Laptop Price Modeling

Timothy Lu
Springboard Capstone 2022

TABLE OF CONTENTS

Problem Statement

The increasing complexity and power of laptops makes their pricing challenging. Without appropriate pricing, companies can make potential customers shy away or lose out on revenue. Understanding consumer desires and what factors may have the greatest impact on their purchasing decisions will allow a company to better price their products and learn what components to focus on. This knowledge is vital so that companies do not waste resources pursuing functionality consumers do not find useful or miss out on great pricing strategy.

I hope to create a machine learning model that can group laptops by their specifications and their price. As we create new laptops, we can then use these groups to help us place our laptop and the pricing that should go along with it.

Data Wrangling

The dataset used was scrapped by a user on Kaggle and contains approximately 900 rows and 23 columns. Keeping these rows intact and making sure that the data is usable is key to any good analysis. There are some columns that we may not find very useful in our analysis so some initial exploration will help us reduce these columns. There was also some missing data that I needed to look deeper into and will discuss in a later section.

Ultimately, I decided that the data should focus on the laptop specifications themselves so I dropped the ratings, star ratings, and review columns as these columns would not provide us with technical data. I also removed the columns for the old price and the discount amount as I am only focusing on the latest price. With that, we were able to begin exploring the data in earnest.

Exploratory Data Analysis

I started looking at features that may be useful for predicting pricing and the ways the different categories interact with each other. This helped me consider what other processing I wanted to do on the data and how I wanted to approach the modeling portion. First, I looked at the price and how pricing was distributed across all the laptops (**Figure 1**). I found the median was at around $825.42 with a few laptops reaching prices of over $4,000. I then followed that with a heatmap that helped look at whether there were any correlations for our numerical measurements with price in the data because that information would be the most helpful for deciding which features to prioritize during the process. Through that process, I found that there was some correlation between the RAM size, SSD size, and graphics card size (**Figure 2**) with price.

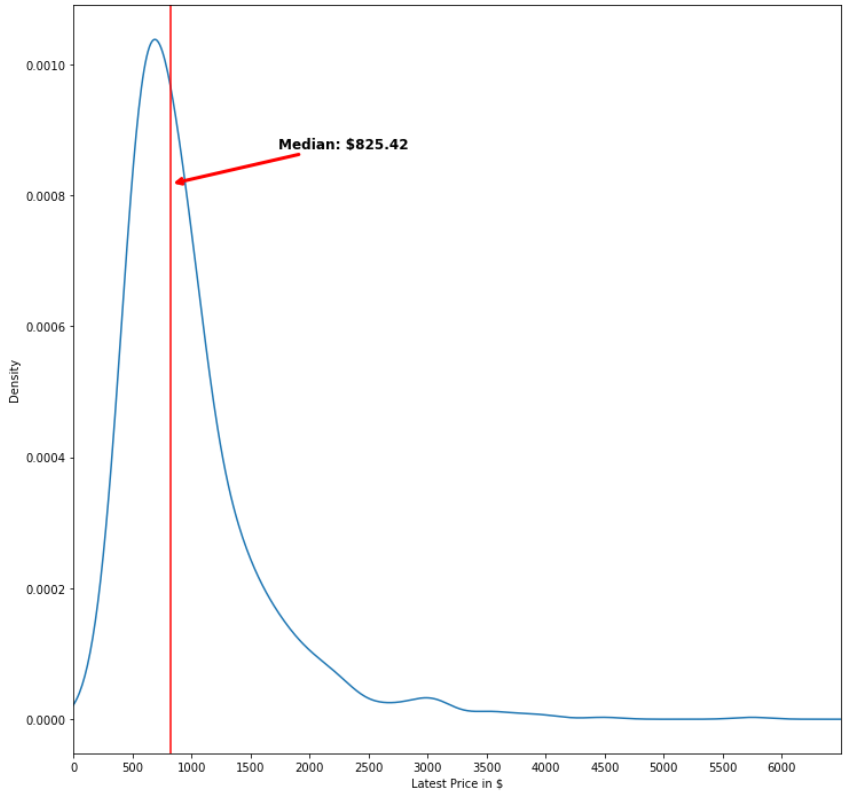

Figure 1. Median price of laptops

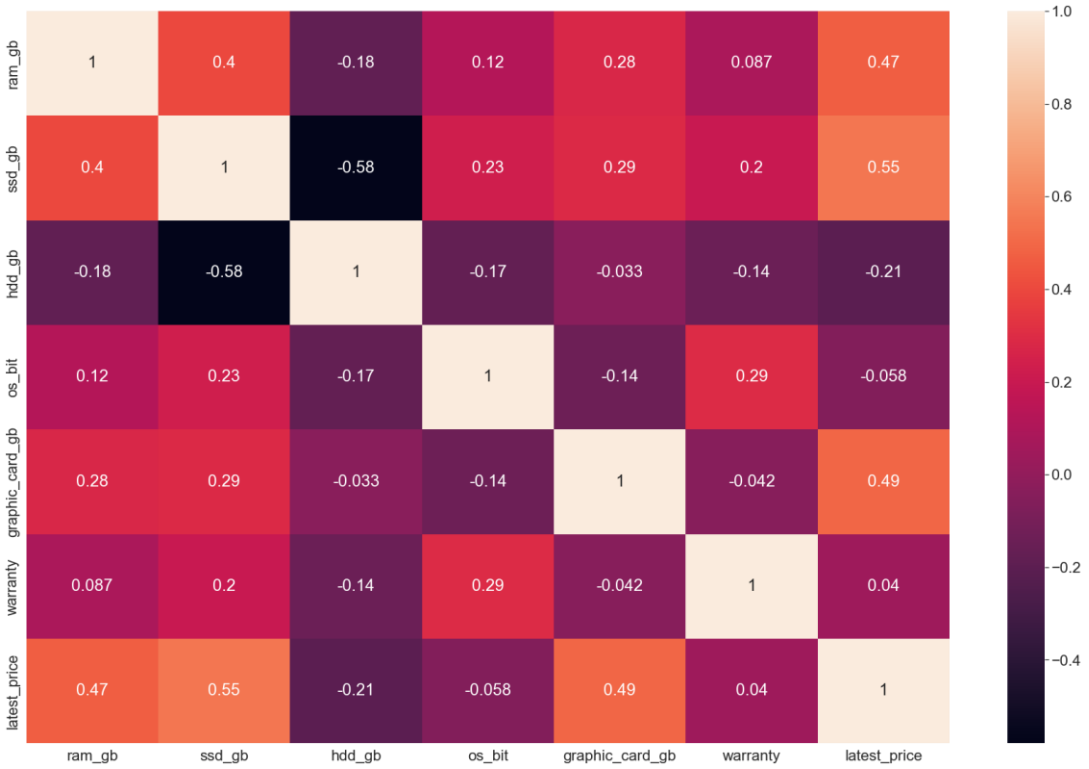

Figure 2 Heatmap of correlations for some features

Following these two analyses, I wanted to break down the information further and look at the categorical information that was collected. What I found was that for the most part, prices were distributed evenly across the board and not many features really stood out in terms of pricing. This led me to believe that it was a combination of features that led to certain prices for the laptops. The trend of laptops with larger storage space being more expensive was shown in the boxplots and was consistent with the positive correlation we saw in the heatmap. The most interesting boxplot was one made of the brands, and we saw a few brands really jump out. Namely, Alienware and Apple really stood out from other brands in terms of pricing (**Figure 3**) so knowing the branding of a laptop could really impact the pricing. The

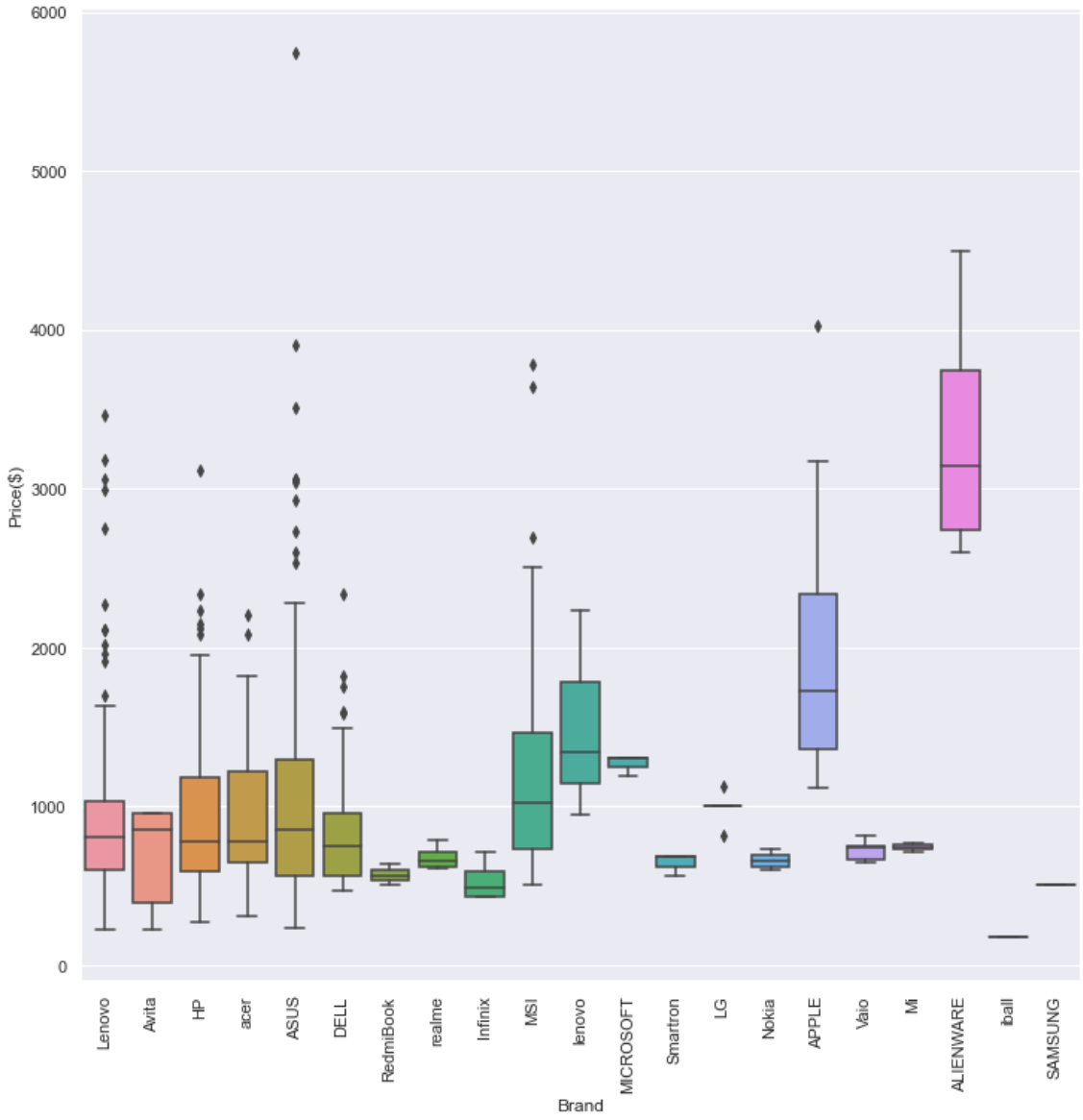

Figure 3 Brand vs Price Boxplot

In-Depth Processing

After finishing my data exploration, it was time to prepare the data for the modeling. The data needed some work before it was ready for machine learning models. I realized in my exploration that the model data may not be particularly useful as it parceled out the laptops into such small clusters and the fact that the model of the laptop did not seem to have a large impact on the pricing. I wanted to avoid the possibility of the model causing overfitting due to how distinct model information.

The other major thing was dealing with missing data. Within the data, it was not inputted a NaN and so pandas did not immediately recognize the missing data. It was typed in as a string and labeled as 'Missing'. After counting, I realized that a large portion of my data (almost 33%) was 'Missing' for the display size and processor generation information. To preserve the amount of data I had, I decided to impute the data instead of deleting those rows. I chose to use the mode for imputation as I believe this would be the most accurate. Utilizing the mean or median may be slightly askew as the numbers are not necessarily continuous measures but rather discrete categories of size or generation. I am hoping the mode is a more accurate representation of the laptops on the market than using the other methods.

Modeling

As part of the modeling process, I tested three different models: 1) K-Prototypes K-Modes clustering algorithm, 2) Isomapping, and 3) XGBoost Decision Trees. I felt that regression was not appropriate given the sparsity and categorical nature of the data. By categorizing and grouping the different laptops together I was hoping to glean information from certain pricing "groups". The goal would be to predict the appropriate price range for a laptop given its specifications. Throughout the model testing I made sure to try different parameters that fit within the model and to optimize the model as much as possible.

We first started with K-prototype which is a K-modes based algorithm because it can use categorical data without special encoding. This gave me some clusters to examine and work with. It works similarly to K-Means but instead of finding the distance between points it looks at the distance between numerical data points and similarities between categorical data points to create a centroid. Upon creation of the model, I found that it worked best when I was using 12 clusters and found some interesting results when looks at the SHAP values (**Figure 4)** to explain the clusters. We saw that graphic card, warranty, latest price, and weight were some of the top ways that the data was categorized. We adapted our code to fit into Anton Ruberts' analysis of K-prototypes and it was very helpful in the final visualization and understanding of the results of the K-prototypes clusters.

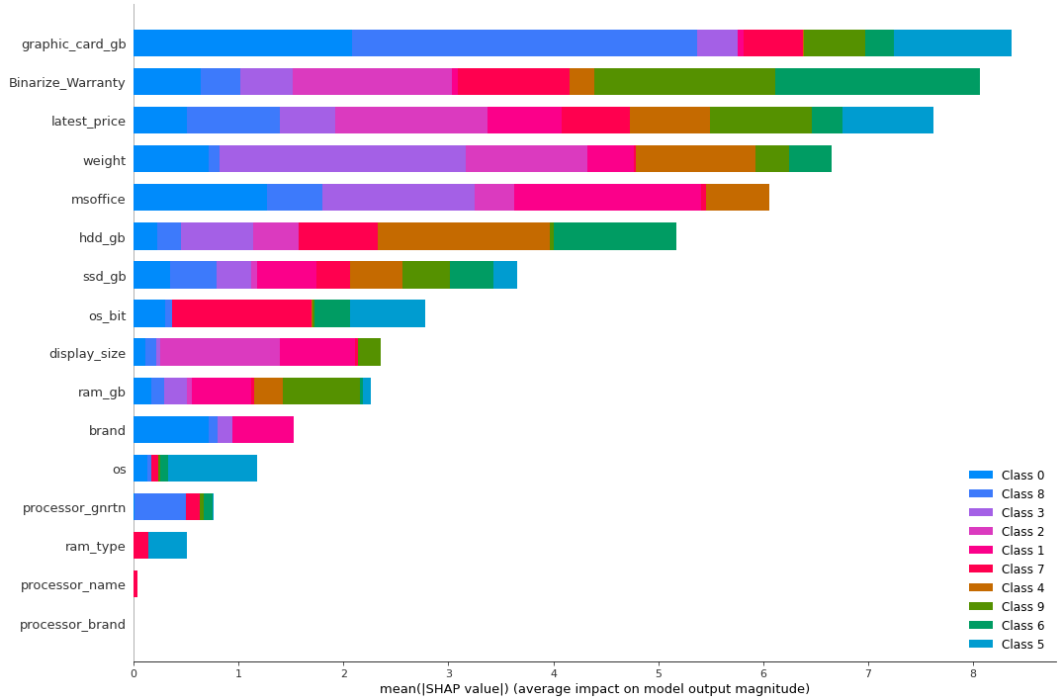

Figure 4 SHAP values for K-Prototype Clusters

The next model we looked at was Isomapping and ultimately it turned out to have challenging interpretability due to the large amount of overlap. Isomapping is a dimensionality reduction algorithm which projects the high-dimensional data into a 2-D or 3-D space depending on how many components we choose to have. I chose to examine both two and three components for Isomapping. Unfortunately, I was unable to really get any meaningful interpretations for the data using Isomap as the components were muddled and somewhat indecipherable (**Figure 5**).

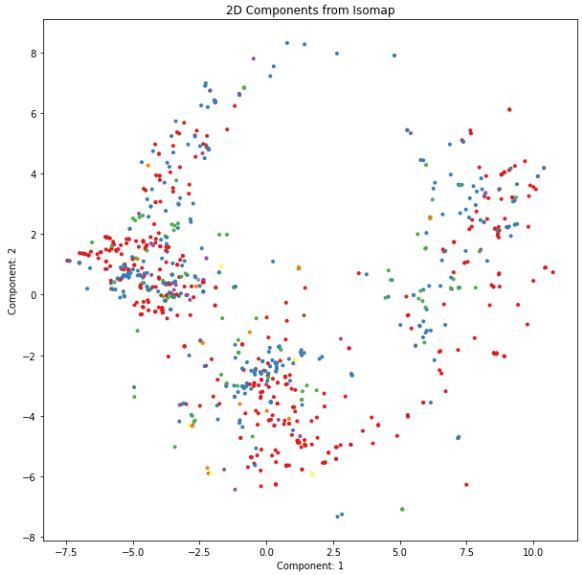

Figure 5 2-D Isomap for Laptop Modeling Data

Finally, I looked at XGBoost which is a boosting algorithm based on decision trees. It aims to create the best “decision tree” pathing for each outcome by minimizing errors from each previous iteration of the tree. I broke up the pricing here into distinct groups to allow the XGBoost Decision tree to find which components led to which pricing decisions. I trained the machine learning algorithm on approximately 67% of my data and use 33% as a training set. I then used Bayesian Optimization to optimize the model. It ended up performing at about a 65% accuracy. We got some interesting insights. Graphics card, display size, SSD size, and RAM size were the most important factors for the model (**Figure 6**).

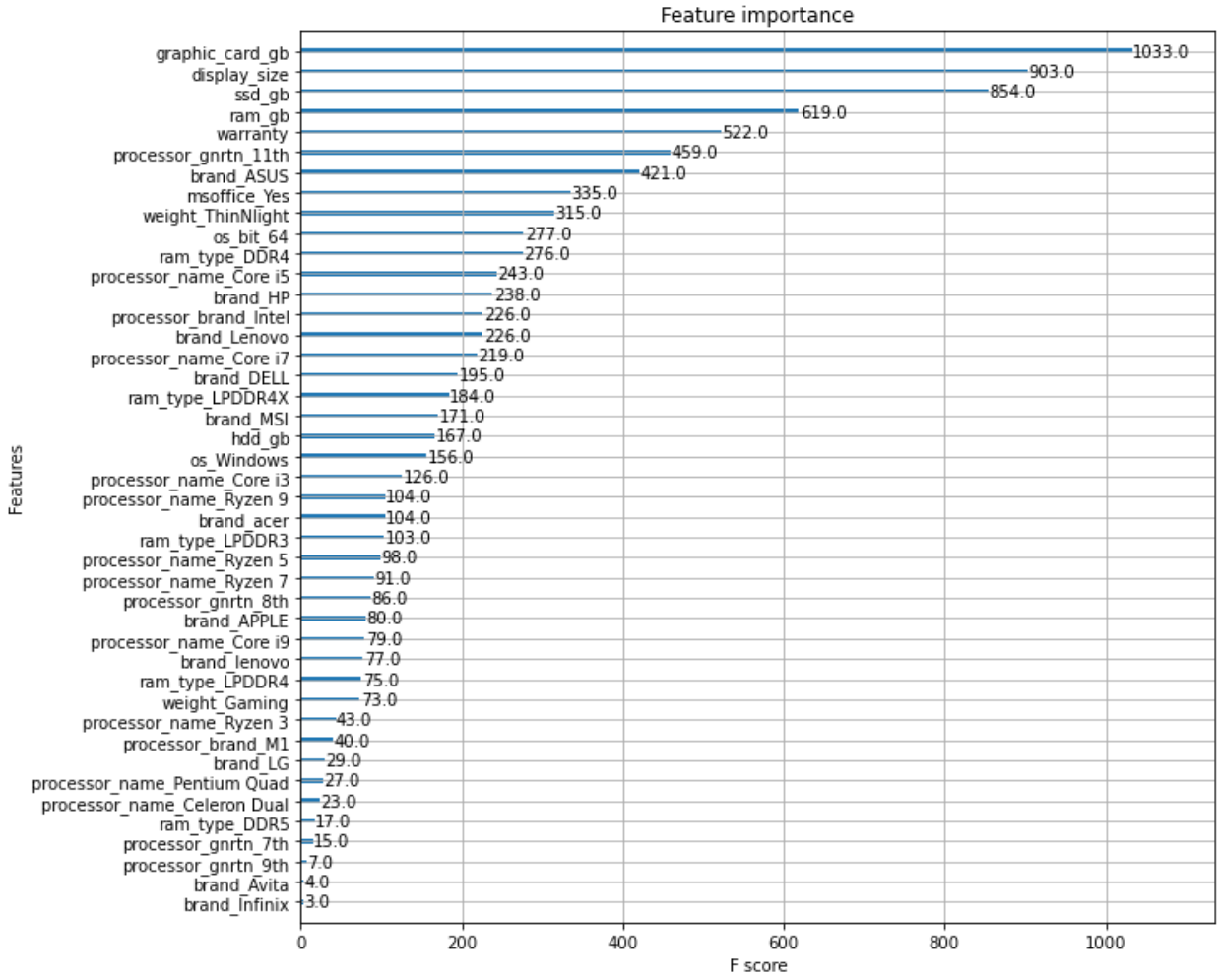

Figure 6 Most important features for XGBoost Decision Trees

Results and Discussion

In the end, I decided that a K-Prototype algorithm was the best for clustering my data. It gave me highly interpretable centroids that allows me to break down which components are grouped together to form a pricing group (**Figure 7**). I also think that using some of the information from the XGBoost model is useful because it gives a small breakdown of how a model decides on the pricing decision. In both the K-

prototype and XGBoost, graphics card size is a big component of how the models decided on the pricing group; I believe that companies should really focus on the graphics cards when it comes to pricing. It seems laptops with higher end graphics cards will be able to sell for a higher price. The K-prototype also found great value in the warranty and weight of laptops. We should consider laptops with warranties and their weight with their respective groups. It is challenging to imagine what consumer opinion is on warranty and weight. Based on domain knowledge, I would believe that laptops that come with a warranty are worth slightly more and those that are lighter are also worth more. Designing a lighter laptop will target a specific group of consumers and can be priced accordingly.

Additionally, the XGBoost model showed that display size and SSD size are good places to delineate the pricing. If it is a larger display, it will be worth more and the same goes for SSDs. The combination of a large display size and large SSD will lead to what consumers consider a higher end laptop. The type of RAM seems to not be a common point of delineation for the model which means that the price is spread more evenly across the type of RAM. That makes sense as a certain type of RAM (DDR4/5) will come in different sizes which as discussed will have a greater impact on the price of a laptop.

```
array([['0.18067226890756166', '0.15155228758170047',
        '0.051470588235294115', '0.022058823529411766',
        '0.6357170319108068', '0.14210969855231814', 'ASUS', 'Intel',
        'Core i5', '11th', 'DDR4', 'Windows', '64', 'Casual', 'No', '1'],
       ['0.21874999999999986', '0.17013888888888873', '0.015625',
        '0.09635416666666667', '0.5776143790849667',
        '0.1474437792056075', 'Lenovo', 'Intel', 'Core i5', '11th',
        'DDR4', 'Windows', '64', 'Casual', 'Yes', '1'],
       ['0.16512059369202217', '0.14069264069264054', '0.0',
        '0.003246753246753247', '0.4405398523045581',
        '0.11045363514989681', 'Avita', 'Intel', 'Core i5', '11th',
        'DDR4', 'Windows', '64', 'ThinNlight', 'No', '0'],
       ['0.11211573236889684', '0.1308016877637127', '0.0',
        '0.0031645569620253164', '0.5740382228840901',
        '0.09158139122205135', 'DELL', 'Intel', 'Core i3', '11th',
        'DDR4', 'Windows', '64', 'ThinNlight', 'Yes', '1'],
       ['0.07142857142857144', '0.03968253968253969',
        '0.5119047619047619', '0.017857142857142856',
        '0.6056022408963588', '0.0747064419225634', 'DELL', 'Intel',
        'Core i3', '11th', 'DDR4', 'Windows', '64', 'ThinNlight', 'Yes',
        '1'],
       ['0.11278195488721802', '0.0', '0.25', '0.42105263157894735',
        '0.6666666666666662', '0.22789670437776688', 'acer', 'Intel',
        'Core i5', '10th', 'LPDDR4', 'DOS', '32', 'Casual', 'No', '0'],
       ['0.13010204081632662', '0.03273809523809523',
        '0.47767857142857145', '0.16071428571428573',
        '0.6341036414565828', '0.09667064419225635', 'ASUS', 'Intel',
        'Core i5', '10th', 'DDR4', 'Windows', '64', 'Casual', 'No', '0'],
       ['0.06666666666666668', '0.04444444444444436',
        '0.25416666666666665', '0.004166666666666667',
        '0.6640522875816991', '0.0650794003115265', 'ASUS', 'Intel',
        'Core i3', '11th', 'DDR4', 'Windows', '32', 'Casual', 'No', '0'],
       ['0.255952380952381', '0.21874999999999992', '0.0',
        '0.5729166666666666', '0.7075163398692812', '0.1792011779595016',
        'ASUS', 'Intel', 'Core i5', '11th', 'DDR4', 'Windows', '64',
        'Casual', 'No', '1'],
       ['0.40476190476190493', '0.2441406250000001',
        '0.041666666666666664', '0.4427083333333333',
        '0.6960784313725492', '0.2857153426791276', 'MSI', 'Intel',
        'Core i7', '10th', 'DDR4', 'Windows', '64', 'Casual', 'No', '0']]
```

Figure 7 Array of K-Prototype Centroids – Presented here in a technical way, we can take these arrays and reverse them back to the original values and format to create a easy to understand table for less technical audiences

To further this research, I would like to find data with less missing information. Missing a lot of the display size data may have biased the results somewhat as there would have been a larger portion of these display sizes. In the future, it may be useful to create a neural network that can take a more complicated series of inputs that can consider things like consumer ratings and preferences to output a price. I think in the future, I would like to use consumer preferences/ratings to help parse out more of what consumers want. As it stands, we can now use this model and these centroids to quickly assess not only what price our laptops should be but also what features we should look to improve next. The next steps would be to collect more laptop data and see how accurately our models can predict their pricing groups.

Diving into consumer behavior will allow us to understand how consumers choose laptops on various features and what they feel is a justifiable price. In doing so, we can combine that knowledge with our features model and see what adjustment can be made to existing laptops to optimize their pricing structure. It is possible we can raise the prices of certain laptops because they have more desirable features to certain consumers while lowering the prices of others with less desirable features. This is an adaptable and flexible model but one that could use further refinement.