \frac{MMRAcquisitionAuctionCleanPrice - MMRCurrentAuctionCleanPrice}{VehicleAge}$$

3. Difference in Aquired v. Current Average Retail Price

$$DifAquiredAvg\_CurAvgRetail = \frac{MMRAcquisitionRetailAveragePrice - MMRCurrentRetialAveragePrice}{VehicleAge}$$

4. Difference in Aquired v. Current Above Average Retail Price

$$DifAquiredAvg\_CurAboveAvgRetail = \frac{MMRAcquisitionRetailCleanPrice - MMRCurrentRetialCleanPrice}{VehicleAge}$$

5. Adjusted Odometer

$$AdjVehOdo = \frac{VehOdo}{VehicleAge}$$

After creating these attributes we then ran another information gain (“IG) test to compare out newly created features and their IG versus their original features. As you can see in figure 3.3, each of the created variables had a higher information gain than their original counterparts.

Figure 3.3

Rank	Attribute	Information Gain
1	WheelTypeID	0.0639567
2	**WheelType**	**0.0639020**
3	Model	0.0320760
4	**DifAquiredAvg_CurAvgAuction**	**0.0248049**
5	SubModel	0.0238734
6	**DifAquiredAboveAvg_CurAboveAvgAuction**	**0.0234160**
7	**DifAquiredAvg_CurAvgRetail**	**0.0204563**
8	**VehicleAge**	**0.0197949**
9	**DifAquiredAboveAvg_CurAboveAvgRetail**	**0.0189444**
10	VehYear	0.0179685
11	MMRAcquisitionAuctionAveragePrice	0.0141110
12	MMRCurrentAuctionAveragePrice	0.0139565
13	MMRCurrentAuctionCleanPrice	0.0137191
14	MMRAcquisitionAuctionCleanPrice	0.0134356
15	VehBCost	0.0131512
16	VNZIP1	0.0125862
17	**AdjVehOd**	**0.0125713**
18	PurchDate	0.0118711
19	MMRCurrentRetailCleanPrice	0.0116824
20	MMRCurrentRetailAveragePrice	0.0116251
21	WarrantyCost	0.0098552
22	MMRAcquisitionRetailAveragePrice	0.0093032
23	MMRAcquisitonRetailCleanPrice	0.0088381
24	BYRNO	0.0071587
25	Trim	0.0071261
26	VehOdo	0.0056366

After examining the results and using our domain knowledge we decided that we would use the 7 most relevant attributes to build our model which were the following (highlighted above):

WheelType, VehicleAge, AdjVehOd, DifAcquiredAvg_CurAvgAuction, DifAcquiredAboveAvg_CurAboveAvgAuction, DifAcquiredAvg_CurAvgRetail, DifAcquiredAboveAvg-CurAboveAvgRetial.

4. Model

As we began the modeling phase of our project, we were focused on building a model that returned class membership probability. In our case this meant the probability that the car would be either a good or bad buy. Because our problem was strictly oriented towards having the mode accurate probability estimates, we were able to narrow our search for models. Logistic Regression has long been considered "the most common procedure" for estimating class probability so that seemed to be a natural starting point (Provost and Fawcett).

Although logistic regression is the most popular in terms of class membership probability estimate, it is by no means the only option and definitely not the best universal learner. The No-Free-Lunch Theorem clearly states that "any learning algorithm has a limited scope of phenomena that it can capture, or an inherent inductive bias, and there can be no universal learner" (Ben-David, Srebro and Urner). For this reason we also chose to ensemble our Logistic Regression model with a Naïve Bayes model to improve the overall predictive ability of our model.

We however decided not to use a decision tree models since class membership probability is often not as accurate with unbalanced data, such as ours, even with smoothing methods such as Laplace. Additionally, probability from decision trees often is skewed towards 0 and 1 (Chawla and Cieslak). For this reason we chose not to include in an ensemble model.

Another model that we considered was a Support-Vector Machine. SVMs are very powerful and although not quite suited for probability estimation due to their loss functions, there have been many academics papers that discuss calibrating the results to produce sufficient probability estimates (Drish). Nevertheless, SVMs are very computationally demanding and require much more memory in training and use than we are able to support. For this unfortunate reason we could not even build a model to compare, let alone deploy one given the business problem to be solved in real-time.

Before we begin analyzing our model-building phase, it is useful to note that our data and model was being tested against a GINI score that was used in the Kaggle

competition as the scoring metric[3]. For this purpose, when evaluating our models, we used the GINI score and AUC metric to chose our model.

4.1 Logistic Regression Model

For our model-building phase we used R exclusively. For building our logistic regression model we used the *glm* package[4] since it is the most robust and popular in the R ecosystem. We first ran a simple logistic regression on the target variable *IsBadBuy* with all 7 of our selected attributes.

Figure 4.1 (full output in Appendix B)

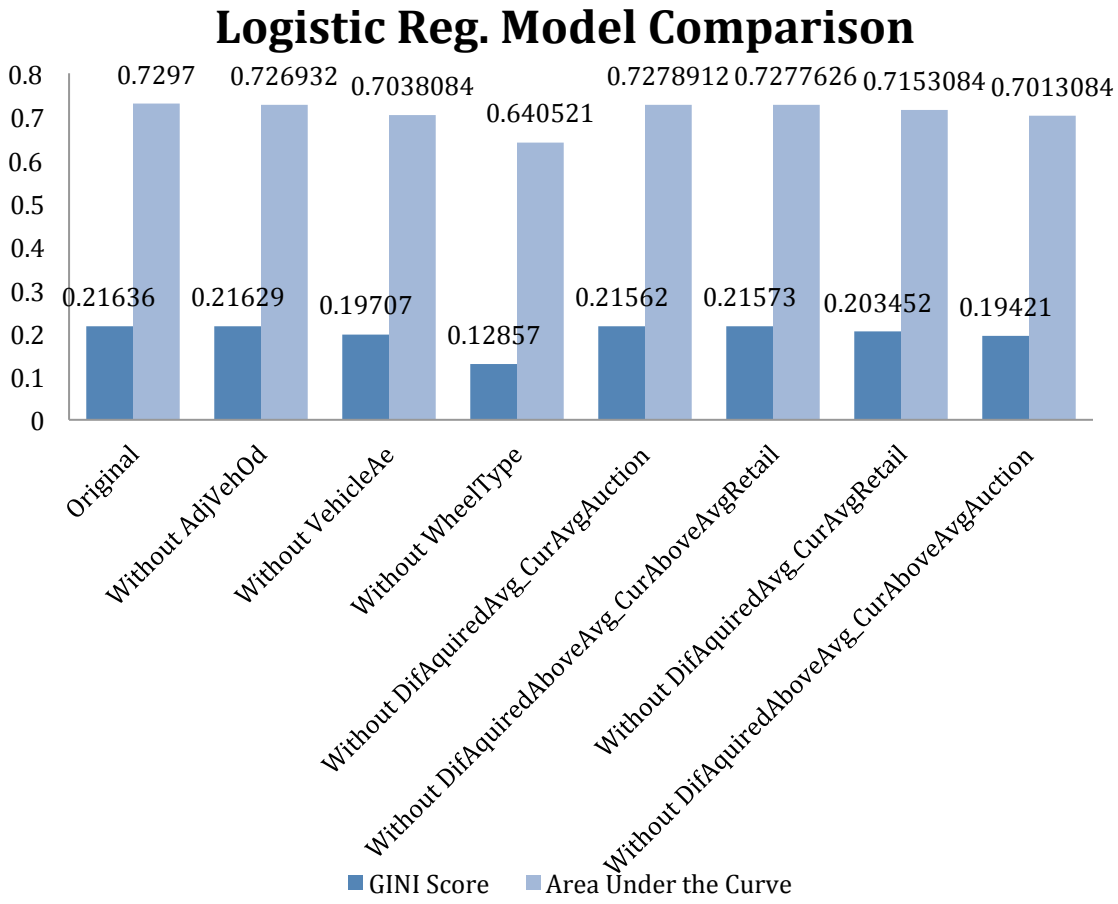
Logistic Regression Model 1		
Coefficients	**Estimate**	**Std. Error**
(Intercept)	-2.08E+00	4.73E-02
VehicleAge	1.67E-01	5.98E-03
WheelTypeCovers	-3.83E-02	1.42E-02
WheelTypeNULL	1.85E+00	2.58E-02
WheelTypeSpecial	8.18E-02	5.98E-02
DifAquiredAvg_CurAvgAuction	-3.41E-04	1.21E-04
DifAquiredAboveAvg_CurAboveAvgRetail	6.05E-05	7.15E-05
DifAquiredAvg_CurAvgRetail	1.49E-04	7.53E-05
DifAquiredAboveAvg_CurAboveAvgAuction	1.15E-04	1.10E-04
AdjVehOd	3.19E-06	1.13E-06

GINI Score: 0.21503
AUC: 0.7279

After building the first logistic regression model we then decided to start optimizing the model by removing selected attributes and checking which one produced the highest GINI score. Below is a graph that shows the GINI score of altered models, ultimately resulting in our original modeling testing the highest.

[3] https://www.kaggle.com/c/DontGetKicked
[4] http://cran.r-project.org/web/packages/glmnet/index.html

Figure 4.2



From the above chart, our original feature selection process seemed to prove successful since the original model recorded the highest scores in GINI and AUC. It should be noted that our original model ranked 300 out of 571 models submitted on Kaggle[5]. While these initial results seemed promising, we decided we could do better and went forward with building a Naïve Bayes probability estimator.

4.2 Naïve Bayes

While Naïve Bayes models are known for making the often-erroneous assumption that all attributes are conditionally independent of each other which often skews probability estimates towards one and zero, we decided to use its value nonetheless since it could possible correct some of the biases that a Logistic Regression model often makes.

[5] https://www.kaggle.com/c/DontGetKicked/leaderboard

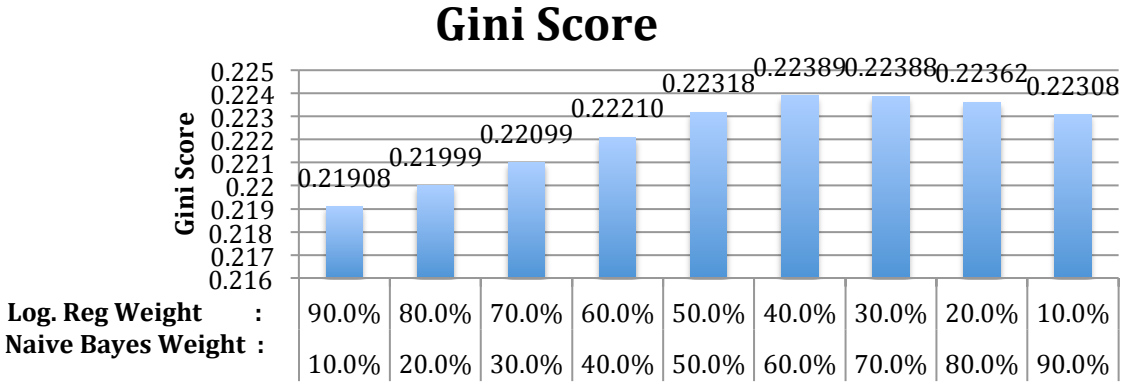
Additionally, the naïve bayes model helps correct a flaw in logistic regression, which is the fact that it cannot output a probability if data is missing. A naïve bayes model does not suffer this same problem and thus can correct instance where a piece of information is missing. With this in mind, we constructed a naïve bayes model separately from the logistic regression model before stacking them. To create the naïve bayes model, we used the *e1071*[6] package in R.

The first model was built using all seven attributes and achieved a GINI score of 0.21774, actually higher than our original logistic regression model, which was surprising. We then tested one more naïve bayes model using Laplace correction, however there was no change since our data set was so large. Given our success of both models, we decided to average their probability estimates in an attempt to mitigate each of the model's biases.

4.3 Stacked Model

After taking the probability estimates and averaging the two equally, we were able to harness both of the model's predicting power while simultaneously reducing their biases. The following graph shows the GINI scores of weighting the Naïve Bayes and Logistic Regression Model's predictions differently:

Figure 4.3



[6] http://cran.r-project.org/web/packages/e1071/index.html

Given the above finding, we decided the optimal weighting would be 40% logistic regression and 60% naïve bayes since we got the highest Gini score of 0.22389. This combined model's Gini score was 0.00789 higher than originally. A small but significant improvement.

4.4 Model Conclusion

Taking a step back, it is important to ask the question, "how does this model help the firm?" It is crucial to always have the end goal in mind of solving your original problem, which is predicting whether a car will be a good buy in the future or a bad buy in the future. The above stacked model helps a buyer at an auction by outputting a probability that the said car will in fact either be good or bad in the future. By doing this, we do not solve the exact problem, but we aid the buyer in his decision and give him an anchoring point.

It is useful to think about this in two examples. Lets say that a buyer is new to the job. He is looking at a used Ford Fusion that is 3 years old and has such and such attributes. He is fairly certain that there is not a high risk that this car is bad, and he in fact is thinking about placing a bid. He then puts the info of the car into our model and sees that in fact there is 73% chance that the car is in fact worthless. With this new piece of information, his thoughts on bidding have changed and we have perhaps saved him from losing on average $6,249 from buying a bad car. Additionally, the model could be used as a conformation tool, when a buyer thinks that the car is either bad or good and can thus confirm it with our model, making him more confident in his decision. While our model cannot claim to *solve* the problem at hand (nor do I believe anyone can), we can confidently say that our model can aid the buyer in his decision, which is exactly what is being asked of us.

5. Evaluation

Evaluating a model is a key step in any data mining process. While we have constantly been evaluating our model throughout each process, this section will focus on

previous evaluation techniques and choices as well as how to evaluate the model in the future as it is deployed.

5.1 Pre-Deployment Evaluation

Evaluating different models and parameter settings is key to selecting the right model. We have constantly been doing that throughout the process using a holdout approach with a test data set, however this section will go through why we chose to use the Gini score as our main metric for evaluation.

Evaluating a model is different in every scenario and is dependent upon not only your data, but the model's objective. Accuracy, true positive rate, precession and the like are almost always concerned with a binary outcome of either a correct or false classification. However, the model we have built and the underlying logistic regression and naïve bayes models output not a pure classification, but the probability of membership to a class. The reason that this is relevant is that all the mentioned evaluation metrics are subject to a cutoff point, meaning we could manipulate the metrics within a range by changing what the necessary probability needs to be to be classified as either good or bad. These metrics are irrelevant however since we are not concerned with predicting whether it is good or bad, but simply how good or how bad the car might be, so we cannot use these traditional evaluation metrics.

Instead we needed to use a metric that takes into consideration the moving thresholds, such as AUC or Gini index. Both of these metrics are closely related to each other as shown below.

Figure 5.1

$$\boxed{Gini = (2 \times AUC) - 1}$$

It should also be noted that the Kaggle competition used a Gini index score, which influenced our evaluation metric since we wanted to compare our model. We could have used AUC, with no difference in results. We chose to use the Gini index score since it allowed us to compare our models against the competition and took into consideration the moving thresholds of our probability estimate. Our final Gini score placed us in the top 47% of competitors and only 0.04 away from top place.

5.2 Post-Deployment Evaluation

Evaluating a model after it is in use is a constant process. While metrics and numbers can be used in the building phase, it can be much harder to use metrics after its deployment depending upon the nature of the model. Because our model outputs a probability estimate, it is hard to measure the accuracy of it as we noted before without setting a threshold, which is impractical in reality. Therefore it is hard to predict an expected improvement by using the model exactly, however we have proposed two alternative methods.

One way to evaluate how effective the model is in the use phase is by taking a look at the baseline rate of buying unsellable cars, which is 12.3%. A business case could be developed by taking a look after using the assisting model for one year (in order to have a large enough sample size) and comparing the baseline rate of unsellable cars to the historic rate of unsellable cars bought at the auction. While this is not a perfect evaluation metric, it could identify if there is any improvement by buyers in identifying which cars are unsuitable. This metric however could be skewed by that particular time period, selling and buying habits, and perhaps the buyers themselves. Nevertheless it would serve a useful benchmark in identifying and large improvements or failures of our model.

Another important metric for evaluating the success of the model is often overlooked and much simpler than people realize. Buying cars at auctions is an art in itself, and many buyers have been doing it for their entire lives. A crucial evaluation metric is the buyer's opinion of the model. By testing out our model, and receiving their input we could greatly improve it or perhaps realize how useless it is. Domain validation is obviously more qualitative in nature as opposed to quantitative nevertheless, it could still prove useful in understanding the effectiveness of the model.

6. Deployment

The deployment phase of the model is when it all comes together and can be seen as one entire product. That doesn't mean the process is over, since it constantly needs to be evaluated and improved as some attributes become relevant and others become

irrelevant as new data presents itself. Nevertheless it is still an exciting time for us as data scientists.

6.1 Use Phase

As we have reiterated through out our analysis, our deployment phase will allow buyers, while they are at auctions, to receive a quick probability estimate that the car they are looking at is in fact worthless. Below we have included an example iPad GUI of what we believe the model would look like from the front-end user perspective.

Figure 6.1

The following is a mere representation of what the model could possibly look like and is more useful as a visualization of the end product. High probabilities would be highlighted in red, while low probabilities would be highlighted in green. Additionally, there would be an import feature for retrieving the market price of the car so it would not have to be manually entered.

6.2 Associated Risks

While there are no glaring risks that our model presents since it is a guide to buyers and not an outright decision based model, it is nonetheless useful to identify some of the possible risks. One risk that Carvana might need to consider is how likely buyers are on relying solely on the model's estimate. It should be well explained and understood by buyers that is it is an aid and worth consulting, but should not be used as a pure decision tool since the dynamics of used car auctions are so complex that a model cannot perfectly fit it. In order to mitigate this risk, a day of training with a demo session would need to take place so all buyers could understand how to use the application and the limitations of the model.

Another risk that Carvana might want to consider is the complexity of the model itself. While we may be able to understand the underlying model, the end user most likely will not. While the front end should be designed so there is as little confusion as possible, buyers might also not use the application if they do not know what is happening behind the front end. This risk could be addressed through a possible information session, but most likely it will take time for them to trust the application and feel comfortable with an outside opinion while making their decisions.

6.3 Conclusion

Despite the above-mentioned risks, we believe that our predictive model can add valuable insight to Carvana and their hundreds of used car auction buyers. While we are aware our deployment is just the beginning of our model, we are nonetheless excited to be a part of the Carvana team. We hope you consider our proposal and the model we have produced and look forward to hearing back from your team.

Works Cited

1. Ben-David, Shai, Nathan Srebro and Ruth Urner. "Universal Learning vs. No Free Lunch results." 28 April 2013 <https://cs.uwaterloo.ca/~rurner/UniversalLearningNIPS2011.pdf>.

2. Chawla, Nitesh V. and David A. Cieslak. "Evaluating Probability Estimates from Decision Trees." 3 May 2013 <http://www.aaai.org/Papers/Workshops/2006/WS-06-06/WS06-06-005.pdf>.

3. Drish, Joseph. "Obtaining Calibrated Probability Estimates from Support Vector Machines." 1 May 2013 <http://cseweb.ucsd.edu/users/elkan/254spring01/jdrishrep.pdf>.

4. Provost, Foster and Tom Fawcett. Data Science for Business: Fundamental principles of data mining and data-analystic thinking. 2012.

Appendix A – Python Script

```
import csv
final =csv.writer(open('training_edit2.csv','wb+'))
training= csv.reader(open("training.csv","rb"), delimiter=',')
###Set Column Names
for row in training:
      final.writerow(row+["DifAquiredAvg_CurAvgAuction"]+["DifAquire
dAboveAvg_CurAboveAvgRetail"]+["DifAquiredAvg_CurAvgRetail"]+["DifAq
uiredAboveAvg_CurAboveAvgAuction"]+["AdjVehOd"])
      break


for row in training:
      if row[22]=="0" or row[22]=="NULL":
            continue
      if row[1]=="1":
            row[1]="TRUE"          ##BAD
      if row[1]=="0":
            row[1]="FALSE"   ##GOOD
### Difference in Price paid 31 and MMRCurrentAuctionAveragePrice 22

      avgCurAucPrice = (int(row[22])+int(row[23]))/2
      avgCurRetailPrice = (int(row[24])+int(row[25]))/2

### Difference in Aquired v. Current Average Auction Price
      if(int(row[5])>0):
            vehAge=int(row[5])
            dif1 = (int(row[18])-int(row[22]))/vehAge
      else:
            dif1 = (int(row[18])-int(row[22]))
### Difference in Aquired v. Current Above Average Retail Price
      if(int(row[5])>0):
            vehAge=int(row[5])
            dif2 = (int(row[21])-int(row[25]))/vehAge
      else:
            dif2 = (int(row[21])-int(row[25]))
### Difference in Aquired v. Current Average Retail Price
      if(int(row[5])>0):
            vehAge=int(row[5])
            dif3 = (int(row[20])-int(row[24]))/vehAge
      else:
            dif3 = (int(row[20])-int(row[24]))
### Difference in Aquired v. Current Above Average Auction Price
      if(int(row[5])>0):
            vehAge=int(row[5])
            dif4 = (int(row[19])-int(row[23]))/vehAge
      else:
            dif4 = (int(row[19])-int(row[23]))
### Veh Miles divided by age
      if(int(row[5])>0 and int(row[14])>0):
            dif5 = int(int(row[14])/int(row[5]))
      else:
            dif5 = int(row[14])
final.writerow(row+[dif1]+[dif2]+[dif3]+[dif4]+[dif5])
```

Appendix B

```
Call:
glm(formula = IsBadBuy ~ ., family = binomial(link = probit),
    data = train_data)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.1725  -0.4899  -0.4040  -0.3173   3.2175

Coefficients:
                                                 Estimate Std. Error z value Pr(>|z|)
(Intercept)                                    -2.079e+00  4.729e-02 -43.953  < 2e-16 ***
VehicleAge                                      1.668e-01  5.978e-03  27.905  < 2e-16 ***
WheelTypeCovers                                -3.834e-02  1.415e-02  -2.709  0.00675 **
WheelTypeNULL                                   1.845e+00  2.577e-02  71.596  < 2e-16 ***
WheelTypeSpecial                                8.184e-02  5.981e-02   1.368  0.17119
DifAquiredAvg_CurAvgAuction                    -3.408e-04  1.213e-04  -2.808  0.00498 **
DifAquiredAboveAvg_CurAboveAvgRetail            6.053e-05  7.150e-05   0.847  0.39724
DifAquiredAvg_CurAvgRetail                      1.494e-04  7.533e-05   1.983  0.04742 *
DifAquiredAboveAvg_CurAboveAvgAuction           1.147e-04  1.096e-04   1.047  0.29527
AdjVehOd                                        3.187e-06  1.132e-06   2.815  0.00488 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 53821  on 72163  degrees of freedom
Residual deviance: 45558  on 72154  degrees of freedom
AIC: 45578

Number of Fisher Scoring iterations: 5
```

Appendix C

List of Features/Attributes Included in Data Set

Field Name	Definition
RefID	Unique (sequential) number assigned to vehicles
IsBadBuy	Identifies if the kicked vehicle was an avoidable purchase
PurchDate	The Date the vehicle was purchased at Auction
Auction	Auction provider at which the vehicle was purchased
VehYear	The manufacturer's year of the vehicle
VehicleAge	The Years elapsed since the manufacturer's year
Make	Vehicle Manufacturer
Model	Vehicle Model
Trim	Vehicle Trim Level
SubModel	Vehicle Submodel
Color	Vehicle Color
Transmission	Vehicles transmission type (Automatic, Manual)
WheelTypeID	The type id of the vehicle wheel
WheelType	The vehicle wheel type description (Alloy, Covers)
VehOdo	The vehicles odometer reading
Nationality	The Manufacturer's country
Size	The size category of the vehicle (Compact, SUV, etc.)
TopThreeAmericanName	Identifies if the manufacturer is one of the top three American manufacturers
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
MMRCurrentRetailCleanPrice	Acquisition price for this vehicle in the retail market in above average condition as of

	current day
PRIMEUNIT	Identifies if the vehicle would have a higher demand than a standard purchase
AUCGUART	The level guarantee provided by auction for the vehicle (Green light - Guaranteed/arbitratable, Yellow Light - caution/issue, red light - sold as is)
BYRNO	Unique number assigned to the buyer that purchased the vehicle
VNZIP	Zipcode where the car was purchased
VNST	State where the the car was purchased
VehBCost	Acquisition cost paid for the vehicle at time of purchase
IsOnlineSale	Identifies if the vehicle was originally purchased online
WarrantyCost	Warranty price (term=36month and millage=36K)