UNIVERSITY OF
GEORGIA

Online Marketplace Price Prediction

ECON 7720: Machine Learning

Professor Thurk

Members

Jake Beinart
Sam Lodinger
Daniel Saul

Table of Contents

I. Executive Summary

Purpose: The purpose of this report is to dive into the data provided by a seller, create an accurate model using machine learning methodologies, and predict the price of a specific product in an online marketplace where buyers and sellers can come together to trade goods and services.

Problem: Some of the most popular online marketplaces include Amazon, eBay, Walmart, and Facebook Marketplace. Due to long, similar product descriptions, and many other factors that affect pricing strategy, it is quite difficult to correctly gauge and predict the price for certain products. For instance, a seller provides a listing of Jacket A and B on an online marketplace. The first thing that a buyer typically sees is the name of the product when searching online. The name of Jacket A is "Nike Dri-Fit, Long-Sleeve Jacket, Men's Large, Brand New." The name of Jacket B is "Long-Sleeve Grey Pullover, Adidas, Womens, Size Small, New." How do we determine the correct price of each item?

Solution: Machine learning is a way of making computers better at solving problems by giving them access to large amounts of data and letting them learn from it on their own. This allows them to find patterns and make predictions or decisions without being explicitly programmed or told how to do so. To predict future sales for this marketplace, the model is trained on historical data such as the price and performance of the products, the seller's reputation, condition the product is in, the demand for the specific products, and other determination features. That trained model is then applied to a testing data set to guarantee accurate and valid results. However, we do not make just one machine learning model; we create multiple different models and test them against each other to make more accurate predictions. We can tune and alter the models by varying the number of features, hyperparameters, and preprocessing data. The types of machine learning models we utilize are linear regression, elastic net, neural networks, and lasso. We selected the lasso model as our final model to make predictions.

Validation: Ultimately, these machine learning models work because given a wide variety of model types, model tuning methods, and a large amount of training data, the models refine their own predicting capabilities over time, making themselves more accurate. The methods used are well known by data science experts and optimized to avoid overfitting or underfitting of the data, meaning our accuracy is not exactly perfect but not the worst at predicting test data. The diverse and large amount of training data we use increases validity as well. With regards to further notes and recommendations, this information can be used by the marketplace to optimize pricing, target the right customers, identify potential trends, and improve their chances of making successful sales.

II. Data Description

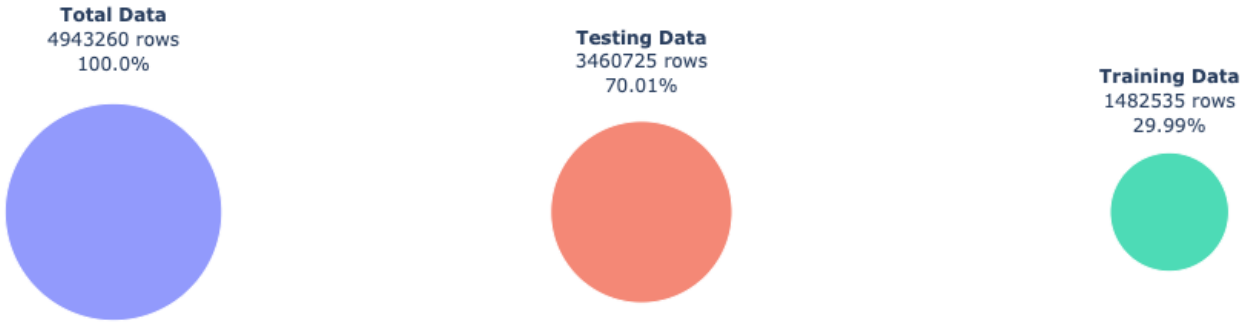


Figure 1. This figure shows the total amount of observations used in this project and how they were split into training and testing sets.

Figure 1 displays the number of nuts and bolts for the tools we are using to predict the sales price of online items. We have approximately 5 million rows, observations, or data points, all meaning the same, of sales data which were randomly distributed into two separate datasets: one to be used for training our machine learning model to make accurate predictions and the other to be used for testing the model we derive. Approximately 70% of the total data, 3.5 million rows, will be allocated for testing whereas 30%, 1.5 million rows, will be used for training. With regards to null or missing values, there are 632,682 rows with no brand name and 6,327 rows with no category name. These null values may pose a threat to model accuracy, since branding and categorizing products is a large price driver. The following visualizations are derived from the training data.

Variable Name	Type	Description	Notes
train_id	int	The listing identifier in training data.	
test_id	int	The listing identifier in testing data.	
name	str	The listing title.	Text that looks like prices (e.g. $20) is replaced with [rm] to reduce leakage.
item_condition_id	int	The condition of the items provided by the seller.	Values are 1, 2 ,3, 4, and 5.
category_name	str	The category of the	

		listing brand_name.	
price (Target Variable)	float	The price that the item was sold for (USD).	This column doesn't exist in the test dataset.
shipping	int	Shipping fee is paid by: Seller (1) and Buyer (0).	
item_description	str	The item description.	Text that looks like prices (e.g. $20) is replaced with [rm] to reduce leakage.

Table 1. This table shows a data dictionary describing the variables in our training data set.

Table 1 acts as our data dictionary; where we can go to gain a better understanding of the columns, variables, or features, also all meaning the same, in our testing and training datasets. This table includes only the base variables from the raw data; additional variables we created can be found in modeling sections and code. From left to right, the variable name is what is shown in Python when creating visualizations and models, the type demonstrates whether a variable is numerical or text, the description gives more insight into the definition of the variable in this context, and the notes are general disclaimers or small things to keep in mind when conducting analysis with these variables. Lastly, with regards to columns, the only differences between the training and testing sets is that the training set includes the `price` and `train_id` variable and the testing set includes the `test_id` but not `price` variable since our goal to is to make our own prediction of sales price.

Table 2. This table shows the descriptive statistics of the price, item condition, and shipping variables from the training data.

Table 2 provides the most basic analysis and statistics of the quantitative variables, price, item condition, and shipping in the training dataset. With regards to the statistics columns, the count is just the total number of instances in our dataset, the mean is the average number of each variable, the `STD` is the standard deviation which tells us how spread out the data is relative to the mean, the minimum and maximum shows the beginning and end of the range of a certain variable, the percentages relate to the percentile of where most of the values in the variables fall. We can see that

	Price	Item Condition	Shipping
Count	1482535.00	1482535.00	1482535.00
Mean	26.74	1.91	0.45
STD	38.59	0.90	0.50
Minimum	0.00	1.00	0.00
25%	10.00	1.00	0.00
50%	17.00	2.00	0.00
75%	29.00	3.00	1.00
Maximum	2009.00	5.00	1.00

the average price is $26.74 with the maximum being $2009.00, meaning that about 75% of the prices we have are relatively low, below $29, but skewness of the data may be present due to the large outliers in price. The average item condition seems to be around level 2 with about 75% of item conditions being 3 or lower. Lastly, we can see that most shipping fees are paid by the seller.

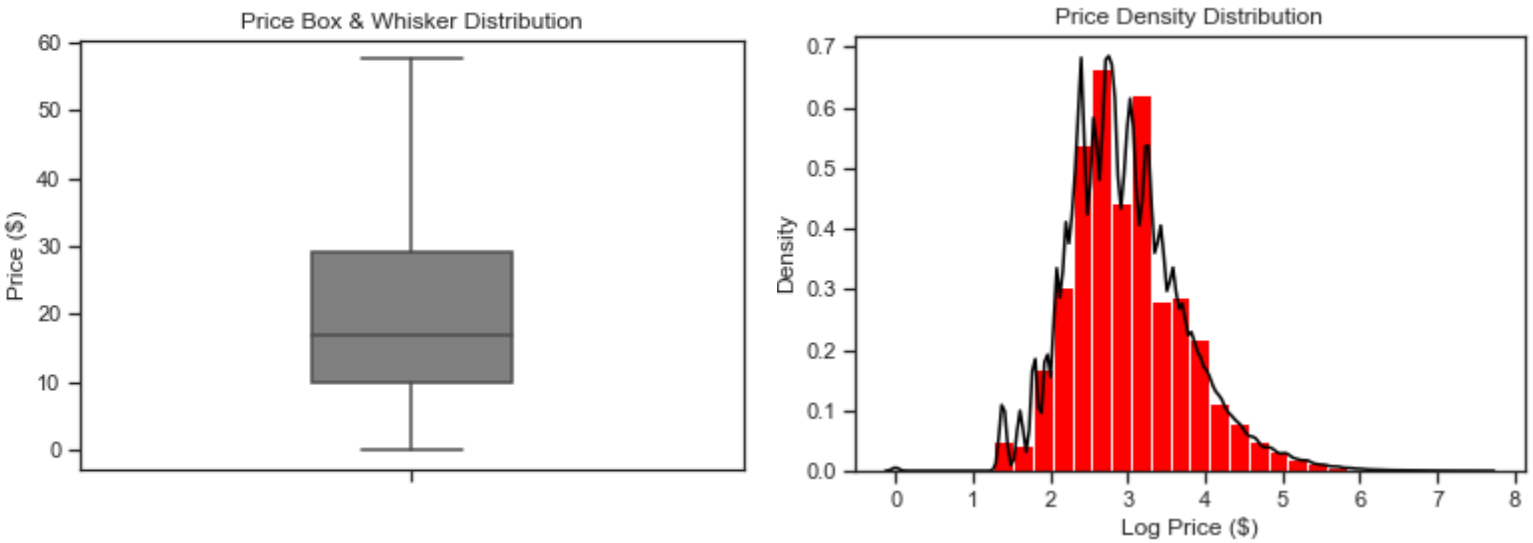


Figure 2. These visuals display a box and whisker plot without outliers for price on the left and a density plot using log price on the right.

Figure 2 takes a more in-depth look at the `price` variable with respect to Table 2, but without strong price outliers in the box and whisker plot that do not make up much of the price distribution and which may affect our model analysis. Assuming the removal of outliers, we can see that the minimum price is $0, maximum is slightly below $60, median price is approximately $18, 75% of prices fall below $30, 25% are above $30, and about 50% of prices fall between $30 and $10. Additionally, we attempt to scale down the price value by applying a logarithmic function to price in the right plot. This allows us to visually understand where prices are distributed density-wise for all listings on an even playing ground. If we did not do this, you would not be able to see price distribution effectively; there is a large amount of skewness towards outliers without scaling down price. The black line illustrates a density plot line for a clearer analysis of the pattern and shape of price. Furthermore, this aids our decision making on whether to utilize log prices or potentially an average price by brands in our model to address skewed data.

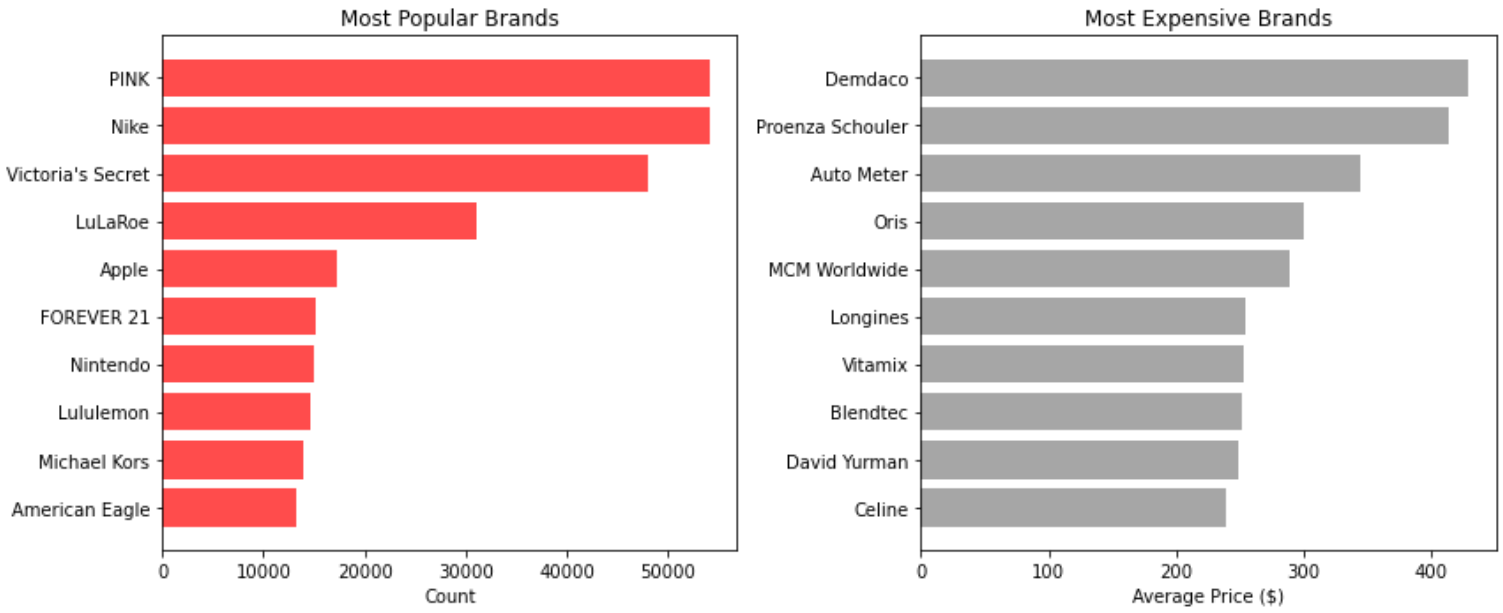
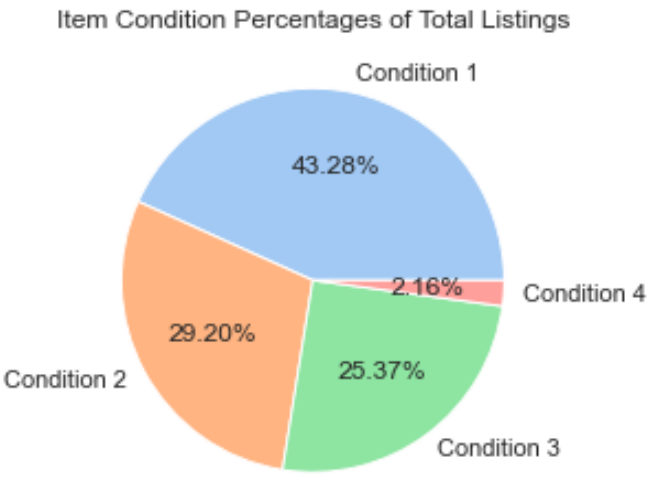


Figure 3. This visual shows the top ten most popular and most expensive brands in our training data set.

Figure 3 investigates the 4,809 unique brands sold in the online marketplace; first, by the top ten highest amount sold, followed by the top ten brands holding the highest average prices on the right. The chart on the left excludes rows with null or missing brand names (632,682 null values out of 1,482,535 total values) to increase our model accuracy since predicting the value of price relies on understanding brand pricing trends and ranges such as vintage or designer. We notice that the most popular brands seem to fall into the category of clothing, technology, and game companies while the most expensive brands, on average, are expensive accessories and clothing, gifts, pressure gauges, and small kitchen appliances. About the price range, the top ten highest average prices move from about $220 to slightly over $400, with the highest listed by luxury bag companies. The average price for missing brand name items is $21.13, quite outside of the top ten. Overall, we can infer that the variance in pricing ranges for different brands are important for our model to understand price matching and calculations.

Figure 4. This chart shows who is responsible for paying the shipping fee and the average price associated with each group.

Figure 4 reveals the number of listings (no missing values) where the buyer or seller pays the shipping fee and the average price on the listing of each designated group. In approximately 650,000 cases in the dataset, the seller pays the shipping fee with an associated average price of $22.57. On the other hand, slightly more than 800,000 cases have the buyer paying the fee with an average price of $30.11. Typically, total sale prices are lower when the seller pays the fee, being a better deal for the consumer. If the buyer pays the fee, the total sale price will be higher since shipping costs must be accounted for. Empirically, many companies require that the buyer pays for shipping because it reduces company expenses, but it reduces demand because buyers will go where the prices are the lowest. We would expect the frequency of the seller paying the shipping fee to be higher in this instance because the average sale price is lower, but it is not, which is an important discrepancy to keep in mind during our model testing.



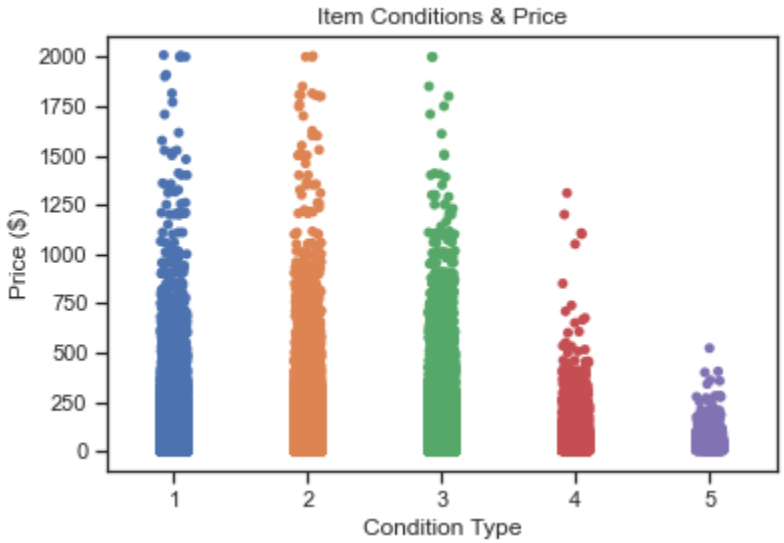


Figure 5. These visuals display the item condition percentage count for all data on the left and the item conditions with the price levels that fall into each category on the right.

Figure 5 covers the item condition variable and price spread comparison. We have assumed that the interpretation of the conditions is that 1 is the best, new condition and 5 means oldest, worst condition. In the left plot, we can see that the percent of listings in condition 1 are the highest, decreasing almost linearly to condition 4. Condition 5 was removed for visual clarity, and it was a very low percentage. The plot on the right displays the spread of prices for each condition type, with no missing values to cause accuracy problems. Each plot demonstrates that the data contains items with better condition types that have higher prices due to being better quality-wise. This variation is likely an indicator in our predictions that the larger condition type, the higher the prices will be.

Tier 1	Count	Percent	Avg. Price
Women	664385	45.01%	$28.89
Beauty	207828	14.08%	$19.67
Kids	171689	11.63%	$20.64
Electronics	122690	8.31%	$35.17
Men	93680	6.35%	$34.71
Home	67871	4.6%	$24.54
Vintage & Collectibles	46530	3.15%	$27.34
Other	45351	3.07%	$20.81
Handmade	30842	2.09%	$18.16
Sports & Outdoors	25342	1.72%	$25.53

Tier 2	Count	Percent	Avg. Price
Athletic Apparel	134383	9.06%	$28.46
Makeup	124624	8.41%	$18.69
Tops & Blouses	106960	7.21%	$18.24
Shoes	100452	6.78%	$41.81
Jewelry	61763	4.17%	$27.5
Toys	58158	3.92%	$21.52
Cell Phones & Accessories	53290	3.59%	$30.14
Women's Handbags	45862	3.09%	$58.2
Dresses	45758	3.09%	$29.45
Women's Accessories	42350	2.86%	$30.93

Tier 3	Count	Percent	Avg. Price
Pants, Tights, Leggings	60177	4.06%	$34.39
Other	50224	3.39%	$23.69
Face	50171	3.38%	$19.74
T-Shirts	46380	3.13%	$19.39
Shoes	32168	2.17%	$24.79
Games	30906	2.08%	$22.65
Lips	30871	2.08%	$18.35
Athletic	27059	1.83%	$55.71
Eyes	26038	1.76%	$15.03
Cases, Covers & Skins	24676	1.66%	$13.17

Table 3. This table shows the top 10 categories and sub-categories split into tiers from the `category_name` variable with respective average prices and percent of totals.

Table 3 takes the `category_name` column with 1,287 unique categories and splits it into three tiers based on specificity of the product. There were a low number of missing values that should not cause an issue. As the broadest tier (10 tiers), Tier 1 it classifies listings by gender, electronics, athletics, home goods and collectibles, and miscellaneous items. Listings related to women is the clear front runner for most products listed (accounts for 45%), and prices vary as expected depending on complexity and luxury. For example, electronics with an average price of $35.17 should be more expensive than kids' items at $20.64. The top ten categories (out of 122 unique categories) in Tier 2 have a strong focus on clothing and body accessories–athletic apparel accounting for the most at 9.06%. In Tier 3 (out of 866 unique categories), the most popular items are clothing and beauty products, a clear pattern in each tier.

III. Model Description

Our preferred model will utilize sentiment analysis and similar products to estimate prices. Sentiment analysis analyzes text data and assigns it a numerical value based on the connotation of the key words and phrases. Figure 7 provides an insight into the fundamental components of sentiment analysis. To properly conduct this analysis, the methods tokenization, stop word filtering, negation handling, stemming, and classification are ways we preprocessed the listing text data provided in the files. Words and phrases with a positive connotation will receive a positive numerical value and words and phrases with a negative connotation will receive a negative numerical value. When more of the words and phrases have a positive connotation, the magnitude of the numerical value assigned to this description will be larger. This will give us a numerical representation of the text data that we received. With this numerical data, we will essentially be able to include the text data to our machine learning model.

Figure 6. This visual shows the components of sentiment analysis.

We also want our model to use similar products to estimate prices. Since there are many brands, categories, and names of products, and conditions we want to be able to group these products together based on these attributes. If many of these attributes are similar, there will be a good comparison to the product which we are trying to price. With this comparable product we will be able to see what previous items similar to this item sold for and estimate a fair price for the item. To accomplish this task, we used the columns of brand name, category name, and name. We grouped the items together by the category and then took the mean of the prices in this group. We repeated this process for the brand name column, the name of product column, and the condition column. We then added these columns to our model to account for this data.

Figure 7 below shows the different accuracies for multiple types of model methods. Most of the models performed relatively similarly with accuracies around 44%. The only model method that performed far worse than all the other methods is the neural network method. This did not narrow our decision down very much, but since many of the models performed very similarly, we decided to prioritize model simplification.

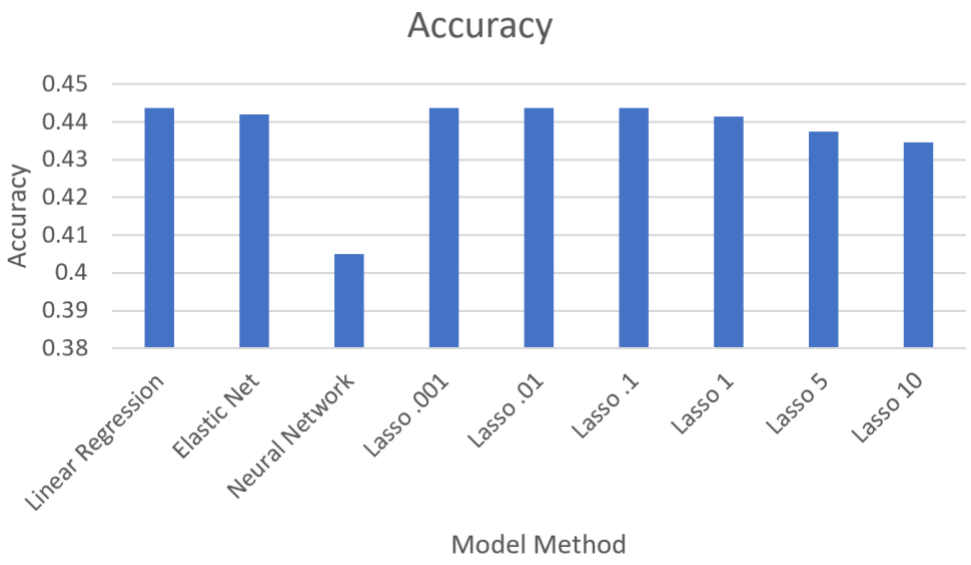


Figure 7. These visuals display the accuracies of different models we tested.

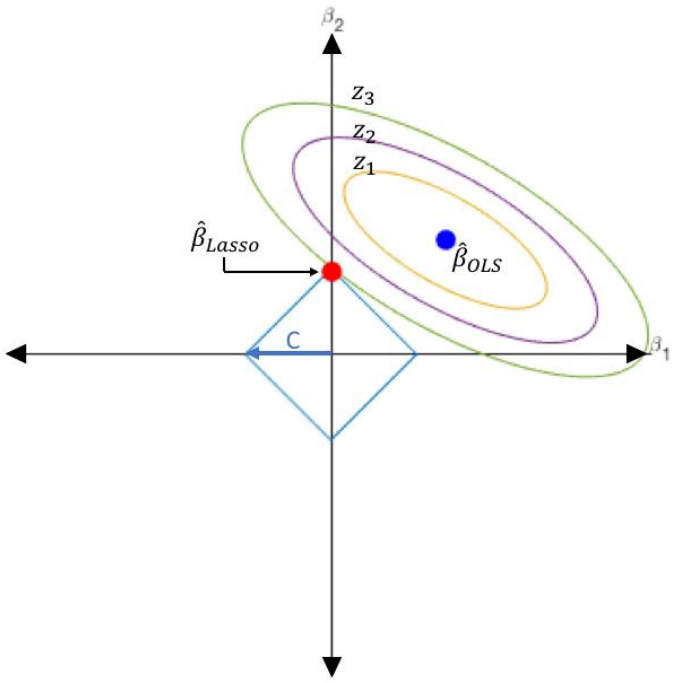


Figure 8. This picture shows the mathematics of a Lasso regression model.

With model simplification as a priority, we decided to make our model a lasso model. Lasso stands for least absolute shrinkage and selection operator. Lasso is a regression model that

punishes features that are not as important to predicting the price of the item by either decreasing the feature's effect on estimating the price or removing the variable from the model entirely. The red point is the lasso coefficients, the blue square is the penalty term shown as a constraint region, the blue point is the RSS (least square) coefficient, and the rings display the contours of RSS as it moves away from the minimum.

We chose this model because it helps us simplify our model without significantly reducing the accuracy of our model. Even though we will be sacrificing some accuracy, the decrease in the number of variables in the model will save us time and money in the future. With less variables in the model, we will be able to estimate prices on large amounts of items without the amount of processing power we would need for more complicated models. This will reduce our workload on our hardware, so the model will run quicker, and we will not need any additional investment into our technology.

IV. Results

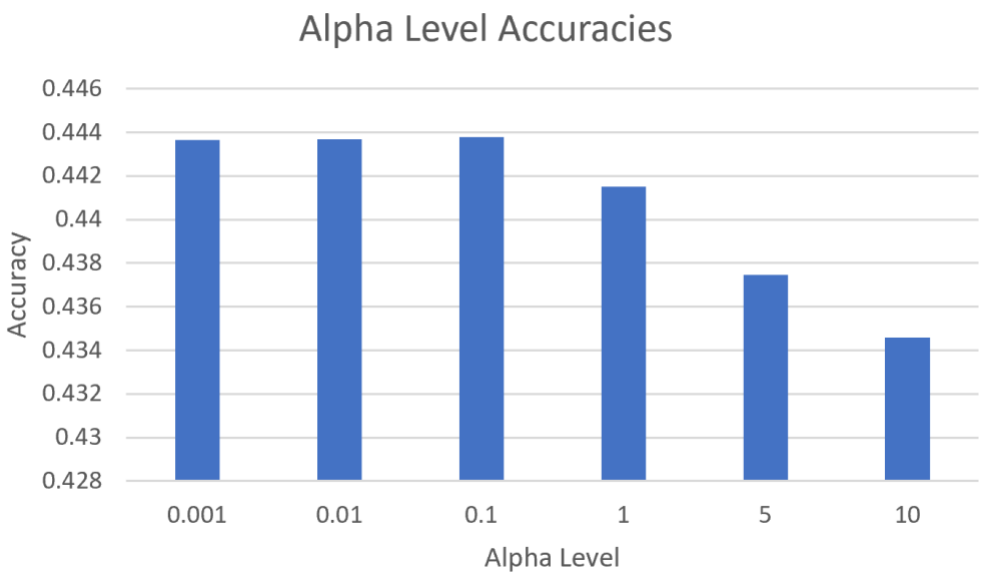


Figure 9. This figure shows the different levels of accuracy for the lasso models we used.

With a lasso model, there is one main function that you can change. This function is called the alpha level. The alpha level decides how strict the punishment should be on variables. The higher that the alpha level is set, the more severe the punishment will be for less effective variables. As the alpha level increases, less of the variables are used in the model. This also means that the higher that the alpha level is, the model will be simpler. This leads to a tradeoff between model simplification and model accuracy. Unfortunately, there is no way to solve for the perfect alpha level, so we tried some alpha levels of different magnitudes and calculated their accuracy to find the ideal tradeoff between model simplification and accuracy. With the results shown on this graph, we decided to use an alpha level of 1 as it still simplifies the model but does not sacrifice as much accuracy as the higher alpha levels do.

	Variable	Lasso .001	Lasso .01	Lasso .1	Lasso 1	Lasso 5	Lasso 10
0	item_condition_id	0.000000	-0.000000	-0.000000	-0.000000	-0.000000	-0.000000
1	shipping	-4.283835	-4.242149	-3.825283	-0.000000	-0.000000	-0.000000
2	description_polarity	1.217714	1.161511	0.599465	0.000000	0.000000	0.000000
3	brand_mean	0.666931	0.666904	0.666627	0.664526	0.661039	0.656838
4	condition_mean	2.166516	2.161505	2.111393	1.700120	0.469712	0.000000
5	cat_mean	0.331008	0.331030	0.331253	0.330590	0.297407	0.275677
6	name_mean	0.708916	0.708932	0.709096	0.711331	0.713733	0.712699

Table 4. This table shows the variable coefficients for each lasso model tested.

Table 4 shows the difference in the coefficients of the variables for each alpha level. This table illustrates that as we increase the alpha level more of the variables are removed from the model. With the lowest alpha level, all six variables are included in the model. When we increase the alpha level to 1, we only use four of the variables. Increasing the alpha level to 10 only keeps three of the variables in the model.

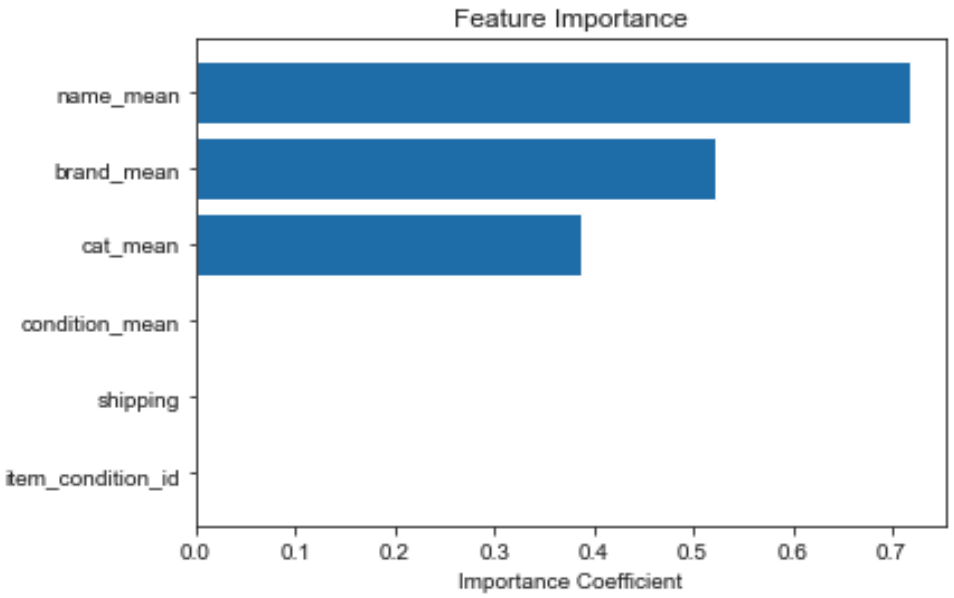


Figure 10. This chart shows the importance of variables used in our model.

These models show that the most important variables to predicting price are the three mean variables that we created. These features are the most important because using past sales of similar products and substitutes gives us a very good idea of a fair price for the item we are trying to price. Due to these conditions, this model is best utilized when trying to price an item that is commonly sold on the platform. With more unique items, there may not be many comparisons and our model may not be as accurate. The other three variables relating to item condition mean, shipping, and item condition ID do not have much to no importance in our lasso models.