Lego Insights and Investment Application

DATA309 Data Science Capstone Project

Joshua Lowe
University of Canterbury
joshpaullowe@gmail.com

Yaquub Ali
University of Canterbury
abdiy2333@gmail.com

Yazeed Almutairi
University of Canterbury
almutairiyazeed22@gmail.com

1 Introduction

1.1 LEGO

LEGO is a globally recognised brand producing plastic pieces that can be assembled to form a wide range of models. LEGO is popular across a wide range of ages which stems from its ability to be themed. From Star Wars to Harry Potter, LEGO engages people with vastly different interests which has allowed it to grow into a multi-billion-dollar business (Fortune Business Insights, 2021). This project will leverage data from multiple sources to provide insight into the market dynamics of LEGO and provide an application to help investors make informed decisions about which sets to invest in.

1.2 Goals

LEGO's partnerships with entertainment franchises contributes to its success by resonating with global audiences (Fortune Business Insights, 2021). This has allowed LEGO to become a viable investment option for investors (Dobrynskaya & Kishilova, 2022). Through data analysis, the resale prices of LEGO sets can be predicted based on a variety of features. Cluster analysis will be used to group LEGO sets based on features and user ratings. Then, through machine learning, a Price Prediction Model will be created to predict the resale price of sets based on features such as piece count, theme, subtheme and minifigure count. Techniques such as random forest, linear regression and XGBoost will be utilised to create predictions of the average yearly return of a Lego set.

A key objective of this project is to use Amazon Web Services (AWS) as a cloud-based data storage to house all datasets required for the entire project. AWS provides scalability and security ensuring large volumes of data can be efficiently stored and accessed (Singh, 2024). Additionally, Dash by Plotly will be utilised to develop and interactive dashboard that integrates with AWS (Jean-Michel D, 2018). The dashboard will serve as the front-end interface allowing users to predict the investment potential of LEGO sets. Additionally, knowledge embedding will be used to make the dashboard more user friendly (Linjuan et al., 2022). This technique allows Large Language Models (LLMs) such as ChatGPT to access the results of the analysis conducted throughout this project. This data pipeline ensures a streamlined, end-to-end workflow where data is pulled from AWS, analysed, and presented through dash. This approach ensures the project is scalable and efficient.

1.3 Constraints

The success of the project depended on obtaining high-quality, consistent data from multiple sources. The data came in a mixture of structured, semi-structured and unstructured data. It ranged from downloadable comma separated value files (CSVs) to APIs. This required careful implementation and cleaning. The most significant obstacle faced was gathering retail and resale price data, which took much longer than expected. It would have been ideal to gather data from websites such as Brickeconomy, Bricklink and Brickinsights, but this was not possible due to paywalls, private API keys or a limited number of API calls. Many of these sites also did not allow scraping using CSS selectors. When attempts were made to do this, it was instantly blocked. We reached out to some websites that had private API keys, however many of these sites did not respond.

2 Data

The five V's of Big data are Volume, Variety, Velocity, Veracity and Value (Hiba et al., 2015). The data sets used in this project are relatively small in size, however the framework still provides a structured approach to understanding the data. The data came from three sources, Brickset, Brickowl and Rebrickable (Brickset Home Page, 2024; Rebrickable - Build with LEGO, 2024; Brick Owl - LEGO Marketplace, 2024; Tosic, 2021)

2.1 Volume

Volume refers to the amount of data which is stored (Hiba et al., 2015). This can range from bytes all the way up to petabytes. More data typically leads to more issues with storage and analysis however, in the case of machine learning, more data is typically better (Sarker, 2021). The Brickset data contains 15,634 rows and 36 columns which is large enough for analysis. It has a substantial number of missing values posing a challenge to get into a format that is suitable for analysis. The Rebrickable dataset has multiple datasets with varying sizes. The two sets that were mainly used for analysis contain information about the sets and the themes. The "output_sets" dataset has 23,131 rows and six columns and the themes dataset contains 465 rows with three columns. None of the Rebrickable datasets contain missing values. The Brickowl data only has 5304 rows and there are no missing values. The overall Volume of the datasets is large enough to create insightful analysis but not big enough to case issues for running analysis on the local devices used throughout the project.

2.2 Variety

Variety is the diversity of data types. This refers to unstructured, semi-structured and structured data, as well as the column types of each dataset (i.e integers or strings). The Rebrickable data is structured, and a schema can be found online meaning joins between various datasets within it can be performed. Both the Brickowl and Brickset datasets do not have a schema but do come in a tabular format with rows and columns, making them semi structured. Across all the datasets there is a variety of data types, particularly string, integer and date types. The varied nature of these datasets means careful data integration is required.

2.3 Velocity

Velocity refers to the speed at which the data is generated. The velocity of the LEGO data is relatively low. All datasets are updated periodically rather than in real-time. However, the dashboard implemented for this project gets input from the user. This means there is some degree of dataflow. When users are using the dashboard, it is essential that the experience is smooth and efficient. There are various techniques to reduce latency and make the front-end user friendly. Machine learning algorithms should be executed in advance, meaning that when a user requires an output form the algorithm the response will be instant.

2.4 Veracity

Veracity is the quality of the data. Missing values, outliers, variance and imbalances all impact the Veracity of the data. Both Brickset and Brickowl contain a large number of missing values. For

Brickset the columns “Launch Data" and “Exit Date” nearly two-thirds of the rows contain missing values. These missing values could have a significant effect on the results if not taken into consideration and dealt with accordingly. For all columns (variables) of the Brickset data and the Brickowl data, the percentage of outliers is less than 1%. This is relatively low and likely has little effect on results of the analysis. The Rebrickable data has a slightly higher outlier percentage with the number of pieces columns having 1.68% outliers. This is again relatively small and will have little effect on the results. None of the columns in any of the datasets display near zero variance. This is to be expected due to the huge variation in values expected when dealing with price data, piece count, set number, theme type, etc. The datasets do contain some classification imbalance, some themes have a lot more sets than others. For instance, “Star Wars” has 822 compared to “The Simpsons” which only has two. This can have a significant effect on the results of the analysis because if the two sets of “The Simpsons” have a dramatic price increase then it would indicate this theme is a strong predictor for resale value but in reality, it's too small of a sample size to be a statistically meaningful indicator.

3.4 Value

Value refers to the actionable insights that can be derived from the data. The combination of Brickset, Rebrickable, and Brickowl data has imperfections, but offers substantial value for exploring LEGO trends and predicting price. The Brickset data in particular had data on retail and resale price allowing for price prediction to be carried out. Each dataset has unique aspects that when used in conjunction provide a detailed dataset to perform analysis on.

3 Methodology

3.1 Data Pipeline

Extract, Transform and Load (ETL) is a data engineering process used to collect data from multiple sources, clean it and then load it into a target system for further analysis/storage (Souibgui et al., 2019). This process ensures that the data is organised, cleaned and structured properly so it can be analysed for meaningful insights. Amazon Web Service (AWS) is a cloud service provided created Amazon to assist with ETL. This allows for an efficient data pipeline which is essential for creating a scalable application. The ETL process overview can be seen in Diagram 1.

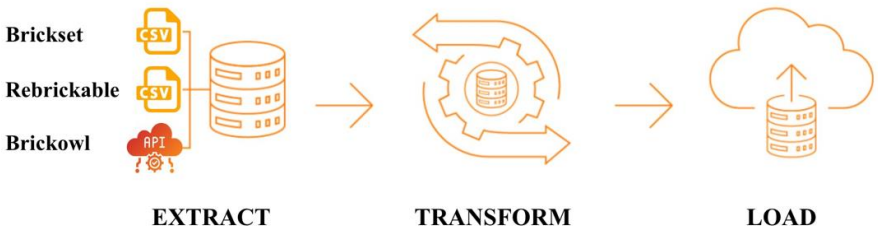


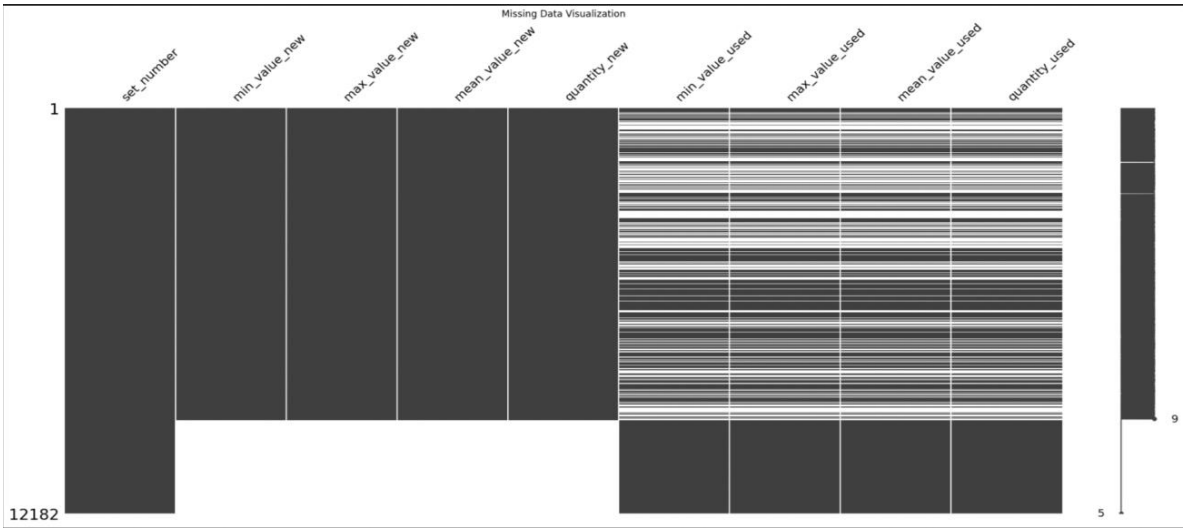
Diagram 1. *Diagram of Extract, Transform and Load Process*

Extraction involves obtaining the data needed for analysis. The Rebrickable data is publicly available and can be downloaded straight from their website in the form of CSVs. The Brickset data was found on another project through GitHub. The raw data was a series of CSVs for each year from 1991 to 2022. The Brickowl data was acquired through a scraping algorithm with the help of Trien Lam Truong, another DATA309 Student. The scraping process included the use of API parameters to access LEGO set data. This was done using the requests library, which was used to send a “GET” request to the Brick Owl API with the required parameters such as API key and data type (sets). At first, a CSV file was created to store unique identifiers (BOIDs) of the LEGO sets. This ensured it could be tracked which sets had been processed. The scraping algorithm then iteratively queried the API for availability data for each BOID that was not already included in the main dataset. Any successful data retrieval was then passed into an S3 bucket for storage.

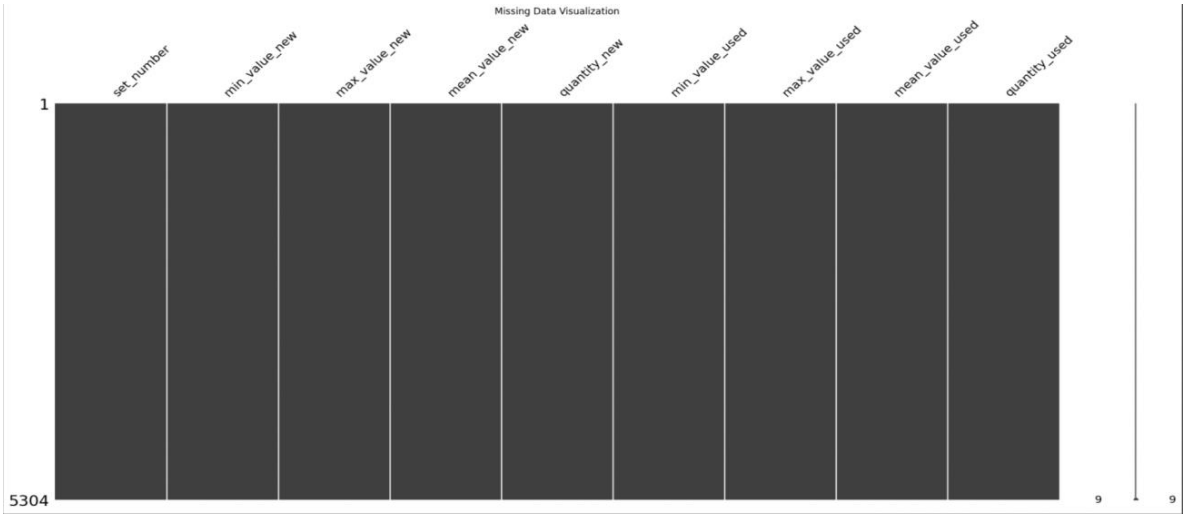
Since this data is all in the form of CSVs and will not be updated throughout the duration of the project, it is “static” data. This makes this step of ETL relatively simple. Instead of setting up a process to regularly update the data, the CSVs can be directly uploaded to the cloud. AWS’s Simple Storage Service (S3) was used to house the data. S3 is a durable, scalable and secure object storage service that allows the data to be stored efficiently. The data is stored in “buckets” and can be organised into “folders” within the buckets. This project's bucket was called “lego-data-bucket” and had various folders based on where the data came from. For instance, the Rebrickable data was stored in a folder called “rebrickable-data”.

Transformation: the data is cleaned in preparation for analysis. The data from the Brickowl API required currency conversion. This utilised the foreign exchange rates API (FXRatesAPI, 2024). This API was used to obtain the latest foreign exchange rates and defined a simple arithmetic function to convert prices from many different currencies into USD. Once all price data was standardised, the sets were separated into "new" and "used" conditions and performed group-wise aggregations to calculate minimum, maximum, and mean prices for each set.

As the dataset contained various missing values, a thorough cleaning process took place to remove any entries with missing values. This process was relatively simple. Firstly, columns that contained many missing entries were dropped as they would not be meaningful to the final analysis. Then all missing entries from the columns were removed using the ‘dropna()’ method. Visualizations 1 and 2 below show missing data visualizations before and after our cleaning process. It clearly shows that before the cleaning process, many fields within our data contained a significant portion of missing elements. After the cleaning process, however, the solid black columns represent the absence of missing data. The number in the bottom left of both vi shows the number of rows in our data. After our cleaning process this number decreased significantly down from 12182 to 5304. Although this is a significant amount, to make meaningful outcomes from our analysis, a smaller dataset was preferred.



Visualisation 1: *Missing Data Visualization Before Cleaning Process*



Visualisation 2: *Missing Data Visualization After Cleaning Process*

Irrelevant columns were removed from all of the datasets to ensure only columns needed for analysis remained. To perform the analysis some additional columns had to be created. Firstly, a column was added that contained the year of retirement for each set. This was derived from the already existing date of retirement column which contained the exact day of retirement. Then a column was created that contained the number of

Loading: due to the static nature of the datasets used throughout this project the cleaned datasets were stored back in the cloud in a different S3 bucket. This provides a simple location for the cleaned data to be retrieved from to perform analysis. Using a different S3 bucket is essential to ensure an efficient data pipeline. It creates a clear distinction between different data states. The raw,

uncleaned data is stored in one bucket while the clean data in another. This makes the clean data easier to access allowing for more efficient analysis.

3.2 Price Prediction

The price prediction model was developed using the Brickset data outlined in section 2. This dataset contained various information about lego sets from 1991 – 2022. Key features included piece count for each set, minifigures included in each set, original retail price, the current resale price, themes and subthemes. The data also included launch dates, the date the set was retired and how many years the set was in retirement.

Given that some of LEGO sets were relased as far back as 1991, it was essential to adjust both retail and resell prices for inflation. This adjustment ensured a fair comparison of prices across different time periods, allowing us to accurately compare the price performance of sets released decades apart. This adjustment allowed us to eliminate distortions caused by changes in the purchasing power of money over time. For further analysis we calculate the percentage growth per year for each set. This captured how much the value of a LEGO set had risen each year since the set has been retired; by adjusting for inflation and calculating percentage growth per year since retirement, we were able to focus on the real investment performance of each set, rather than being misled by the effects of general price level increases. This was especially important for older sets, such as those released in the 1990s and early 2000s, where inflation might otherwise exaggerate their apparent growth in value.

The dataset included fields for launch date and retirement date for each LEGO set, which were parsed to extract the year of release. Years in retirement was also a separate column to make it easier to use instead of having to deal with date formatting. Before this data was used for analysis, we made sure there were no missing values. The final dataset contained 14,980 rows and 0 of these were missing values. Categorical variables such as Theme and Subtheme were converted to numeric using one-hot-encoding (Samuels, 2024)., allowing us to use these field as inputs in machine learning algorithms. Additionally, Theme_Subtheme was created to make sure only a unique combination of these was plotted, since we had some issues were some themes had the same subtheme names, for example, a subtheme called Miscellaneous shows up multiple different themes such as creator expert and Pirates of the Caribbean.

Outliers in the resale prices were observed in the data, where certain sets had extremely high resale values and some sets had very low resale values. However, no outlier removal was performed as the resale price variability was considered an inherent feature of the dataset, which reflects on the demand and rarity for LEGO sets on the secondary markets. One hot encoding was applied to convert theme and subtheme into multiple binary variables. Some subthemes had many unique values, and some had very few to ensure these subthemes with limited data did not skew the results of our plots we decided to only plot those subthemes that had more than 10 entries, initially we attempted to do this without taking into account how many entries in the dataset but this leaded to misleading results where 9/10 subthemes that were plotted as the best predicted percentage growth only had 1 or 2 entries in the dataset. By setting a minimum threshold we ensured that the plots were based on more data which provided a more accurate representation of trends and patterns for predicted percentage growth for subthemes. Any columns that did not show sufficient variance were reviewed, but no significant near zero variance issues were identified.

When working with this dataset there were a few ethical considerations to be considered. However, the Brickset data was publicly available, and all data was used was compliant with public licensing. This data did not contain any private or sensitive information.

Several machine learning methods were used to predict the percentage growth per year of LEGO sets. Linear regression was used to predict the percentage growth based solely on features like piece count, minifigure count and retail price. However, the model performed relatively poor. Linear regression was useful for understanding basic relationships such as piece count on resale price, but it lacks the flexibility to model the more complex interactions that were present in the data such as theme and resale price.

A random forest model was used to capture nonlinear relationships between features and predicted percentage growth. The random forest model outperformed linear regression in every scenario. Feature importance was evaluated with pieces, retail price and minifigures in predicting percentage growth per year. The model showed that the percentage growth was largely determined by pieces and retail price, but specific themes and subthemes also played a role. While random forest performed well it struggled with extreme values (sets with very high resale prices).

XGBoost was also used due to its ability to capture complex relationships between features (XGBoost Documentation, 2024). XGBoost provided similar results to random forest. This method performed well in scenarios where nonlinear interactions were key such as the relationship between theme and percentage growth. While XGBoost was highly effective it can be sensitive to hyperparameters, which required careful finetuning to avoid any overfitting.

The following metrics were used to evaluate model performance. MAE (Mean Absolute Error) measures the average magnitude of errors between predicted and actual values. The lower the value the better the performance. RMSE (Root Mean Squared Error) emphasizes larger errors by squaring the difference between actual and predicted values, the lower the score the better. R^2 (Coefficient of Determination) measures how well the model captures variance in the data, with 1 being perfect and 0 meaning no explanation power (EUMeTrain, n.d.).

The random forest model was selected as the best performing machine learning method due to its balance between flexibility, robustness and interpretability. Pre-processing to optimise this method included one hot encoding for categorical variables, scaling was not necessarily due to it being a tree-based algorithm.

3.3 Clustering Analysis

The clustering aspect of our project was done using the same data as our price predictions, as we were hoping to generate results that supported each other's conclusions/findings. The first step involved 'OneHotEncoding', which is a method used to convert categorical data into a numerical format (Samuels, 2024). This was applied to 'Theme', which allows for more effective processing of categorical data. From here, a new data frame is created which includes this encoded data. Next, the numerical features, such as percentage growth per year, pieces, and years in retirement (which were my chosen features for my clusters) were scaled using 'StandardScaler'. This ensures that the

numerical features will all have a mean of zero along with a standard deviation of one. This allows the clustering algorithm to avoid being more biased towards certain features that may have larger numerical values. Both 'OneHotEncoding' and 'StandardScaler' were made available for use through 'Skicit-learn', a machine learning, free-to-use, library in python. Once the features are ready, the scaled numeric features and the encoded categorical data are combined into one single data frame which the Kmeans clustering algorithm is implemented on to separate the LEGO data into three distinct clusters. The Kmeans clustering algorithm was also implemented through 'skicit-learn'. The distinct clusters represent different investment characteristics. From here, we wanted to understand what sets the clusters apart in terms of their characteristics.

Through the use of 'Matplotlib' a visualization library in Python, we generated various visualizations to shed light on the unique characteristics of each cluster. Various visualizations were made to understand the clustering results and unique characteristics of each. A stacked bar chart was made to show the distribution of LEGO themes across the clusters, giving an insight into which themes dominate each cluster. Another plot shows the average price growth year by cluster, highlighting the differences in yearly appreciation between the clusters. Along with this, a bar plot summarized the average price growth per year (%) and years in retirement for each cluster, giving a comprehensive view of how these factors vary across groups.

The clusters were then named according to their distinct features. For example, our first cluster was named 'Low Price, Moderate Growth'. Our second cluster was named 'Large Premium Sets, High Growth', and our third cluster was named 'Retired Collectibles, Modest Growth'. The dataset was then updated to include these cluster names, to allow for easier interpretation for anyone who may come across our project. Finally, we created a summary table for each cluster, which calculates the mean and standard deviation for our chosen features for each cluster.

Regarding the data used for the clustering, as it was the same data used for the price predictions, which was generated from Brickset, which is publicly available, there were very few ethical considerations to consider.

3.4 Knowledge embedding

Knowledge embedding is a process that converts data into dense vector representations. This allows machine learning models to interpret and analyse this information (Linjuan et al., 2022). In this project the data used included various results from the analysis performed on LEGO, such as the average number of parts per set per year, and the RMSE, MAE and r^2 values for certain machine learning algorithms that were used (AI Jason, 2023). This process was used to create an interactive tool integrated into the dashboard allowing users to ask questions specific to this project.

The OpenAI Embeddings module was used to transform the dataset into vectors. Each result from the dataset was represented as a vector and is stored in a knowledge base. To manage the vectors and conduct efficient similarity searches, a FAISS (Facebook AI Similarity Search) index was used. This enables fast and scalable similarity search across large amounts of data.

An overview of the process is provided in Diagram 2. When the user asks a question in the dashboard the system turns the question into a vector. It can then assess similarities between the question and the results. If vector for of a question is close in the vector space to the vector form of a certain result, then it has a close similarity rating. The question along with the top three most similar results are parsed through OpenAI's API to ChatGPT. ChatGPT then uses the question and the relevant results to generate a response for the user.

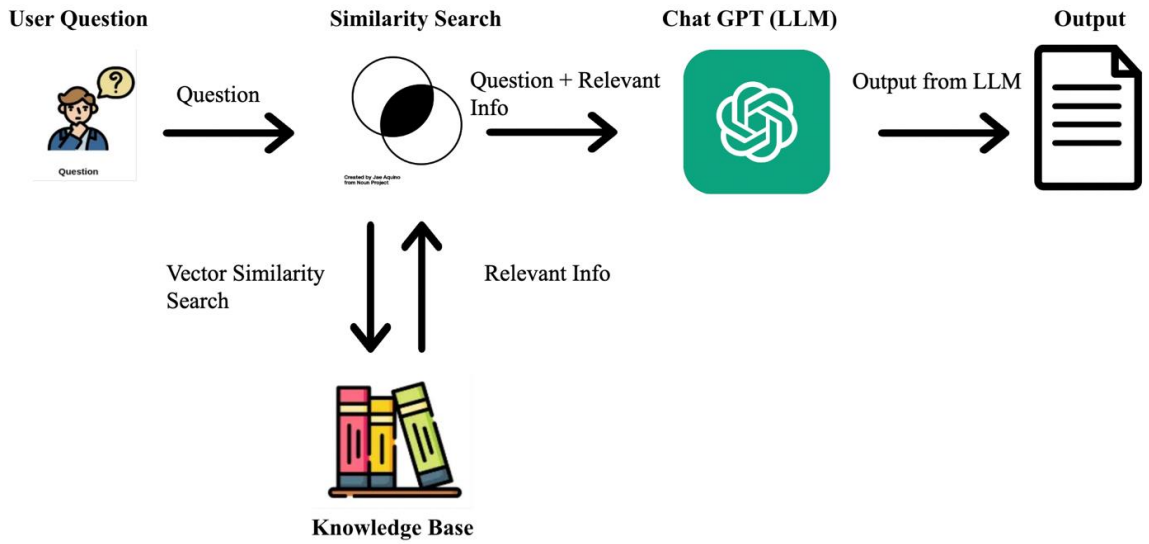


Diagram 2. *Diagram of Knowledge Embedding Process*

3.5 Dashboard

The Dashboard was implemented using Dash, a python framework by Plotly used to build web applications (Jean-Michel D, 2018). The application is locally hosted on a MacBook Air 2020. The interface has several interactive elements that allow users to input data and receive a response.
The first input field is where users can enter a LEGO set ID. Once the user press's a "Predict" button the code will access the results of the random forest algorithm and show the predicted percentage output for the given LEGO set. If the set ID that is entered is not available in the dataset it returns "Set ID not found in the dataset."

There is a textbox where the user can type any questions. This is where the knowledge embedding is used. Upon the user pressing a "Generate Response" button, knowledge embedding is used to create a response for the user.

The algorithm used to generate the percentage yearly growth prediction follows the exact same procedure as outlined in section 3.2. The knowledge embedding is likewise the same as described in section 3.4.

A diagram for the architecture of the dashboard is shown in Diagram 3. The dashboard integrates machine learning models, natural language processing and uses external data sources from AWS S3 buckets. The user interacts with the dashboard, this is detected by the local MacBook which hosts the dashboard and a request is sent to S3 to retrieve the relevant data. Analysis is then performed on the data and the output is displayed back to the user. This all happens in real time as the user is using the dashboard. To reduce latency in the system when the dashboard is initially set up the random forest algorithm is run. This means that when the user wants to generate a price prediction the algorithm does not have to be re-run. If the algorithm was rerun every time the user inputted a value it would take a lot of time to generate a response. With this implementation the output is generated for the user instantaneously.

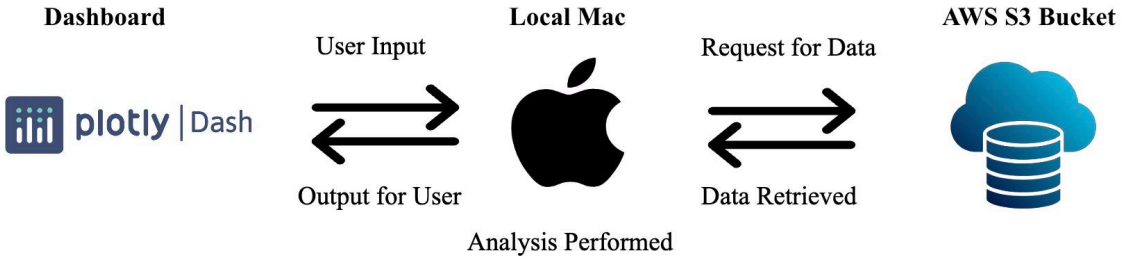


Diagram 3: *Diagram of End-to-End Application*

3.6 Ethics

There are minimal ethical concerns in this project due to the public nature of most of the data. The Rebrickable and Brickset datasets are readily available and downloadable. These datasets consist of structured and semi structured data with no personal or sensitive data. This means the sets can be used without ethical consideration. The acquisition of the Brickowl data was completed through web scraping which raises some ethical concerns. It is possible the websites terms and conditions were violated. The data once again contains no personal or sensitive data, merely data on LEGO sets. This means the data can be used for the project. To mitigate this ethical concern mostly Rebrickable and Brickset data were used to conduct the analysis. All the data used allowed us to comply with public licencing.

3.7 Residuals

The residuals were calculated as the difference between the actual and predicted values from the random forest model. Residual analysis was performed to determine any patterns in the errors. No significant outliers were removed, but heteroscedasticity was noted, where residual spread increased with the larger predictions.

Residuals were analysed using histograms, scatter plots and QQ plots. Random forest which produced tight residual distributions around zero was the method due its ability to capture nonlinear relationships in the data.

4 Results

4.1 Price Prediction

In this section, we present the results of our analysis for price prediction models to estimate the percentage growth of LEGO themes per year, what variables have the highest effects on prediction percentage growth on aftermarkets and if there are any apparent trends in the data. We begin by highlighting key insights generated from visual representations of the data, such as trends and patterns that influence our understanding of the LEGO market. Following that, we assess the performance of the chosen machine learning model, including its ability to anticipate previously unseen data outcomes. Finally, we explore the reasoning for selecting the Random Forest model, including its applicability for our dataset and the difficulties it may have. This thorough investigation provides a solid platform for making informed investment selections in the LEGO industry.

In Figure 1, we evaluated the performance of three different models, Linear Regression, Random Forest, and XGBoost—using features like pieces, theme, subtheme, retail price, minifigs, years in retirement, and launch year to predict the percentage increase per year for LEGO sets.

Linear Regression showed an MAE of 3.301, RMSE of 6.148, and R^2 of 0.342. This model struggles to fully capture the complexities of the relationships in the data, likely because it assumes a linear relationship between the features and the target variable, limiting its effectiveness in a dataset with nonlinear interactions.

Random Forest demonstrated the best performance, with an MAE of 0.605, RMSE of 1.975, and R^2 of 0.932. These scores indicate that Random Forest accurately captures the percentage increase per year, with the R^2 showing that it explains 93.2% of the variance in the data. Its ability to model nonlinear relationships makes it particularly effective for this task.

XGBoost also performed well, with an MAE of 1.908, RMSE of 3.394, and R^2 of 0.799. While slightly underperforming compared to Random Forest, XGBoost still demonstrates strong predictive power. Its ability to model complex interactions is reflected in its relatively low error metrics and high R^2.

Regarding feature importance, Random Forest identified retail price as the most important feature, with a high importance score of 8.245, indicating that sets with higher original prices tend to experience stronger percentage growth. Pieces were also significant, with an importance score of 1.909.

Years in retirement was another key feature, with an importance score of 1.705 in Random Forest, reinforcing the idea that sets that have been retired for longer often become more valuable due to scarcity and demand.

XGBoost placed more emphasis on specific themes, particularly Theme_City, with an importance score of 0.151, suggesting that certain themes contribute more significantly to percentage growth. However, it downplayed the importance of retail price and pieces compared to Random Forest, indicating that it captures growth trends differently across the data.

Model	MAE	RMSE	R^2
Linear Regression	3.301	6.148	0.342

Random Forest	0.605	1.975	0.932
XGBoost	1.908	3.394	0.799

Feature	Importance_RF	Importance_XGB
Pieces	1.909	0.003
Years_in_retirement	1.705	0.021
Retail_price	8.245	0.005
Theme_city	7.895	0.151
Minifigures	4.689	0.002

Figure 1: *Evaluating performance for three different machine learning methods with these specific features pieces, theme, subtheme, retail_price, minifigs, years_in_retirement and launch_year and feature importance scores for the best performing machine learning methods*

In Figure 2, we evaluated the performance of four machine learning models—Linear Regression, Random Forest, XGBoost and a baseline model—using features such as theme, subtheme, retail price, pieces, and minifigs to predict the percentage increase per year of LEGO sets. We dropped the extra features such as years in retirement from figure 1 as we believe these are the best features to predict the future percentage growth of LEGO sets.

The baseline model, which provides a reference point, resulted in an MAE of 4.223, RMSE of 7.585, and R^2 of -0.001. The negative R^2 shows that this model doesn't explain the variance in the data at all, making it ineffective for predictions.

Linear regression showed an MAE of 3.191 and an RMSE of 6.158 with an R^2 of 0.341. These scores indicate linear regressions is not unsuitable for predicting percentage growth of LEGO sets. Mainly because linear regression assumes a linear relationship within the features.

Random forest had a much stronger performance, achieving an MAE of 0.699 an RMSE of 2.074 and R^2 of 0.925. The massive drop from linear to random forest shows us that it was able to capture those nonlinear relationships between the features selected.

XGBoost had a slightly lower performance compared to random forest, with an MAE of 2.377, RMSE of 4.046 and R^2 of 0.715. The slight increase in both MAE and RMSE shows that it underperformed compared to random forest. The lower R^2 reflects that XGBoost did not handle the dataset as well as expected, although XGBoost typically excels in situations like this, however in this case It did not.

Both random forests and XGBoost the most important feature in predicting percentage growth per year, was pieces. In random forest pieces had an importance score of 3.417 and while the XGBoost

score is much lower at 0.004 it still plays an important role. Retail price was the second with random forest having a score of 1.272 and XGBoost having a score of 0.003, this indicates that retail price is not as influential as piece count in determining growth rate per year. Minifigures also played a role, especially in the random forest model with a score of 9.202.

Model	MAE	RMSE	R^2
Baseline Model	4.223	7.585	-0.001
Linear Regression	3.191	6.158	0.341
Random Forest	0.699	2.074	0.925
XGBoost	2.377	4.046	0.715

Feature	Importance_RF	Importance_XGB
Pieces	3.417	0.004
Retail_price	1.272	0.003
Minifigures	9.202	0.003
Theme_dimensions	2.616	0.062
Theme_city	2.342	0.028

Figure 2: *Evaluating performance for three different machine learning methods with specific features Theme, Subtheme, retail_price, pieces and minifigs and feature importance scores for the best performing machine learning methods*

In this analysis, a Random Forest model was used to predict the percentage growth of LEGO sets based on various features like pieces, retail price, minifigures, theme and subtheme. The model was trained on 80% of the data and tested on the remaining 20%. The evaluation metrics showed the following results:

Mean Absolute Error (MAE):	0.852
Root Mean Squared Error (RMSE):	2.23
R^2 Score:	0.913

These scores indicate that while the model captures some of the variance in the data, with an R^2 of 0.132, a significant portion of the variation in LEGO set growth rates remains unexplained. However, the model still provides insights into which features most influence the annual percentage growth of LEGO sets.

In figure 3, pieces emerged as the most important feature by large margin explaining about 30%. Which could indicate larger sets with large piece numbers appreciate the most in value over time. Retail price was the second most important feature explaining around 15%. This could mean the more expensive sets tend to increase overtime as well. Minifigures also played a role in predicting growth per year, this is due to some minifigures being quite unique and rare which could contribute to the growth of the set overtime.

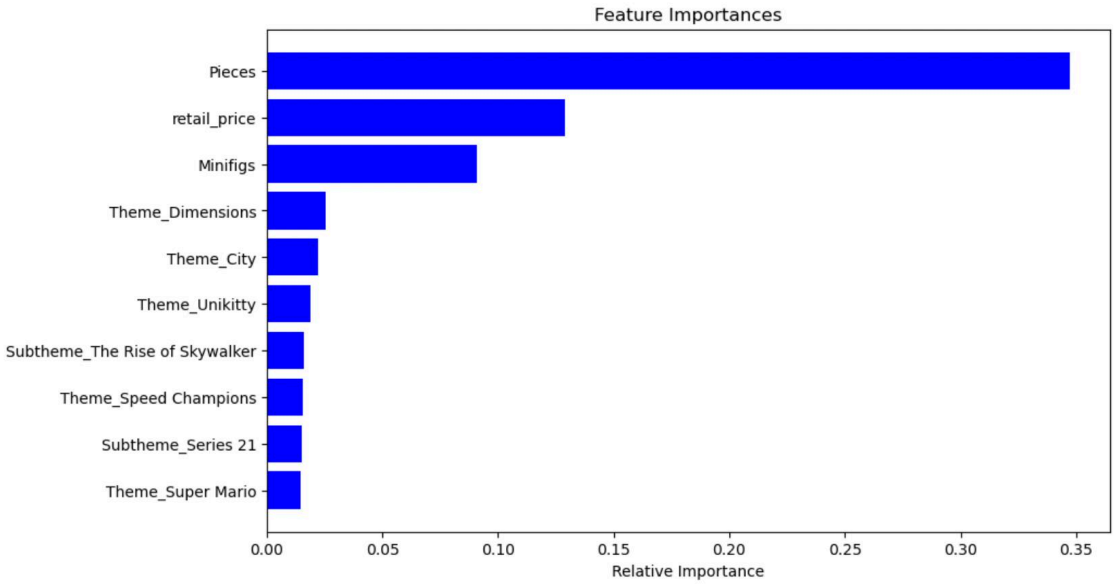


Figure 3: *Feature importance in predicting percentage growth for LEGO*

In figure 4, we observe the top 10 LEGO themes by their predicted percentage growth per year, which indicates how much these themes have increased in value over time.

Speed Champions takes the top spot with the highest average annual growth, followed closely by Jurassic World and Minecraft. These themes appear to consistently appreciate, suggesting strong demand in the resale market.

Other themes such as Creator Expert, Ideas, The Hobbit, The Lord of The Rings and Pirates of the Caribbean also exhibit notable growth, reinforcing their potential as solid investments for LEGO collectors.

Based on the RMSE of 0.276 the data means the predicted percentage growth is off by roughly 0.27% and the R^2 score of 0.998 means 99% of the variance in the data is explained. Which means this model is predicting percentage growths per year extremely well for themes.

However, it is important to consider the dataset's variability, as the themes Monkie Kid and Ghostbusters both have fewer than three entries in the data, which could skew their average percentage growth. Such a small sample size may not provide a reliable picture of long-term trends, and the high growth rates observed could be driven by outliers or specific sets rather than a consistent pattern across the theme.

MAE	0.075
RMSE	0.276
R^2	0.998

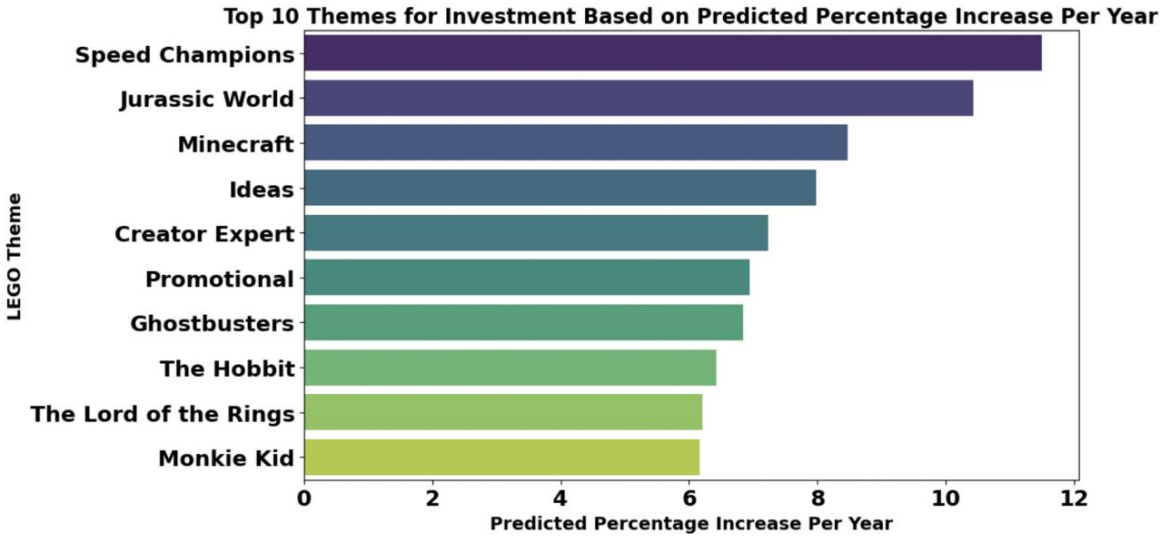


Figure 4: *Top 10 LEGO themes by predicted percentage growth per year*

In figure 5, we see the top 10 subthemes with the highest average percentage growth per year. As we can see this graph was dominated by speed champions with 4 out of the 10 subthemes belonging to speed champions. With those being Mclaren, Ford, Porsche, Cheverolet. Jurassic World and Ninjago both had multiple subthemes with those being Fallen Kingdom, Sons of Garamadon, Legend of Isla nubar, hunted and the hands of time.

Based on the RMSE of 0.408 the data means the predicted percentage growth is off by roughly 0.40% and the R^2 score of 0.997 means 99% of the variance in the data is explained. Which means this model predicts percentage growths per year extremely well for subthemes.

To ensure a more reliable analysis, subthemes with fewer than 10 entries were filtered out. This adjustment was necessary because, without this filter, all of the top 10 subthemes initially had less than 5 entries, which could have significantly skewed the results. By focusing only on subthemes with a larger number of entries, we obtain more dependable data, making the findings more representative of overall trends in the LEGO secondary market.

MAE	0.075
RMSE	0.408
R^2	0.997

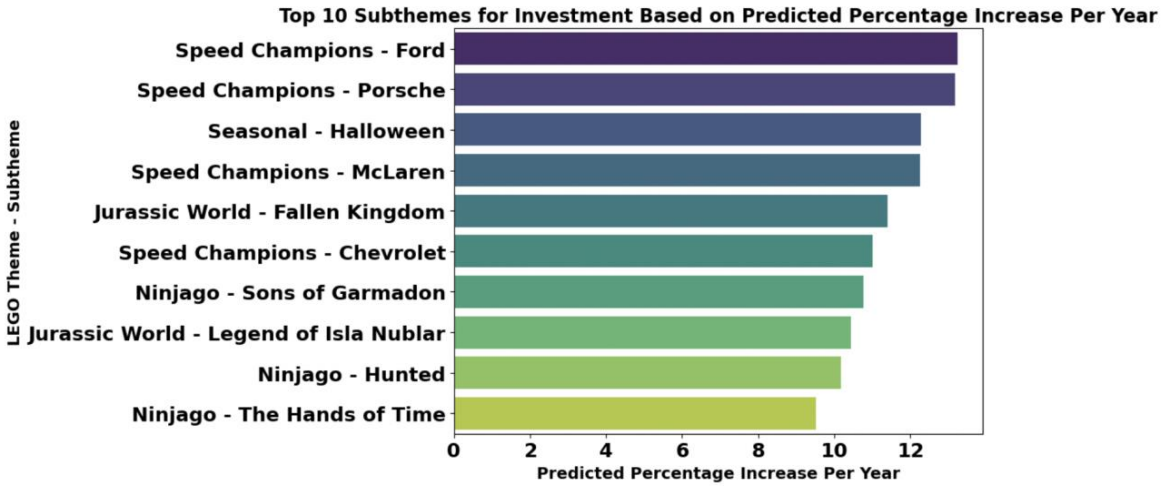


Figure 5: *Top 10 Lego subthemes by predicted percentage growth per year*

Cluster Analysis

This section focuses on the results of our clustering analysis and is aimed at identifying distinct investment profiles within the LEGO market. We analyse features like yearly price growth, resale price, number of pieces, and years in retirement to uncover which variables are most influential in distinguishing between different clusters of LEGO sets. The analysis includes a series of visualizations, which gives a clear overview of trends and patterns within the data, giving beneficial insights into how specific LEGO sets perform in the resale market. We will then evaluate the effectiveness of the KMeans algorithm in categorizing LEGO sets into meaningful clusters. We will also touch on the decision to use Kmeans, along with the challenges that came with it, and the methods explored to determine how many cluster groups we will use.

From figure 6 and 7 below, we can see that there are some clear differentiation factors between our clusters. Figure 6 shows the relationship between the number of pieces in a LEGO set and its retail price in USD, with points color-coded to represent the different clusters identified through the Kmeans algorithm. Figure 7 shows general summary statistics for each cluster. We can see that cluster 2 is dominated by high-priced and large-piece sets, and clearly stands out with sets that have an average of 1402 pieces, a mean resale value of $412.16 USD, and a retail price of $140.34 USD. Cluster 1 and 3 look to have relatively similar piece counts at around 200 and retail prices at around $26, however, cluster three has a significantly higher average resale price at $109.23 compared to cluster 1's $43.74. This could potentially mean that cluster 1 and 3 mostly include more generally affordable sets, with decent investment potential, though not as premium as cluster 2.

Figure 6: *Scatter Plot of Retail prices against pieces per cent, by cluster.*

Cluster Name	Average Price Growth (%)	Average Resale Price (USD)	Average Pieces	Average Years in Retirement	Average Retail Price
Cluster 1: Low Price, Moderate Growth	4.08%	$43.74	203.38	6.15	$26.25
Cluster 2: Large Premium Sets, High Growth	6.90%	$412.16	1401.82	8.55	$140.34
Cluster 3: Retired Collectibles, Modest Growth	3.81%	$109.23	207.67	18.66	$27.96

Figure 7*: Summary Statistics Per Cluster*

From figure 8, we can see how the distribution of top themes varies across clusters. Cluster 1 shows a more balanced distribution of themes, indicating that sets from a wide variety of themes are present here. This cluster seems to have a mix of sets from both popular themes and more niche ones. In Cluster 0, Collectable Minifigures dominate with 488 sets, followed by City (357 sets) and Star Wars (347 sets). These themes are widely popular and probably attract a wide range of collectors and investors. Cluster 1 looks to be dominated by Star Wars, along with a small portion of sets from Collectable Minifigures, Marvel Super Heroes and Harry Potter. Like Cluster 1, Cluster 3 also has a relatively even distribution of themes, however with the majority being from Star Wars, and Town. These sets might not see as much price growth as others, for reasons that will be discussed later in the report, but they continue to be popular for buyers.

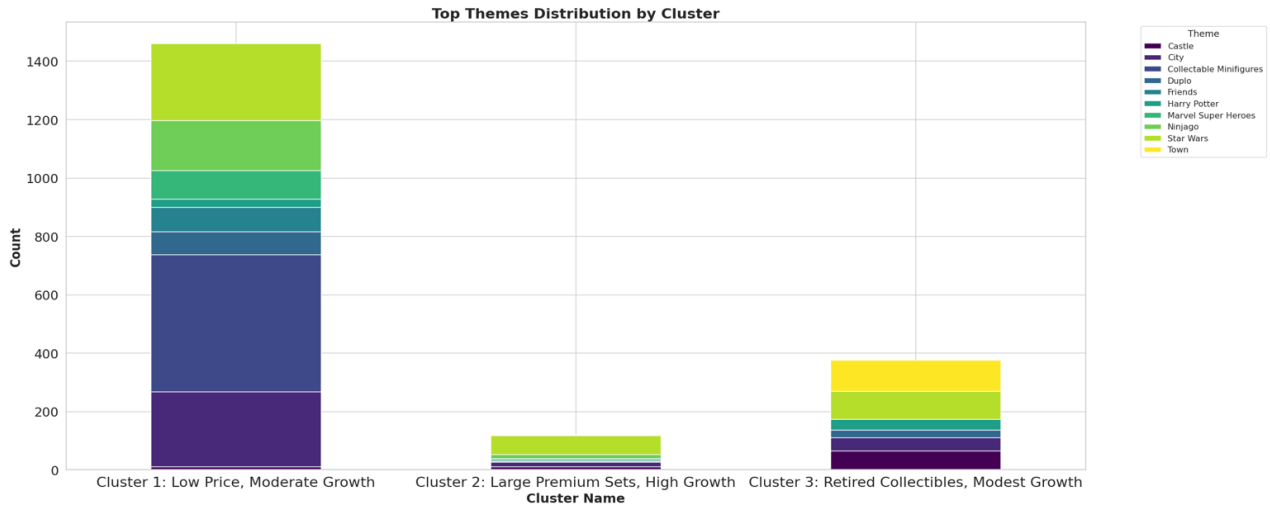


Figure 8: *Stacked Bar Chart of Theme Distribution By Cluster*

Figure 9 shows the price growth percentage per year for each cluster against each other. Figure 10 shows the average price growth per year against years in retirement for each cluster. Figure 9 clearly shows that Cluster 2 has the highest price growth per year at 6.9%, which is close to double cluster 1 and 3, at 4.08% and 3.81%, respectively. As seen earlier in figure 7, the relatively large and premium nature of the sets within cluster 2 could be driving this. Star Wars, Harry Potter, and Marvel Super Heroes are examples of themes that drive this extreme growth, suggesting they are highly sought after in the resale market despite their significantly higher prices. Figure 10 also shows that sets within this cluster have been in retirement for an average of 8.55 years. Cluster 1, with a 4.08% price growth per year, comes second. As seen in figure 9, sets in cluster 1 have on average, been in retirement for the least amount of time at 6.15 years. Cluster 3, on the other hand, with 3.81% yearly price growth, mainly includes sets that have been in retirement for a very long time at an average of 18.66 years.

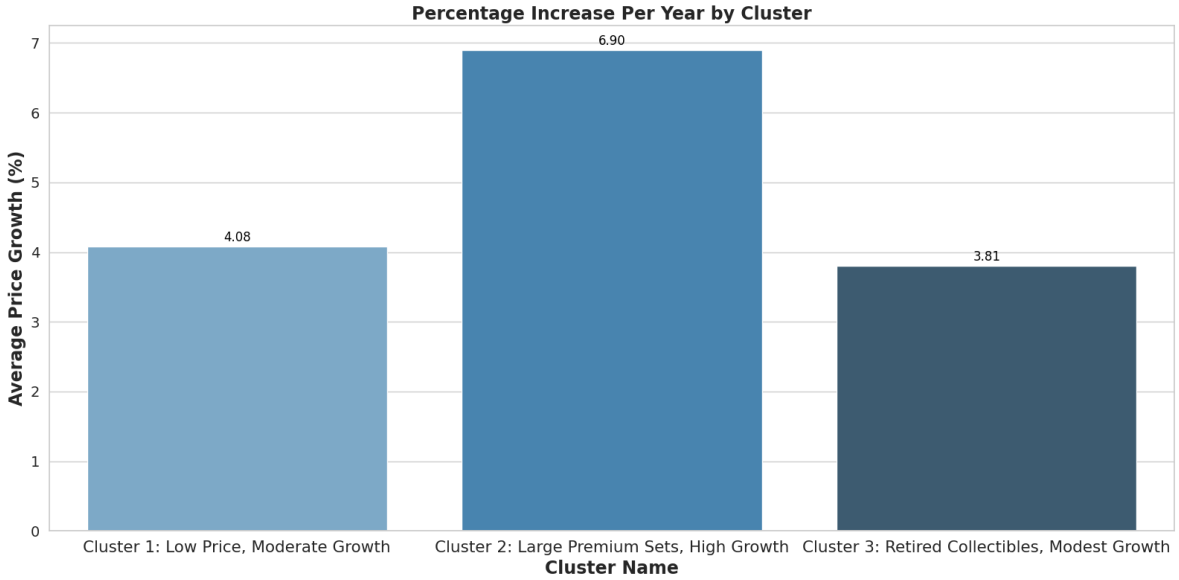


Figure 9: *Percentage Increase Per Year By Cluster*

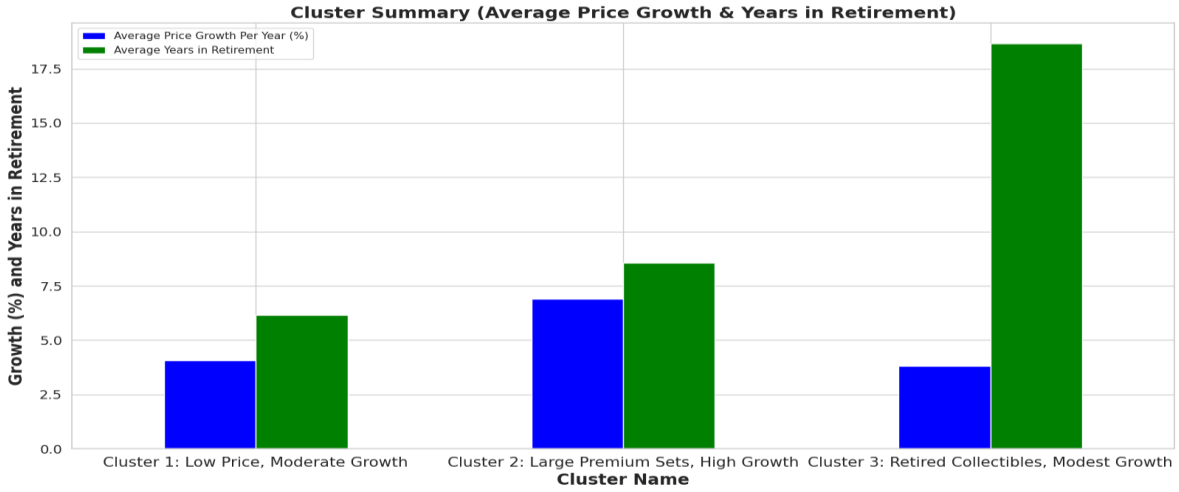


Figure 10: *Average Price Growth Per Year Against Years In Retirement By Cluster*

As certain clusters contained disproportionately larger amounts of data than others, we wanted to get a general measure of which themes performed the best across all our clusters. From Figure 11, we can see that the majority of the top-performing themes seem to belong to either popular TV shows, movies, and video games. For example, we can see that the theme 'Speed Champions' dominates with what looks to be around 13% annual price growth, followed closely by 'Jurassic World' and 'Minecraft'. Looking through the rest of the themes on the left we also see other fan favourites such as 'Star Wars' and 'The Lord Of The Rings'. Overall, I believe that cluster analysis performed very well on our data and did a good job of segmenting our clusters based on unique, and easily understandable characteristics.

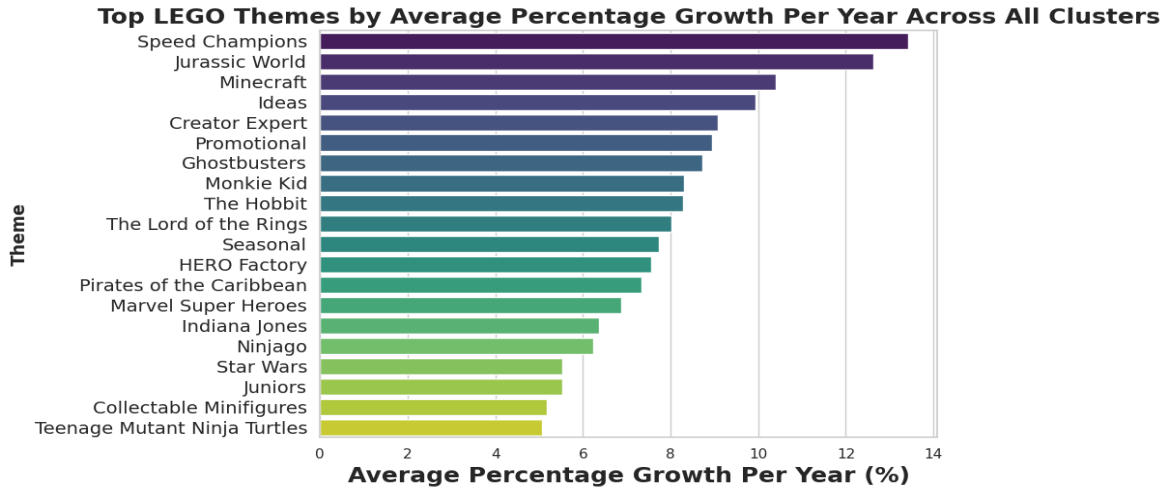


Figure 11: *Average Percentage Growth for Individual Themes Per Year Across All Clusters*

4.5 Dashboard and Knowledge Embedding

The dashboard was successfully implemented using Dash as seen in figure 12. The user has inputted the 'set_number' 1817-1 and the system has generated a prediction of 2.57%. This provides users with an interactive, real-time and seamless experience. This allows them to get a rough idea of how well a certain set will perform in the future allowing them to make more accurate predictions. When a user enters a set number that is not in the database the response "Set ID is not valid for predictions" is generated for the user, as illustrated in figure 12.

As well as implementing a price prediction tool the dashboard has a text box where the user can ask questions. This is where the knowledge embedding is used. If the user asked a normal ChatGPT model what the average number of pieces per set for the year 2019 was, it would have no idea. Throughout the process of knowledge embedding described in section 3.4 an accurate answer can be generated for the user.

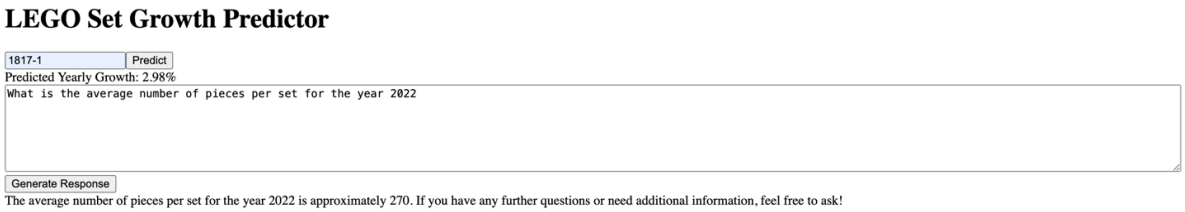


Figure 12: *Screenshot of Dashboard Implementation, Price Predictor*

The dashboard operates as a fully end to end system. User input triggers a sequence of actions that acquire relevant data stored in AWS S3 buckets, transferring the necessary variables into the model

before generating the output. The design ensures smooth data retrieval processing and prediction. The price prediction is generated instantly, dash analytics estimate that it takes around seven milliseconds to generate a price prediction and around one second to generate a response using knowledge embedding. This is extremely fast and allows the user to have a positive experience.

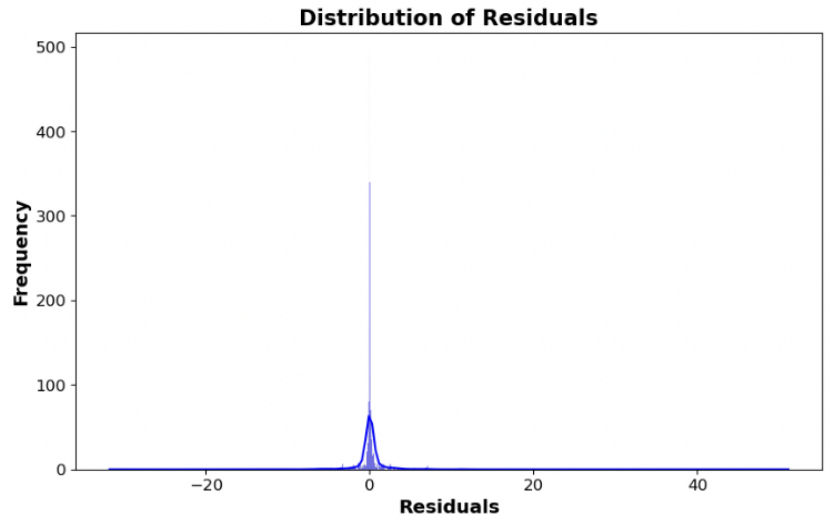
4.6 Residuals

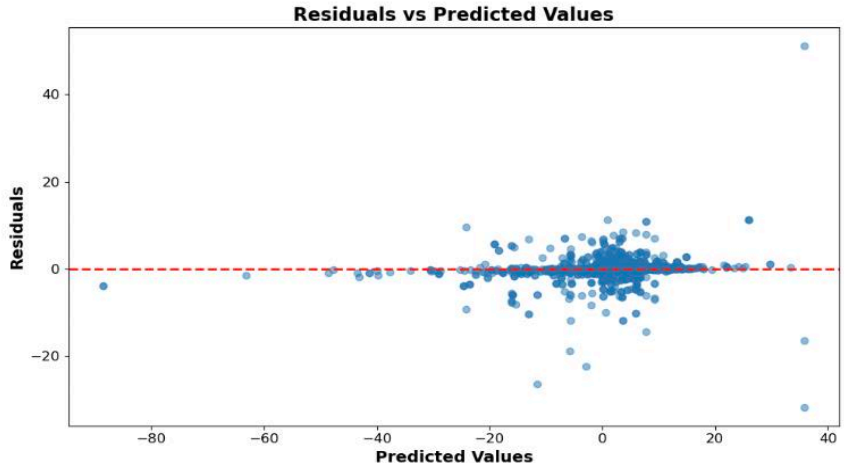
In figure 13, we can see the residuals plotted against our predicted values.

Residual Distibution: The residuals are tightly centred around 0, as shown in the residual histogram. This suggests that the model's predictions are relatively accurate, but there are some outliers, likely affecting the overall spread.

Residuals vs Predicted Values: The scatter plot reveals some heteroscedasticity, meaning the spread of residuals increases as the predicted percentage increases, meaning the model might underperform for those large predictions

QQ Plot: the QQ plot shows some deviation from normality, especially on both ends at the tails. Which indicates that the residuals are not perfectly normally distributed, which may affect the assumptions of certain statistical test. However, this is not a major issue for tree-based models like random forest.


Distribution of Residuals
Frequency
Residuals
500
400
300
200
100
0
−20
0
20
40


Residuals vs Predicted Values
Residuals
Predicted Values
40
20
0
−20
−80
−60
−40
−20
0
20
40

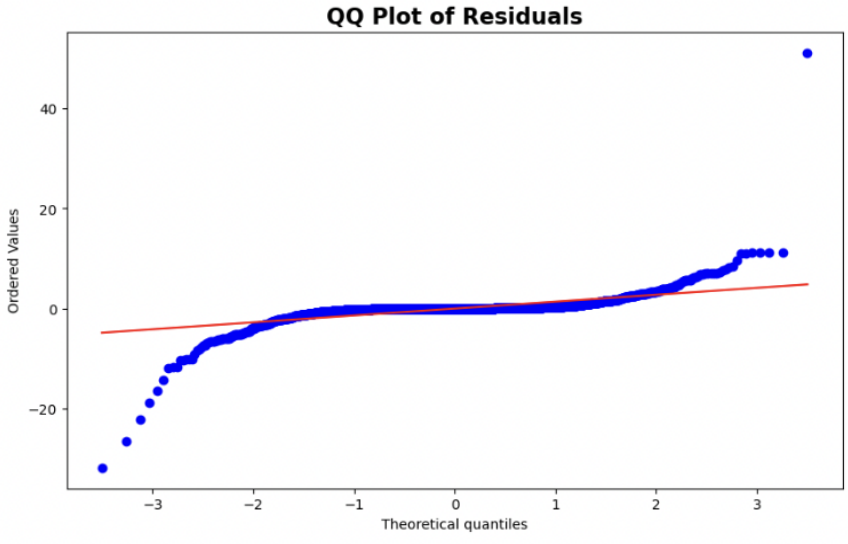

QQ Plot of Residuals
Ordered Values
Theoretical quantiles
40
20
0
−20
−3
−2
−1
0
1
2
3

Figure 13: *Distribution of residuals for predicted percentage growth*

5 Conclusion

5.1 General Conclusions

Throughout our report, we believe that we showed enough evidence, reinforced with statistics, to highlight which factors have the most effect on a LEGO set's investment potential. Through our clustering, we were able to highlight which characteristics drive higher yearly growth rates. From these were the theme a set belongs to, and its size (piece count). Cluster 2, which dominated in terms of yearly growth, primarily contained sets with large piece counts with an average of 1402 pieces per set. Along with this, the majority of sets within this cluster belonged to themes with large fan bases such as Star Wars and Marvel Superheroes. These findings go hand in hand with nerdcube.ue's LEGO investment guide (*Investing in LEGO: The Ultimate Guide for Collectors and Investors*, 2023). The guide states that special sets dedicated to iconic films or books, tend to experience the highest growth on the secondary market. Along with this, it also states that huge sets, with larger piece counts are more profitable than medium-sized sets, which again, goes hand in hand with our clustering analysis.

Along with our clustering analysis, our price prediction models also clearly highlight which factors influence a LEGO set's investment potential. Our model's results show a predicted annual growth of up to 12% for subthemes such as Speed Champions - Ford (Figure 5) and 12% growth for the Speed Champions theme (Figure 4) we also saw similar high growth predictions across various LEGO themes and subthemes, aligning closely with previous research done by others. Studies like Shanaev et al. (2019) reported annual LEGO investment returns averaging around 8-12%, supporting the statistical validity of our predictions. This demonstrates that our model is statistically meaningful and effectively captures trends in the LEGO market, making it valuable for LEGO investors seeking high-growth potential sets.

Additionally, our model's high R^2 scores of 0.998 for themes and 0.997 for subthemes (Figures 4 and 5) further reinforce the confidence we have in these findings, as they reflect nearly perfect alignment between predicted and actual data. Comparing this with broader alternative investments like art, which Renneboog and Spaenjers (2012) also explored, we found LEGO to offer compelling and stable growth rates that compete well with traditional investment such as stocks and real estate.

These strong predictions done through price prediction models and cluster analysis reflect that themes such as Speed Champions, Jurassic World, Minecraft, ideas and subthemes such as Speed Champions – Ford, Speed Champions – Porsche, Jurassic World – Fallen Kingdom offer significant returns, making them excellent candidates for investment portfolios.

One of the primary objectives of this project was to create a scalable application that integrated both a front-end and back-end system. This goal was achieved through the effective use of AWS and Dash. AWS was used for cloud storage allowing all the data to be readily accessed for analysis and for the dashboard. This system could accommodate larger datasets than were used for this project meaning the application is sailable.

The front-end of the application was developed using Dash. This allowed for the creation of an interactive web application using Python. The user could seamlessly use the price prediction model, and the knowledge embedding developed to generate insights. The combinate of the AWS back-end and Dash allowed for a very smooth interface which would almost instantly generate responses.

No one in the group has experience working with either AWS or Dash. This meant there was a steep learning curve to obtain the skills needed to implement this application. Through the study of best practises for AWS we were able to create a robust data pipeline using an Extract Transform Load process. The dashboard was simple in nature but very functional. The experience of learning this tool has well equipped us to create end-to-end applications in the future. This is detrimental for data science jobs in the industry.

5.2 Future work

A primary direction for future work would be to refine the price prediction model. This could be achieved in a number of different ways. Firstly, it would be effective to create a month-by-month price prediction so investors could more accurately see how a LEGO set would increase or decrease in value in the future. Additionally, it would be wise to consider a broader range of predictors (features) in future models. Currently the bulk of the prediction from piece count, minifigure count, retail price, theme and subtheme. Considering in more detail exactly what pieces were contained in each set could lead to a more accurate prediction. This price prediction model could be extended to create a more informative investing "suite". This suite could contain information about sets that are going to retire soon and sales of sets. This would allow investors to buy the set at a lower price, likely instantly increasing the return on investment of the set

Given that AWS is used for data storage, the system is highly scalable. Future work could leverage this scalability to process larger datasets and handle live data streams. For instance, live data could be acquired on when sets go on sale and the live resale prices of sets. To improve the knowledge embedding process a vector database could be created. Currently the knowledge base mainly consists of one large CSV that is then vectorised ready to be search. This is somewhat inefficiently as there are a lot of missing values. The similarity search still performs well because overall the CSV is quite small, however as more data is used this could slow down the speed in which a response is generated for the user.

Currently the Dash dashboard is user friendly and very functional however it is not visually appealing. Future work could improve on the visual elements of the dashboard to make it more inviting for users. Aesthetically pleasing designs can significantly impact user engagement and satisfaction. Incorporation modern design principles such as a colour scheme, good typography and a nicer layout could all lead to a more visually appealing dashboard (Eline Jongmans et al., 2022). Ideally the theme of the dashboard would be colourful as this would resonate well with LEGOs branding which is visually striking.

5.3 Problems

Some issues we encountered with predicting percentage growth for subthemes is that without filtering the data to ensure a minimum number of entries the results were skewed significantly towards those subthemes with very minimal entries. To ensure that this did not affect our results we

had to only use subthemes that had more than 10 entries in our dataset, which provided much more meaningful results in our graph which can be seen in figure 5.

In our initial project objectives, one of our primary goals was to conduct a sentiment analysis by collecting customer reviews from various sources. At the beginning of the project, we successfully implemented a script to scrape Amazon reviews for LEGO sets. However, after a period of time, the script had stopped working, and despite multiple attempts, we were unable to resolve the issue. Consequently, we explored alternative platforms that could provide customer reviews. One promising site we identified was Brick Insights. Unfortunately, all their data was accessible only through an API, and despite reaching out to them over several weeks, we received no response. As a result, we were forced to discontinue the sentiment analysis aspect of our project.

Another constraint is the limited understanding of AWS and Dash within the group. Both AWS and Dash are complicated services requiring a large amount of understanding to implement. Furthermore, as a group we have a limited understanding of the LEGO product line, themes and market trends which increased the amount of time required to understand the results of the analysis.

5.4 Ethics

Our data sources—Rebrickable, Brickowl and Brickset provided publicly available information through downloadable CSVs or public API keys. This data contained various data about LEGO such as retail price, resale value, piece counts and minifigure counts. None of the data contained any sensitive or personal information, ensuring that that were no privacy concerns or ethical issues. By using public data, we maintained ethical integrity throughout the project.

6 References

AI Jason. (2023, July 26). *"How to give GPT my business knowledge?" - Knowledge embedding 101*. YouTube. https://www.youtube.com/watch?v=c_nCjlSB1Zk

Brick Owl | Brick Owl - LEGO Marketplace. (2024). Brickowl.com; Brick Owl. https://www.brickowl.com/

Brickset home page. (n.d.). Brickset.com. https://brickset.com/

Dobrynskaya, V., & Kishilova, J. (2022). LEGO: THE TOY OF SMART INVESTORS. *Research in International Business and Finance*, *59*(1), 101539. https://doi.org/10.1016/j.ribaf.2021.101539

Eline Jongmans, Jeannot, F., Liang, L., & Damperat, M. (2022, July). *Impact of website visual design on user experience and website evaluation: the sequential mediating roles...* ResearchGate; Taylor & Francis. https://www.researchgate.net/publication/361717406_Impact_of_website_visual_design_on_user_experience_and_website_evaluation_the_sequential_mediating_roles_of_usability_and_pleasure

Fortune Business Insights. (2021, December). *Toys Market Size, Share & Growth | Global Industry Trends [2027]*. Www.fortunebusinessinsights.com. https://www.fortunebusinessinsights.com/toys-market-104699

Hiba, J., Shnain, A., Hadishhaheed, S., & Haji, A. (2015, January 1). *(PDF) BIG DATA AND FIVE V'S CHARACTERISTICS*. ResearchGate. https://www.researchgate.net/publication/332230305_BIG_DATA_AND_FIVE_V

Jean-Michel D. (2018, February 26). *Building a dashboard with Dash (plotly), AWS and Heroku*. Medium; Towards Data Science. https://towardsdatascience.com/building-a-dashboard-with-dash-plotly-aws-and-heroku-jean-michel-d-d102e26ac8c1

Linjuan, F., Yongyong, S., Fei, X., & Hnghang, Z. (2022). Knowledge Graph Embedding Based on Semantic Hierarchy. *Cognitive Robotics*. https://doi.org/10.1016/j.cogr.2022.06.002

Rebrickable | Rebrickable - Build with LEGO. (n.d.). Rebrickable.com. https://rebrickable.com/

Sarker, I. H. (2021). Machine Learning: Algorithms, Real-World Applications and Research Directions. *SN Computer Science*, *2*(3), 1–21. Springer. https://doi.org/10.1007/s42979-021-00592-x

Singh, G. (2024, January 1). *A REVIEW PAPER ON AWS*. https://www.researchgate.net/publication/377339733_A_REVIEW_PAPER_ON_AWS

Souibgui, M., Atigui, F., Zammali, S., Cherfi, S., & Yahia, S. B. (2019). Data quality in ETL process: A preliminary study. *Procedia Computer Science*, *159*, 676–687. ScienceDirect. https://doi.org/10.1016/j.procs.2019.09.223

XGBoost Documentation. (2024). *XGBoost Documentation*. https://xgboost.readthedocs.io/en/stable/

EUMeTrain. (n.d.). *Variation of convective cloud features*. EUMeTrain. https://resources.eumetrain.org/data/4/451/english/msg/ver_cont_var/uos3/uos3_ko1.htm

Tosic, M. (2021). *michaeltosic/lego-analysis: Notebook that uses data analysis methods to analyze the history of LEGO sets between 1991 and 2021*. GitHub. https://github.com/michaeltosic/lego-analysis/tree/master

(2024). Fxratesapi.com; FXRatesAPI. http://fxratesapi.com

Samuels, J. I. (2024, January 5). *One-Hot Encoding and Two-Hot Encoding: An Introduction*. Https://Www.researchgate.net/. https://doi.org/10.13140/RG.2.2.21459.76327

Shanaev, S., Shuraeva, A., & Ghimire, B. (2019). *The Rise of the Collectibles Market: Evidence from LEGO Sets. SSRN.* https://ssrn.com/abstract=3508756

Renneboog, L., & Spaenjers, C. (2012). *Hard Assets: The Returns on Rare Diamonds and Gems. The Quarterly Journal of Finance, 2(04), 1250014.* https://doi.org/10.1142/S2010139212500144

Investing in LEGO: The Ultimate Guide for Collectors and Investors. (2023, May 8). NerdCube. https://www.nerdcube.eu/guides/investing-in-lego/