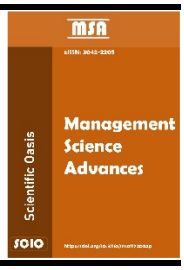
SCIENTIFIC OASIS

Management Science Advances

Journal homepage: www.msa-journal.org
eISSN: 3042-2205



Product Success Classification for E-Commerce Summer Sales: Evidence from the Wish Dataset

Raman Kumar[1] , Vladimir Simic[2,3,*], Rupinder Kaur[4], Harkirat Kaur[4,5], Jasleen Kaur[4,6], Kanishka Sharma[4,7], Vivek Kumar[8], Vivek John[9]

1 Department of Mechanical and Production Engineering, Guru Nanak Dev Engineering College, Ludhiana, Punjab, India
2 University of Belgrade, Faculty of Transport and Traffic Engineering, Vojvode Stepe 305, 11010 Belgrade, Serbia
3 Department of Computer Science and Engineering, College of Informatics, Korea University, Seoul 02841, Republic of Korea
4 Department of Information Technology, Guru Nanak Dev Engineering College, Ludhiana, Punjab, India
5 Department of Computer Science and Engineering, Graphic Era (Deemed to be University), Clement Town, Dehradun 248002, India
6 University Centre for Research and Development, Chandigarh University, Gharuan Mohali, Punjab, India
7 Chitkara University Institute of Engineering and Technology, Chitkara University, Rajpura 140401, Punjab, India
8 Computer Science & Engineering, Noida Institute of Engineering & Technology, Greater Noida, India
9 Department of Mechanical Engineering, Uttaranchal Institute of Technology, Uttaranchal University, Dehradun, India

ARTICLE INFO

Article history:
Received 5 January 2026
Received in revised form 25 March 2026
Accepted 25 April 2026
Available online 27 April 2026

Keywords:
E-Commerce; Product Classification; Machine Learning Algorithms; Wish Platform; Random Forest; K-Nearest Neighbors; Product Management

ABSTRACT

This study addresses the task of accurately classifying products on the Wish platform using machine learning (ML) algorithms. The Wish platform is an online e-commerce platform where sellers and buyers transact. This study aims to evaluate and compare the performance of different ML algorithms for product classification. The main methods employed in this study involved collecting a dataset from the Wish platform, consisting of various product attributes. Four ML algorithms, namely Logistic Regression (LR), Random Forest (RF), Decision Tree (DT), and K-Nearest Neighbors (KNN), were implemented and trained on the dataset. Performance metrics such as accuracy, precision, recall, F1-score, and area under the ROC curve (AUC) were used to evaluate the algorithms. The results of this study showed that RF achieved the highest accuracy (78.93%) and AUC score among the evaluated ML algorithms. LR and KNN also demonstrated competitive performance, while DT had relatively lower accuracy. Feature importance analysis revealed the most influential feature for all models, providing insights into the key factors contributing to product classification on the Wish platform. These findings have implications for decision-making on e-commerce platforms, enabling more accurate classification and targeted strategies for successful product promotion and inventory management.

1. Introduction

Managing, storing, and tracking inventory goods are all parts of inventory management. Supply chain management includes inventory management, as it encompasses all activities from when an

* *Corresponding author.*
E-mail address: vsima@korea.ac.kr

https://doi.org/10.31181/msa31202650

item enters the shop until it is dispatched [1]. It promotes efficiency, resource management, and yield maximization. Structure, method, and system are the three levels on which inventory management operates. The crucial elements of a firm, such as marketing and finance, are safeguarded by inventory management. Managers perform various actions that cost money when holding inventory. The price may fluctuate while perishable goods are being kept. Inventory modeling is essential since decisions impact every aspect of the business, including product intake and production, cash management, cost reduction, and profit maximization. Controlling inventory is necessary to ensure the company has the proper items to prevent shortages and excess stock, which account for a sizeable portion of all corporate expenditures [2].

Optimization of inventory management in e-commerce is becoming more dependent on artificial intelligence (AI), machine learning (ML), and deep learning (DL) [3-5]. Businesses can supply the appropriate items at the right time using AI-based systems that estimate client demand and identify patterns [6]. ML algorithms may be utilized to optimize inventories and find abnormalities in consumer behavior [7,8]. Inventory tracking may be automated by utilizing ML, resulting in precise real-time stock-level updates. Businesses may maximize their revenues and efficiency while lowering expenses by utilizing the potential of AI. Improving inventory management with ML has several advantages [9]. The ability of machine learning to speed up decision-making and automate procedures is one of its main advantages. This can speed up the calculation of the necessary inventory and increase inventory accuracy. Additionally, it can aid in early issue detection and resolution, ultimately saving time and money [10]. The large amount of data required for efficient analysis is a consideration when using ML to manage inventory. Large data sets can also be time-consuming and require specialized knowledge. As a result, several gaps still need to be filled in applying ML to inventory management. Although using ML for inventory management will present numerous difficulties, the outlook is promising. Technology advancements are gradually but steadily altering how organizations run [11]. ML has the potential to significantly improve inventory management in the future with the appropriate strategy and a sharp emphasis on business demands [12].

Selukar *et al.* [13] developed deep reinforcement learning (DRL) strategies to manage perishable product stock. Real-world cost and time characteristics were included in that model. Inventory costs and spoilage rates were decreased due to the simulation experiments. Zhang [14] intended to increase the management effectiveness of e-commerce platforms and help online retailers create a proper sales plan. It was essential to conduct online research for sales forecasting analysis. It demonstrated how precise forecasting of goods sales might increase management effectiveness and operational profitability on an e-commerce platform. A unique forecasting model for online clothes sales was developed using data mining technology.

Tsai & Chen [15] identified printed sources utilizing QR codes; several classifiers were implemented. Experiments were conducted using colored QR codes to test whether the residual bottleneck method was required for printed change detection. The results showed that the reduced-residual model could outperform all other investigated parameters for color printer source detection without including residual bottleneck blocks. Kezel [16] proposed a model-based approach as a viable choice for modeling client demand. This method used theoretical arguments to evaluate demand, and each model option was linked to particular assumptions that must be met for the model to be legitimate. A requirement was generally modeled as a Poisson process across the retail industry, particularly when a negative binomial distribution was used. Leenatham & Khemavuk [17] recognized that demand forecasting was crucial for logistics operations because it effectively supported numerous tasks and enhanced commercial competitiveness in a rapidly changing marketplace. The results revealed that conventional approaches often used statistical techniques such as exponential

smoothing or autoregressive integrated moving-average (ARIMA) models to predict product demand. However, most of the research focused on numerical data, such as time series and qualitative data, which significantly impacted product requirements in the real world. Sudimonto *et al.* [18] analyzed the uniqueness of research in inventory control and the usage of ML techniques. ML techniques were typically used to forecast, categorize, and timeline production. Belhadi *et al.* [19] reported that the importance of supply chain resilience and performance increased during recent disruptions caused by outbreaks and crises. The automation, unification, and globalization of the supply chain have raised interest in using cutting-edge information-processing tools, such as ML, to enhance supply chain performance. Qu *et al.* [20] used a decision-support system to optimize retail pricing and revenue for retail items, accounting for price, holidays, incentives, stock, and other regional factors. For the previous 2.5 years, it used sales data from major retailers across 45 regions.

Online retailers have a broad array of inventory management issues, including demand fluctuations, reverse logistics, overstock, managing SKUs, keeping inventory counts, multi-channel shoppers, bullwhip impact, and distressed inventories [21]. To minimize these hazards, online retailers adopted wholesale models, inventory categorization, hybrid strategies, pre-purchase stock, and stockless policies, which only buy products when customers place orders. The survey also showed that better inventory management is crucial for improving customer satisfaction, which dramatically helps e-commerce firms. Li *et al.* [22] revealed that, with the ongoing expansion of cross-border e-commerce firms' business activities and the rising operating costs of overseas warehousing, the cross-border e-commerce enterprise resource planning (ERP) system is responsible for managing, coordinating, and optimizing global supply chain warehousing. On the cross-border e-commerce ERP platform, the product sales forecast and inventory optimization strategy realized by the ML algorithm can successfully define the primary factors and use the sales record big data. The results indicate that the ML approach boosted the optimal inventory equilibrium efficiency and had strong predictive power. Pallathadka *et al.* [23] implemented AI in the e-commerce and financial industries to attain enhanced consumer experiences, effective supply chain oversight, increased productivity, and reduced waste. People have utilized these models in businesses and governmental organizations to anticipate and learn from data. ML models were being developed for the food business's complexity and range of information.

Qi *et al.* [24] found that enhancing the timeliness of product delivery in e-commerce strongly depends on product sales forecasting and presented a new deep neural framework for e-commerce sales forecasting. They built a sales residual network on top of the decoder to quantify the influence of competing interactions when a promotion campaign was initiated for a target item or just some substitutable counterparts. Several tests were run on two real-world datasets from the Taobao e-commerce site in various domains. Their findings demonstrate that the proposed framework performed better than modern deep learning alternatives and previous baselines regarding forecasting accuracy.

Boute *et al.* [25] reported considerable promise for sequential decision-making, including early advancements in inventory control using DRL. However, the vast number of options available when constructing a DRL algorithm and the significant computing work required to tune and assess each option may make it difficult to use in actual applications. With access to considerable historical data, Qi *et al.* [26] looked at a data-driven multiperiod inventory replenishment problem with uncertain demand and vendor lead time. They suggested an end-to-end system that outputs the recommended replenishment amount directly from input features using deep learning models with no intermediate steps. Jackson *et al.* [27] reported that inventory control had been a hot topic for debate in operations research and industrial engineering for over a century. Recent studies confirm that machine learning

can significantly improve e-commerce decision-making by enabling accurate product preference prediction and demand-driven inventory planning. For example, Random Forest models have shown strong predictive capability for consumer preference categories, with price and physical attributes emerging as dominant features [28]. Similarly, retail demand forecasting has been demonstrated as an effective approach for inventory optimization in cross-border e-commerce settings, improving inventory turnover and reducing inventory-related costs [29]. However, real-world e-commerce datasets frequently contain label noise and inconsistencies, which can reduce classification reliability and operational performance, motivating the need for robust classification pipelines and improved data quality handling [30].

Recent years have seen a rapid expansion of e-commerce, with platforms like Wish gaining popularity among customers worldwide. These systems' performance significantly depends on accurate product categorization and effective inventory management. Inventory management entails maintaining appropriate stock levels to fulfill consumer demand and reduce carrying costs and stockouts. Product categorization aims to classify items so personalized suggestions, enhanced search results, and focused marketing tactics can be implemented precisely. Inventory management is challenging during seasonal sales events like summer sales, when there are substantial swings in demand for items on e-commerce platforms. A delicate balance may need to be struck to maintain adequate stock levels for high-demand commodities without overstocking low-demand ones. Furthermore, optimizing user experience and profitability depends on effectively identifying items during such occasions.

1.1 Aims of the Study

The problem addressed in this study is accurately classifying products as successful or unsuccessful on the Wish platform based on various attributes. The objective is to compare and evaluate different ML algorithms to identify the most effective model for product classification. The research aims to achieve high accuracy and performance in distinguishing between positive and negative instances, providing insights for effective product management and decision-making on the Wish platform. The following research objectives were laid out as:

i. for product classification on the Wish platform, compare the performance of different ML algorithms, including LR, RF, DT, and KNN;
ii. evaluate the ML algorithms based on performance metrics such as accuracy, precision, recall, F1-score, and area under the ROC curve (AUC);
iii. identify the most influential features for product classification using feature importance analysis;
iv. provide recommendations for selecting the most effective ML algorithm for accurate product classification on the Wish platform;
v. explore limitations, future scope, and potential areas of improvement for further research in product classification and algorithm selection.

The rest of the paper is organized as follows: Section 2 presents materials and methods to accomplish the desired results, including a description of ML methods and a workflow methodology diagram. Section 3 describes a case study of summer sales and the optimal classification of summer sales. It includes a comparative analysis of machine learning algorithms based on performance metrics and hyperparameter tuning or optimization. Section 4 includes further discussion of the results, followed by conclusions in Section 5, along with limitations and future scope.

2. Materials and Methods

The workflow of the present research is depicted in Figure 1. The first stage in the analytical process is to prepare the data. This comprises cleansing, transforming, and dividing the data into sets for training, validation, and testing. The last step is to choose the data's most crucial features. Several methods, including dimension reduction, feature importance, and correlation analysis, may be utilized to accomplish this. The next stage is to choose the best model for the classification problem after selecting the features. Classification models come in various shapes and sizes, including logistic regression, decision trees, random forests, support vector machines, and neural networks.

The model must then be trained using the training set after being chosen. The model is optimized to discover the ideal parameters for the selected objective function. The validation set follows to evaluate the trained model. This step is crucial to prevent the model from overfitting the training set of data. Tuning the hyperparameters will further enhance the model's performance. The model can be used in production after being trained and tested. This step entails incorporating the model into the operational procedure and tracking its effectiveness over time. It is essential to interpret the model's outputs to understand the data and the classification task. Techniques like feature importance, Shapley Additive Explanations (SHAP) values, and partial dependence graphs are used.

Four ML algorithms were utilized to classify profitable and non-profitable products of the summer sales on the Wish website for clothes. ML algorithms include Logistic Regression (LR), K-Nearest Neighbors (KNN), Decision Tree (DT), and Random Forest (RF).

2.1 Logistic Regression

ML employs the LR classification technique. A logistic function is used to model the dependent variable. There are only two valid classes because the dependent variable has a dichotomous structure. LR is used to forecast an outcome from a categorical dependent variable. The consequences must therefore be discrete or categorical. It provides a probability value between 0 and 1 rather than precise integers like 0 and 1. It can be true or false, 0 or 1, yes or no, etc. The exponential constant e^b has a value of 2.718 in Eq. (1). It is used to represent the sigmoid function a:

$$a = \frac{1}{(1+e^b)} \tag{1}$$

If an is significantly negative, the value of b (predicted value) will be close to zero. The value of b is projected to be close to one if the value of a is also significantly positive.

2.2 K-Nearest Neighbor Classification

The KNN algorithm calculates fresh items by similarity and records all available data. This indicates that as new data is generated, it can be swiftly classified using the KNN method. The KNN algorithm saves the dataset throughout the training phase, and when it receives new data, it assigns it to a category that is quite similar to the new data. According to "feature similarity", which denotes how closely a new parameter resembles the points in the training set, the KNN algorithm predicts the values of new data points. Any implementation of an algorithm requires a dataset. Ergo, in the first stage of KNN, loads both the training and test data. Then, we choose the *K*-value, or the number of closest data points, with any positive integer allowed as *K*. Subsequently, we determine the distance between each row of training data and test data using the Euclidean distance (*ED*):

$$ED = \sqrt{(a_2 - a_1)^2 + (b_2 - b_1)^2} \tag{2}$$

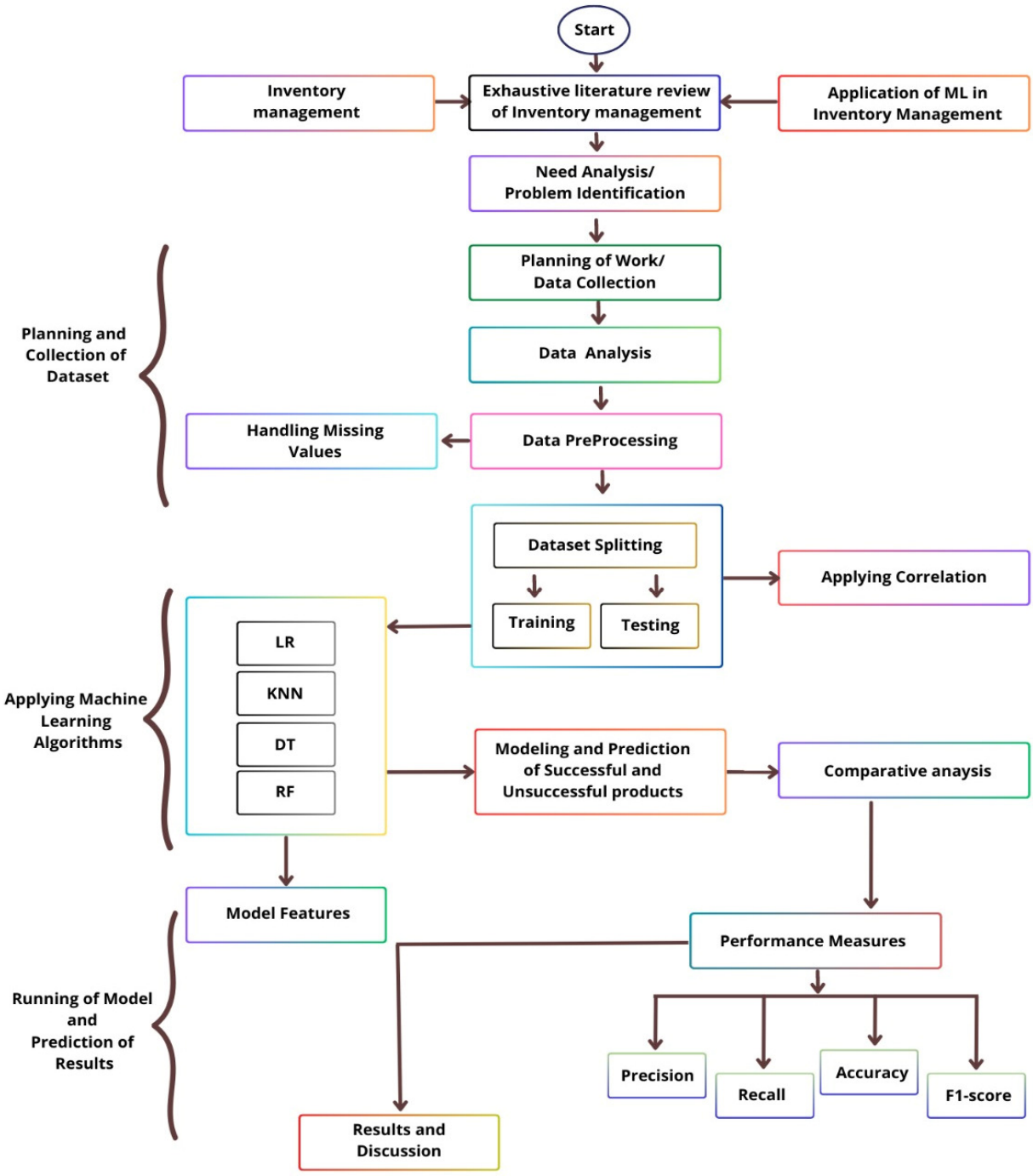


Fig. 1. Workflow of the present research

2.3 Decision Tree Classification

DT can be used to solve classification and regression problems. DTs are a supervised learning model that works best for classification issues. It is a tree-structured classifier with internal nodes denoting data set characteristics, branches denoting formulas, and each leaf node denoting a classification outcome [31]. The entire training set is first regarded as the root. Feature values should be categorical; if continuous, they are discretized before model construction. Entries are distributed recursively based on attribute values. Calculus used to create the decision tree algorithm: Entropy

must be understood before moving on to information gain. Entropy is shown in Eq. (3), and in various situations, entropy is a measure of disorder, or uncertainty:

$$H(s) = -Plog_2p^+ - (-Plog_2p^-) \quad (3)$$

where p^+ is the percentage of the positive class and p^- is the percentage of the negative class. How a DT divides the data depends on entropy [32]. It influences the bounds that a decision tree draws. Entropy values range from 0 to 1, where the lower the value, the more reliable the system is.

2.4 Random Forest Classification

RF is a supervised learning algorithm that can perform dataset classification and regression. RF is a tree-based algorithm that uses the quality features of multiple decision trees to make decisions. First, RF leads to creating independent decision tree branches during the training phase. The overall prognosis is then obtained by combining the predictions from all tree branches. The ensemble learning technique is the name given to RF, which employs a set of findings to make a judgment. Researchers can combine machine-learning algorithm predictions into a single framework using ensemble learning. The "forest" term for the ensemble technique originates from the training of several decision trees [33,34]. The random forest's output, which RF uses, is the average of each prediction tree:

$$F(p) = 1 - \sum_{n=1}^{k} r^2 \left(\frac{n}{p}\right) \quad (4)$$

The bagging or bootstrapping technique, on which RF is developed, is used to build a wide range of data samples from training phases randomly chosen with replacement. Each subset of data is used to create regression trees, which is referred to as a parallel process.

2.5 Performance Measures

The important performance metrics for LR, DT, KNN, and RF are:

i. *Precision* (*Pre*) – Precision, as shown in Equation (5), is the ratio between the true positives (T_P) and all the positives, which contains T_P and false positives (F_P). For the given problem statement, that would be the measure of classification of products that are correctly identified as successful and unsuccessful out of all the products being sold. Precision also gives us an estimate of the relevant data points [35].

$$Precision(Pre) = \frac{T_P}{T_P+F_P} \quad (5)$$

ii. *Recall* (*r*) – The recall is the measure of the model correctly identifying T_P as shown in Eq. (6). Thus, recall tells how many are correctly identified for all the successful and unsuccessful products in the market. Recall also measures how accurately the model can identify the relevant data [36].

$$Recall(r) = \frac{T_P}{T_P+F_N} \quad (6)$$

iii. *F1-score* – It is the harmonic mean of the precision and recall, as shown in Eq. (7) [37].

$$F1 - score = 2\left(\frac{Pre \times r}{Pre + r}\right) \quad (7)$$

iv. Accuracy (a) – It is the ratio of the total number of correct predictions to the total number of predictions [34,38].

$$a = \frac{T_P + T_N}{T_P + T_N + F_N + F_P} \quad (8)$$

Using accuracy as a defining metric for our model makes sense intuitively, but more often than not, it is advisable to use precision and recall, too. There might be other situations where our accuracy is very high, but our precision or recall is low. Ideally, we would like to avoid any problems where our product is successful, but our model classifies it as unsuccessful; i.e., aim for high recall. Although we do strive for high precision and high recall value, achieving both at the same time is not possible.

3. Case Study of Summer Sales

3.1 Pre-Processing

Collecting data from various sources is known as a dataset, and all stages of object recognition, such as the training stage and the performance evaluation of recognition algorithms, require the utilization of an appropriate dataset. The primary objective is to instruct the network to identify the traits that separate one class from another. The parent dataset is taken from the Wish platform on Kaggle, consists of 42 parameters, and is shown in Figure 2.

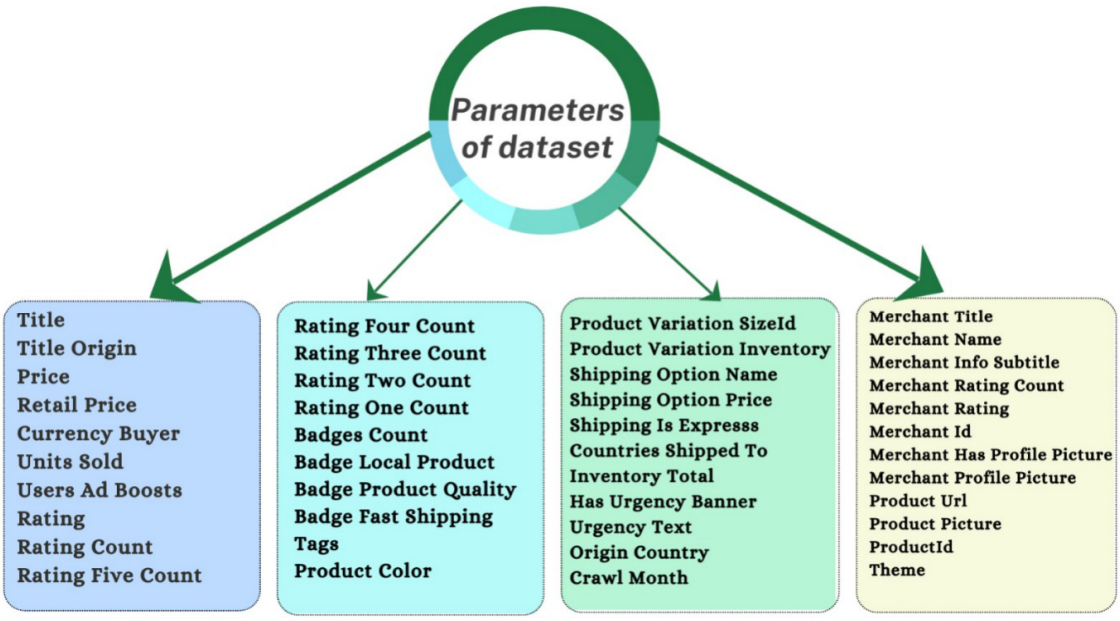


Fig. 2. Forty-two attributes of the dataset

Wish is an American online e-commerce platform for transactions between sellers and buyers. Instead of merely employing a search bar format, the platform visually customizes the purchasing experience for each customer. Vendors can post their goods on Wish and do direct business with customers. Wish does not stock the products themselves or handle returns; instead, it works with the providers of payment services to handle payments. The description of 42 attributes is shown in Figure 3.

Attribute	Definition	Type
Title	Title for localized for european countries	object
TitleOrigin	Original english title of the product	object
Price	Price you would pay to get the product	float64
RetailPrice	Reference Price for similar articles on the market	int64
CurrencyBuyer	Currency of the Prices	object
UnitsSold	Number of units sold	int64
UsesAdBoosts	Whether the seller paid to boost his product	int64
Rating	Mean product rating	float64
RatingCount	Total no. of ratings of the product	int64
RatingFiveCount	No. of 5-star ratings	float64
RatingFourCount	No. of 4-star ratings	float64
RatingThreeCount	No. of 3-star ratings	float64
RatingTwoCount	No. of 2-star ratings	float64
RatingOneCount	No. of 1-star ratings	float64
BadgesCount	No. of badges the product or the seller have	int64
BadgeLocalProduct	Badge that denotes a local product.	int64
BadgeProductQuality	Badge awarded for good evaluations	int64
BadgeFastShipping	Badge awarded for rapid shipping	int64
Tags	Tags set by the seller	object
ProductColor	Product's main color	object
ProductVariationSizeId	One of the available size for a product	object
ProductVariationInventory	Inventory the seller has	int64
ShippingOptionName	Shipping name	object
ShippingOptionPrice	Shipping Price	int64
ShippingIsExpress	Whether the shipping is express or not	int64
CountriesShippedTo	No. of countries this product is shipped to	int64
InventoryTotal	Total inventory for product's variations	int64
HasUrgencyBanner	Whether there was an urgency banner	float64
UrgencyText	Text appears over products in the search results	object
OriginCountry	Country from where the product originated	object
MerchantTitle	Merchant's displayed name	object
MerchantName	Merchant's canonical name	object
MerchantInfoSubtitle	The subtitle text on a seller's info section	object
MerchantRatingCount	Number of ratings of this seller	int64
MerchantRating	Merchant's rating	float64
MerchantId	Merchant unique id	object
MerchantHasProfilePicture	Whether there is a `merchantProfilePicture` url	int64
MerchantProfilePicture	Custom profile picture of the seller	object
ProductUrl	URL to the product page	object
ProductPicture	Picture of the product	object
ProductId	Product identifier	object
Theme	The search term used (Summer)	object
CrawlMonth	Meta: for info only.	object

Fig. 3. Description of 42 attributes

In Python, "int64" refers to a data type for signed 64-bit integers with values between -9223372036854775808 and 9223372036854775807. Large numbers that can be used in calculations and other mathematical operations are represented using this data type. For effective large-scale numerical computing, the Python library "numpy" can hold 64-bit integer arrays. In Python, a kind of data representing floating-point integers with 64 bits of accuracy is called float64. A wide range of values, including both extremely small and very big numbers, can be represented by this data type. The Python programming language and its libraries offer a wide range of mathematical operations and functions that can be used to manage float64 numbers. Applications requiring great precision in computation and data analysis for science frequently employ this data type.

Any form of value represented as an object is called an object data type in Python. This implies that it can refer to any Python data type, including strings, integers, lists, tuples, dictionaries, etc. The object data type is the base class for all other data types, and all values in Python are objects. Since it can represent anything in Python, it is also referred to as a generic type.

3.2 Features Extraction

Feature selection selects a subset of relevant features or parameters from a dataset. There are 42 parameters for classifying successful and unsuccessful products in the present case. The main objective of feature selection is to improve the accuracy and efficiency of the model by reducing the number of irrelevant or redundant features that can negatively impact model performance. Various techniques, such as filtering, wrapper, and embedded methods, are used to filter important, irrelevant, or redundant features. The embedded method was utilized to filter 42 parameters. Embedded techniques incorporate feature selection into the model-building process using regularization techniques that penalize including irrelevant or redundant features. The feature extraction library is imported from sklearn, an open-source data analysis library in Python. It resulted in the extraction of 11 important parameters out of 42.

Descriptive statistics are essential for inventory management of the nine important parameters out of the 11 selected for ML classification to summarize and analyze large amounts of data about inventory levels, sales, and other relevant variables. Two parameters are not included in the descriptive analysis. One is the product "title", and the second "is_successful" is without numerical values. The descriptive statistics are shown in Table 1 for nine product pricing and inventory management variables. It indicates that 1573 items or products are available on the Wish website for summer sales.

Table 1
Descriptive statistics of significant parameters of inventory management

Variable	Price (€)	Retail price (€)	Uses ad boosts	Badge product quality	Product variation inventory	Inventory total	Merchant rating count	Merchant rating	Merchant profile picture
Total Count	1573	1573	1573	1573	1573	1573	1573	1573	1573
Mean	8.33	23.29	0.43	0.07	33.08	49.82	26,496	4.03	0.14
StDev	3.93	30.36	0.50	0.26	21.35	2.56	78,474	0.20	0.35
Variance	15.46	921.60	0.25	0.07	455.96	6.57	6,158,240,183	0.04	0.12
Minimum	1	1	0	0	1	1	0	2.33	0
Median	8	10	0	0	50	50	7936	4.04	0
Maximum	49	252	1	1	50	50	2,174,765	5	1
Range	48	251	1	1	49	49	2,174,765	2.67	1
Mode	8	7	0	0	50	50	32,168	3.88	0
N for Mode	282	177	892	1456	907	1563	15	15	1347
Skewness	1.32	2.74	0.27	3.25	−0.57	−16.48	15.89	−1.03	2.03

3.3 Machine Learning Approach to Predict Successful and Unsuccessful Products

Four ML Algorithms, such as LR, KNN, DT, and RF, were applied to forecast successful and unsuccessful products. Before applying ML techniques, the dataset was separated into two groups. The model is trained using the training dataset, and the method's efficacy is evaluated using the testing dataset. With a cumulative ratio of 100 %, it was advised to indicate the ratio of training to testing samples as a percentage, such as 90:10, 85:15, 80:20, and 70:30 [39]. The 75:25 ratio was chosen as the best ratio for this inquiry. The training and testing dataset ratio was kept constant across all techniques, including LR, KNN, DT, and RF, to ensure that the models were trained and tested on the same dataset and that their efficacy could be evaluated on a comparable basis. Similar strategies were previously adopted for training and testing datasets [40,41]. As a result, the models were trained using the 1179 products, i.e., 75 % of 1573 products, and their effectiveness was tested using 394 products, i.e., 25 % of 1573 products.

The ML methods were executed using Python library version 3.11.1. Table 2 presents the time different ML algorithms took to execute on a Wish website summer sales dataset. KNN took the least time of 0.0041 seconds, and less time is desirable because the algorithm can process data more quickly, which can be important when working with large datasets or trying to achieve real-time performance. The results are consistent and expected based on the computational complexity of each ML algorithm.

Table 2
Execution time of the ML algorithms

Machine learning algorithms	Time (seconds)
Decision Tree	0.02
Random Forest	0.22
K-Nearest Neighbors	0.0041
Logistic Regression	0.03

KNN is a relatively simple algorithm that operates on the entire dataset, so it should be faster than the other algorithms, which have more complex computations. DT and LR are relatively simple algorithms that can be executed quickly, especially compared to ensemble methods like RF, which require building multiple DTs and combining their results. RF is expected to take the longest because it involves building multiple DTs and combining their results. The execution time of the RF is also affected by the number of trees in the ensemble and the depth of the trees. The execution times of the ML algorithms are reasonable and suggest that they are suitable for processing datasets of moderate to large size. However, it is essential to note that execution time is just one factor when selecting an algorithm for a particular task. Other factors, such as the accuracy and interpretability of the ML algorithms, should also be considered.

3.4 Optimization of Hyperparameters for DT, KNN, and RF

Optimizing hyperparameters is essential in building ML models to improve their performance (Table 3). It is an iterative process, and it is necessary to balance model complexity and overfitting. The optimal hyperparameters may vary depending on the specific dataset.

Table 3
Hyperparameters and search space of ML algorithms

Algorithms	Hyperparameters	Search Space
Logistic Regression	C	Continuous values in a specified range, such as [0.001, 1000]
	Penalty	Categorical choice between "l1" and "l2"
	Solver	Choice between "newton-cg", "lbfgs", "liblinear", "sag", "saga"
Random Forest	n_estimators	Integer values typically range from 10 to 1000
	Max_features	Categorical choice or integer values, such as "auto", "sqrt", "log2"
	Max_depth	Integer values typically range from 1 to 100
	Min_samples_split	Integer values typically range from 2 to 20
Decision Tree	Max_features	Categorical choice or integer values, such as "auto", "sqrt", "log2", or a range of integers
	Max_depth	Integer values typically range from 1 to 100
	Min_samples_split	Integer values typically range from 2 to 20
K-Nearest Neighbors	n_neighbors	Integer values typically range from 1 to 20
	Weights	Categorical choice between "uniform" and "distance"
	Algorithm	Categorical choice between "auto", "ball_tree", "kd_tree", "brute"
	Leaf_size	Integer values typically range from 10 to 100

DT has a significant hyperparameter, maximum depth (Max_depth), which controls the tree's depth. Start with a small value and gradually increase it to find the optimal depth, preventing overfitting, and specify the maximum levels the tree can grow. Figure 4 shows the graph of the accuracy of the DT model versus Max_depth, which varied from 1 to 10. It was found that Max_depth of 8 is optimal for DT, with an accuracy of 72.34.

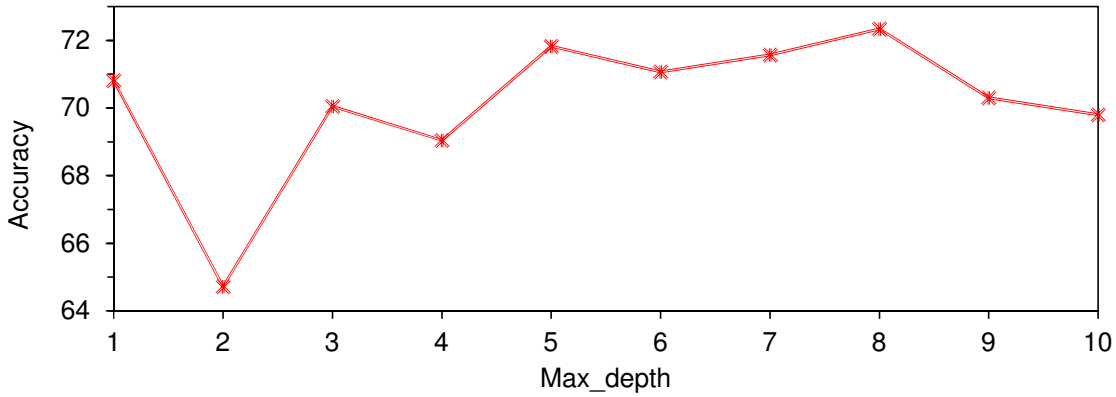


Fig. 4. Accuracy versus the hyperparameters in the case of DT

KNN has a vital hyperparameter, "Number of Neighbors" (n_neighbors). It refers to the number of nearest neighbors to consider when making predictions for a new data point. The choice of the number of neighbors determines the flexibility or smoothness of the KNN model. Smaller values of n_neighbors result in more flexible models, as fewer nearby neighbors influence the predictions. Larger values of n_neighbors result in smoother decision boundaries and less sensitivity to individual data points. Properly selecting n_neighbors is crucial for achieving the best balance between model complexity, overfitting, and accuracy on unseen data. Figure 5 shows the variation of n_neighbors on the accuracy of the KNN model to classify data. At 90 n_neighbors, the KNN model attained a maximum accuracy of 72.58. This is an optimal value, as accuracy reduces with a further rise in n_neighbors.

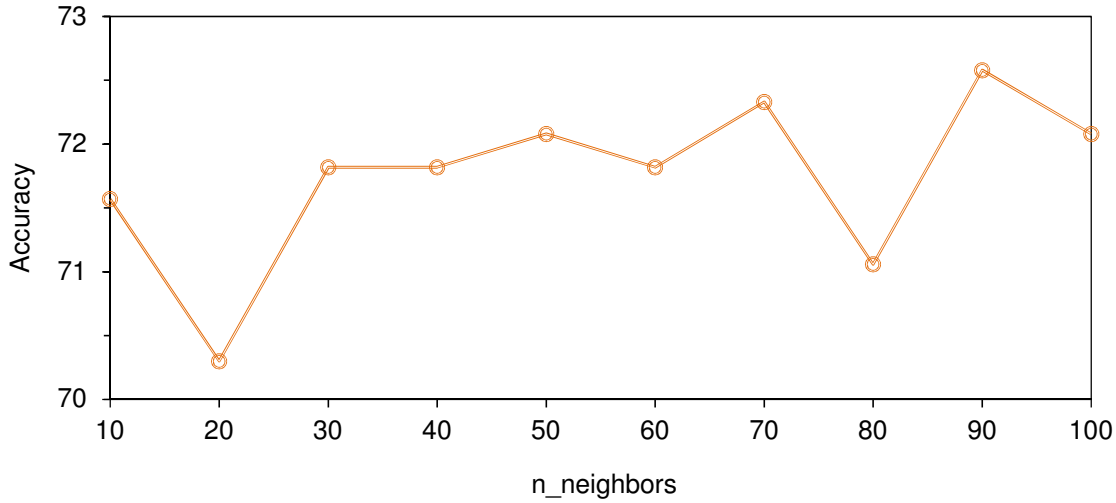


Fig. 5. Accuracy versus the hyperparameters in the case of KNN

RF has a vital hyperparameter n_estimators, which determines the number of trees in the forest. Increasing the number of trees can improve performance up to a certain point. Figure 6 depicts how

the rise in n_estimators affects the accuracy of the RF classification model. At n_estimators of 10, accuracy is 74.41, and it fluctuates and goes up and down with a surge in n_estimators. So, 100 n_estimators is the optimal value, with an accuracy of 78.93 for the RF model.

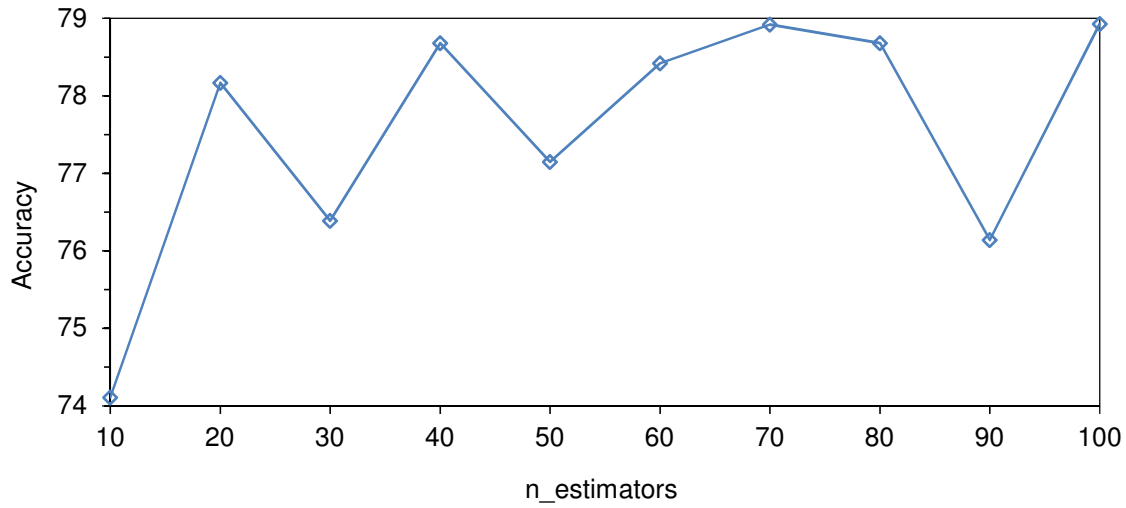


Fig. 6. Accuracy versus the hyperparameters in the case of RF

3.5 Performance Metrics for Product Classification

Performance metrics are crucial for assessing how well ML models execute tasks involving product classification. The exact objectives and specifications of the product categorization job determine the performance measures that should be used. It is advised to consider various measures to fully grasp the ML model's performance. The performance metrics are defined by Eqs. (5)–(8).

3.5.1 Logistic regression

The LR binary classification model's performance on a dataset with 394 instances is displayed in Table 4. The LR model determined whether each event was “successful” or “unsuccessful”, and the dataset contains the actual classification. There were 279 cases accurately labeled as “unsuccessful”, giving the model a precision of 0.75; i.e., 75% of all instances were labeled as “unsuccessful”. The recall score quantifies the percentage of real positives the model appropriately detected. The recall for “unsuccessful” was 0.93, meaning that 93% of all instances of “unsuccessful” in the dataset were classified correctly by the model. At a precision of 0.6, the LR model correctly identified 36% of “successful” events. The model missed many instances of “success” since the recall for “successful” was low at 0.25. The model's overall accuracy was 0.73, meaning it correctly classified 73% of occurrences regardless of class. The LR model performs better for “unsuccessful” instances than “successful” ones based on these criteria.

Table 4
Performance metrics for product classification by LR at optimal hyperparameters

	Precision	Recall	F1-score	Support
“Unsuccessful”	0.75	0.93	0.83	279
“Successful”	0.6	0.25	0.36	115
Accuracy			0.73	394

3.5.2 Decision tree

The performance of a DT binary classification model is displayed in Table 5. The precision for the "unsuccessful" class was 0.8. Only 50% of the cases were predicted to be "successful" with a precision of 0.5. With a recall of 0.78 for the "unsuccessful", the model accurately recognized 78% of the actual "unsuccessful" cases, and 0.53 for the "successful" cases. The model's overall accuracy was 0.71.

Table 5
Performance metrics for product classification by DT at optimal hyperparameters

	Precision	Recall	F1-score	Support
"Unsuccessful"	0.8	0.78	0.79	279
"Successful"	0.5	0.53	0.51	115
Accuracy			0.71	394

3.5.3 K-nearest neighbors

Table 6 displays the results of a KNN binary classification model that forecasted the success or failure of a specific event at optimal hyperparameters. In total, 394 occurrences occurred, of which 115 were projected to be successful and 279 to be unsuccessful. The accuracy for the "unsuccessful" category was 0.79. With a recall of 0.75 for the "unsuccessful" category, the model correctly identified 75% of the actual unsuccessful events. The harmonic mean of precision and recall for the category "unsuccessful" was 0.77, giving that category an F1-score in that range. The precision for the "successful" category was 0.46, meaning that only 46% of the projected "successful" events were. The recall for the "successful" category was 0.51 as well, meaning that only 51% of the real successful occurrences could be properly identified by the model. "Successful" events had an F1-score of 0.48. The model's overall accuracy was 0.68, which indicates that the model accurately predicted 68 % of the occurrences. In contrast to successful events, the performance was better at predicting unsuccessful ones.

Table 6
Performance metrics for product classification by KNN at optimal hyperparameters

	Precision	Recall	F1-score	Support
"Unsuccessful"	0.79	0.75	0.77	279
"Successful"	0.46	0.51	0.48	115
Accuracy			0.68	394

3.5.4 Random forest

Table 7 displays how well an RF classification model performed on a test data sample at optimal hyperparameters. With an overall accuracy of 0.77 in this instance, the RF model "successfully" anticipated the outcome in 77% of the test cases. The RF model appears to be more accurate at predicting unsuccessful outcomes than successful ones, as the precision and recall for the "unsuccessful" class are greater than for the "successful" class. Yet, the F1-scores for both groups are comparable, indicating that the model's precision and recall are fairly evenly distributed. Although the RF model seems to be functioning satisfactorily, there may be potential for enhancement, especially in foreseeing positive outcomes.

Table 7
Performance metrics for product classification by RF at optimal hyperparameters

	Precision	Recall	F1-score	Support
"Unsuccessful"	0.82	0.86	0.84	279
"Successful"	0.62	0.54	0.58	115
Accuracy			0.77	394

3.6. Calibration Curve

A calibration curve is a tool used in ML to evaluate the effectiveness of a binary classification model by comparing predicted probabilities to actual probabilities for the positive class. In other words, it considers the degree to which a model's anticipated probability corresponds to a successful event's actual probability. The calibration curve is produced for various projected probabilities by graphing the expected probabilities of the positive class (*y*-axis) against the actual fraction of positive events (*x*-axis) [36]. The calibration curve's points must be on the diagonal line $(y = x)$ for the model to be fully calibrated, indicating the projected probabilities must coincide with the actual probabilities. The calibration curves are shown in Figure 7.

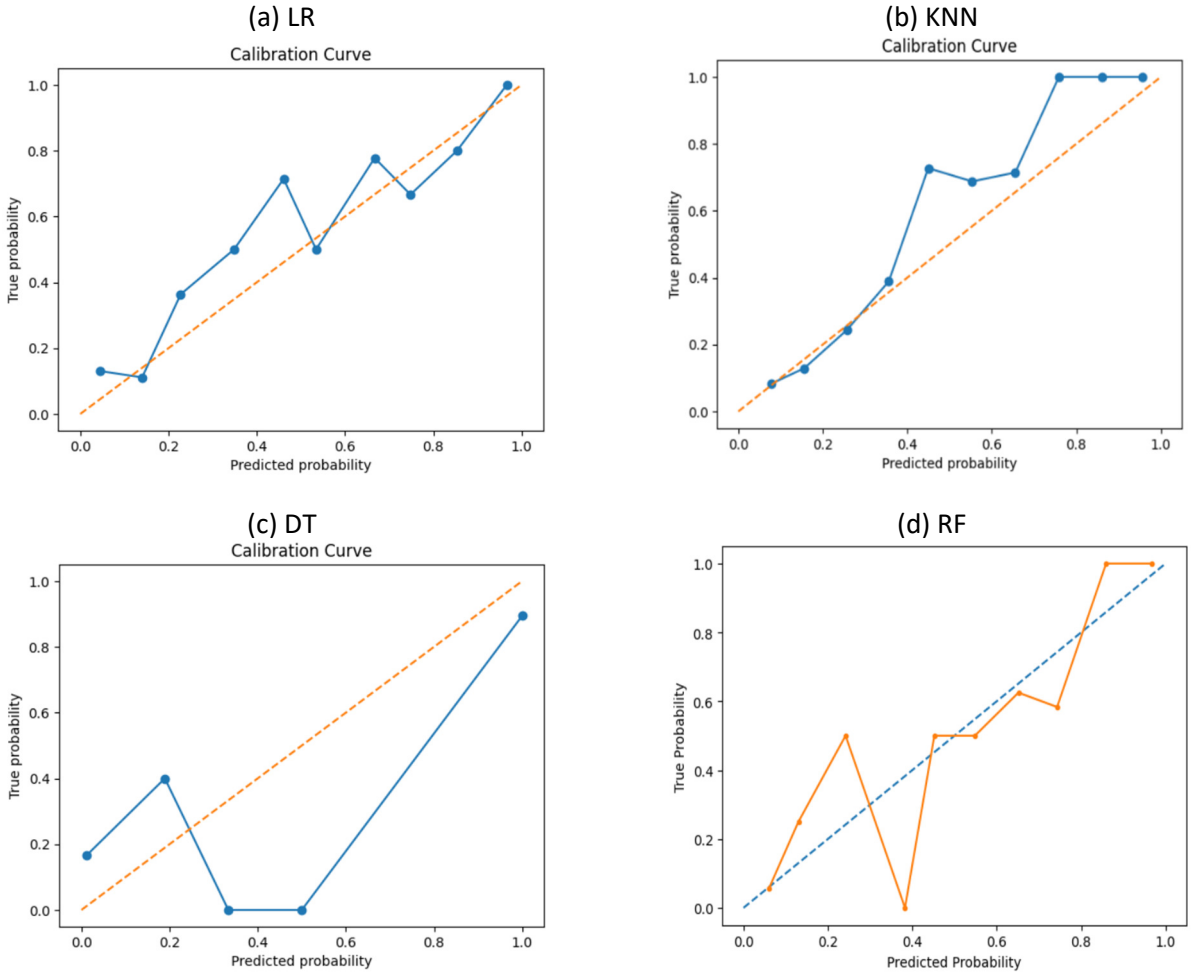


Fig. 7. Calibration curves

3.7. Receiver Operating Characteristic Curve

An illustration of a binary classification model's performance on a graph is called a Receiver Operating Characteristic (ROC) curve. The graph contrasts the true and false positive rates at various threshold levels. The proportion of negative cases wrongly categorized as positive is the false positive rate, whereas TPR is often called sensitivity or recall. The best ROC curve would have a steep slope at the beginning, signifying a high TPR and a low FPR, whereas the poorest model would be a diagonal line, representing guesswork. A model is said to be flawless if it has an area under the ROC curve (AUC) of 1, while a model with a 0.5 AUC is just marginally more accurate than guessing at random. In machine learning, ROC curves help assess a model's performance and contrast various models [42,43].

Figure 8 shows ROC curves for LR, KNN, DT, and RF. RF has a maximum AUC score of 0.818, followed by KNN with an AUC score of 0.7245, LR of 0.7232, and DT of 0.6741. The AUC scores for the different ML models indicate their performance in classifying products. Based on these scores, the RF model performs the best among the evaluated models in distinguishing between positive and negative instances. It has the highest AUC score, indicating that it can better separate the classes than the other models. The KNN model also performs relatively well, followed by LR and DT.

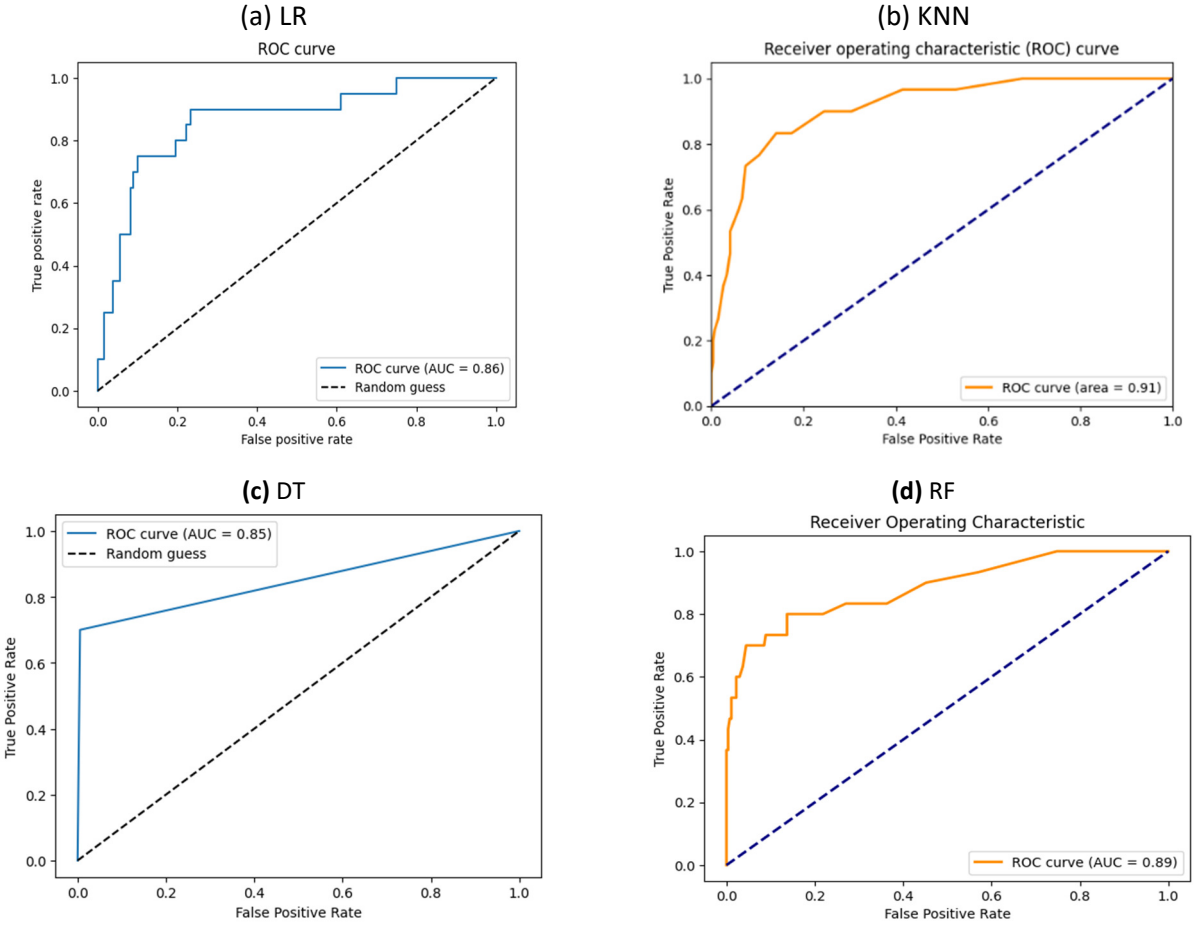


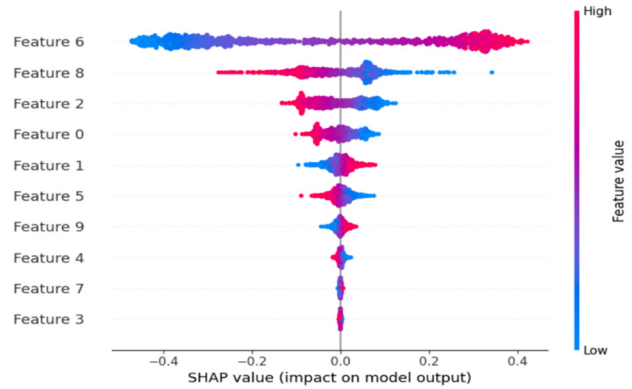
Fig. 8. ROC curves

3.8 Explainability: Shapley Additive Explanations

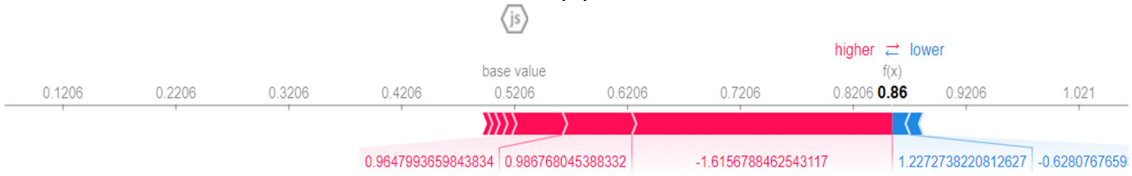
A technique for explaining the output of any ML model is SHAP (Figure 9). It is an integrated framework combining multiple existing techniques for explaining model predictions and expanding them to give more precise and coherent explanations [44]. The basis of SHAP is the game theory idea

of Shapley values, which offers every component of a model a contribution value. These numbers in ML represent the contribution of each feature to the model's prediction. SHAP generates a set of explanations for each prediction, explaining how each attribute contributed to the forecast.

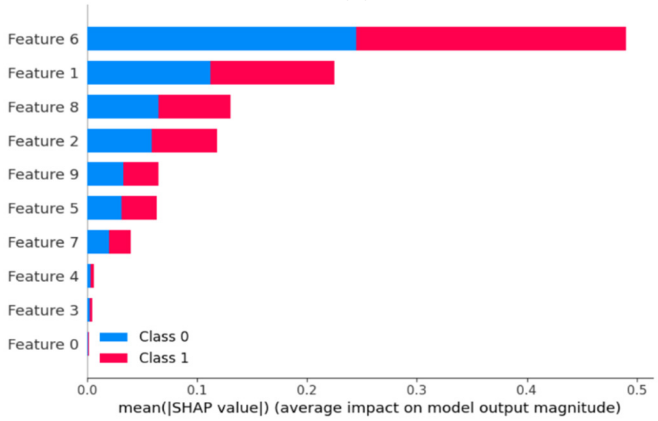
(a) LR



(b) KNN



(c) DT



(d) RF

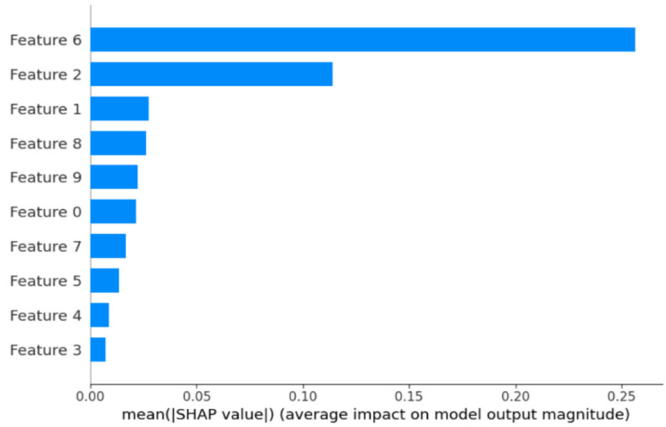


Fig. 9. SHAP for LR, KNN, DT, and RF

Figure 9 shows SHAP for LR, KNN, DT, and RF. Feature 6 is the most important for all ML models. Any attribute or characteristic, such as age, income, education level, or any other pertinent data dependent on the area or issue the developer is working on, could be represented by feature 6. To comprehend the precise context and interpretation of feature 6 in the given dataset, the inventory total is used as feature 6.

4. Discussion

Figure 10 displays the precision, recall, and F1-score for two classes (i.e., "successful_1" and "unsuccessful_0") regarding LR, DT, KNN, and RF. The LR has a low F1-score of 0.36 and a high precision for unsuccessful of 0.75. However, a low recall for the successful category of 0.25 and a low F1-score for the successful class indicate that the model is more accurate at predicting unsuccessful than successful products. The model has a higher chance of properly predicting successful cases but is more likely to generate false positive predictions. The DT model has a greater chance of properly recognizing successful cases and a higher possibility of making false positive predictions. It has a higher recall for the successful category of 0.56 but a poorer accuracy of 0.48 and an F1-score of 0.51. With equivalent precision and recall rates for both categories, KNN performs reasonably well. However, its F1-score, including both classes, is only 0.49 for successful products and 0.77 for unsuccessful products. In addition, RF has a great F1-score for both classes and strong performance in both recall and precision. Overall, it is the best-performing model, outperforming DT and LR regarding recall for successful cases (0.55) and precision for unsuccessful cases (0.81), respectively.

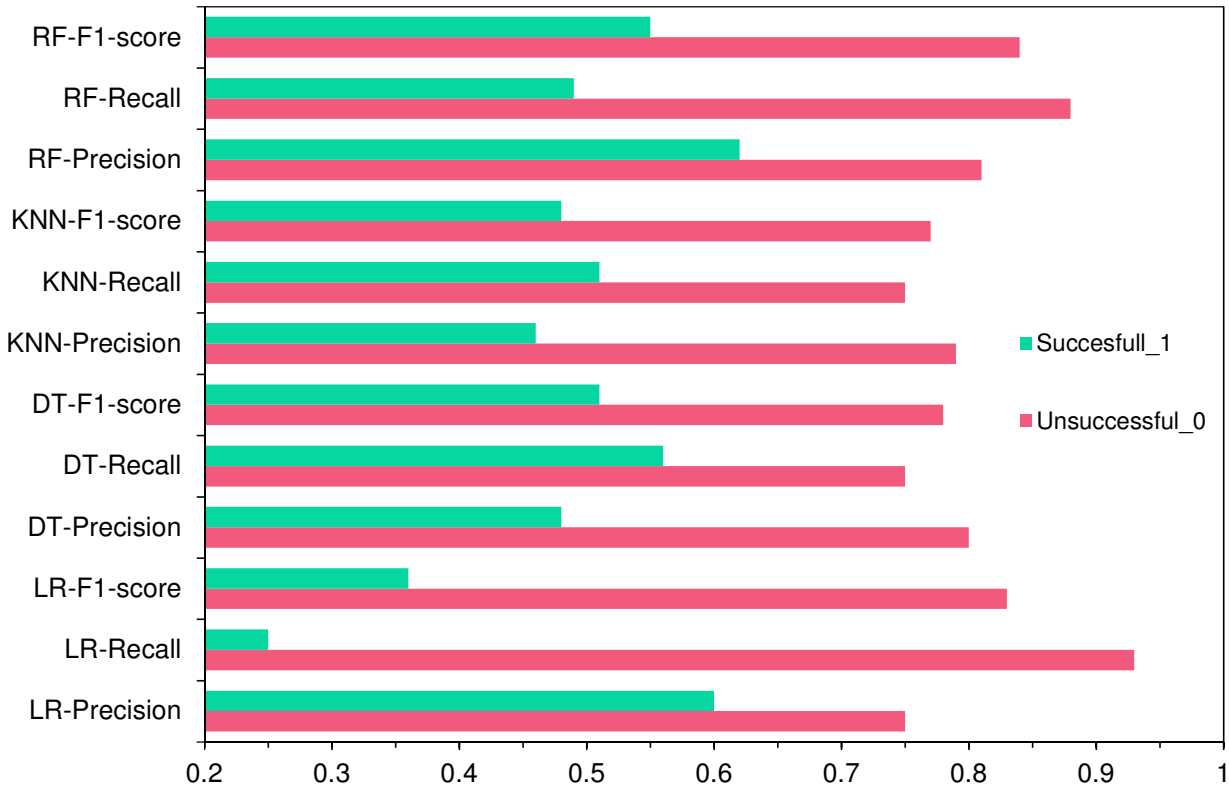


Fig. 10. Comparative analysis of ML classification models regarding precision, recall, and F1-score

By comparing the results from Figure 10, the performance of the different ML algorithms can be assessed for the given classes:

i. *"Unsuccessful_0" class* – RF has the highest precision of 0.81, recall of 0.88, and F1-score of 0.84 among all the models. This indicates that RF performs well in correctly identifying instances of the "Unsuccessful_0" class, with a relatively low rate of false positives and false negatives. LR also shows decent performance with a precision of 0.75 and a recall of 0.93. It has a high recall, meaning it identifies a high proportion of actual "Unsuccessful_0" instances, but its precision is slightly lower compared to RF. KNN and DT have lower precision, recall, and F1-scores for this class, indicating that they might not perform as well in correctly identifying "Unsuccessful_0" class instances
ii. *"Successful_1" class* – LR has the highest precision of 0.6 and an F1-score of 0.36 among the models. However, its recall of 0.25 is relatively low, indicating that it struggles to identify actual instances of the "Successful_1" class. RF has a higher recall of 0.49 compared to LR for this class, but its precision of 0.49 and F1-score of 0.55 are slightly lower. KNN has a higher recall of 0.51 compared to LR and RF for this class, but its precision of 0.46 and F1-score of 0.48 are lower

Comparative analysis from Figure 10 indicates that RF generally performs better across both classes, with higher precision, recall, and F1-scores for the "Unsuccessful_0" class and competitive performance for the "Successful_1" class. LR performs well for the "Unsuccessful_0" class but struggles with the "Successful_1" class. KNN shows a relatively balanced performance for both classes, but its precision and F1-scores are slightly lower than RF and LR. While considering overall performance, RF appears to be the better ML algorithm among the evaluated models for the given task involving product classification.

Figure 11 shows the accuracy ratings of the four ML models. The RF model has the greatest accuracy rating. It correctly classified 78.93 % of the test set instances. The LR model has the second-highest accuracy score of 0.73. This shows that the LR model could correctly classify 73 % of the test set's cases. The accuracy ratings for the DT and KNN models were 72.34 % and 72.58 %, respectively. The accuracy metric provides a general overview of the performance of the models, showing how well they classify the instances overall. RF has the highest accuracy, indicating the best overall predictive performance among the evaluated models. It is advisable to consider additional evaluation metrics and analyze the specific requirements and characteristics of the classification task to assess the models' performance comprehensively.

There are several factors to justify why one model performs better than the others in terms of accuracy for product classification. The complexity of a model can impact its performance. RF and DT models are more complex than LR and KNN. RF and DT models can capture complex relationships and interactions within the data, improving accuracy in certain cases. Product classification tasks may involve non-linear relationships between features and target classes. RF and DT models can capture non-linear patterns in the data. In contrast, LR and KNN models may struggle to handle non-linearity unless appropriate transformations or feature engineering techniques are applied. RF is an ensemble learning method that combines multiple decision trees, whereas DT, LR, and KNN are standalone models. The ensemble nature of RF allows it to reduce the variance and overfitting that can occur with individual decision trees. This can contribute to RF's improved accuracy compared to DT. RF has the advantage of providing feature importance measures. It can identify the most relevant features for product classification, enabling the model to focus on the most informative aspects. This feature selection capability can enhance RF's accuracy by considering the most influential features while making predictions. The KNN algorithm determines class membership by comparing the features of a given instance with its nearest neighbors. The distribution and density of the data points can

influence the effectiveness of KNN. If the products in the dataset exhibit clear clusters or groups, KNN may perform well. However, if the data distribution is complex or overlapping, KNN's accuracy may be compromised.

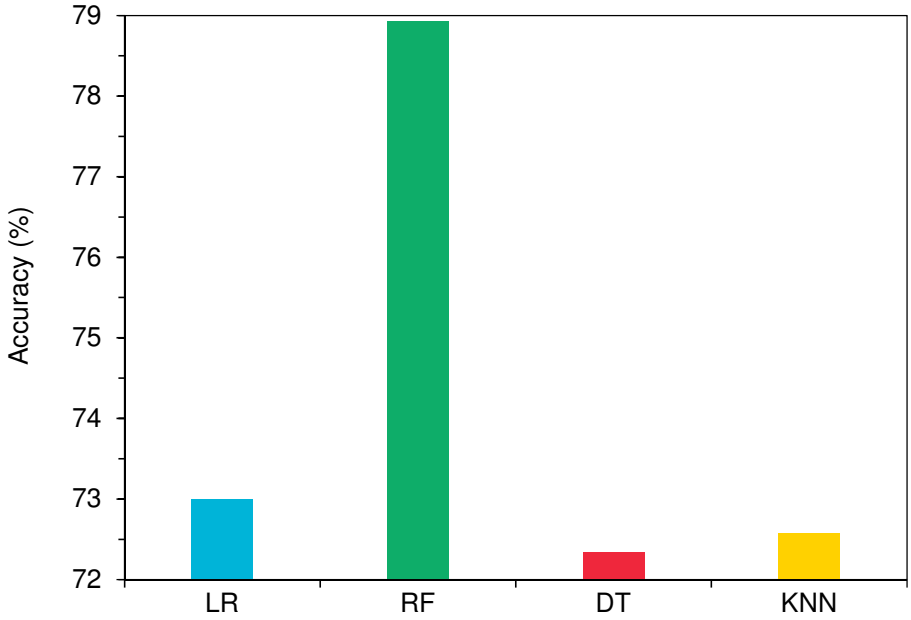


Fig. 11. Comparative analysis of ML classification models regarding accuracy

It is important to note that the performance of ML models is highly dependent on the dataset's characteristics, feature engineering, hyperparameter tuning, and other factors specific to the task at hand. Therefore, the superior accuracy of RF in product classification could be attributed to its ability to capture complex relationships, handle non-linearity, leverage ensemble learning, and identify important features. However, it is always recommended to perform thorough experimentation and analysis to determine the most suitable model for a specific classification task.

4.1 Implications of the Study

i. *Enhanced inventory management* – This study's results can improve inventory management procedures for small and medium-sized businesses using e-commerce sites such as Wish. Using demand forecasting methodologies and optimizing stock levels using machine learning-driven insights, enterprises may mitigate the expenses of overstocking and stockouts, resulting in enhanced operational efficacy and financial gains.
ii. *Targeted marketing strategies* – Using ML algorithms to classify products accurately enables the implementation of focused marketing campaigns. Businesses may target their marketing efforts to promote successful items and modify methods for unsuccessful ones by knowing which products have a higher chance of succeeding or failing. Conversion rates and total sales success may both be enhanced by this focused strategy.
iii. *Improved customer experience* – A better customer experience results from precise product classification and effective inventory management. Consumers will likely locate more pertinent items, get tailored suggestions, and encounter fewer availability-related problems: higher consumer happiness, more frequent purchases, and a favorable view of the brand result from this.

iv. *Cost reduction* – Businesses may reduce excess inventory, storage, and logistics expenses by utilizing machine learning algorithms for product classification and inventory management. For e-commerce businesses, maximizing stock levels and simplifying inventory procedures result in cost savings and increased profitability.
v. *Strategic decision-making* – Businesses can make data-driven decisions thanks to the study's insights. Businesses may efficiently manage resources, prioritize product promotions, and make well-informed decisions on inventory stocking levels, pricing tactics, and marketing campaigns by comprehending the critical aspects that impact product success and failure.
vi. *Future research and innovation* – The study creates more e-commerce and machine learning research and innovation opportunities. To further improve inventory management and product classification accuracy, research may be conducted in areas including deep learning models, multi-class classification scenarios, sophisticated feature engineering approaches, and optimization tactics.
vii. *Competitive advantage* – Companies that successfully use the study's conclusions have an edge over rivals in the e-commerce market. Enhanced customer satisfaction, focused marketing tactics, and better inventory control procedures all support long-term viability in a cutthroat marketplace.

The study's consequences go beyond product classification and inventory management; they also affect many facets of e-commerce operations and strategic decision-making for companies using Wish and other similar platforms.

5. Conclusions

Small and medium-sized enterprises can maintain inventory levels while reducing human labor by using demand forecasting, allowing them to spend less money on keeping inventory. Since the stocks are ordered depending on demand, the forecasting technique can reduce overstock and stock-outs of particular items. This study comprehensively analyzed ML algorithms for product classification on the Wish platform. The quantitative results demonstrated that RF achieved the highest accuracy of 78.93% and an AUC score of 0.81799, outperforming the other models. LR and KNN also showed competitive performance with 73% and 72.58% accuracy, respectively, while DT had a relatively lower accuracy of 72.34%.

Furthermore, feature importance analysis using SHAP indicated that feature 6 was the most influential feature for all ML models. This insight can provide valuable information for understanding the key factors contributing to classifying successful or unsuccessful products on the Wish platform. These findings have implications for product management and decision-making on e-commerce platforms, enabling more accurate classification and targeted strategies for successful product promotion and inventory management. The outcomes of this study were based on the particular dataset and context of the Wish platform. However, it is crucial to highlight that future research might examine how generalizable these conclusions are across various e-commerce platforms and datasets. Additionally, investigating more sophisticated methods, such as DL models, might improve the precision and effectiveness of product categorization in subsequent research.

This study has some restrictions. First, the investigation was done using a particular dataset from the Wish platform, which might not accurately reflect the variety of items and situations seen in other e-commerce platforms. Furthermore, the study ignored multi-class classification in favor of a binary classification challenge. The effectiveness of the algorithms on various datasets might be examined

in more detail, and the investigation could be expanded to multi-class classification scenarios. Additionally, investigating the effects of feature engineering methods and hyperparameter optimization may help the models perform even better. In the future, categorical embeddings in neural networks may be employed to increase the model's accuracy because this field of neural networks is still in its infancy and still needs further research. Huge logistics, warehousing, and distribution issues related to inventory management are possible future research areas.

Conflict of Interest

The author declares no conflict of interest.

Acknowledgment

The author received no external funding for this research.

References

[1] Stranieri, F., Fadda, E., & Stella, F. (2024). Combining deep reinforcement learning and multi-stage stochastic programming to address the supply chain inventory management problem. *International Journal of Production Economics, 268*, 109099. https://doi.org/10.1016/j.ijpe.2023.109099.

[2] Canyakmaz, C., Özekici, S., & Karaesmen, F. (2024). Risk management through financial hedging in inventory systems with stochastic price processes. *International Journal of Production Economics, 270*, 109189. https://doi.org/10.1016/j.ijpe.2024.109189.

[3] Helmi, R. A. A., Johar, M. G. M., & Hafiz, M. A. S. B. M. (2023). Online phishing detection using machine learning. In *2023 1st International Conference on Advanced Innovations in Smart Cities* (ICAISC) (pp. 1-4). IEEE. https://doi.org/10.1109/ICAISC56366.2023.10085377.

[4] Helmi, R. A. A., Elghanuni, R. H., & Abdullah, M. I. (2021). Effect the graph metric to detect anomalies and non-anomalies on Facebook using machine learning models. In *2021 IEEE 12th Control and System Graduate Research Colloquium* (ICSGRC) (pp. 7-12). IEEE. https://doi.org/10.1109/ICSGRC53186.2021.9515227.

[5] Kaur, R., Kumar, R., & Aggarwal, H. (2025). Systematic Review of Artificial Intelligence, Machine Learning, and Deep Learning in Machining Operations: Advancements, Challenges, and Future Directions. *Archives of Computational Methods in Engineering, 32*(8), 4983-5036. https://doi.org/10.1007/s11831-025-10290-z.

[6] Lin, S. W., & Lu, W. M. (2024). Using inverse DEA and machine learning algorithms to evaluate and predict suppliers' performance in the apple supply chain. *International Journal of Production Economics, 271*, 109203. https://doi.org/10.1016/j.ijpe.2024.109203.

[7] Do, N., Tham, J., Azam, S., & Khatibia, A. (2020). Analysis of customer behavioral intentions towards mobile payment: Cambodian consumer's perspective. *Accounting, 6*(7), 1391-1402. https://doi.org/10.5267/j.ac.2020.8.010.

[8] Aldousari, A. A., Delafrooz, N., Ab Yajid, M. S., & Ahmed, Z. U. (2016). Determinants of consumers' attitudes toward online shopping. *Journal of Transnational Management, 21*(4), 183-199. https://doi.org/10.1080/15475778.2016.1226658.

[9] Mgm, J., Yajid, M. S. A., & Khatibi, A. (2020). A situation analysis of integrated logistics berhad Malaysia. *Systematic Reviews in Pharmacy, 11*(1), 734-741. https://doi.org/10.5530/srp.2020.1.94.

[10] Pallathadka, H., Mustafa, M., Sanchez, D. T., Sajja, G. S., Gour, S., & Naved, M. (2023). Impact of machine learning on management, healthcare and agriculture. *Materials Today: Proceedings, 80*, 2803-2806. https://doi.org/10.1016/j.matpr.2021.07.042.

[11] Hammann, D. (2024). Big data and machine learning in cost estimation: An automotive case study. *International Journal of Production Economics, 269*, 109137. https://doi.org/10.1016/j.ijpe.2023.109137.

[12] Tirkolaee, E. B., Sadeghi, S., Mooseloo, F. M., Vandchali, H. R., & Aeini, S. (2021). Application of machine learning in supply chain management: a comprehensive overview of the main areas. *Mathematical Problems in Engineering, 2021*(1), 1476043. https://doi.org/10.1155/2021/1476043.

[13] Selukar, M., Jain, P., & Kumar, T. (2022). Inventory control of multiple perishable goods using deep reinforcement learning for sustainable environment. *Sustainable Energy Technologies and Assessments, 52*, 102038. https://doi.org/10.1016/j.seta.2022.102038.

[14] Zhang, B., Tseng, M. L., Qi, L., Guo, Y., & Wang, C. H. (2023). A comparative online sales forecasting analysis: Data mining techniques. *Computers & Industrial Engineering, 176*, 108935. https://doi.org/10.1016/j.cie.2022.108935.

[15] Tsai, M. J., & Chen, T. M. (2022). A Non-Bottleneck Residual Approach for QR Code Printed Source Identification. In *2022 IEEE Eighth International Conference on Big Data Computing Service and Applications* (BigDataService) (pp. 132-136). IEEE. https://doi.org/10.1109/BigDataService55688.2022.00028.

[16] Kerzel, U. (2023). Demand Models for Supermarket Demand Forecasting. *International Journal of Supply & Operations Management, 10*(1), 89-104. https://doi.org/10.22034/ijsom.2022.109044.2145.

[17] Leenatham, A., & Khemavuk, P. (2020). Demand forecasting using artificial neural network based on quantitative and qualitative data. In *2020 1st International Conference on Big Data Analytics and Practices* (IBDAP) (pp. 1-6). IEEE. https://doi.org/10.1109/IBDAP50342.2020.9245614.

[18] Sudimanto, Lumban Gaol, F., Warnars, H. L. H. S., & Soewito, B. (2022). Inventory Control with Machine Learning Approach: A Bibliometric Analysis. In: Ranganathan, G., Bestak, R., Palanisamy, R., Rocha, Á. (eds) *Pervasive Computing and Social Networking. Lecture Notes in Networks and Systems*, vol 317. Springer, Singapore. https://doi.org/10.1007/978-981-16-5640-8_21.

[19] Belhadi, A., Mani, V., Kamble, S. S., Khan, S. A. R., & Verma, S. (2024). Artificial intelligence-driven innovation for enhancing supply chain resilience and performance under the effect of supply chain dynamism: an empirical investigation. *Annals of Operations Research, 333*(2), 627-652. https://doi.org/10.1007/s10479-021-03956-x.

[20] Qu, T., Zhang, J. H., Chan, F. T., Srivastava, R. S., Tiwari, M. K., & Park, W. Y. (2017). Demand prediction and price optimization for semi-luxury supermarket segment. *Computers & Industrial Engineering, 113*, 91-102. https://doi.org/10.1016/j.cie.2017.09.004.

[21] Patil, H., & Divekar, B. R. (2014). Inventory management challenges for B2C e-commerce retailers. *Procedia Economics and Finance, 11*, 561-571. https://doi.org/10.1016/S2212-5671(14)00221-4.

[22] Li, J., Wang, T., Chen, Z., & Luo, G. (2019). Machine learning algorithm generated sales prediction for inventory optimization in cross-border E-commerce. *International Journal of Frontiers in Engineering Technology, 1*(1), 62-74.

[23] Pallathadka, H., Ramirez-Asis, E. H., Loli-Poma, T. P., Kaliyaperumal, K., Ventayen, R. J. M., & Naved, M. (2023). Applications of artificial intelligence in business management, e-commerce and finance. *Materials Today: Proceedings, 80*, 2610-2613. https://doi.org/10.1016/j.matpr.2021.06.419.

[24] Qi, Y., Li, C., Deng, H., Cai, M., Qi, Y., & Deng, Y. (2019). A deep neural framework for sales forecasting in e-commerce. In *Proceedings of the 28th ACM international conference on information and knowledge management* (pp. 299-308). https://doi.org/10.1145/3357384.3357883.

[25] Boute, R. N., Gijsbrechts, J., Van Jaarsveld, W., & Vanvuchelen, N. (2022). Deep reinforcement learning for inventory control: A roadmap. *European Journal of Operational Research, 298*(2), 401-412. https://doi.org/10.1016/j.ejor.2021.07.016.

[26] Qi, M., Shi, Y., Qi, Y., Ma, C., Yuan, R., Wu, D., & Shen, Z. J. (2023). A practical end-to-end inventory management model with deep learning. *Management Science, 69*(2), 759-773. https://doi.org/10.1287/mnsc.2022.4564.

[27] Jackson, I., Tolujevs, J., & Kegenbekov, Z. (2020). Review of inventory control models: a classification based on methods of obtaining optimal control parameters. *Transport and Telecommunication Journal, 21*(3), 191-202. https://doi.org/10.2478/ttj-2020-0015.

[28] Çardak, H., Bardak, S., Bardak, T., Capraz, O., Ozcetin, S., & Kızılırmak, S. (2025). Predicting Consumer Preferences for Furniture Products on E-commerce Platforms: An Analysis Using Machine Learning and Favorite Listing Data. *BioResources, 20*(4), 9768-9784. https://doi.org/10.15376/biores.20.4.9768-9784.

[29] Feng, J., Su, L., & Liu, X. (2025). Research on Inventory Management of Cross-Border e-commerce Through Retail Demand Forecasting. *Journal of The Institution of Engineers (India): Series C, 106*(6), 1755-1761. https://doi.org/10.1007/s40032-025-01251-3.

[30] Zhang, M., Fu, J., Zhang, Y., Ma, Q., & Chen, J. (2025). Integrated label correction for e-commerce data: Boosting accuracy with baseline attention and enhanced Bayesian updating. *Computers in Industry, 173*, 104392. https://doi.org/10.1016/j.compind.2025.104392.

[31] Li, Z., & Tham, J. (2023). Network accounting information security based on classification and regression tree algorithm. In *International Conference on Algorithms, High Performance Computing, and Artificial Intelligence* (AHPCAI 2023) (Vol. 12941, pp. 112-119). SPIE. https://doi.org/10.1117/12.3011569.

[32] Thakur, V., Kumar, R., Kumar, R., Singh, R., & Kumar, V. (2024). Hybrid additive manufacturing of highly sustainable Polylactic acid-Carbon Fiber-Polylactic acid sandwiched composite structures: Optimization and machine learning. *Journal of Thermoplastic Composite Materials, 37*(2), 466-492. https://doi.org/10.1177/08927057231180186.

[33] Abdullah, M. I., Hao, S. K., Abdullah, I., & Faizah, S. (2023, August). Parkinson's disease symptom detection using hybrid feature extraction and classification model. In *2023 IEEE 14th Control and System Graduate Research Colloquium* (ICSGRC) (pp. 93-98). IEEE. https://doi.org/10.1109/ICSGRC57744.2023.10215477.

[34] Dissanayake, K., & Md Johar, M. G. (2021). Comparative study on heart disease prediction using feature selection techniques on classification algorithms. *Applied Computational Intelligence and Soft Computing, 2021*(1), 5581806. https://doi.org/10.1155/2021/5581806.

[35] Chhetri, B., Goyal, L. M., & Mittal, M. (2023). How machine learning is used to study addiction in digital healthcare: A systematic review. *International Journal of Information Management Data Insights, 3*(2), 100175. https://doi.org/10.1016/j.jjimei.2023.100175.

[36] Dissanayake, K., & Johar, M. G. M. (2023). Two-level boosting classifiers ensemble based on feature selection for heart disease prediction. Indonesian Journal of Electrical Engineering and Computer Science, 32(1), 381-391. https://doi.org/10.11591/ijeecs.v32.i1.pp381-391.

[37] Chikkalingaiah, S., Padmanabha Rao Hari Prasad, S. A., & Uggregowda, L. D. (2023). Classification techniques using gray level co-occurrence matrix features for the detection of lung cancer using computed tomography imaging. *International Journal of Electrical & Computer Engineering, 13*(5). https://doi.org/10.11591/ijece.v13i5.pp5135-5146.

[38] Mohammadi, S., Dehghani, M., Khatibi, A., Sanderman, R., & Hagedoorn, M. (2015). Caregivers' attentional bias to pain: does it affect caregiver accuracy in detecting patient pain behaviors?. *Pain, 156*(1), 123-130. https://doi.org/10.1016/j.pain.0000000000000015.

[39] Pawanr, S., Garg, G. K., & Routroy, S. (2023). Prediction of energy efficiency, power factor and associated carbon emissions of machine tools using soft computing techniques. *International Journal on Interactive Design and Manufacturing (IJIDeM), 17*(3), 1165-1183. https://doi.org/10.1007/s12008-022-01089-4.

[40] Garg, A., Lam, J. S. L., & Gao, L. (2016). Power consumption and tool life models for the production process. *Journal of Cleaner Production, 131*, 754-764. https://doi.org/10.1016/j.jclepro.2016.04.099.

[41] Mondal, S., & Goswami, S. S. (2024). Rise of Intelligent Machines: Influence of Artificial Intelligence on Mechanical Engineering Innovation. *Spectrum of Engineering and Management Sciences, 2*(1), 46-55. https://doi.org/10.31181/sems1120244h.

[42] Sarkar, A., Goswami, S. S., & Sahoo, S. K. (2026). AI-Powered Threats and Solutions: A Theoretical Analysis of Risks, Governance, and Ethical Safeguards. *Applied Research Advances, 2*(1), 1-23. https://doi.org/10.65069/ara2120267.

[43] Sitinjak, C., Simic, V., & Simanullang, W. F. (2025). Promoting the Adoption Dynamics of Autonomous and Shared Autonomous Vehicles: A Scientific Mixed-Methods Approach. *International Scientific Spectrum, 1*(1), 1-29. https://doi.org/10.66972/iscis1120253.

[44] Ibrahim, Z., Rahman, N. R. A., & Johar, M. G. M. (2019). A job satisfaction of emotional intelligence, leadership, employee performance with information technology. *International Journal of Recent Technology and Engineering, 8*(2), 712-718. https://doi.org/10.35940/ijrte.B1148.0982S919.