Project Design

I wanted to predict the price of a lego set based on features of that lego set. There were some features that seem obvious like the number of lego pieces in the set and the suggested age range, but I discovered deeper insights like how the wording on the product description affected price. I collected data from 21 countries in total and after converting currencies to USD, I still found discrepancies in pricing for the same lego sets.

There isn't much of a business case to be made for this analysis, but one could be made with additional information provided from lego. If we knew how many units were sold of each of these products in a certain time frame, we could help identify the things customers are willing to pay more for. This could help inform pricing strategy or new product development depending on who would be willing to pay for the study.

Tools

Jupyter Notebook - I did all of my scraping, EDA, modelling, and visualization in Jupyter Notebooks.
Pandas - I stored and manipulated my data in pandas dataframes.
Statsmodels - I ran all my model in statsmodels because I liked the p-values and confidence intervals it provided with my coefficients.
NLTK - I used this package to make sense of the product description field. It was particularly helpful in converting the text into different parts of speech.
Selenium - I used selenium to browse the lego website and pull down all the necessary info for this analysis.

Data

I scraped data from the lego website. The website has an option to select which country you are from, so I repeated this process for each country.

1) Go into every theme (Star Wars, Architecture, etc.) and get a list of every available product
2) From that list of links, go into every product page and scrape all fields of interest.

From there I had a data set with around 12500 rows and 14 columns. From those 14 columns I engineered around 5-10 other features that I used to model price.

Algorithms

OLS - I liked this model because it got me results with coefficients that had p-values and confidence intervals. When I split the data between training and test sets, I got fairly consistent performance so I didnt think overfitting was a problem. I had issues with heteroskedasticity, so I

ended up using log transformations and dropping some of the more extreme lego sets (Millenium Falcon costs $800 and the average set price is $47).

Lasso / Ridge Regression - I tested these models, but they didn't offer any additional predictive power or a reduction in overfitting so I didn’t end up using them.