LILAC (LEGO Investment-Linked Awesome Classifier)

1. Problem Statement

This project was inspired by a random encounter with the website bricklink.com. Bricklink is the world's largest secondary marketplace to buy and sell LEGO products. After some research, it was discovered that not only can LEGO be a viable alternative investment, but its annualized investment returns (if all newly released sets are purchased each year) for the past 10 years have also been very comparable to the S&P500 index.

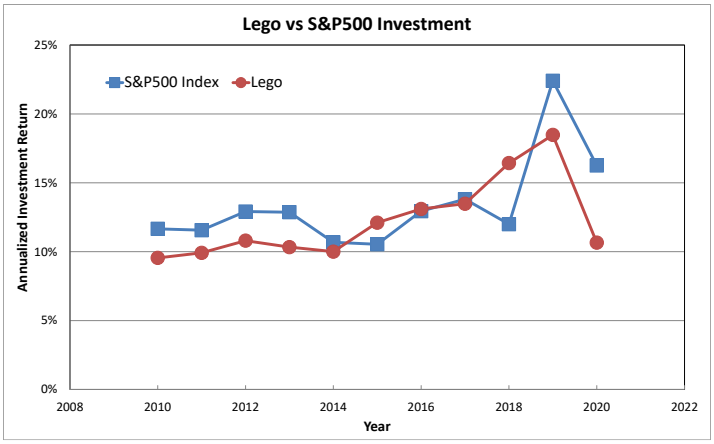

Figure 1 Comparison of Annualized Investment Returns between LEGO and S&P500 Index.

Just like some stocks perform better than others, picking the right LEGO set will also generate better returns. With that, the business problem was defined as such:

Can one leverage data science (machine learning in particular) to identify good LEGO investment opportunities to outperform the general stock market?

2. Background

There are two main reasons why this is good problem to apply data science techniques:

- Data Science is about discovering the unseen patterns and relationships. LEGO design is art. Not only are the possibilities of what can be built by LEGO endless, there are also no apparent reasons for one LEGO build to worth more, or less, value than another in the market.
- Data Science is also about generating business values. Studying this problem will help raise awareness of another type of investment opportunities. Solving it could help people formulate the optimal strategies to maximize their investment returns in the business.

The idea of LEGO as an alternative investment has been mentioned in a few online articles but it is till not known to a lot of people. There are websites such as BrickEconomy which track the value growths of the LEGO sets. However, little work has been shared around what factors are impactful to the valuation of a LEGO set in the secondary market. In the world of LEGO investment, it is easy to fall in the trap of believing that a set you like personally will also be desirable to other people.

3. Data

The data used in this project was 100% scraped using both Selenium and BeautifulSoup from a variety of LEGO-themed public websites: BrickEconomy, Bricklink, BrickOwl, LEGO.com, Rebrickable, Brickset, and Brickinsights.

Relevant information gathered included: original retail value, secondary market current value, primary market availability, various designs (color, theme, parts, etc.), and review score.

Rebrickable is the only website that offers an API and its data came in with a JSON format. Data from other websites came in with a variety of formats including list, dictionary, and dataframe. All the data was cleaned, saved into separate csv tables, and then fed the MySQL DBMS (schema shown below) before any further analysis was carried out in Python. At this stage, the database contains ~2.25 million rows, with each row representing a unique part in a LEGO set, for a total 9,931 LEGO sets.

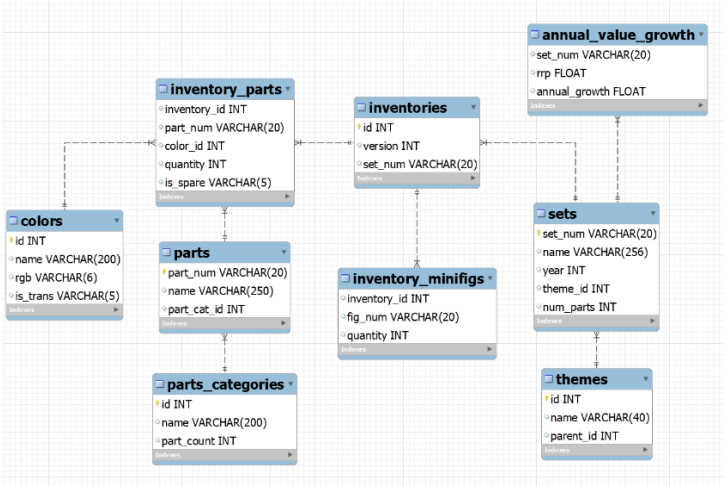

Figure 2 Lego Database Entity Relationship Diagram

4. Exploratory Data Analysis (EDA)

In additional to some generally exploration of LEGO history, the most important task during this stage of the project was to decide on the features for modelling. Based on the problem statement, the dataset was first divided into 2 classes using 8% annual value growth (historical average annual return of S&P 500) as the cut-off. The generally methodology followed in the project was the Univariate Analysis. The process involved looking through all features in the database one at a time and examining its relevance (predictive power) to the target variable. Techniques used are Descriptive Statistics, Bar Charts, and Box Plots.

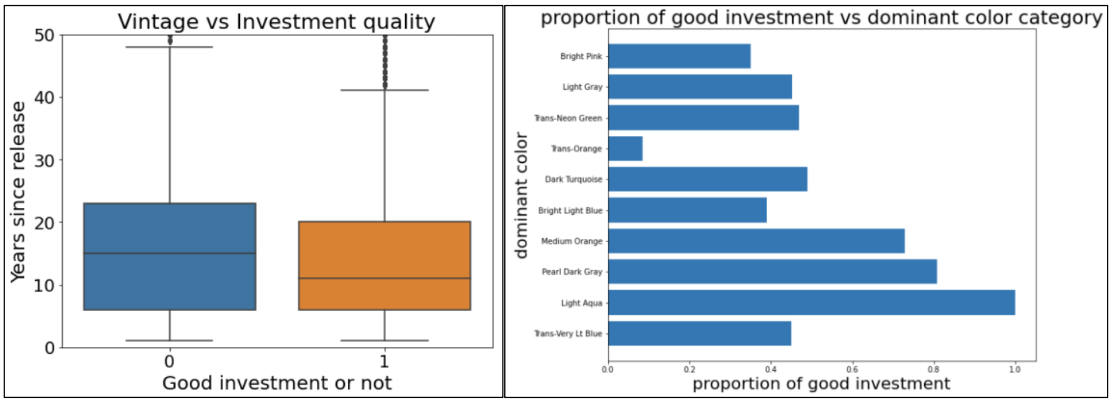

Figure 3 Feature Engineering Examples

However, not all features in their original forms are relevant to the target variable. Feature engineering is also required. For example, the above figure shows that two new features of LEGO vintage and Dominant color, engineered from Release Year and Parts Color, respectively, appear to have predictive powers.

5. Modelling

Three groups of classifiers with different complexities have been tested: Standalone Models, Ensemble models, and Deep Learning models. Using Precision score as the evaluation metric (for an investment-related problem, it is important that the predicted positive is actually a true positive), their performances are summarized as follows:

Table 1 Modelling Results

Model	Test Precision	Model	Test Precision
FF Neural Network	0.75	LinearSVC	0.70
SVM (rbf)	0.74	XGBoost	0.69
RandomForest	0.72	Naïve Bayes	0.69
Logistic Regression	0.70	KNN	0.66

6. Findings and Conclusions

The Feedforward Neural Network has performed the best with a Precision Score of 0.75, while most other models tested were also able to achieve a score between 0.70 and 0.75. Using results published in the literature on other investment-related problems, these are pretty good numbers. However, because the top performing models (Neural Network and SVM) are both non-linear and they lack interpretability. Efforts were turned to the linear models and tree-based models to draw insights on the features.

Table 2 Left – Logistic Regression; Middle – LinearSVM; Right - RandomForest

Top Positive	Top Negative
More parts	Higher rrp
Retired status	Retail available
More unique parts	Generic part categories
Theme: Classic Space	Theme: Bulk Bricks
Theme: Series 19 minifigures	Theme: Universal Building Set
Theme: Black Knights	Theme: Basic

Top Positive	Top Negative
More parts	Higher rrp
Retired status	Retail available
Is Parent Theme	Generic part categories
Theme: Classic Space	Theme: Bulk Bricks
Theme: Heroes	Theme: Hidden Side
Theme: Villains	Theme: Dimensions

Most Relevant
Years since release
Review Score
$ per part
Unique part categories
Num of parts
Unique parts

Bothe Logistic Regression and Linear SVM models have identified similar feature impacts. For example, both models have suggested that bigger sets are better. However, bigger sets also tend to be more expensive and Higher Retail Price is a negative feature. One needs to find the balance when making investment decisions. Another interesting finding is that some of the negative theme features include “Bulk Bricks”, “Universal Building Set”, and “Basic”. All these are simply suggesting a lack of uniqueness. It is probably the reason why those associated LEGO sets are not very desirable in the secondary market.

7. Business Applications and Next Steps

The model will be tested on the newly announced 2021 LEGO sets, and a set of investment recommendations will be made. In the meantime, the best-performing Neural Network is far from optimized and will continue to be tuned. Lastly, equally important to model optimization is being able to interpret the model results. The framework of SHAP (SHapley Additive exPlanations) will be studied. This effort may also be extended to potentially applying the Recursive Feature Elimination (SVM-RBF-RFE) framework on the second-best performing model (rbf SVM).

8. Neural Network Optimization and Interpretation using SHAP *

*Please note that this section was added after the project was completed.

A scikit-learn wrapper was applied over the Keras model before feeding into a Pipeline and two sequential GridSearchCV's for optimizing the following hyper-parameters: initializer, regularizer, regularization_rate, optimizer, learning_rate, dropout, and epochs. After tuning, the performance of the Neural Network classifier on the test dataset improved from 0.75 to 0.78.

Achieving a high accuracy on the prediction provides just half of the answer to the business question; no analysis is complete without drawing insights from the models. SHAP (SHapley Additive exPlanations) is one of the creative frameworks that can be used to interpret Neural Network models. It draws upon the classic Shapley values from game theory and was first proposed by Lundberg and Lee (https://arxiv.org/abs/1705.07874) in 2017.

The feature importance according to their mean SHAP values is summarized in the following plot:

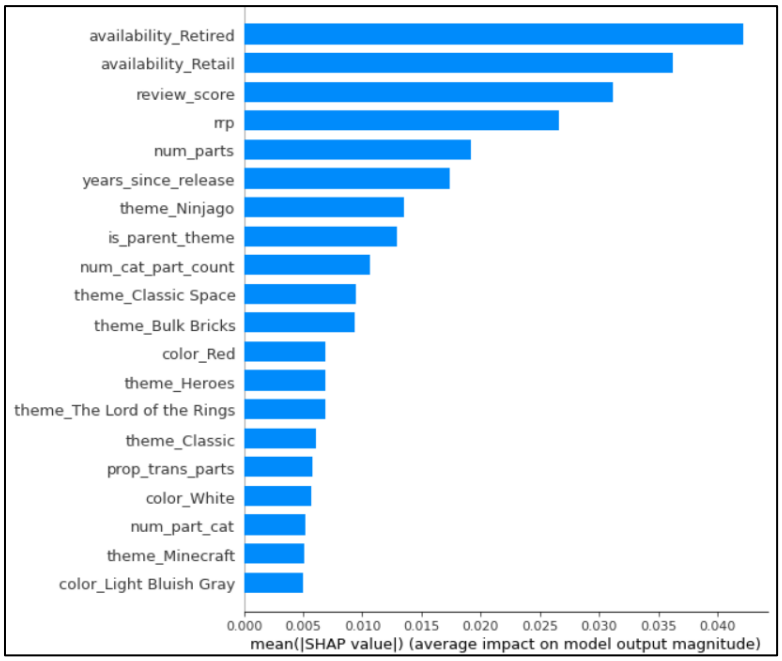

Figure 4 Feature importance in optimized Neural Network model using SHAP framework

It is interesting and encouraging to see that among the 10 most impactful features identified by the Neural Network model, only "theme_Ninjago" was not captured by the linear or tree models. Although the non-linear Neural Network model is considerably more complex than the linear or tree models, the impacts of the features are relatively consistent.

Thinking of this from a different perspective, the alignment of the interpretation results also somewhat confirmed the validity of applying the SHAP framework on Neural Network models.