3. CHAPTER 3

RESEARCH METHODOLOGY

3.1. Overview

This chapter outlines the methodological approach undertaken to investigate the factors influencing BSR and to develop predictive models based on structured product data. The methodology is carefully designed to align with the study's research objectives and to address the gaps identified in the literature review.

The dataset used in this study comprises real-time Amazon Best Sellers data, specifically focusing on the Software product category. It includes records from multiple countries and categories, reflecting diverse market conditions. The data was collected using Python-based web scraping techniques and contains key product attributes such as title, price, star rating, number of reviews, and BSR. These variables were selected for their potential influence on product visibility and consumer behaviour.

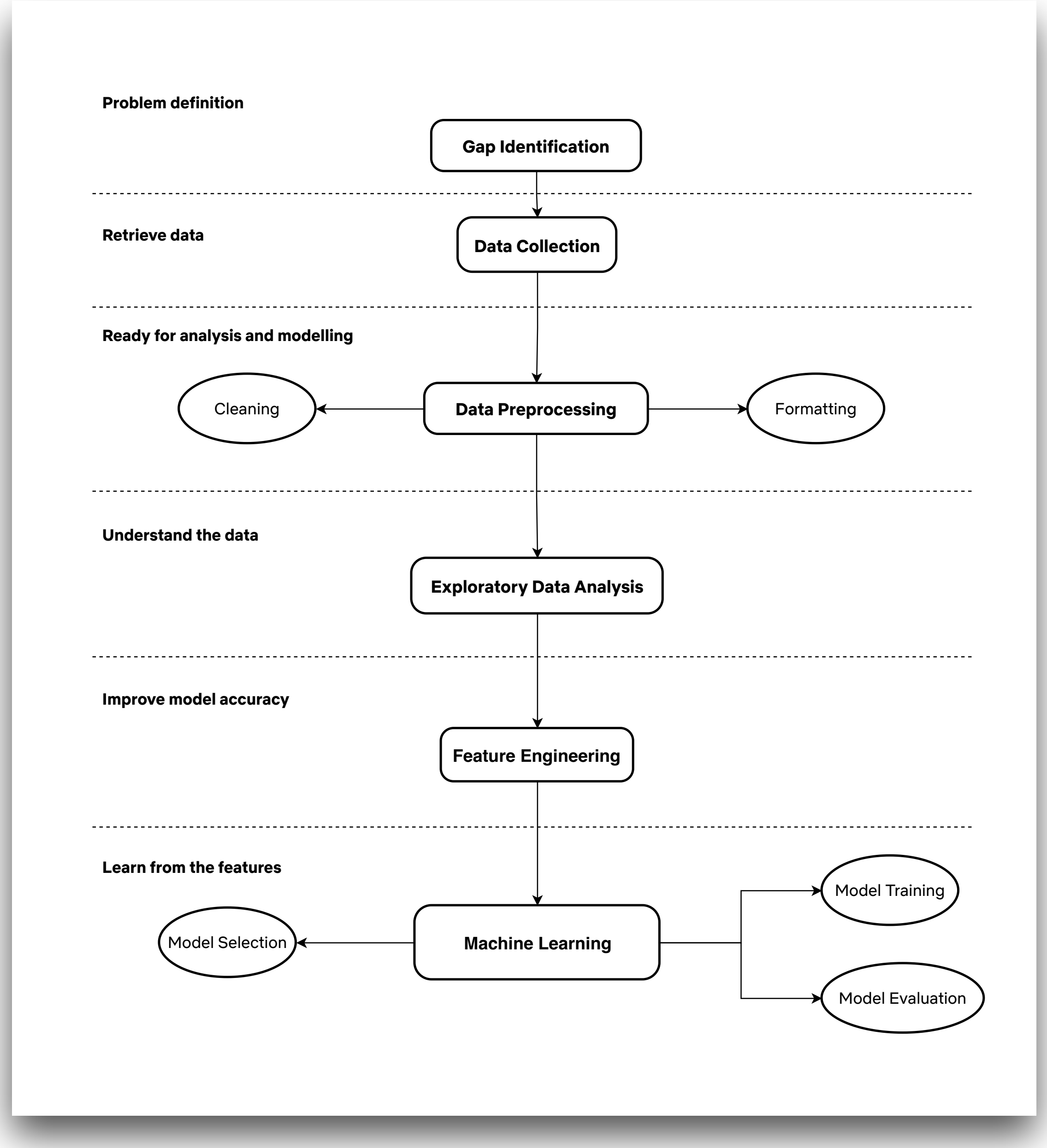
The methodology includes several sequential stages: selecting a suitable research design, collecting and describing the dataset, cleaning and preprocessing the data, engineering relevant features, building and evaluating machine learning models, and selecting appropriate tools and metrics for analysis. Each stage is intended to ensure that the model is not only statistically sound but also practically relevant for real-world applications, particularly for sellers and analysts seeking to optimise product visibility on Amazon.

3.2. Research Design

The research adopts a quantitative, data-driven design that centres on predictive modelling using structured product-level data. Given the nature of the problem—forecasting

BSR based on measurable product attributes—a quantitative approach is most appropriate, as it enables statistical evaluation and comparison of model performance.

This study follows a supervised machine learning framework, wherein BSR serves as the target variable (dependent variable), and features such as product price, star rating, number of reviews, and engineered variables act as predictors (independent variables). The objective is to uncover patterns and relationships between these features and the product's BSR, which is updated dynamically on Amazon's platform.



To ensure robustness, the research design incorporates the following key components:

- Exploratory Data Analysis (EDA) to assess variable distributions, detect anomalies, and understand inter-feature correlations.

- Feature Engineering to enhance model performance by creating new, informative features from the raw dataset.

- Model Comparison between linear and non-linear algorithms—including Linear Regression, Decision Tree Regressor, and Random Forest Regressor—to determine which method best captures the complexity of the data.

- Evaluation Metrics such as R^2, RMSE, and MAE to quantitatively assess model accuracy and reliability.

This design enables the researcher to not only build a predictive model but also to critically evaluate its performance, generalisability, and interpretability—key aspects that address the practical needs of Amazon sellers and platform analysts.

3.3. Data Collection

The data used in this study was collected through Python-based web scraping from Amazon's public Best Sellers listings, with a specific focus on the Software category. The scraping process was automated to extract real-time information from product detail pages across multiple Amazon marketplaces.

The dataset includes structured product-level attributes that are publicly visible and potentially influential in determining Best Seller Rank (BSR). Key variables collected include:

- Product Title

- Price

- Star Rating

- Number of Reviews

- Country of Marketplact

- Best Seller Rank (BSR)

This method of data collection ensures the dataset is both current and representative of real-world product rankings. All data was obtained from publicly accessible pages for academic purposes, without violating Amazon’s terms of service.

3.4. Dataset Description

The dataset used in this research comprises structured, real-time information on Amazon Best Sellers, specifically within the Software category. The data was collected via Python-based web scraping and includes listings from the Amazon marketplaces in multiple countries, allowing for comparative analysis across regions. A total of 2,423 product records were initially gathered, covering key attributes that are visible to consumers and potentially influential in Amazon’s ranking algorithm.

The dataset contains the following core features:

Variables	Description
Product Title	the name or description of the product
Product Price	listed price at the time of scraping
Star Rating	average customer review score
Number of Reviews	total count of submitted customer ratings

Country	the Amazon marketplace region (e.g. US, UK)
Rank	the product’s relative position within its category, which serves as the target variable for prediction

After preprocessing and the removal of rows with missing or inconsistent data, the final cleaned dataset used for model training consisted of approximately 2,000 complete records. This refined dataset provided a reliable foundation for exploratory analysis, feature engineering, and the development of predictive models.

3.5. Data Preprocessing

Before applying machine learning models, several preprocessing steps were conducted to ensure the quality, consistency, and suitability of the dataset for analysis. This process involved data cleaning, transformation, and preparation of variables for feature engineering and modelling

3.5.1. Handling Missing Values

The initial dataset contained missing values in key columns such as product_price, product_star_rating, and product_num_ratings. Records with missing values in any of these essential fields were removed to maintain data integrity. After cleaning, approximately 2,000 complete records remained.

3.5.2. Data Type Conversion

The product_price column initially included currency symbols and non-numeric characters. These were removed using regular expressions, and the values were converted to floating-point numbers. Similarly, columns such as product_star_rating, product_num_ratings, and rank were explicitly cast to numerical types to support mathematical operations.

3.5.3. Outlier Handling

To avoid distortion in visualisation and statistical analysis, products with extreme price values (e.g. over $200) were excluded during exploratory visualisation stages. However, the full range of prices was retained for modelling to preserve the generality of the dataset.

3.5.4. Feature Transformation

To address skewness in the distribution of review counts, a logarithmic transformation was applied using the log1p function, which computes log(1 + x). This transformation improved the stability and scaling of the feature for regression tasks.

These preprocessing steps ensured that the dataset was clean, numerically consistent, and ready for feature engineering and model development. They also reduced the risk of biased or unreliable results due to missing, malformed, or unscaled data.

3.5.5. Encoding Categorical Variables

The categorical variable country was converted into a numerical format using one-hot encoding. This transformation created binary indicators for each country, allowing the regression models to learn region-specific effects without introducing ordinal relationships. The encoding process excluded the first category to prevent multicollinearity. This step was essential for incorporating geographic context into the model without distorting the numeric relationships among features.

3.6. Modelling Approaches

To predict Amazon Best Seller Rank (BSR), this study adopts a supervised regression modelling approach, using both linear and non-linear algorithms to explore the relationships between structured product attributes and sales rank. The aim is to determine which machine learning models can most accurately capture the complex patterns within the data and deliver reliable BSR predictions.

3.6.1. Linear Regression

Linear Regression was used as a baseline model due to its simplicity and interpretability. It assumes a linear relationship between the input features—such as price, star rating, and number of reviews—and the target variable (BSR). While it provides insight into the directional influence of each feature, its performance is limited in datasets where relationships are non-linear or involve interactions between variables.

3.6.1. Decision Tree Regressor

The Decision Tree Regressor was implemented to capture non-linear patterns and conditional relationships in the data. This model splits the dataset into decision nodes based on feature values, allowing for flexible modelling of complex feature-target dynamics. Decision trees are intuitive and visually interpretable, but they are prone to overfitting if not properly tuned or pruned.

3.6.2. Random Forest Regressor

Random Forest, an ensemble learning method, was selected to improve predictive accuracy and robustness. It constructs multiple decision trees and averages their predictions, reducing the risk of overfitting and improving generalisability. This model also provides insights into feature importance, making it suitable for both prediction and interpretability.

Each model was trained on the same dataset using a standard 80/20 train-test split to ensure fair comparison. The models were evaluated using regression metrics including R^2, RMSE, and MAE, as detailed in Section 3.7. This multi-model approach allows for a comprehensive understanding of the modelling landscape and the identification of the most effective algorithm for BSR prediction.

3.7. Evaluation Metrics

To assess the performance of the machine learning models developed in this study, three standard evaluation metrics for regression tasks were used: R-squared (R^2), Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE). These metrics provide complementary insights into how accurately the models predict Amazon Best Seller Rank (BSR) based on product attributes.

3.7.1. R-squared (R^2)

R^2 shows how much of the variation in BSR can be explained by the model. A value closer to 1 means the model fits the data well, while a value near 0 means it does not explain much. It's useful for judging the overall fit of the model.

$$R^2 = 1 - \frac{\sum (y_i - \hat{y}_i)^2}{\sum (y_i - \bar{y})^2}$$

3.7.2. Root Mean Squared Error (RMSE)

RMSE measures the average size of the prediction errors, giving more weight to larger errors. It tells us how far off the predictions are, on average, in the same units as BSR. A lower RMSE indicates better performance.

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2}$$

3.7.3. Mean Absolute Error (MAE)

MAE calculates the average of the absolute differences between the predicted and actual values. Unlike RMSE, it treats all errors equally. It gives a straightforward idea of how far off the predictions are, on average.

$$MAE = \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y}_i|$$

These metrics were chosen to evaluate the models from multiple perspectives: R^2 for overall model fit, RMSE for penalising large errors, and MAE for interpretability and robustness. The combination of these metrics ensures a comprehensive evaluation of model performance and supports fair comparison across different algorithms.

3.8. Tools and Technologiies

This research employs a range of programming tools and data science libraries to support data collection, preprocessing, analysis, modelling, and visualisation. All development and experimentation were conducted in a Python-based environment, which offers flexibility, scalability, and strong community support for machine learning tasks.

3.8.1. Programming Language

- Python 3.11 was used as the core programming language due to its extensive support for data analysis, machine learning, and web scraping. Python’s readable syntax and mature ecosystem make it particularly well-suited for applied data science research.

3.8.2. Integrated Development Environment (IDE)

- PyCharm was used for writing, testing, and debugging code. It provides an efficient interface for managing Python projects and integrating version control and virtual environments.

3.8.3. Libraries and Frameworks

- pandas and numpy were used for data manipulation, cleaning, and numerical operations.

- matplotlib and seaborn were employed for data visualisation and exploratory analysis.

- scikit-learn served as the primary machine learning library, providing access to a wide range of algorithms and evaluation tools, including:

- LinearRegression

- DecisionTreeRegressor

- RandomForestRegressor
- Performance metrics such as r2_score, mean_absolute_error, and mean_squared_error

3.8.4. Web Scraping

Python-based scraping tools were used to collect real-time Amazon product data. Though the specific module (e.g. requests, BeautifulSoup, or Selenium) is not listed in the script, a custom scraping script was developed to extract structured product-level information from public listings.

3.8.5. Hardware and Runtime Environment

The experiments were conducted on a personal computer with macOS, using local computation. The dataset size was moderate and did not require distributed computing or cloud infrastructure.

This suite of tools and technologies provided a robust, reproducible environment for executing each stage of the research methodology—from raw data collection to final model evaluation.

3.9. Summary

This chapter outlined the methodological framework used to investigate the relationship between structured product attributes and Amazon Best Seller Rank (BSR), and to develop predictive models using machine learning techniques. The chapter began with an overview of the research design, which followed a quantitative, supervised learning approach to address the research problem.

Data was collected via Python-based web scraping, focusing on the Software category across multiple Amazon marketplaces. Key variables such as product price, star rating, number of reviews, and BSR were extracted and preprocessed to ensure consistency and analytical reliability. The data underwent cleaning, transformation, and feature engineering to enhance the quality of the inputs.

Several machine learning models were implemented—namely Linear Regression, Decision Tree Regressor, and Random Forest Regressor—to predict BSR. Each model was evaluated using R^2, RMSE, and MAE to compare predictive accuracy and interpretability. The results highlighted the limitations of linear models in capturing the complex dynamics of BSR and demonstrated the superiority of non-linear models, particularly Random Forest.

The next chapter will present the experimental results, visualisations, and a comparative analysis of model performance, leading to insights that support the research objectives.