Web Scraping and Linear Regression

Model- 2

Prepared by:

Sara AlAbdulsalam

Fatimah AlShammari

Introduction:

Online shopping today has become an important part of every business. Products presentations and alignments is a significant part because it impacts how the audience perceive the brands. Another important part for the brands owners is products pricing estimation that decides to higher or lower the product price boundary which is based on the product producing expenses and on the audience reviews and ratings. One of the most popular shopping sites is iHerb that has more than 30K products [1] and we chose to help iHerb by implementing two machine learning models specifically linear regression, one to predict the products ratings and another to predict product price, then reflect the results on the products distribution on the site making the higher rating products appears first and on top to the audience, and choose the appropriate price for a product. The model will be trained and tested on data scraped from iHerb website. In this document we will show you insights extracted from the data, whether linear regression model is suitable to solve the problem, implementation and results.

Data Description:

A model performance deepens heavily on the data it was trained on. To acquire the data, we used web scraping on iHerb website. The features of the products dataset are the following:

- Name: the name of the product

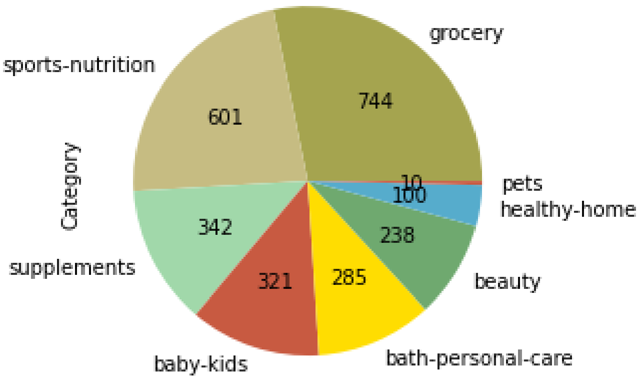
- Categories: the name of the category the product belongs to (e.g., Beauty, Supplements, Grocery)
- Size: the size measurement of the products (measures with mg, g, kg…)
- Number of reviews: the number of customers reviews on the product

The target will be:

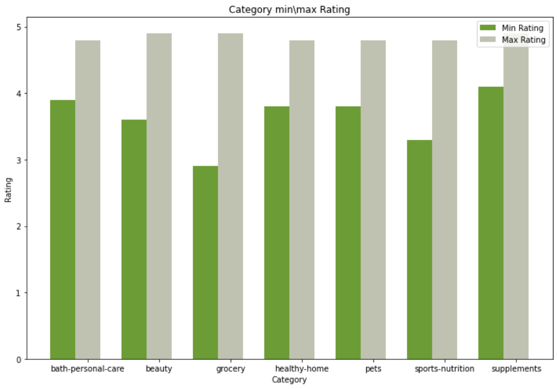
- Ratings: the average rate of the product.
- Price: product cost

We gathered 2641 products, and performed simple EDA:

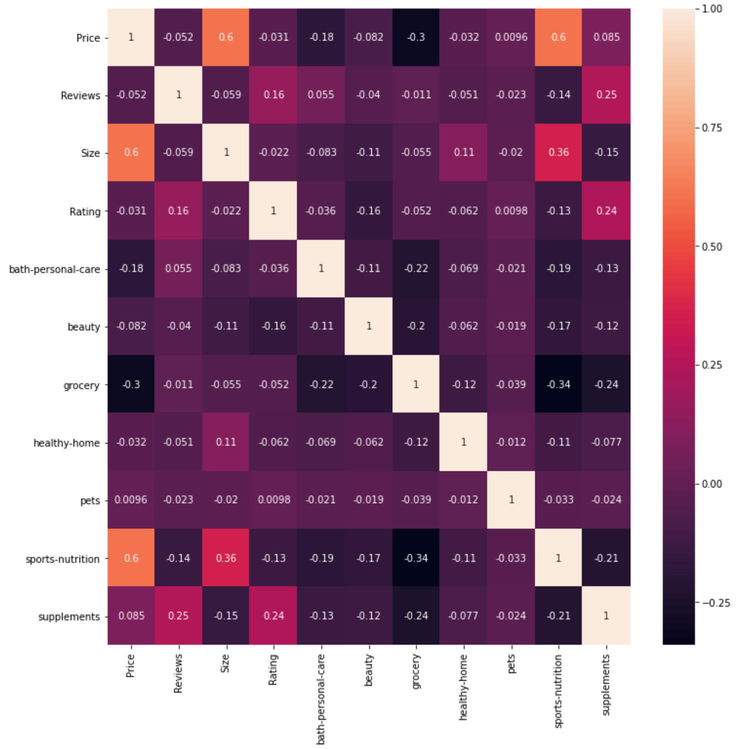
The below pie chart shows the distribution of products category:



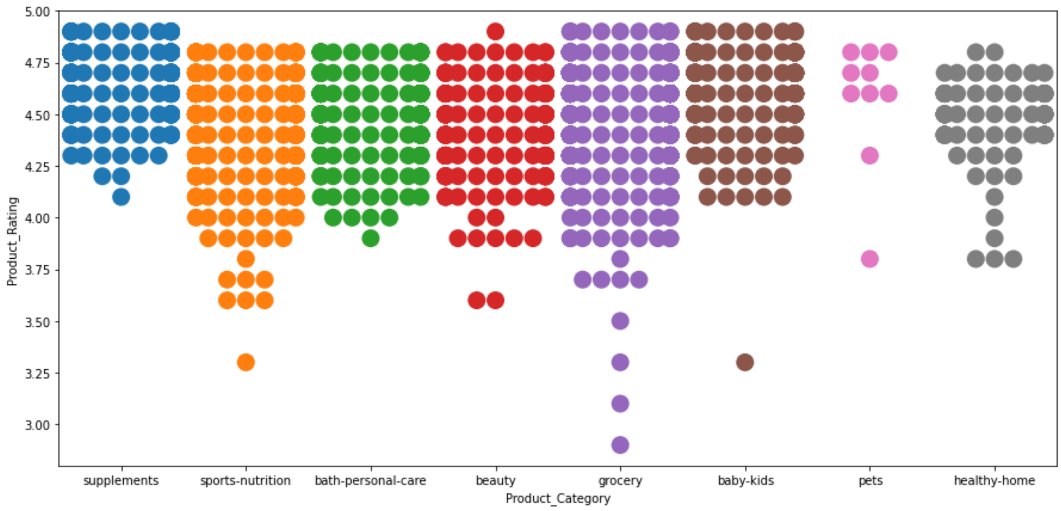
Bar chart of the maximum and minimum rating of each category, as we can see all the products have high rating (no less than 2.9 out of five):



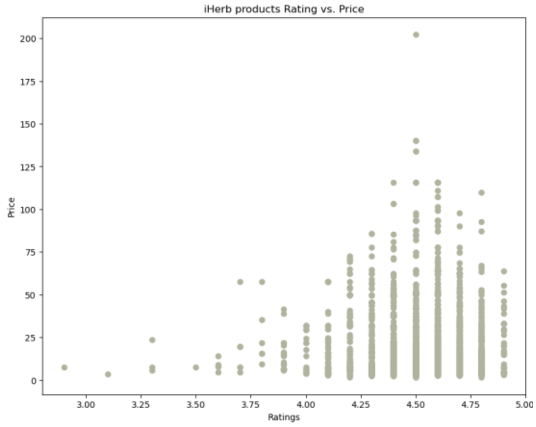
Heatmap of the features correlation, which shows that the rating has no strong nor moderate correlation with the other variables therefor linear regression is not a suitable model for predicting it:



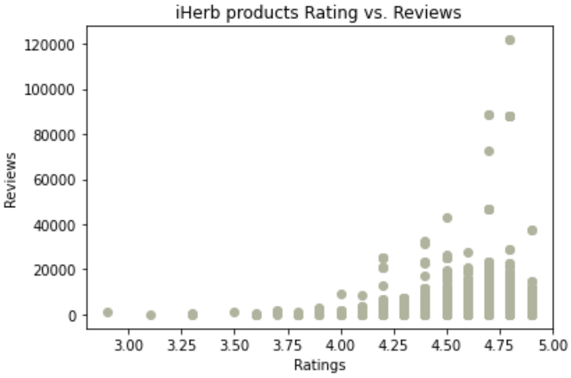
Swarm plot of categories and ratings, we can see that the products are stacked on the higher rating boundary:



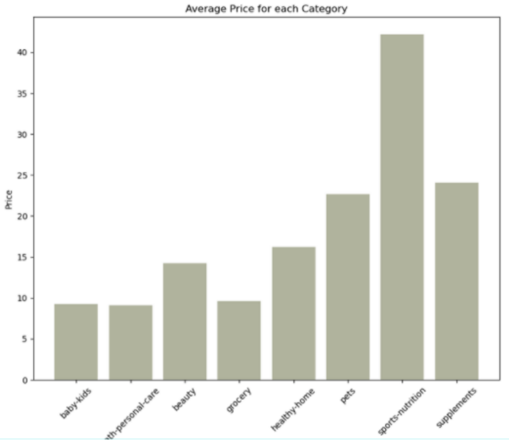
Scatter plot of price and ratings:



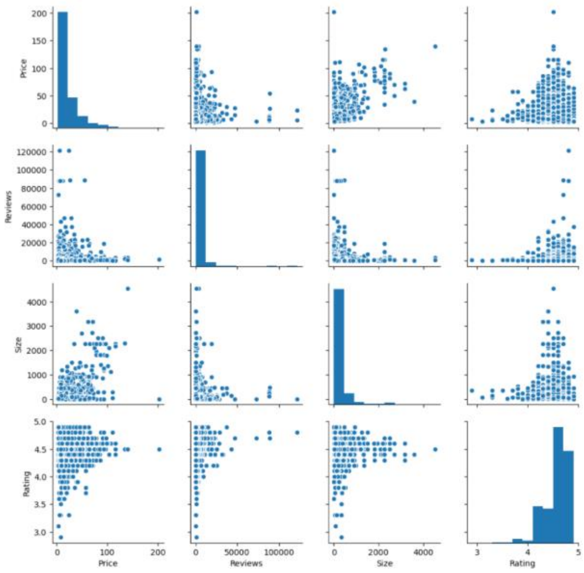
Scatter plot of products ratings vs reviews:



Bar chart of the average price for each categories which highlights the face that model products are between 10 and 40 price range:



Pair plot of all the numerical features:



Pre-proccing:

- Extract product size from its name, for example:

Doctor's Best, Lutein from OptiLut, 20 mg, 120 Veggie Caps

➔ size = 20 mg

- Convert categorical variable (category) to dummy variables

- Drop rows with null values

Tools Description:

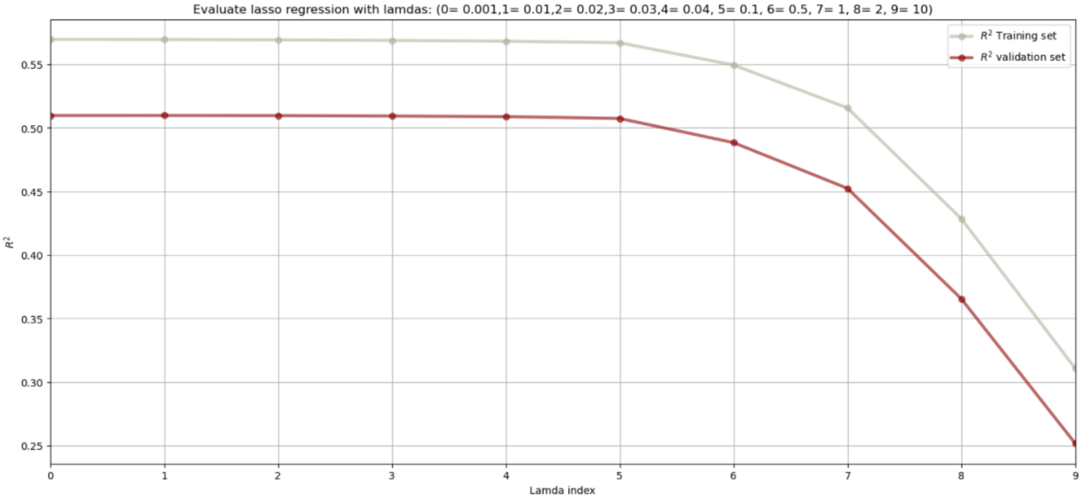
To achieve our goal, we will analyze and explore the data in Python by using Jupyter, and we will use different packages such as: BeautifulSoup, SKLearn, statsmodels, Pandas, Matplotlib, Seaborn, and numpy.

Algorithms:

- LASSO Feature Engineering: using lasso model coefficients of features we chose the features with the highest coefficients, and dropping the features with lower coefficients since they might be considered a noisy data for the model.

- LASSO Regression: Least Absolute Shrinkage and Selection Operator linear regression, we chose LASSO Regularization to drop any noisy collinear features. We split the dataset to 40% for training, 30% for validation and 30% for testing. Making multiple tries of selecting features then fitting the model then evaluate it on the validation set, the best validation score is approximately 69%.

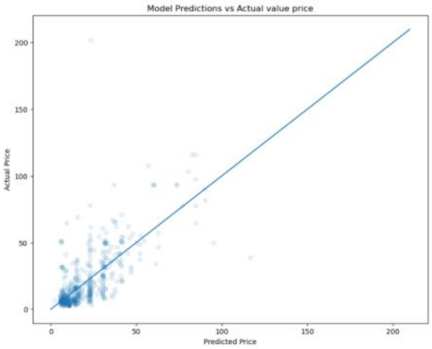
  To choose alpha value we used multiple tries:



  Which outputs that alpha = 0.03 is a good choice. To make sure we used LASSO cross validation with k equal five we got 0.033 as an optimal value.

  - LASSO Model -trained on all features-:

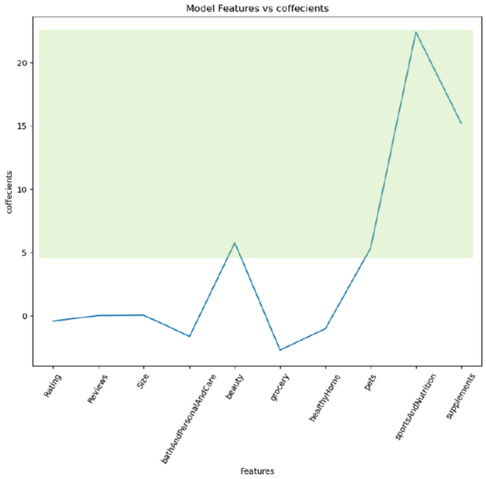
    Model predictions:



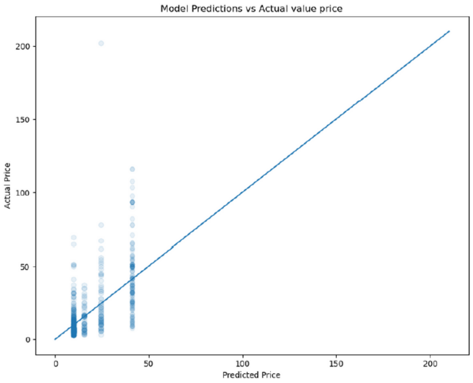
R^2 score on validation set: 50.484%

  - LASSO Model -trained on high coefficient features-:

    High coefficients features:



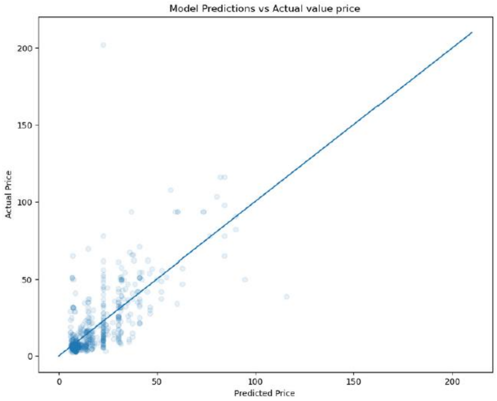
Model predictions:



R^2 score on validation set: 37.908%

- LASSO Model -trained on high coefficient and highly correlated features-:

  Model:



R^2 score on validation set: 50.58%

After choosing the best features which are the ones with highest correlation with the dependent variable (Price) and highest coefficient we re-trained the model with 70% of the data using SKLearn and StatsModel.

- Linear Regression: attempts to model the relationship between two variables by fitting a linear equation to observed data.
- We spitted the dataset into 70% training and 30% testing.
  - Sklearn

    We use cross validation score to evaluate the training.

    We did two linear regression models, one before feature selection and another one after feature selection,

    The result for First model:

    - Training accuracy: 0.5696

    - Validation score: 0.5615

    - Testing accuracy: 0.6977

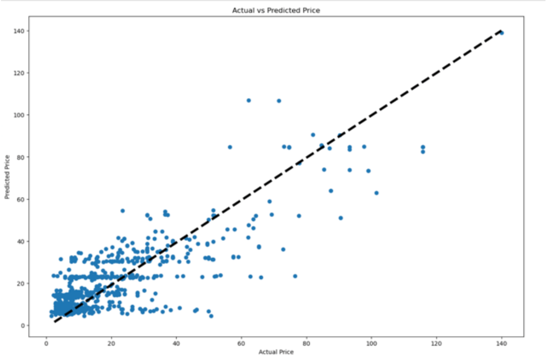
    The result for Second model:

    - Training accuracy: 0.5681

    - Validation score: 0.5615

    - Testing accuracy: 0.6982

    Below figure express the model prediction against the actual value of product’s price.



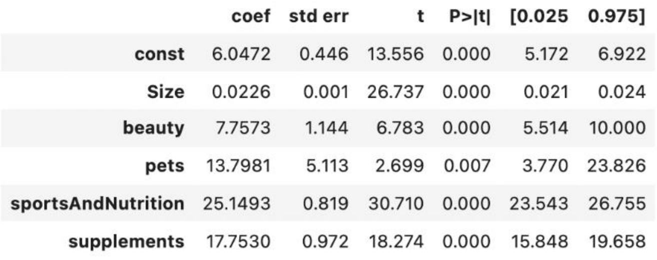
  - Statsmodels

    We use the training set after feature selection, and we get the results below:

    r-squared: 0.568

    Adjusted r-squared: 0.567

    All p-values for each attribure are less than 5%:

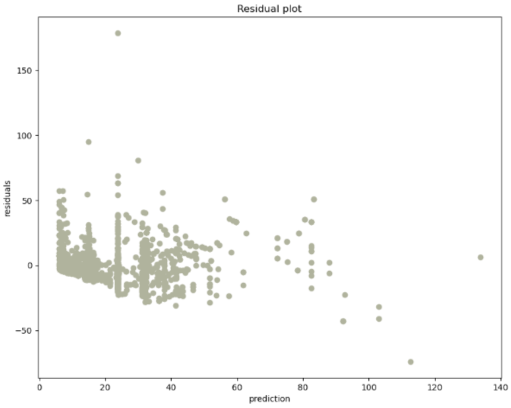


	coef	std err	t	P>|t|	[0.025	0.975]
const	6.0472	0.446	13.556	0.000	5.172	6.922
Size	0.0226	0.001	26.737	0.000	0.021	0.024
beauty	7.7573	1.144	6.783	0.000	5.514	10.000
pets	13.7981	5.113	2.699	0.007	3.770	23.826
sportsAndNutrition	25.1493	0.819	30.710	0.000	23.543	26.755
supplements	17.7530	0.972	18.274	0.000	15.848	19.658

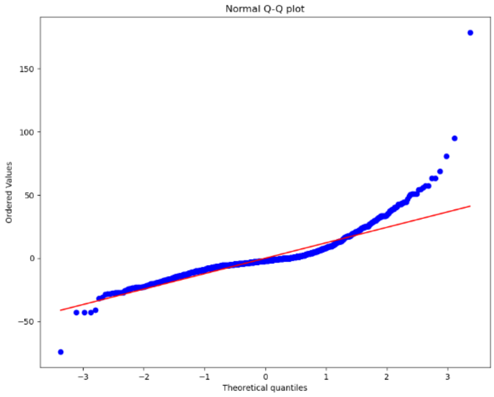
- Assumptions:
  - Assumption 1: regression is linear in parameters and correctly specified.
  - Assumption 2: residuals should be normally distributed with zero mean.

To check for assumption 1 and 2 look at the figures below:

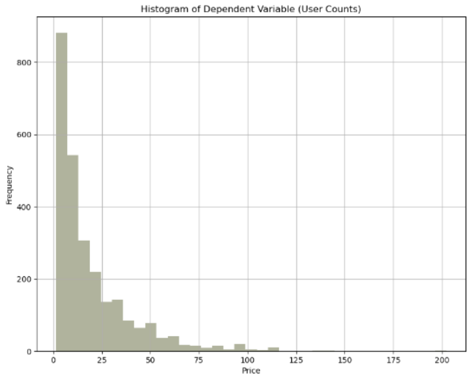
Residual plot:



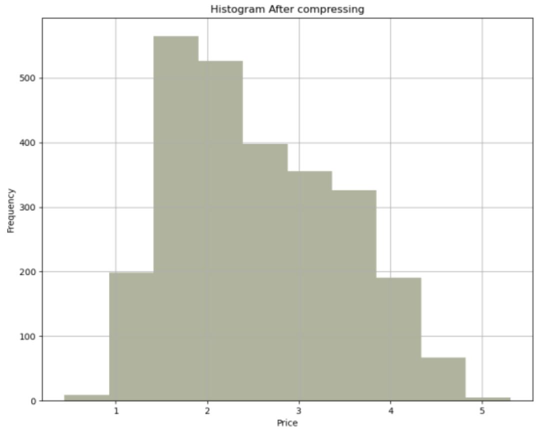
Q-Q Plot: right skewed



Bar chart for price to check the distribution among products.

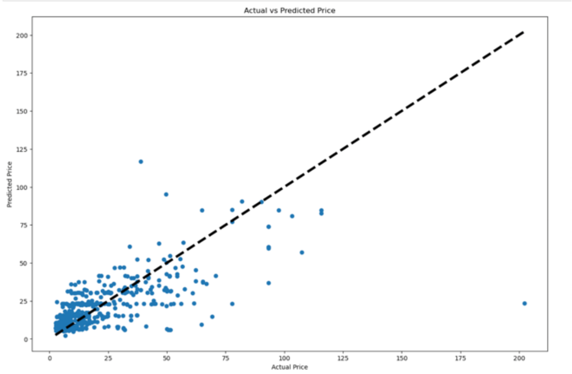


Bar chart after shrinking price values by using log function to check the distribution among products.



- Assumption 3: error terms must have constant variance.

To check for assumption 3:



- Assumption 4: errors are uncorrelated across observations.

- Assumption 5: no independent variable is a perfect linear function of any other independent variable (no perfect multi-collinearity).

Conclusion:

We aim to improve the user experience and site sales by using a linear regression algorithm to predict product ratings, making the products with the highest rating appear first. By training and testing the model on iHerb website dataset. In this document, we reviewed the problem that we want to solve, a description of the data we will work on, and finally the tools that we will use.

References:

[1] https://www.iherb.com/info/about