The Impacts of Conditions, Auction Rules and Sellers' Information on Ebay's Car Auctions*

Shuanger Chen † Ante Du ‡ Zeyu Li § Xiaozhe Wang ¶

December 13, 2018

Abstract

This paper is based on Gregory Lewis's *Asymmetric Information, Adverse Selection and Online Disclosure: The Case of eBay Motors*. We made further progress towards analyzing its Ebay transaction data. By applying various methods of machine learning, we tested the variables that might affect the bid price and seller's revenue. The majority of data variables present the assessment of vehicle conditions, yet some variables provide detailed information about how those auctions were conducted. The data also includes some information about the sellers.The aim of this paper is to reveal relations between those variables and auctions and using several machine learning methods to make prediction about the auction revenue.

*Special thanks to Professor Lorenzo Magnolfi, Professor Christopher Sullivan and Dan Mcleod for their patient instructions

†schen586@wisc.edu

‡adu3@wisc.edu

§zli739@wisc.edu

¶xwang978@wisc.edu

1 Introduction

Our research purpose is to find out how the auction itself would impact revenue of sellers.The data we use is from Lewis (2011)'s paper "*Asymmetric Information, Adverse Selection and Online Disclosure: The Case of eBay Motors*". It contains data about the auction format and data about car characters. To better pursue our goal, variables about car characters are taken as control variable, and we use unsupervised learning method to deal with it. Variables about auction format are major elements in the model we constructed through supervised learning.

After manually do regression on all 90 variables that describes auction, we pick 25 of them as our final full set of covariates. Then, we use backwards stepwise subset selection combined with 10-fold cross validation to further eliminate our covariates. The minimum test sample MSE lies in a model using 9 variables, *inspection*, $featured$, *numbids*, *miles*, *phone*, *photos*, *length*, *dealer*, *webpageebizauto*. Basically, the more information sellers provide, the more revenue they would gain. Cars sold by dealer may end up with lower prices.

To get a more precise inspection and prediction, we choose 5 variables about car to cluster the observations. What we find out is that *bookvalue* dominated the clustering process, so we conduct clustering again only using *bookvalue*. The result is similar with the one above. On each group we divided, we develop a model through backward subset selection combined with 10-fold cross validation. One common variable, *photos*, appears on every model and would improve revenue by about 2% overall. The MSE for these five groups are from 0.9475 to 1.0689.

We use both Lasso and Ridge methods to penalize the magnitude of coefficients in order to get a better prediction ability in test samples. Both λ of ridge and lasso is close to 0, and the MSEs are the same, 1.014. Ridge and lasso did not improve our model much in this case.

Random Forest is also a way to improve our research. After performing random forest on our model, MSE decline a portion. *Descriptionsize*, *photos* and *bookvalue* are by far the three most important variables in predicting seller's revenue.

Other than grouping our observation, we tried to group our covariates by Principal Component Analysis to cut down our dimensionality. But the cumulative explained proportion does not reach 80% until the component adds up to 12. And does not reach 90% until 18 groups of components.

Overall, adding more covariates to our model do improve the prediction, and the more information sellers display, they would get a better price. Besides that, *bookvalue* do explain a lot about how much money the seller would actually get.

2 Data and Variables

2.1 Data

The data we use is from Lewis (2011)'s paper *"Asymmetric Information, Adverse Selection and Online Disclosure: The Case of eBay Motors"*. The data was scratched from eBay car, a car auction market has a trade volume of nearly 50,000 cars each month, far beyond those of its competitors. This dataset is a good one to investigate into winning bids and make predictions in the following ways. Firstly, in this dataset, both car characteristics and information about auctions were precisely recorded, which provides us with sufficient variables to learn from. Secondly, due to the large volume in eBay, the dataset provides us with 30,289 winning bids where we can construct our training and test samples in many ways.

2.2 Variables

Variables we use can be summarized into two types. Car characteristics and auction formats. In this paper, we mainly focus on the predicting ability of auction information on the revenue. As a result, we try to include car characteristics as control variables in many ways, including clustering in unsupervised learning and Principle Component Analysis (PCA).

2.2.1 Revenue

The revenue of an auction is the price, which depends on the auction format. In eBay's mechanism, final price is the second highest bid among all bidders. In the data, the variable *Biddy*2 indicates the second highest bid(s) that the goods had received during the auction's active period. We note here that not every car auction ends with a second highest price: some of them have an *buyitnow* option which means the seller set a higher price which the buyer can buy the car at once without waiting for the auction to end, as long as she pays a *buyitnow* price. Fortunately, this won't be a problem with using biddy2 as the revenue since the sell variable has already exclude those sold by *buyitnow*. After keeping observations only when sell equals to one, we have already dropped those end up with *buyitnow*. Meanwhile, another concern would be what happens if there is only one bidder wining, which means she only needs to pay the reserved price (if there is one). In our data, however, we do not know

the exact reserve price the seller set for this auction. Since it is only a small amount (13.8%) in our whole dataset, we only use auctions with a second highest bid in our sample.

2.2.2 Auction

The data set provide us with detailed information about the auction it includes. Firstly, *numbids* is the number of bidders in an auction, which is what we are interested in primarily. Secondly, we can calculate the length and which quarter an auction is in by the starting and ending dates. In particular, we use the ending data as an indicator for quarters because we believe that when bidders bid, they should know that they will not get the car until the auction ends, therefore, the price they are willing to pay should be a reasonable price at the date the auction ends. Moreover, a large set of indicators for information the seller provides including number of photos (*photos*), phone number provided (*phone*), length of text description (*text*), whether the seller is a power seller (*pwrseller*).

2.2.3 Cars

Car information is the deciding factor of the auction price. Among which we notice the car's *bookvalue* is highly correlated with the revenue. inspection means that a car has been inspected or not before it is listed. The car's *miles*, *year* it was produced, whether it is $featured$, and its model describes a car's features. In addition, there are plenty of dummies of car's *scratch*, *ding*, *dent* and *broken* status. Table 1(Summary Statistic) reports the descriptive statistics of variables we use.

TABLE 1: Summary Statistics

Statistic	N	Mean	St. Dev.	Min	Pctl(25)	Pctl(75)	Max
biddy2	26,115	7,452.599	15,093.340	1.000	1,850.000	9,200.000	1,780,300.000
inspection	30,289	0.193	0.394	0	0	0	1
miles	30,289	115,914.700	96,805.080	1	61,612	142,616	500,000
featured	30,289	0.071	0.257	0	0	0	1
bookvalue	17,723	7,488.085	6,630.193	889.000	2,770.000	10,255.500	45,097.000
numbids	30,289	17.476	11.349	1	9	25	123
phone	30,289	0.367	0.482	0	0	1	1
pwrseller	30,289	0.106	0.308	0	0	0	1
photos	25,692	16.121	9.808	1.000	9.000	23.000	95.000
year	30,289	1,988.446	12.605	1,950	1,979	1,998	2,007
addedinfo	30,289	0.292	0.455	0	0	1	1
text	30,289	3,136.257	4,219.610	29	728	3,364	62,703
doors	30,289	2.520	0.870	2	2	4	4
startingdate	30,289	16,960.950	70.705	16,844	16,897	17,022	17,088
endingdate	30,289	16,967.740	70.719	16,853	16,904	17,029	17,094
length	30,289	6.797	1.558	3	7	7	10
age	30,289	18.554	12.605	0	9	28	57
sellerage	30,289	2.951	2.175	−0.548	1.129	4.603	16.241
season1	30,289	0.182	0.386	0	0	0	1
season2	30,289	0.375	0.484	0	0	1	1
season3	30,289	0.376	0.484	0	0	1	1
season4	30,289	0.067	0.249	0	0	0	1

3 Supervised Learning and Model Selection

3.1 Selecting covariates

Due to the large number of our variables, we first manually do regressions on models and eliminate variables from the full model and to a model that with smallest number of variables and compare the R-squared of each model. In model 1, we include all our variables including dummies for the defects (49 out of 90), dummies for models and other characteristics for cars and auction. The second model dropped these defect dummies, and the third model only includes characteristics for cars and auctions.

Table 2: Comparison of Models

	Full Model	Model 2	Model 3
Num. of covariates	90	41	25
R squared	0.769	0.765	0.750

The stepwise regressions show that while dropping out variables, the R squared keeps at a relatively stable level (around 0.75). Indicating that dropping those variables will not have huge influence on our prediction. As a result, we will use Model 3 as our full set of our covariates in the following sections.

3.2 Subset Selection

After picking up the variables we are interested and take most of our prediction tasks, we then use subset selection on our covariates. Since our number of variables are relatively small after the selection in the last chapter, relative to our numbers of observations, we implement of our selection in the "backwards" method, which allows us to start from the full model, and observe change in our criteria (R squared or BIC) when stepwise dropping out covariates.

Subset selection can result in a model containing a subset of covariates. However, a high R-squared or BIC does not guarantee a precise prediction in the test sample, or other samples.

To better select a model in this bias-variance tradeoff, we use 10-Fold Cross Validation to select the best model. The minimum test sample MSE lies in a model using 9 variables. Table 3 shows the comparison between the selected model from backward selection and Model 3.

Table 3: Backward Subset Selection

	Dependent variable:	
	ln(biddy2)	
	(1)	(2)
inspection	0.139***	0.139***
	(0.024)	(0.025)
miles	−0.00000***	−0.00000***
	(0.00000)	(0.00000)
featured	0.334***	0.358***
	(0.037)	(0.038)
numbids	0.014***	0.014***
	(0.001)	(0.001)
phone	0.194***	0.228***
	(0.022)	(0.022)
photos	0.024***	0.028***
	(0.001)	(0.001)
webpageebizautos	−0.131	−0.203
	(0.082)	(0.300)
length	0.063***	0.060***
	(0.006)	(0.006)
dealer	−0.022	0.026
	(0.021)	(0.022)
Constant	7.608***	−0.831
	(0.051)	(1.754)
Observations	11,038	11,038
R^2	0.223	0.238
Adjusted R^2	0.222	0.237
Residual Std. Error	1.006 (df = 11028)	0.997 (df = 11015)
F Statistic	350.972*** (df = 9; 11028)	156.617*** (df = 22; 11015)
Test Sample MSE	0.960	1.020

Notes: This table reports both the model selected by backward selection (Column 1) and model with the full set of covariates (Column 2). In Column 2, other covariates not in Table 1 are included but not presented. Robust standard errors are clustered at the community (village) level and are in parentheses; *** p<0.01, ** p<0.05, * p<0.1.

Column 1 in Table 3 reports the coefficients of 9 covariates selected by backward selection and cross validation. One extra bidder can result in a 1.4% rise in revenue and presenting a phone number for potential buyers to contact the seller can increase the revenue by 19.4%. In addition, one more photo added can contribute in 2.4% more revenue. Length of auction has a positive impact on price, too. 1 day longer can increase the revenue by 6.3%. For car characteristics, a featured car can result in a 33.4% higher in price compared with non-featured cars. Finally, miles do have a negative impact on revenue.

4 Clustering

4.1 Methodology

Since we are interested in how the revenue is impacted by the auction format, but what also affect the revenue a lot are car factors. So, we decided to cluster cars into different groups. Then, conditional on these groups, we can see how the auction format impacts sellers' income by performing ten folds cross validation subset selection.

In clustering, we divided our observations into five groups so that the observations within each group are quite similar to each other, while there might be huge difference between groups. The reason we divided into 5 groups is that we do not have a too many observations, 5 would be a reasonable number of groups for twelve thousand observations.

4.2 Results

In our first clustering, as seen in the table 4, we included *inspection*, *miles*, *bookvalue*, *doors*, *age* which indicate car situation. We excluded *model* and *maker* which are not numeric, because clustering requires all numeric variables. From the Table 4, we found doors is not a

TABLE 4: Mean of each clustered group

Group Number	Inspection	Miles	Bookvalue	Doors	Age	Biddy2
1	0.2662281	90349.03	6691.254	2.746272	9.763377	6606.558
2	0.2100157	146553.58	4541.088	2.911904	11.740692	3798.192
3	0.1250000	484933.39	3879.909	2.634615	12.961538	3477.467
4	0.3016340	29302.25	14077.613	2.659804	6.127451	17583.575
5	0.1978691	217357.61	4047.713	3.072298	12.563166	2809.667

major clustering factor, and *inspection*, *miles*, *age* have correlation with *bookvalue*. In other words, *bookvalue* had already contain most of the information. Therefore, only *bookvalue* would be enough for clustering our observations.

We conduct stepwise subset selection within each subset, as we have done for the whole dataset choosing variables using the cross validation method. Even though seller's revenue has strong positive relation with *bookvalue*, we did not put it in the regression model, because we have already conditional on it. Different subsets were selected in different groups. The result of each groups model is shown in Table 5.

In the first group, which has the second highest *bookvalue*, the subset selection result shows that only the variable photo is picked. As we expected, if the auction provides car's *photo*, seller's revenue would increase by 2.49%. The second group has lowest average *bookvalue*. Subset selection within this group gives us a optimal subset of two variable——*photo* and *season*1. Coefficient of *photo* is similar with that in first group. Sellers may gain 4.84% more in season one.The third group has highest average *bookvalue*, and *text*, *descriptionsize*, *inspection*, $featured$, *photos* was picked. $Text$ and *descriptionsize* have small coefficients, that is because they have large numbers. An interesting finding is that they have opposite coefficient signs. It would not make sense in econometrics, but in machine learning, we decide to count them in for better prediction. If seller provide inspection for the car, their

revenue would increase by 6.71%. The coefficient of $featured$ is 0.5342, meaning if the car is featured, the revenue will go up dramatically by 53.42%. The forth group has second lowest average *bookvalue*. The coefficient of $featured$ is also large in this group, 0.4343. *webpageauction*123 also have a large coefficient. It indicates that selling cars on webpage auction123, seller will gain 47.33% more. And seller gain about 7% more when post the car on ebayhosting instead of other webpages. Inspection plays an important part in this model. Providing photo has about the same impact as above.The fifth group has median average *bookvalue*. *Featured*, *numbids*, *phone*, *photos* and *descriptionsize* have positive affect on this group, while *season*1, *webpageauction*123, *webpagecarad*, *webpageebayhosting*, have negative impact on the revenue in this group.

TABLE 5: MSE of each groups regression

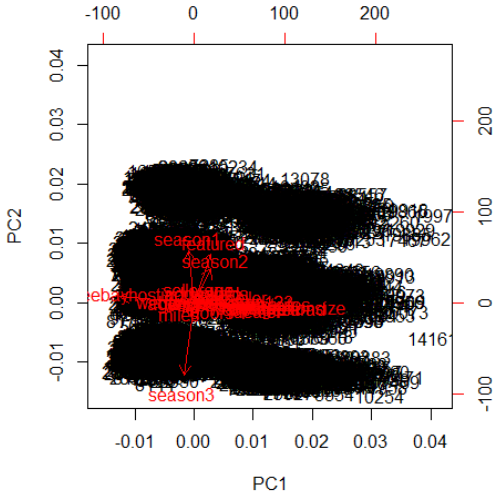
group number	group 1	group 2	group 3	group 4	group 5
mse	1.006	1.021	0.960	1.061	0.947

The MSE for these five groups are from 0.9475 to 1.0689. It indicates the prediction is decent. In short, providing *photos* on the website will almost surely help the seller. But some *webpages* may have different influence on different groups. Since some website may have expertise on selling inexpensive car, and, when expensive cars were put on for sale, buyer may assume the car is inexpensive. Overall, the more details sellers provide, the more they will gain.

TABLE 6: regression on best fit covariate in each clustered group

	Dependent variable:				
	ln(biddy2)				
	(1)	(2)	(3)	(4)	(5)
text			−0.0001		
			(0.00004)		
descriptionsize			0.00000		0.00000
			(0.00001)		(0.00001)
inspection			0.067	0.331***	0.185*
			(0.210)	(0.066)	(0.104)
phone				0.165**	0.086
				(0.066)	(0.109)
featured			0.534	0.434***	0.056
			(0.348)	(0.100)	(0.176)
numbids					0.010**
					(0.004)
webpageauction123				0.473**	−0.237
				(0.207)	(0.412)
webpagecarad					−0.361
					(0.348)
webpageebayhosting				0.072	−0.312
				(0.093)	(0.462)
photos	0.025***	0.026***	0.025**	0.018***	0.026***
	(0.003)	(0.002)	(0.010)	(0.004)	(0.006)
season1		0.048			−0.041
		(0.048)			(0.118)
Constant	8.072***	8.045***	8.063***	7.938***	8.052***
	(0.068)	(0.041)	(0.167)	(0.129)	(0.499)
Observations	820	2,672	147	1,283	477
R^2	0.059	0.061	0.065	0.081	0.123
Adjusted R^2	0.058	0.060	0.032	0.077	0.105
Residual Std. Error	1.004 (df = 818)	1.011 (df = 2669)	1.001 (df = 141)	1.033 (df = 1276)	0.985 (df = 466)
F Statistic	51.724*** (df = 1; 818)	86.285*** (df = 2; 2669)	1.972* (df = 5; 141)	18.762*** (df = 6; 1276)	6.562*** (df = 10; 466)
Note:					*p<0.1; **p<0.05; ***p<0.01

FIGURE 1: Principle Component Analysis



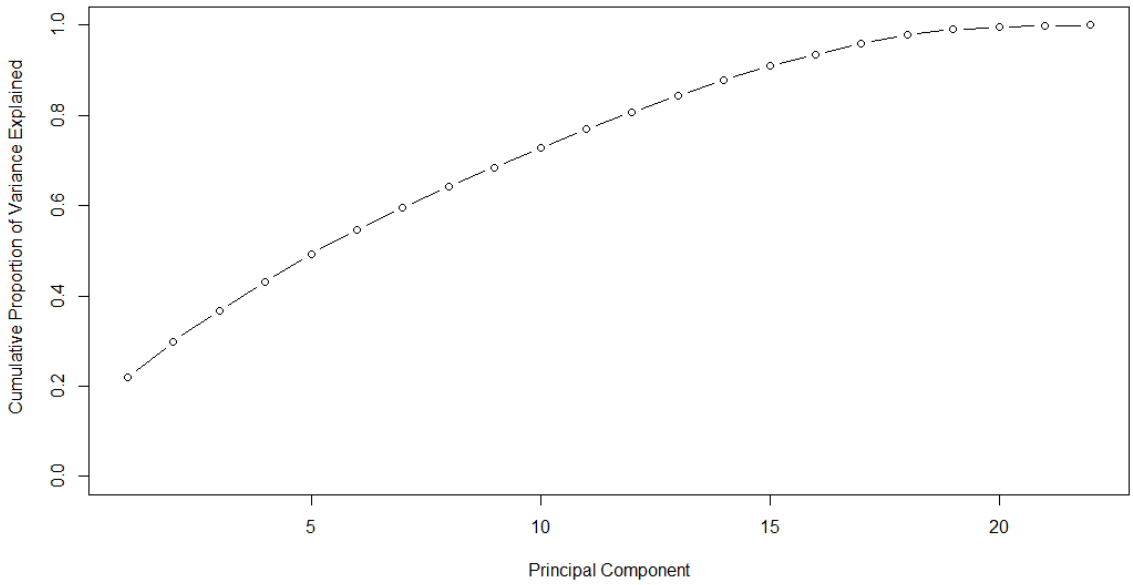
5 Principal component analysis

We have selected 22 variables in to conduct our research, but the dimensionality is a big problem. We would like to find a low-dimensional representation of the data that captures as much of the information as possible. Intending to obtain a two-dimensional representation of the data that captures most of the information, we plot the observations in a low-dimensional space. It finds a low-dimensional representation of a data set that contains as much as possible of the variation.

As we can see in Figure 1, it does not turn out very well. That might because of we have to many observations.

The figure shows us that the cumulative explained proportion does not reach 80% until the component adds up to 12. And does not reach 90% until 18 groups of components. It shows that principal component did not do much in lowering the dimensionality here.

FIGURE 2: Principle Components and Ability in Explanations



6 Lasso and Ridge

Penalized regression is another way to reduce variance thus enhance performance in test samples. We use both Lasso and Ridge methods to penalize the magnitude of coefficients in order to get a better prediction ability in test samples. Similar to the subset selection approach, we use 10-Fold Cross Validation to select the best tuning parameter ϵ.

Both λ in Lasso and Ridge is relatively small, indicating that the scale of penalization is not too much. In addition, both Lasso and Ridge give similar prediction error in test samples, with an MSE of 1.014, which is slightly higher than our MSEs in the subset selection model.

TABLE 7: Tuning Parameters and MSEs of Lasso and Ridge

	Lasso	Ridge
λ^*	0.000066	0.0022
MSE	1.014	1.014

7 Random Forest

7.1 Methodology

We can consider random forest process as the process of decorrelating the tree partially. By choosing split candidates from a random sample of m predictor without consider a majority of the available predictors, overcome the problem that averaging many highly correlated quantities does not lead to substantial reduction in variance. Our goal in this part is to use random forest to predict auction revenue based on 500 observations of succeed auctions that have the largest variance in the training set. We randomly divided the observations into training data and test data and apply random forest model to the training set for values of the number of splitting variables. We run regression contains all predictors that be considered for each split of the tree and regression that only pick 7 predictors for each split of the tree, then compare the result of each other.

The variable we get show importance is IncNodePurity which measure of total of the decrease in node impurity that results from splits over that variable, average over all trees (this was plotted in figure4). In the case of regression trees, the node impurity is measured by the training RSS, and for classification trees by the deviance. The importance we get mean the features that are more closely related with dependent variable and contribute more for variation of the dependent variable.

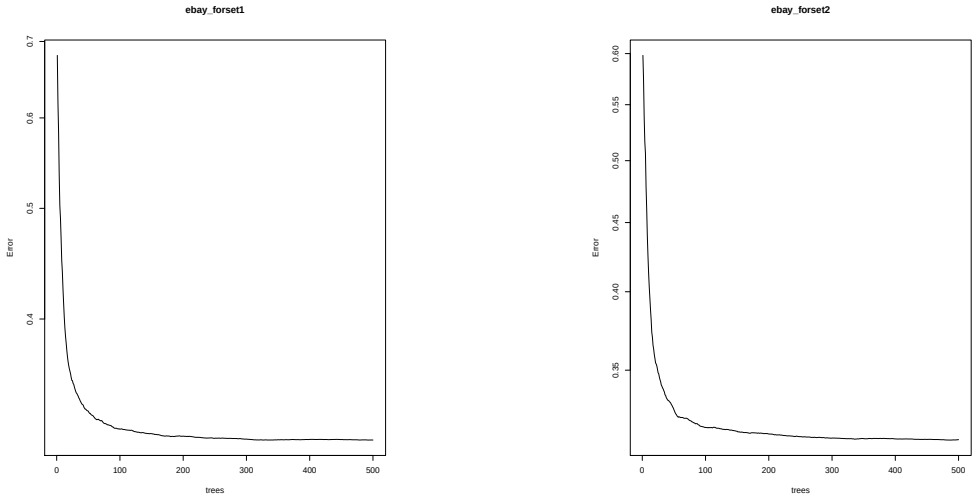


FIGURE 3: Best Number of Trees to do split

7.2 Result

The result shows in the figure 3. The error rate of a single tree is more than 60% and the null rate is more than 70%. We see that using 400 trees is sufficient to give good performance. Compared the figure and the MSE in Table 8 of bagged regression tree with which growing a random forest proceed with mtry equal 7, we can find the latter one has smaller MSE than the former one. This indicates that random forest yielded an improvement over bagging in this case.

The results of importance indicate that across all of the trees considered in the random forest, descriptionsize, photos and bookvalue are by far the three most important variables. For these are more useful, we will tend to split these mixed label nodes into pure single class node.

FIGURE 4: importants

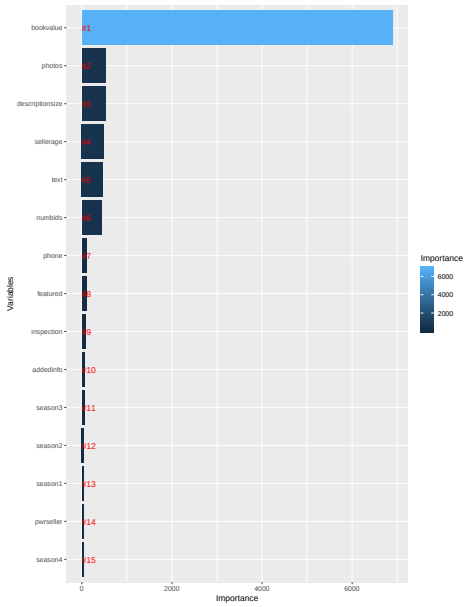


TABLE 8: Random Forest MSE

	Baggage	RF
MSE	0.3070008	0.3048053

8 Mostly Commonly Observed Car Model

presents numbers of each model. There are 6,319 Mustang cars in our data, which is the most commonly observed model. In Table XXXXXX, we present regression results using the same covariates we pick in the backward selection process (see Table XXXX). The coefficients show that almost all covariates we select have a statistically and substantially significant impact on the revenue. One extra bidder would rise the revenue by 1.33%. One day longer in auction length would result in 6.63% increase in revenue. Compared with seller not providing phone

Table 9: Car Models

Model	Freq.	Percent	Cum
3-Series	242	0.8	0.8
Accord	2,315	7.64	8.44
Altima	241	0.8	9.24
Camaro	3,666	12.1	12.1
Camry	1,308	4.32	4.32
Civic	2,516	8.31	8.31
Corolla	606	2	35.97
Corvette	3,618	11.94	47.91
F-100	660	2.18	2.18
F-150	2,501	8.26	8.26
F-250	1,989	6.57	64.91
F-350	1,346	4.44	69.36
Maxima	387	1.28	70.64
Mustang	6,319	20.86	91.5
Silverado 1500	599	1.98	93.48
Silverado 2500	273	0.9	94.38
Tacoma	606	2	96.38
Thunderbird	1,097	3.62	100
Total	30,289	100	

numbers, auctions with a phone number end up with 20.4% higher in price. One more photo would contribute 2.27% increase in revenue. In addition, having car inspected, less miles, featured cars, not using Ebizautos will all make revenue higher. Dealers, however, does not have a significant impact on the revenue.

These results provide helpful suggestions for Mustang sellers to design an auction. Firstly, webpages with more precise car information (photos, texts) can better eliminate asymmetric information problem (Lewis, 2011). Secondly, a longer auction length is good for sellers to attract buyers' attention. Moreover, inspections and features will both make cars popular.

9 Conclusion

In our paper, we predict revenue of auctions (winning bid payment) base on auction and product characteristics. After dropping the covariates that have no huge influence on our prediction, we use subset selection in supervised learning part and found that integrality and

TABLE 10

VARIABLES	(1) biddy2log
inspection	0.154***
	(0.032)
miles	-0.000***
	(0.000)
featured	0.287***
	(0.046)
numbids	0.013***
	(0.001)
phone	0.204***
	(0.029)
photos	0.023***
	(0.002)
webpageebizautos	-0.381***
	(0.126)
length	0.066***
	(0.009)
dealer	0.037
	(0.029)
Constant	7.575***
	(0.071)
Observations	4,680
R-squared	0.195

Robust standard errors in parentheses
*** p<0.01, ** p<0.05, * p<0.1

precise of information are important to predict auction revenue better. Another supervised learning method, lasso and ridge, does not penalize much which also indicate predict better with adequate information. The same finding was gotten in our principle component analysis part in unsupervised learning, for it is hard to get weighted correlation between variables in this dataset. Concerned another part of unsupervised learning, clustering, core was to figure out different fit best variables in 5 different level car group and predict base on them, then conclude same way. Other than focus on result, the random forest model analysis the advantage of it by the result. Finally, auction revenue vary by number of bidders, time of years, text, photos, dealer, webpage, phone obviously.

We only used one model for every method basically, the second model can be applied to test our result in first model in the future research. For instance, bootstrap would be a good way to prove our result in cross validation in resampling part.

10 References

[1] Lewis, Gregory. "Asymmetric information, adverse selection and online disclosure: The case of eBay motors." *American Economic Review* 101.4 (2011): 1535-46.

[2] James, Gareth, et al. "An Introduction to Statistical Learning: with Applications in R (Springer Texts in Statistics)." (2013).

[3] Friedman, Jerome, Trevor Hastie, and Robert Tibshirani. "The elements of statistical learning." (2001).

11 Appendix

```
#clear workspace
rm(list=ls())

#print results
print_results=T

#prepare the package we need
library(readr)
library(tidyverse)
library(dummies)
library(Lahman)
library(modelr)
library(nnet)
library(fastDummies)
library(chron)
require("data.table")
require("ggplot2")
require("knitr")
require("glmnet")
require("leaps")
library(randomForest)
library(cluster)
library(stargazer)
options(scipen=999)

# input data
ebay <- read_csv("/Users/ante/Desktop/690\ final\ project/20090124_data/690.csv")

#################################################################
#######
#initialize some useful values and add some variables we need in the later

# add id to to let operation, like merge data easier in the later
ebay$id <- 1:nrow(ebay)

# pick month up to know this auction happened what time in a year
ebay <- ebay %>%
  mutate(kaishi = substr(startdate,1,9),
         jieshu = substr(enddate,1,9))
```

```
ebay <- ebay %>%
  mutate(startd = as.Date(kaishi, "%b-%d-%y"),
         endd = as.Date(jieshu, "%b-%d-%y"))

ebay <- ebay %>%
  mutate(startmonth = substr(startd,6,7),
         endmonth = substr(endd,6,7))

# divide month to 4 seasons dummy
ebay$season1 <- 0
ebay$season1[(ebay$endmonth == '01')|(ebay$endmonth == '02')|(ebay$endmonth
=='03')] <- 1
ebay$season2 <- 0
ebay$season2[(ebay$endmonth == '04')|(ebay$endmonth == '05')|(ebay$endmonth
=='06')] <- 1
ebay$season3 <- 0
ebay$season3[(ebay$endmonth == '07')|(ebay$endmonth == '08')|(ebay$endmonth
=='09')] <- 1
ebay$season4 <- 0
ebay$season4[(ebay$endmonth == '10')|(ebay$endmonth == '11')|(ebay$endmonth
=='12')] <- 1

# get dummy variables that we need for model, trans and webpage
ebay <- cbind(ebay, as.data.frame(lm(season3 ~ model, data = ebay, x=TRUE)$x))
ebay <- select(ebay, -'(Intercept)')

ebay <- cbind(ebay, as.data.frame(lm(season3 ~ trans, data = ebay, x=TRUE)$x))
ebay <- select(ebay, -'(Intercept)')

ebay <- cbind(ebay, as.data.frame(lm(season3 ~ webpage, data = ebay, x=TRUE)$x))
ebay <- select(ebay, -'(Intercept)')

#Logarithm Value
ebay$biddy2=log(ebay$biddy2)

# get rid of dependent linear problem between year and age
ebay <- ebay[,-c(which(colnames(ebay)=="year"))]

##################################################################
###########
# screen the information we need in the data set
#Dataset we will use
data = ebay[,c(which(colnames(ebay)=="biddy2"),
               which(colnames(ebay)=="inspection"):
```

```
                    which(colnames(ebay)=="descriptionsize"),
                  which(colnames(ebay)=="doors"),
                  which(colnames(ebay)=="webpage"):
                    which(colnames(ebay)=="condition"),
                  which(colnames(ebay)=="length"):
                    which(colnames(ebay)=="dealer"),
                  which(colnames(ebay)=="season1"):
                    which(colnames(ebay)=="season3"),
                  which(colnames(ebay)=="text"),
                  which(colnames(ebay)=="webpageauction123"):
                    which(colnames(ebay)=="webpageebizautos")
                  #which(colnames(ebay)=="model")
)]

data = data[,-c(which(colnames(data)=="condition"),
                  which(colnames(data)=="relistflag"),
                  which(colnames(data)=="reserve"),
                  which(colnames(data)=="buyitnow"),
                  which(colnames(data)=="title"),
                  which(colnames(data)=="sell"),
                  which(colnames(data)=="webpage")
)]

datamodel = ebay[,c(which(colnames(ebay)=="biddy2"),
                    which(colnames(ebay)=="inspection"):
                      which(colnames(ebay)=="descriptionsize"),
                    which(colnames(ebay)=="doors"),
                    which(colnames(ebay)=="webpage"):
                      which(colnames(ebay)=="condition"),
                    which(colnames(ebay)=="length"):
                      which(colnames(ebay)=="dealer"),
                    which(colnames(ebay)=="season1"):
                      which(colnames(ebay)=="season3"),
                    which(colnames(ebay)=="text"),
                    which(colnames(ebay)=="modelAccord"):
                      which(colnames(ebay)=="modelThunderbird"),
                    which(colnames(ebay)=="webpageauction123"):
                      which(colnames(ebay)=="webpageebizautos")

)]

datamodel = datamodel[,-c(which(colnames(datamodel)=="condition"),
                          which(colnames(datamodel)=="relistflag"),
                          which(colnames(datamodel)=="reserve"),
```

```
                                    which(colnames(datamodel)=="buyitnow"),
                                    which(colnames(datamodel)=="title"),
                                    which(colnames(datamodel)=="sell"),
                                    which(colnames(datamodel)=="trans"),
                                    which(colnames(datamodel)=="model"),
                                    which(colnames(datamodel)=="webpage")

)]

#the whole dataset
datafull = ebay[,c(which(colnames(ebay)=="biddy2"),
                           which(colnames(ebay)=="inspection"):
                             which(colnames(ebay)=="descriptionsize"),
                           which(colnames(ebay)=="doors"),
                           which(colnames(ebay)=="webpage"):
                             which(colnames(ebay)=="condition"),
                           which(colnames(ebay)=="length"):
                             which(colnames(ebay)=="dealer"),
                           which(colnames(ebay)=="season1"):
                             which(colnames(ebay)=="season3"),
                           which(colnames(ebay)=="text"),
                           which(colnames(ebay)=="modelAccord"):
                             which(colnames(ebay)=="modelThunderbird"),
                           which(colnames(ebay)=="ding_good"):
                             which(colnames(ebay)=="broken_pics"),
                           which(colnames(ebay)=="webpageauction123"):
                             which(colnames(ebay)=="webpageebizautos")

)]

datafull = datafull[,-c(which(colnames(datafull)=="condition"),
                                  which(colnames(datafull)=="relistflag"),
                                  which(colnames(datafull)=="reserve"),
                                  which(colnames(datafull)=="buyitnow"),
                                  which(colnames(datafull)=="title"),
                                  which(colnames(datafull)=="sell"),
                                  which(colnames(datafull)=="trans"),
                                  which(colnames(datafull)=="model"),
                                  which(colnames(datafull)=="webpage")

)]

#Compare Datasets we get from two front steps by do linear regression on each other
seperately
```

```
z = summary(lm(data))

f = summary(lm(datamodel))

g = summary(lm(datafull))

stargazer(lm(data))

# compare R square for each other
z$r.squared

f$r.squared

g$r.squared

#####################################################################
############
#principle component analysis(unsupervise learning)

data1 = data[,c(which(colnames(data)=="miles"),
                which(colnames(data)=="featured"),
                which(colnames(data)=="numbids"),
                which(colnames(data)=="phone"),
                which(colnames(data)=="photos"),
                which(colnames(data)=="webpageebayhosting"),
                which(colnames(data)=="length"))]
#drop observations that contains NA
datapca= drop_na(data)

#drop varible we do prediction on and variable that might absorb other variable effect
datapca = datapca[,-c(which(colnames(datapca)=="biddy2"),
                      which(colnames(datapca)=="bookvalue"))]

#center the variables to have mean zero
pr.out =prcomp (datapca , scale =TRUE)

#contain the corresponding priciple component loading vector
pr.out$rotation

# plot two principle components
biplot (pr.out , scale =1, pc.biplot = FALSE)

#obtain variance explained by each principle component and compute proportion of it
pr.var =pr.out$sdev ^2
```

```
pve=pr.var/sum(pr.var )

#compute the cumulative sum
cumsum(pve)

#plot PVE explained by each component, as well as the cumulativ PVE
plot(pve , xlab=" Principal Component ", ylab=" Proportion of Variance Explained ",
ylim=c(0,1) ,type='b')
plot(cumsum (pve ), xlab=" Principal Component ", ylab ="
       Cumulative Proportion of Variance Explained ", ylim=c(0,1) ,
       type='b')

#####################################################################
####################
# subset selection

#drop variable that might absorb other variable effect and observation that contain NA
datacv <- data[,-c(which(colnames(data)=="bookvalue"))] %>%
  drop_na()

# divide data to training data and test data randomly
set.seed (1)
train=sample (c(TRUE ,FALSE), nrow(datacv ),rep=TRUE)
test =(! train )

#perform backward stepwise selection to select best fit variables
fit.best=regsubsets(biddy2~.,data=datacv[train ,],
                            method = "backward", nvmax = 24)
plot(fit.best, scale = "adjr2")
summary(fit.best)

#make a model matrix from the test data
test.mat=model.matrix(biddy2~.,data=datacv[test ,])

#we extract the coefficients from the regfit.best and use them to form the predictions
and compute test MSE
val.errors =rep(NA ,10)
for(i in 1:10){
  coefi=coef(fit.best ,id=i)
  pred=test.mat [,names(coefi)]%*% coefi
  val.errors [i]= mean((datacv$biddy2[test]-pred)^2)
}

# use smallest MSE to pick fittest model variables
```

```
which.min(val.errors)
coef(fit.best, 10)

# use the subset we get from the subset selection to do regression
set.seed(1)

train=sample (c(TRUE ,FALSE), nrow(datacv),rep=TRUE)
test =(! train )

lm2 <- (lm(biddy2~inspection+miles+featured+numbids+phone
                +photos+webpageebizautos+length+dealer, data = datacv[train,]))

# get mse
mse2 <- mse(lm2, datacv[test,])

#use the whole dataset after we handled to do regression
datar <- datacv

mod3 <- lm(biddy2~., data=datar[train,])

# get mse
mse3 <- mse(mod3, datacv[test,])

#summary regression and compare R square with each other
o = summary(mod3)
r = o$r.squared
q = summary(lm2)
h = q$r.squared
#################################################################
#####

#Lasso and Ridge
d = cbind(data$biddy2,data1)

d = d[,-14]

d= d[complete.cases(d[,1]),]

#Drop NAs
d <- d %>%
   drop_na()


d_shuffle = d[sample(nrow(d)),]
```

```
#Using Cross Validation to pick the best tuning parameter
cv = function(lambda,ridge) {

   sum = 0
   for (k in 1:10) {

      test_indices = seq(floor(seq(1,nrow(d_shuffle),length.out = 11))[k],
                                 floor(seq(1,nrow(d_shuffle),length.out = 11))[k+1],1)

      test_x = as.matrix(d_shuffle[test_indices,2:ncol(d_shuffle)])
      train_x = as.matrix(d_shuffle[-test_indices,2:ncol(d_shuffle)])

      test_y = d_shuffle[,1][test_indices]
      train_y = d_shuffle[,1][-test_indices]

      if (ridge) {
         fit = glmnet(train_x,train_y,alpha=0,lambda=lambda)
         sum = sum + sum((predict(fit,test_x,s=lambda) - test_y)^2)
      } else {
         fit = glmnet(train_x,train_y,alpha=1,lambda=lambda)
         sum = sum + sum((predict(fit,test_x,s=lambda) - test_y)^2)
      }
   }
   return(sum)
}

ridge_lambda = optimize(cv,c(0,1),ridge=T)$minimum #Optimal lambda for Ridge
lasso_lambda = optimize(cv,c(0,1),ridge=F)$minimum #Optimal lambda for Ridge
############################################################

# divide data to training data and test data randomly
set.seed (1)
train=sample (c(TRUE ,FALSE), nrow(d),rep=TRUE)
test =(! train )

#Ridge regression
lambda = ridge_lambda
fit_r = glmnet(as.matrix(d[train,-1]), as.matrix(d[train,1]),alpha=0,lambda=lambda)
ridge_mse = mean((predict(fit_r,as.matrix(d[-train,-1]),s=lambda) - d[-train,1])^2)

#Lasso regression
lambda = lasso_lambda
fit_l = glmnet(as.matrix(d[train,-1]), as.matrix(d[train,1]),alpha=1,lambda=lambda)
```

```
lasso_mse = mean((predict(fit_l,as.matrix(d[-train,-1]),s=lambda) - d[-train,1])^2)

stargazer(lm2, fit_r)
#################################################################

#clustering(unsurpervise learning)

#drop the observations with NA
ebaycar <- ebay %>%
  drop_na()
ebaycar$id <- 1:nrow(ebaycar)

# use car characterize to cluster
ebaycar <- ebaycar %>%
  select(inspection,miles,bookvalue,doors,age,biddy2,id)
ebaycar.cluster<- ebaycar[, 1:5]
ebay.cluster <- kmeans(ebaycar.cluster, centers = 5, iter.max = 20, nstart = 25)

#see variable we try to predict situation after we cluster
table(ebay.cluster$cluster, ebaycar$biddy2)

#aggregate cluster with original dataset
ebay.cluster$cluster <- as.factor(ebay.cluster$cluster)
aggregate(ebaycar, by = list(ebay.cluster$cluster),FUN = mean)

#repeat the process that using car characterize to cluster with different car variables
ebay.cluster2 <- ebaycar[,3]
ebay.cluster2 <- kmeans(ebay.cluster2, centers = 5, iter.max = 20, nstart = 25)
table(ebay.cluster2$cluster, ebaycar$biddy2)
ebay.cluster2$cluster <- as.factor(ebay.cluster2$cluster)
aggregate(ebaycar, by = list(ebay.cluster2$cluster),FUN = mean)
ebaycar <- data.frame(ebaycar, ebay.cluster2$cluster)

# merge cluster id to ebay dataset and sort it
ebaygroup <- ebaycar %>%
  select(id, ebay.cluster2.cluster)
ebaytopart <-
  left_join(ebay,ebaygroup, by = "id" )
sortebaycar <- ebaycar[order(ebaytopart$ebay.cluster2.cluster),]

ebaytopart1 <- subset(sortebaycar,sortebaycar$ebay.cluster2.cluster == 1)
ebaytopart2 <- subset(sortebaycar,sortebaycar$ebay.cluster2.cluster == 2)
ebaytopart3 <- subset(sortebaycar,sortebaycar$ebay.cluster2.cluster == 3)
ebaytopart4 <- subset(sortebaycar,sortebaycar$ebay.cluster2.cluster == 4)
```

```
ebaytopart5 <- subset(sortebaycar,sortebaycar$ebay.cluster2.cluster == 5)
ebaytopart <- ebaytopart %>%
  drop_na()

#use for loop to do subset selection for each cluster sub group
for( i in levels(ebaytopart$ebay.cluster2.cluster)){
  subdata = subset(ebaytopart, ebaytopart$ebay.cluster2.cluster == i)

  # divide data to training data and test data randomly
  set.seed(2)
  train = sample(c(TRUE, FALSE), nrow(subdata), replace = TRUE)
  test = (!train)

  #perform backward stepwise selection to select best fit variables
  fit.best.clu = regsubsets(biddy2 ~ text + inspection + featured + numbids + phone +
pwrseller + photos + season1 + season2 + season3 + season4 + addedinfo +
descriptionsize +sellerage + webpage,data = subdata[train, ], method = "backward",
nvmax = 10)
  summary(fit.best.clu)
  #make a model matrix from the test data
  test.mat = model.matrix(biddy2  ~ text + inspection + featured + numbids + phone
+ pwrseller + photos + season1 + season2 + season3 + season4 + addedinfo +
descriptionsize +sellerage + webpage, data = subdata[test,])

  #we extract the coefficients from the regfit.best and use them to form the predictions
and compute test MSE
  val.error = rep(NA, 10)
  for(j in 1:10){
    coef = coef(fit.best.clu, id = j)
    pred = test.mat[,names(coef)]%*% coef
    val.error[j] = mean((subdata$biddy2[test] - pred) ^2)
  }

  # use smallest MSE to pick fittest model variables
  print(which.min(val.error))
  m = which.min(val.error)
  summary(fit.best.clu)
  print(coef(fit.best.clu, m))
  stargazer(coef(fit.best.clu,m))
}

# do regression in subgroup with different best fit model variables we get seperately and
compare each mse
```

```
#subgroup1
lm_c1 <- lm(biddy2 ~   photos, data = ebaytopart[ebaytopart$ebay.cluster2.cluster == 1,])
print(summary(lm_c1))
mse1 <- mse(lm_c1,   ebaytopart[ebaytopart$ebay.cluster2.cluster == 1,] )

#subgroup2
lm_c2 <- lm(biddy2 ~ photos + season1,
            data = ebaytopart[ebaytopart$ebay.cluster2.cluster == 2,])
mse2 <- mse(lm_c2,ebaytopart[ebaytopart$ebay.cluster2.cluster == 2,])
print(summary(lm_c2))

#subgroup3
lm_c3 <- lm(biddy2 ~ text + descriptionsize + inspection + featured + photos
            , data = ebaytopart[ebaytopart$ebay.cluster2.cluster == 3,])
mse3 <-mse(lm_c3,ebaytopart[ebaytopart$ebay.cluster2.cluster == 3,] )
print(summary(lm_c3))

#subgroup4
lm_c4 <- lm(biddy2 ~   featured + webpageauction123 + webpageebayhosting + inspection + phone + photos,
            data = ebaytopart[ebaytopart$ebay.cluster2.cluster == 4,])
print(summary(lm_c4))
mse4 <- mse(lm_c4,ebaytopart[ebaytopart$ebay.cluster2.cluster == 4,])

#subgroup5
lm_c5 <- lm(biddy2 ~ season1 + featured + numbids + phone + photos + inspection +
            descriptionsize + webpageauction123 + webpagecarad + webpageebayhosting
            , data = ebaytopart[ebaytopart$ebay.cluster2.cluster == 5,])
print(summary(lm_c5))
mse5 <- mse(lm_c5,ebaytopart[ebaytopart$ebay.cluster2.cluster == 5,])

stargazer(lm_c1, lm_c2, lm_c3, lm_c4, lm_c5)

stargazer(mse1, mse2,mse3,mse4,mse5)

#############################################################################
#########################

#random forest

set.seed(101)
dim(ebay)
```

```
ebay <- ebay %>%
  drop_na()
train = sample(1:nrow(ebay),9000)

#bagging part(predict with all variables)
ebay_forset1 <- randomForest(biddy2 ~ text + inspection + featured + numbids + phone
+ pwrseller + photos + season1 + season2 + season3 + season4 + addedinfo +
descriptionsize +sellerage  + bookvalue, data = ebay, subset = train)

#growing random dorest proceeds
ebay_forset2 <- randomForest(biddy2 ~ text + inspection + featured + numbids + phone
+ pwrseller + photos  + season1 + season2 + season3 + season4 + addedinfo +
descriptionsize +sellerage + bookvalue, data = ebay,mtry = 7, subset = train)

#plot tree and error relation
plot(ebay_forset1, log="y")
plot(ebay_forset2, log = "y")

#view importance of variables
importance <- importance(ebay_forset1)
View(ebay_forset1$importance)

# do predict using random forest with test data
pred_ebay <- predict(ebay_forset1, newdata = ebay[-train,])
pred_ebay2 <- predict(ebay_forset2, newdata = ebay[-train,])

# visualization the importance
varImportance <- data.frame(variables = row.names(importance), Importance =
round(importance[,], 2))
rankImportance <- varImportance %>%
  mutate(Rank = paste0('#', dense_rank(desc(Importance))))
ggplot(rankImportance, aes(x= reorder(variables,Importance),
                           y = Importance, fill = Importance))+ geom_bar(stat =
'identity')+
  geom_text(aes(x = variables, y = 0.5, label = Rank),
            hjust = 0, vjust = 0.55, size = 4, colour = 'red')+
  labs(x = 'Variables') + coord_flip()

ebay.test <- ebay[-train,]
table(pred_ebay, ebay.test$biddy2)

#compare mse
mean((ebay.test$biddy2- pred_ebay)^2)
mean((ebay.test$biddy2 - pred_ebay2)^2)
```

```
#####################################################################
#################

#pick up the most commonly observed car model in the data
for(i in names(ebay)[99:115]){
   print(length(which(ebay[,i] == 1)))
}

ebayMustang <- subset(ebay, modelMustang == 1)

#do regression with other characterize with it to see what fit better
lmmodel <- lm(biddy2 ~   inspection + featured + numbids + phone + photos + miles
                      + length + webpageebizautos + dealer, data = ebayMustang)
summary(lmmodel)
stargazer(lmmodel)
```