Mathematical Biosciences and Engineering

MBE, 16(6):8121–8133.
DOI: 10.3934/mbe.2019409
Received: 21 May 2019
Accepted: 03 September 2019
Published: 06 September 2019

http://www.aimspress.com/journal/MBE

Research article

Recommendations to improve dead stock management in garment industry using data analytics

Poonkuzhali Sugumaran[1] and Vinodhkumar Sukumaran[2,3,*]

[1] Department of Information Technology, Rajalakshmi Engineering College, Chennai 602105, India
[2] Department of Computer Science & Engineering, Rajalakshmi Engineering College, Chennai 602105, India
[3] Anna University, Chennai 600025, India

* **Correspondence:** Email: vinodkumar.s@rajalakshmi.edu.in.

Abstract: The garment industry has huge potential when it comes to ability to integrate IT components for efficient data analytics. Rapid changes in trends and the excessive presence of fashion variants lead to unsold garments, termed as dead stock, which affect the profitability of organizations. Hence, in the garment business, markdown planning has been an imperative for efficient management of dead stock. This research work deals with a data analytics model for improving sales by making timely suggestions to retailers to provide offers and discounts to reduce dead stock by markdown optimization. The model consists of two modules, namely a classification module and a gain optimization module. In the first module, a hybrid classifier ID3, with the AdaBoost algorithm, is built to classify garments for sales recommendation, from an apparel dataset taken from the UCI repository. The predictor categorizes the garments into moving stock and dead stock. Finally, the gain optimization module uses linear programming and bandit learning of upper confidence bounds with the Chernoff-Hoeffding inequality algorithm, to bundle dead stock with fast-moving garments by giving optimal discounts that maximize revenue. The hybrid classifier provides 98% accuracy, and thereby, the analytics improve turnover, as well as balance supply and demand in the garment industry.

Keywords: bundling pair; dead stock; garments; hybrid classifier; linear programming; moving stock; recommendation

1. Introduction

The garment industry has tremendously expanded business over the past decade, in both the local and overseas markets. Garment manufacturing is an assembly-oriented activity, and deals with a great range of raw materials, product types, production volumes, supply chains, retail markets, and associated technologies. Dead stock is one of the major issues in the garment industry. In the garment industry, sales data classifications show diverse market trends. Some clusters or segments of sales may grow, while others decline. Unsold garments that remain in warehouses, see zero sales in a defined number of months, and retard business growth, can be referred to as “dead stock”. It affects business cash flow, takes up valuable warehouse space and freezes earnings that otherwise should be dedicated to the purchase of revenue-generating products. Dead stock occurs either in the form of raw materials, or as finished products, i.e. the dresses manufactured by the garment industry. Inventory doesn't just die overnight. Distributors must know what its main causes are and how to fix it. Most times, dead inventory can be pinpointed to poor purchasing decisions. It's difficult for distributors to avoid stocks of dead inventory if they don't have controls in place during the purchasing process. Dead stock can be managed by using data analytics to categorize dead stock and moving stock.

Data analytics tends to be very useful in understanding market trends, and linear programming (LP) is used to optimize gain by helping businesses decide on the products and the quantities among moving, and dead stock, to bundle for sale, to maximize profit. It is easy to turn cash into inventory, but doing vice versa is a challenge in the market. Data mining techniques like classification and prediction can be used to find meaningful patterns for future predictions, that help businesses efficiently utilize dead stock without incurring losses, and increase turnover. The proposed system identifies dead stocks from sets of stocks based on attributes of the data with a class label using the hybrid classifier model (ID3 with AdaBoost ensemble classifier) to provide better recommendations for stock maintenance. To obtain the advantages of both ID3 and the AdaBoost classifier, and to have better performance in terms of accuracy, precision and recall, a hybrid model is proposed. With it, it is also possible to predict what type of materials can be purchased and stocked for future sales.

Classification and Prediction is a useful approach in distinguishing between the sales frequency of different items on the basis of known attributes. For example, a material made of cotton will be in high demand during summer, whereas woolen materials will be dead stocks during the season. These recommendations help to predict which attributes lead to dead stocks in the future so that a seller need not buy predicted kinds of dresses from a distributor, or a manufacturer need not manufacture dresses with the predicted properties. The proposed work looks into the opportunities there may be to automate tasks with an inventory management system, to identify dead stocks, and provide timely options to retailers for providing discounts to reduce dead stock by optimizing gain using linear programming with the UCB algorithm.

The main contribution towards this work is to:

- Isolate dead stock from remaining moving stock given in a dataset using machine learning classifiers (AdaBoost with ID3).
- Subject extracted dead stocks further to bundling with moving stock using the UCB algorithm and by creating sales offers.

- Use linear programming techniques to determine the number of units to be sold of each product bundle, to optimize profit for the organization.

2. Related work

Ignaciuk et al. [1] presented a control system approach using a state space model to help manipulate and maneuver demand and supply in order to avoid delays. The system shows considerable high availability and lower delay in catering to demand. Many statistical tests and cost-based analyses were conducted to ensure the same, based on topology that was simulated out of the existing system players. Bounou et al. [2] presented their work in inventory management system, a multi-pronged, decision-based system. The system takes the risk of dead stock into account. Unstable demand and risk factors lead to a decision support system aimed at avoiding excessive wastage of capital incurred in the buying and storing of goods. Bayesian networks are created to formulate the risk, and, thereby, inventory management is performed. Guidotti et al. [3] suggested that personalized services are becoming the order of the day. The author proposed a similar supermarket scenario, and presented possible criteria with which the consumer would make a purchase decision. Based on these criteria, recurrence of a decision on consecutive scenarios is calculated. Temporal analysis, and prediction the recurrence of purchase decisions, are analyzed herein.

Sun et al. [4] suggested a decision modeling paradigm to predict the games consequences. Krakozov et al. [5] delivered an automated business process to increase the effectiveness of business plans made. The system is an integrated decision support system which is composed of a great many tools, and is guided by established business processes. The expert system considers what promotional activities are carried out in the selling process. Wani et al. [6] developed a sustainable process for imperishable business trends using big data analytics. Predictive techniques are used in order to suggest what criteria satisfy the consumer. It also considers profit margin in the sales process, to guide the business process. Dharmapriya et al. [7] provided a decision support framework for supply network configuration, which considers market fluctuations and a supply chain-based decision framework to meet the present challenges of the business framework.

Das et al. [8] suggested a survey of several recommendation systems used to suggest ecommerce sale choices. Zhang et al. [9] stated the need for recommendation systems in several fields involving deep learning. The framework takes into account much heuristic information, and contextual demand attributes of the consumer. Mugdha et al. [10] suggested that recommender systems have become part and parcel of day-to-day life. Simple means are used to implement the recommendation systems in ecommerce applications. Matrix-based approaches are used to enhance the same, and to improve classification accuracy, thereby. Sehgala et al. [11] suggested that large amounts of data can be sorted using optimized recommendation approaches like collaborative filtering. Further break-down based on user group can also help to enhance the outcome of the recommendation engines. The optimized techniques increase the precision of the recommendations to do with user choices based on past preferences. Kulkarni et al. [12] improvised the hybrid model for collaborative filtering over recommendation systems. Jeble et al. [13] provided a literature review of how big data helps in decision making and also makes suggestions by predictive analysis to managers of industries to allow them to make informed decisions to improve the ROI of their businesses. Anusha et al. [14] proposed a framework for identifying items frequently purchased by customers, and for recommending the same to new customers. Jasim [15] used three machine learning classification algorithms such as the Naïve Bayes, the decision tree and the artificial neural network on this dataset to implement recommendation systems for garments using the weka tool. Ullah et al. [16] proposed a recommendation system using three classifiers such as the Naïve Bayes, the multilayer perceptron and the K-star. The author has done both binary and multiclass classifications using the same classifiers.

In existing systems, many of the recommendation systems have been built based on collaborative filtering and individual classifiers like the Naïve Bayes and the decision tree, for managing inventories and supply chains. The performance of those algorithms is also very poor. These aforementioned problems have motivated some to apply the hybrid classification algorithm in the proposed system, thereby managing dead stocks of material to manage customer needs, and predicting the occurrence and recommending things for the purpose of satisfaction of customers in the textile industry.

3. Materials and method

The data analytics model for managing dead stocks in garment industry is depicted in Figure 1. The architecture briefs that the system takes input from the inventory management system of the distributor. Then, the data is pre-processed to make it suitable for analyzing. After data preparation, a decision tree classifier is built using the training data to classify garments as recommended or not. Using the classifier, based on the recommendations, a prediction engine finds preferred garments by customer, as well as by dead stock that needs to be cleared quickly. Finally, a gain optimization algorithm is used in order to generate pairs of dead stocks and bestselling garments in a gain-optimized manner.

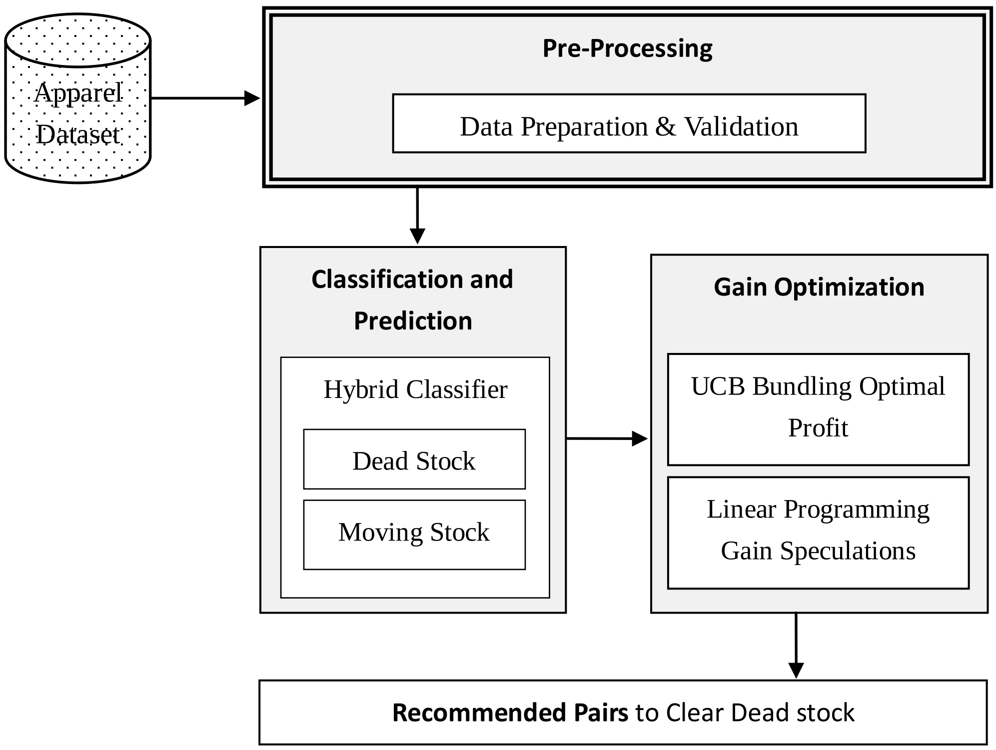

Figure 1. Framework of the data analytics models to manage dead stock.

3.1 Classification and prediction module

In this module, a hybrid classifier is built over the training dataset to find whether a garment is recommended or not. The proposed hybrid classifier is constructed using the decision tree classifier ID3, and the ensemble classifier AdaBoost algorithm, as it classifies the dead stocks and moving stock more accurately than other individual classifiers. If a garment is recommended, then it considered to be the moving stock preferred by most customers, else the garments which are not recommended are declared as dead stock.

Pseudo code for hybrid classifier:

ID3_Classifier (D)
{
Check for the stopping criterion for constructing the tree node
{
Return terminal node with the "Target class" if all the instances belong to same class
Return terminal node as "Failure" if the predicting attribute is empty.
Return terminal node with the "most frequent class value" if all the instances are empty.
} else
{
Compute the gain ratio information for every input attribute A

$$Info(D) = -\sum_{i=1}^{m}(p_i log_2(p_i))$$

$$Entropy(A) = -\sum_{j=1}^{v}(|D_j|/|D| * Info(D))$$

$$InfoGain(A) = Info(D) - Entropy(A)$$

where, p_i *is the probability that an arbitrary instance in AD*
$|D_j|/|D|$ *is weight of the* j^{th} *partition*
E (A) is the entropy of the attribute A
Gain (A) is the information gain of attribute A.
Identify the attribute with highest information gain as Split Attribute
Split D into sublist based on Split Attribute
Perform recursive call for the sublist.
}
}
Adaboot_with_ID3(D)
{
Assign the same weights to all the instances in D.
Train an ID3 classifier over the instances using the weights.
Examine mis-classified instances and update their weights.
Train another classifier with the new instances, re-weight for misclassified instances.
Iterate for N classifiers.
Construct strong classifiers based on a vote of the (weighted) ID3 classifiers.
}

3.2 Gain optimization module

This module is used to find how much optimal discount can be given for a garment while forming a bundled pair in the process of dead stock clearance. It is mandatory that the dead stock be sold at any cost above its cost price, with optimal profit. So the parameters like operational costs, manufacturing costs, storage costs and rental expenses, are considered to fix the selling price and discount rate, and to form best bundled pair to get optimal gain from the product.

Profit = selling price – cost price
Manufacturing price = manufacturing costs + operating expenses + profit
Operating expenses = labour + transportation costs + storage costs + tax + rent for space**
Cost price = manufacturing costs + operating expenses
Selling price = manufacturing costs + operating expenses + profit
List price = selling price + discount
Discount = list price – selling price

Linear programming (LP) and UCB algorithms are used to generate pairs of dead stock and bestselling garments in a gain optimized manner. First, the UCB algorithm is applied in order to bundle pairs of bestselling and dead stock items, and to fix discount rates and offers. Then, the linear programming algorithm is applied with the aim of performing gain optimization over the bundles formed. This is done by maximizing the objective function, considering linear inequalities subject to constraints. In this manner, dead stock clearance sales always go with offers on bundles of related items.

Pseudo code for fixing discounts by Upper Confidence Bound 1 using Chernoff-Hoeffding inequality:

for J = 1 to n
{
Assume n_k *represents the frequency of the event J*
Repeat the event J maximizing payoff X such as

$$\overline{X_k} + \sqrt{2\,log(T)/n_k}$$

Register the reward points X (k, j)
Update the mean payoff for the event using Chernoff-Hoeffding inequality

$$P(|X - \mu| > t) \leq 2e^{-2t^2/n(b-a)^2}$$

}

Pseudo code for finding optimal profit by linear programming:

i. *Assume input attributes as* $cost\ price = c_{ij}, discount\_rate = d_j, Profit = p_i$
ii. *Assume output variables as* $garment\_type = g_j$

n = number of non-negative variables
m = Number of constraints

iii. Maximize linear objective function as linear equation assuming the constraints m_i *asprior formulated.*

$$\begin{gathered}
Maximize\ Z = c_1 g_1 + c_2 g_2 + \ldots\ldots..+ c_n g_n \\
Subject\ to\ d_{11} g_1 + d_{12} g_2 + \ldots\ldots..+ d_{1n} g_n = p_1 \\
d_{21} g_1 + d_{22} g_2 + \ldots\ldots..+ d_{2n} g_n = p_2 \\
\ldots\ldots.. \\
d_{m1} g_1 + d_{m2} g_2 + \ldots\ldots..+ d_{mn} g_n = p_m \\
x_1, x_2, \ldots\ldots.., x_n \geq 0
\end{gathered}$$

iv. Add slack variable to convert each inequality to equality.

v. Then solve the n-dimensional problem in iterations.

The idea behind using Chernoff-Hoeffding Inequality is to arrive at the optimal value in the linear programming by using very simple expected values developed from the sample set. The UCB algorithm assumes rewards for the trials and reward points. The algorithm is harnessed to find the best profitable conditions conducive for sales.

4. Results and discussion

4.1 Dataset

In this study, the dresses dataset is taken from the UCI repository posted by Muhammad Usman & Adeel Ahmed, and students at Air University (dresses attribute sales dataset) [17]. This garment dataset contains 501 instances and 13 attributes of dresses and their recommendations towards optimal sales, and is listed in Table 1. Pre-processing is done for the dataset to fill in the missing values and resolve inconsistencies and redundancies. Feature relevance analysis is done to extract the relevant attributes and to know the impact of those attributes in recommendations.

Table 1. Attribute description.

Attribute	Values
Style	Bohemia, Brief, Casual, Cute, Fashion, Flare, Novelty, OL, Party, Sexy, Vintage, Work
Price	Low, Average, Medium, High, Very-High
Rating	1–5
Size	S, M, L, XL, Free
Season	Autumn, Winter, Spring, Summer
Neck Line	O-neck, Backless, Board-neck, Bow-neck, Halter, Mandarin collar, Open, Peter-pan collar, Ruffled, Scoop, Slash-neck, Square collar, Sweetheart, turndown collar, V-neck
Sleeve Length	Full, Half, Half sleeves, Butterfly, Sleeveless, Short, Three quarter, Turndown, Null
Waise line	Dropped, Empire, Natural, Princess, Null
Material	Wool, Cotton, Mix
Fabric Type	Shafoon, Dobby, Popline, Satin, Knitted, Jersey, Flannel, Corduroy
Decoration	Applique, Beading, Bow, Button, Cascading, Crystal, Draped, Embroidery, Feathers, Flowers
Pattern type	Solid, Animal, Dot, Leapardetc
Recommendation	0, 1

4.2 Evaluation of hybrid classifier

After pre-processing, classification of the dead stock is performed using the hybrid classifier ID3 with AdaBoost algorithm, as it improves performance when compared with the individual classifier. Using training, the dataset decision tree is constructed in order to classify whether the garment is recommended or not, by representing them with a 0 or a 1. Validation of this classifier is done on the testing dataset, and finally identifies whether garments are the fastest selling ones preferred by customers, or dead stock.

The quality of the recommendation system is measured using the following metrics. The performance parameters can be explained as below:

TN: Indicates that dead stock is incorrectly detected as moving stock.

TP: Indicates that dead stock is correctly detected as dead stock.

FN: Indicates that moving stock is incorrectly detected as dead stock.

FP: Indicates that moving stock is correctly detected as moving stock.

Accuracy: It gives correct predictions whether a garment is afast-selling one or dead stock, among all the clothes purchased.

$$Accuracy = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + +\mathrm{FP} + \mathrm{TN} + \mathrm{FN}}$$

Precision: It refers to the percentage of the correct predictions made by the model.

$$Precision = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}$$

Recall: It refers to the percentage of total relevant results correctly predicted by the model.

$$Recall = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}$$

Table 2. Performance evaluation of hybrid classifier with individual classifier.

Classifier	Response time	Precision	Recall
SVM	0.01	0.38	0.38
Naïve Bayes	0.01	0.42	0.49
LogiBoost	0.19	0.41	0.44
ID3	0.4	0.536	0.566
AdaBoost	0.2	0.638	0.644
ID3+AdaBoost	0.08	0.981	0.981

The comparative results of the proposed hybrid method with individual decision tree and ensemble classifier in terms of response time, precision and recall, are shown in Table 2. The proposed model is compared with SVM, Naïve Bayes, LogiBoost, ID3 and AdaBoost. From the obtained results, it is proved that the hybrid classifier (ID3 with AdaBoost) achieves high results in terms of precision, recall and response time. The response time take by the classifier SVM and Naïve Bayes, is relatively lower than the hybrid model, whereas LogiBoost, ID3 and AdaBoost take higher response times than the proposed work. Since the hybrid model consists of two algorithms, it takes greater response time, but achieves high performance as it can be visualize through precision and recall values.

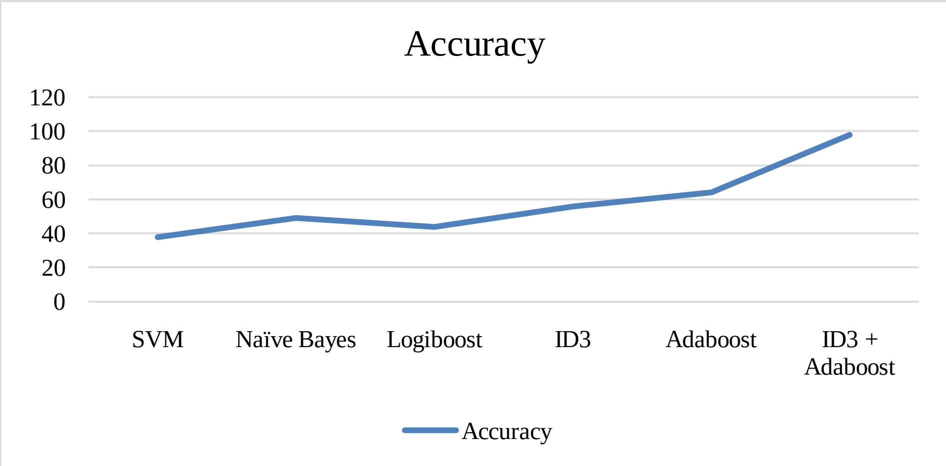

Figure 2. Comparative summary of accuracy.

In Figure 2, the accuracy of the different classifiers has been compared. It shows that the hybrid model produces results of relatively higher accuracy than the remaining classifiers. In comparison to the existing method, the maximum accuracy rates produced by the Jasim [15] is 62.66%, 65.78% and 57.33%, using the Naïve Bayes, the decision tree, and the artificial neural network, correspondingly. Ullah et al. [16] achieve accuracy rates of 72%, 84% and 81%, by using Naïve Bayes, the multilayer perceptron, and the K-star, correspondingly.

The proposed hybrid classifier is also evaluated based on scalability. The comparison for performance based on accuracy, precision, recall, F-measure and ROC area by varying scalability constant in the range of 100 units, is shown in Table 3.

Table 3. Performance comparison for hybrid classifier based on scalability.

Instance	Accuracy	Precision	Recall	F-Measure	ROC Area
100	98	98.6	98	98	100
200	99	99	99	99	100
300	100	100	100	100	100
400	95	95	95	95	99
500	98.6	98.6	98	98	99

From Table 3, it is observed that the performance measure produces only slight variations even if the system is scaled based on 100 instances for each unit. For 300 samples, this model achieves a full 100 percent result. Again, it drops slightly, then, improves again, for 500 samples. Hence, it is easy to conclude that this proposed hybrid classifier is capable of classifying a higher number of samples with better results.

4.3 Gain optimization

The gain optimization is performed using linear programming and the UCB algorithm, based on number of pieces that can be sold, and how much profit can be made under the mentioned constraints.

4.3.1 Bundling of pairs using the UCB algorithm

Among 13 input attributes, style, price, rating, size, season, and material, alone have higher weightage based on correlation analysis with respect to the target class. Significant attributes such as material, season, size and style, have been considered for the purpose of bundling pairs of dead stock with best-selling products, which determine show products are sold. For each pair, the gain is calculated, and the maximum gain is found iteratively with different products being considered. The dead stock and moving garment combo with the maximum gain is suggested to the sales department for sales offers. If the combo does not attract many customers, then the discounts and pairs can be revised, and alternative suggestions can be made. For example, a pair of two casual shirts made of cotton can be paired, where one is a best-selling item, and other is dead stock. The recommended garment from Table 4, which sells best, can be bundled with dead stock made of cotton specified in Table 5. This can help clear dead stock with optimal gain (highlighted in Table 4 and Table 5).

Table 4. Moving stock.

S. No	Style	Price	Size	Season	Material	Recommendation
1	Brief	Average	L	Spring	Silk	Yes
2	Flare	Average	Free	Spring	Cotton	Yes
3	Cute	Low	M	Spring	Chiffon	Yes
4	Casual	Low	M	Spring	Silk	Yes

Table 5. Dead stock.

S. No	Style	Price	Size	Season	Material	Recommendation
1	Casual	Low	Free	Summer	Cotton	No
2	Casual	Low	Free	Spring	Cotton	No
3	Vintage	High	L	Autumn	Polyster	No
4	Bohemian	Low	XL	Summer	Chiffon Fabric	No

For the samples taken in Tables 4 and 5, if one shirt has a cost price of Rs. 800, and is selling well, and other is a casual cotton shirt worth Rs. 1,000, and is dead stock, then a pair is bundled, made up of these two. Even if a discount of 5% on one and 4% on the other, is given, and it is sold at Rs. 2,400, still, a profit is made. The iterations using bandit learning of upper confidence bounds with Chernoff-Hoeffding inequality algorithm is used to bundle dead stock with fast-moving garments by giving optimal discounts that maximize revenue.

4.3.2 Profit optimization using linear programming

Linear programming is used to calculate how many pieces need to be sold, and how much profit can be made. Let us take up a profit optimization problem as an example, and fix the discount for a bundled product. An organization has two garment products to be bundled to optimize profit. For a total expense of Rs. 15,000, garment A costs Rs. 120 per piece, and garment B costs Rs. 210 per piece, to manufacture. A total of 75 pieces are to be sold, with 110 pieces of garment A, and 30 pieces of garment B on hand to be bundled. The profit has to be maximized, at Rs.1.30 profit per piece on garment A, and Rs. 2 profit per piece on garment B. The profit is optimized using linear programming.

The mathematical model for the above profit optimization linear programming model is given below:

Let X1 and X2 be decision variables denoting garment A and garment B.

The objective function is obtained by profit per piece. Since there are 110 pieces of garment A, a total profit of Rs. 143 can be made at Rs. 1.30 per piece. Similarly, Rs.60 can be made as profit on 30 pieces of garment B, at Rs. 2 as the profit per piece. Let the objective function to maximize profit be:

$$Max\ z = 143\ X1 + 60\ X2$$

The constraints are framed considering the cost per piece to make the product, and the number of pieces to be sold. There is also a constraint in the form of the total number of garments A and B, together. So the constraints equations are given as:

$$\text{Subject to: } 120\text{X1} + 210\text{X2} \leq 15000$$
$$110X1 + 30X2 \leq 4000$$
$$X1 + X2 = 75$$

On solving the linear programming model, we get a solution to the problem of determining the number of units to be sold of each bundled garment to attain the maximum profit. This is shown below in Table 6.

Table 6. Summary of findings of linear programming.

Product	Cost Price (Rs)	No Pieces available	Profit per piece(Rs)	No of pieces to be sold	Profit
Garment-A (X1)	120	110	1.3	21	Rs. 6,243
Garment-B (X2)	210	30	2	54	

The outcome of linear programming is this: out of the available amount Rs.15,000, and total available Rs.4,000 product, it is possible to sell 21 pieces of Garment-A and 54 pieces of Garment-B which makes a profit of Rs. 6,243. It gives a fair idea on how the bundle can be beneficial. This offers an idea for the bundling process decision making regarding how many pieces have to be sold, and how much profit can be made from the bundling process.

5. Conclusion

The data analytics model has a greater effect on managing dead stock. The decision making given on moving dead stock products based on the purchasing pattern of customers, can help to clear dresses that have the chance of turning into dead stock. The system discussed in the paper performs efficient classification and prediction of dead stock using a hybrid algorithm combining both the ID3 and the AdaBoost algorithms to attain an optimum accuracy level of 98%. The analytics over profit optimization with linear programming and bandit learning of upper confidence bound with Chernoff-Hoeffding Inequality helps to find the best bundle suggestion. This can improve the turnover of all small-scale, medium-scale and large-scale garment industries by managing dead stock efficiently. This work can be enhanced by applying other optimal algorithms for effective dead stock management in all sectors in the future.

Acknowledgments

The authors would like to thank the reviewers for their helpful comments and valuable suggestions. We thank Mrs. J.Jeyalakshmi from Rajalakshmi Engineering College for her support in implementing the R-code for compiling the results. We would like to thank Mr. Nalankilli for Grammar correction.

Conflict of interest

All authors declare no conflicts of interest in this paper.

References

1. P. Ignaciuk, *Base-stock distributed inventory management in continuous-review logistic systems-control system perspective*, 2017 22nd International Conference on Methods and Models in Automation and Robotics, (2017), 1027–1032. Available from: https://ieeexplore.ieee.org/abstract/document/8046971.
2. O. Bounou, A. E. Barkany and A. E. Biyaali, *Bayesian model for spare parts management*, 10 th International Colloquium on Logistics and Supply Chain Management, (2017), 204–208. Available from: https://ieeexplore.ieee.org/document/7962899.
3. R. Guidotti, G. Rossetti, L. Pappalardo, et al., *Market basket prediction using user-centric temporal annotated recurring sequences*, 2017 IEEE International Conference on Data Mining, (2017), 895–900. Available from: https://ieeexplore.ieee.org/abstract/document/8215574.
4. Y. Sun, C. Liang, S. Sutherland, et al., *Modeling player decisions in a supply chain game*, 2016 IEEE Conference on Computational Intelligence and Games (CIG), (2016), 1–8. Available from: https://ieeexplore.ieee.org/abstract/document/7860444.
5. G. S. Karakozov, G. B. Virabyan, S. V. Verlinski, et al., *Construction of decision support system in business design based on integration of information technology*, 2016 6th International Conference on Computers Communications and Control, (2016), 240–243. Available from: https://ieeexplore.ieee.org/abstract/document/7496767.
6. H Wani and N Ashtankar, *Big data in supply chain management*, 2017 4th International Conference on Advanced Computing and Communication Systems, (2017), 1–4. Available from: https://ieeexplore.ieee.org/abstract/document/8014602.
7. U. S. Dharmapriya, S. B. Kiridena and N. Shukla, *A review of supply network configuration literature and decision support tools*, 2016 IEEE International Conference on Industrial Engineering and Engineering Management, (2016), 149–153. Available from: https://ieeexplore.ieee.org/abstract/document/7797854.
8. D. Das, L. Sahoo and S. Datta, A Survey on Recommendation System, *Int. J. Comput. Appl.*, **160** (2017), 6–10.
9. S. Zhang, L. Yao, A. Sun et al., Deep Learning based Recommender System: A Survey and New Perspectives, *ACM Comput. Surv.*, **52** (2017), 1–35.

10. A. Mugdha and L. Vina, Survey: Collaborative Recommender Systems Using Multiclass Co-Clustering, *Int. J. Innovative Res. Comput. Commun. Eng.*, **5** (2017), 452–457.
11. S. Sehgala , S. Chaudhrya , P. Biswasa ,et al., A new genre of Recommender systems based on modern paradigms of data filtering, *Proc. Comput. Sci.*, **92** (2016), 562–567.
12. K. Kulkarni , K. Wagh, S. Badgujar, et al., A Study Of Recommender Systems With Hybrid Collaborative Filtering, *Int. Res. J. Eng. Technol.*, **4** (2016), 2216–2219.
13. S. Jeble , S. kumari and Y. Patil, Role of big data and predictive analytics, *Int. J. Autom. Log.*, **2** (2016), 307–331.
14. K. Anusha, C. Yashaswini and S. Manishankar, Segmentation of Retail Mobile Market Using HMS Algorithm, *Int. J. Electr. Comput. Eng.*, **6** (2016), 1818–1827.
15. D. S Jasim, Data mining approach and its application to dresses sales recommendation, Available from: https://www.researchgate.net/publication/293464737.
16. M. A. Ullah, A Model for Predicting Outfit Sales: Using Data Mining Methods, in *Emerging Technologies in Data Mining and Information Security*, Springer, (2019), 813.
17. U. Muhammad and A. Adeel, Dresses Attribute Sales Dataset. Available from: https://archive.ics.uci.edu/ml/datasets/dresses_attribute_sales.