EAI Endorsed Transactions
on Scalable Information Systems
Research Article EAI.EU

Predicting Probable Product Swaps in Customer Behaviour: An In-depth Analysis of Forecasting Techniques, Factors Influencing Decisions, and Implications for Business Strategies

Mohit M Rao[1] and Virendra Kumar Shrivastava[2, *]

[1, 2] Department of Computer Science and Engineering, Alliance College of Engineering and Design, Alliance University, Bangalore, India, 562106

Abstract

Introduction: Factors influencing product swap requests and predict the likelihood of such requests, focusing on product usage, attributes, and customer behaviour, particularly in the IT industry.
Objectives: Analyse customer and product data from a leading IT company, aiming to uncover insights and determinants of swap requests
Methods: Gather product and customer data, perform data processing, and employ machine learning methods such as Random Forest, Support Vector Machine, and Naive Bayes to discern the variables influencing product swap requests and apply them for classification purposes.
Results: Analysed a substantial dataset, comprising 320K product purchase requests and 30K swap requests from a prominent social media company. The dataset encompasses 520 attributes, encompassing customer and product details, usage data, purchase history, and chatter comments related to swap requests. The study compared Random Forest, Support Vector Machine, and Naïve Bayes models, with Random Forest fine-tuned for optimal results and feature importance identified based on F1 scores to understand attribute relevance in swap requests.
Conclusion: Evaluated three algorithms: support vector machine, naive Bayes, and Random Forest. The Random Forest, fine-tuned based on feature importance, yielded the best results with an accuracy of 0.83 and an F1 score of 0.86.

Keywords: Prediction, Product swap, Feature Selection, Random Forest, ranking, chi-square test, Support Vector Machine, Machine Learning, Artificial Intelligence

Received on 18 July 2023, accepted on 10 September 2023, published on 03 October 2023

doi: 10.4108/eetsis.4049

1. Introduction

Most of the companies with multiple product stack offer product swaps as an add-on to their services. Even service-based companies offer product or service swaps within a period. For example, if a customer is buying network service support from a leading service provider, they might offer to swap the product for telecom support services within a certain period.

These swap requests can also be due to various reasons. It might be due to uncertainty in business or due to inefficiency in understanding stakeholders' needs, identifying risks and mitigating them [1]. One of the important factors is unawareness of product features, The customer would have assumed telecom support a part of network service support. This might be due to vagueness in the statement of work (SOW) signed or due to miscommunication or technical inefficiency. Another important reason is the relevancy of the product with respect to time. For example, during the great reshuffle period (July - Sep 2021), LinkedIn reported a 31% increase in business-to-business (B2B) buyer job transitions [2]. Another example would be the growth of online education sector companies' posts covid, the sector witnessed

*Corresponding author. Email: virendra.shrivastava@alliance.edu.in over a 37% increase in daily registered visitors during May 2020 [3].

To address the problem, the research identifies and analyses all the factors that contribute to product swap requests. There are research and publications which forecast the optimal rate of returns for e-commerce [4]. When it comes to IT product swaps, the prediction can be a little vague due to the nature of the product or service. The customer behaviour for each product and service will be different and this plays a vital role in forecasting the product swap return. To bridge this gap, this research will drill down to a granular level and analyses the product returns for talent product suites of the firm.

The purpose of the research work is to understand the factors affecting a product swap request and to predict the probability of a customer requesting for swap. All the product attributes are matched against the customer attributes to identify pattern (if any) between these. Only two product suits are selected for this research and are mentioned as product “A” and product “B” throughout the work. Non-PII (Non-Personal Identifiable Information) attributes which are allowed for data analytics are chosen to identify the pattern. PII (Personal Identifiable Information) data is not collected, stored, or processed for this research [5]. If a pattern is identified between a product and the customer who has requested the swap, the machine learning algorithm is then used to train the data between customers who swap and customers who have not requested the swap and this model is then used to predict the new probable swap requests.

There is research which focuses on customer attributes and purchase history [6-10], but this work focuses on product usage, product attributes and customer attributes. When it comes to IT services or products, there are provisions to request for swap even after using the product for a certain period. The research work also deals with multiple variants of data, it analyses the customer attributes and behaviour and the product attributes to identify/derive a pattern of probable swaps. Attributes which contribute to swap are identified using any of the statistical methods. Clustering is done to create different buckets of customers and products, and then a pattern is figured out of it to classify future deals.

The purpose of this research is to gain insights into customer and product behaviour related to swap requests and to identify the key factors that influence such requests by utilizing machine learning algorithms. The study utilizes customer and product data obtained from a prominent IT firm. The dataset contains customer attributes, including customer behaviour, purchase history and product attributes including characteristics of products, similar product stack, product sales history and product usage information. A three-year data set spanning from 2019 to 2022 is leveraged for the research, and the data is stored in Hadoop cluster and are analysed using Apache Spark (By leveraging PySpark).

2. Motivation and Related Work

Irrespective of the industry, the problem of the user or customer wanting to change the product or service is very common. There are publications, that addressed the returns in the electronic retailing (e-tailing) industry using TabNet [11], a deep learning-based algorithm. Research evaluated Decision tree, XGBoost, LightGBM, CatBoost and TabNet and later recorded the best results and outperformed all other models. Another research paper addresses the problem using graph theory [12]. It proposes HyGraph a graphical representation to tackle customer-product return prediction. Based on the graphical representation, the paper describes an algorithm called LoGraph which finds a cluster near an input node by only looking at a small neighbourhood of this node within the graph.

Choosing the right methodology and algorithm is another key aspect of the research work [13]. The paper evaluated nine algorithms for one hundred datasets. “The result shows that a simple algorithm such as K-NN could achieve good accuracy if the optimum parameter settings are used. Also, the result shows that there exist some relationships between data sets’ characteristics and algorithms’ parameter settings, and this is used to help people decide the parameter settings for a given algorithm”. Another research work [14], discussed the practical aspects and challenges of using SVM for similar problem statements and datasets. The work demonstrated how SVM using a kernel motivated by Fisher Linear discriminant outperformed standard linear SVM for the face recognition task. Fine-tuning SVM and optimizing the parameters play a vital role in implementing SVM for practical use. Optimizing the parameters to improve the SVM classification accuracy and speed play a vital role [15]. The nature of the dataset is a key factor while choosing the algorithm [16]. SVM is also seen to be used while predicting credit default swaps prices [17]. For this research, a high dimensional dataset, which is slightly imbalanced, and skewed towards one class is used. A similar problem statement is addressed [18] using the Random Forest of oblique decision trees. The research work recorded better results for imbalanced high dimensional datasets over SVM. Similar research [19] [20] demonstrated how random forest can be fine-tuned for a high dimensional dataset with large number of features relative to the sample size. Deep learning is one of the widely used that offers variety of models to deal with textual data analysis [21-24] and image classifications [25]. Deep neural network models, such as Convolutional Neural Networks, have shown great potential in image analysis and have emerged as a powerful tool for feature extraction and classification tasks [26-28]. These networks are designed to automatically learn and extract relevant features from raw data, eliminating the need for manual feature engineering.

Accuracy, precision, recall and F1 score are measured to evaluate the performance of the model. For this work, accuracy and F1 score is equally important, while accuracy helps in analysing how many predictions were correct; precision and recall determine how good the model is at a specific category and how many times the model detected the same [29]. Using the right factors from the multi-dimensional dataset helps get better results. The work uses statistical techniques [30] [31] to evaluate the features that most contribute to product swap requests. The Chi-Square test is one of the best methods to check if two variables are associated with each other [32], Chi-Square test is used to evaluate whether there is an association between two variables and their associated p-value [33]. G-test [34], is used to determine if the features extracted by the ML model significantly differed from the derived parameters. The utilization of graphs or a minimum-spanning tree can also serve as representations for this concept. References [35-39] provide detailed descriptions of how FMST (Fuzzy Minimum Spanning Tree) and other methodologies which can be applied in this context. These approaches offer alternative perspectives and techniques for addressing the problem at hand.

3. Methodology

This research work is focused on analysing a large dataset containing 320K product purchase requests and 30K swap requests obtained from a leading social media firm. The dataset consists of 320K records, and 520 attributes related to the product purchase. This includes customer attributes, product attributes, product details, product usage data, purchase history and chatter comments related to product swap requests. The dataset is stored in the Hadoop ecosystem and is span across 5 different tables. Primary keys are defined for each table and are also used to collect the swap history and map the swaps with products and respective customers.

The data is collected in real-time in the customer relationship management tool and is stored in Hadoop. All the required data points are collected, and no further data collection exercise needs to be done for the research.

3.1. Data Pre-processing

The first task is to identify a tool that can-do large-scale data processing [40]. As HDFS is used to store data, a few options to process data were, presto, pig and spark. Apache Spark is chosen for data collection and pre-processing. Both Presto [41] and Spark are capable of processing huge workloads. Presto was designed by Facebook to process their huge workloads. Both have rich integration capabilities and connectors and have CLI as well as python modules. Spark [42] has a little upper hand in terms of processing big data in Python with the “PySpark” module. These modules consist of several predefined functions for easy data processing like the PCA function, impute function, etc. Apache spark also has better ML libraries compared to Presto ML.

Apache Pig [43] was another option for processing big data. But compared to Spark, Pig has lesser runtime capability and performance speed [44]. Pig also has limitations in scalability as well as lacks direct machine learning capabilities. So, considering all these factors, Apache spark is implemented as shown in Figure 1. However, this data is not ready for data analytics, there are missing data points, too many dimensions, skewness and the data points are getting stored in multiple tables and systems. This is then aggregated and cleaned using different techniques mentioned below. The processed dataset is then passed to Machine Learning algorithm for training.

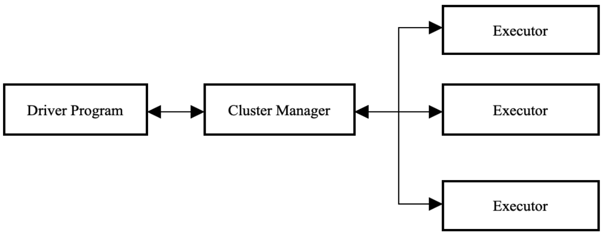

Figure 1. Apache Spark with multiple executors

3.2. Data Aggregation and Data Cleaning

The data required for this research work is fetched from multiple tables, and the first task is to aggregate the data. This is divided into three parts,

- Customer attributes and product attributes of customers who purchased product A and Customers who requested the swap from product A to B
- Customer attributes and product attributes of customers who purchased product B and requested the swap from B to A
- Customer usage history for products A and B (before and after swap)

The above information is fetched by filtering and joining multiple tables from HDFS. The first step in data cleaning is to fix the structural errors, as data has been pulled from multiple tables, there are inconsistencies; naming conventions, typos, incorrect formats etc. These inconsistencies cause mislabelled categories or classes. For example, if an attribute does not apply to a row, this is mentioned as “NA”, “Not applicable” or simply left blank. The next step is to remove duplicates, there are a lot of duplicate records added while joining tables.

Another important challenge in data cleaning is handling the missing values, there are multiple options for the same, including deleting the entire row, replacing with mean or mode, assuming the attribute as a unique category or predicting the missing value using some algorithms. For this work, all the numerical variables are replaced with the mean of a similar “customer industry”, this field is chosen as per expert advice. Categorical features are replaced with a new category “unknown”.

3.3. Data Aggregation and Data Cleaning

As the information is fetched from multiple tables, no attributes have been avoided during the data collection process. Each of the above sets of data has over five hundred columns which leads to the curse of dimensionality. This brings the need for dimensionality reduction techniques, to transform the data from a high-dimensional space into a lower-dimensional space so that it retains some meaningful properties of the original data.

Feature Selection v/s Feature Extraction:

There are a couple of methods to reduce the dimensionality, this work discusses the option of choosing between a feature section or feature projection (feature extraction) to reduce the dimensionality. While feature selection filters irrelevant or redundant features from the dataset, feature selection keeps a subset of original features. Feature projection or feature extraction creates a new, smaller set of features which still captures useful information from the data. There are supervised algorithms that have built-in feature selection capabilities like the random forest, but considering the variety of data, it is better to choose a stand-alone feature selection method like variance threshold or correlation threshold. These methods will remove the features that don't add much value to the target, but at the same time require manually setting or tuning a threshold, which can be tricky for our data with over five hundred features. Setting the threshold too low will remove useful features while setting the threshold high will add unwanted features, setting an optimum number of thresholds will be tricky.

Like feature selection, there are algorithms which already have built-in feature extraction, which extract increasingly useful representations of raw input through each hidden neural layer. In this research work, the dataset is not a straightforward simple text dataset. Hence, stand-alone feature extraction methods like Principal component Analysis (PCA) or Linear Discriminant Analysis (LDA) [45] are more suited. Linear Discriminant Analysis is a supervised technique and needs labelled data which makes the process more tedious and situational. Product swaps for all customers won't contain enough data points for training. Principal component Analysis on the other hand is an unsupervised algorithm that creates a linear combination of original features. Dimensionality can be reduced by limiting the number of principal components to keep based on cumulative explained variance. It is faster and easy to implement [46], which means the testing of the algorithms are done in short time. It also offers several variations and extensions and is available as a built-in function within the pyspark python library.

The whole data processing steps is done using Apache Spark by leveraging "pyspark" and other python modules. Hosted notebook with GPU is used to connect to HDFS and performs data cleaning and data transformation. Spark also has a UI feature to monitor the execution of tasks.

3.4. Unforeseen Risks

PCA is used for dimensionality reduction for the dataset which has around 498 columns (excluding ID's, keys, names, and other unimportant fields). This looks possible and normal with pyspark, but most of these columns are string type and are categorical variables. There are a couple of major challenges in implementing the same with PCA.

- PCA is known to be efficient for continuous variables and not that great for categorical data [47].
- After applying one-hot encoding, there are around 900000 columns, new columns are created from each category in categorical variables. Pyspark can't process more than 65535 features [48].

There are alternate approaches to reducing the dimensions. The random forest algorithm, which has an inbuilt dimensionality reduction mechanism is a good option. Regardless of the dimensionality reduction technique being used, there are still 900 thousand columns and pyspark won't be able to process such a huge volume of data. This leads to incorrect predictions. There are multiple options to mitigate this risk; count vectorizer, frequency encoding, and choosing variables manually from a distribution plot are a few to mention [49]. The earlier options choose only the majority of occurrences from a given field. For example, if a column has one thousand values, and if two values make up 70% of the data, frequency encoding will only have three distinct values, it will combine all other values into a single datapoint. This leads to feeding incorrect information to the model, especially while processing a huge volume of sales data. Sometimes a minimum occurrence of a value will have high weightage in the predicting outputs.

The second option of plotting the number of distinct values in categorical variables and manually emitting the columns from the dataset. There were around twelve fields with more than 70K distinct values. These were mostly free text fields, and eliminating such categorical variables mitigated the risk of having too many one-hot encoded categorical columns for processing.

3.5. Classification Algorithms

Once the data pre-processing tasks are complete, the next step is to gain insights into the attributes that affect swaps and to derive a pattern between all historical contracts and swaps. As the scope is limited to two products, the swap data is limited to less than 100K records and much lesser dimensions than the original dataset. This brings down the options. Based on the dataset characteristics [50], the options are narrowed down to the Decision Tree, Random Forest, Naïve Bayes, Support Vector Machines and Stochastic Gradient Descent. Considering the sample size, Stochastic Gradient Descent [51] is more appropriate for large samples and requires a few numbers of hyperparameters and it is sensitive to feature scaling. The decision tree is a simple classification algorithm which can handle both numerical and categorical data, but

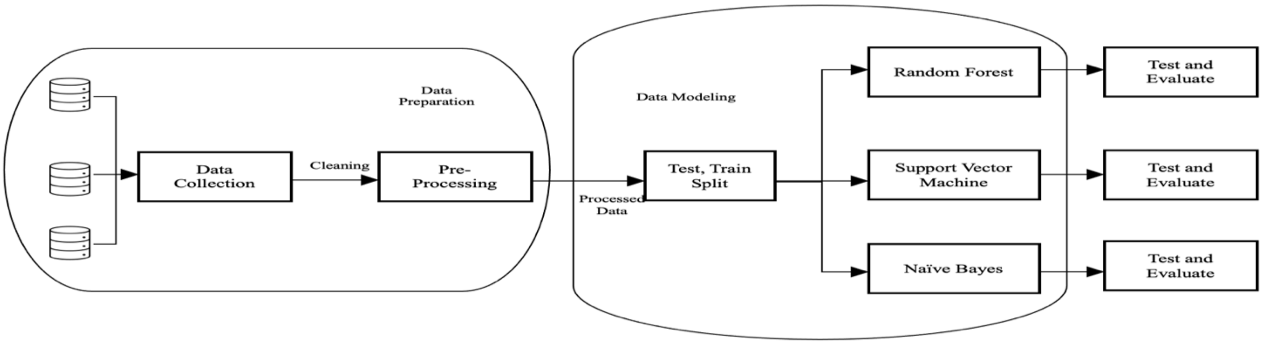

Figure 2. Data Modelling Methodology

considering the dimensions, even after principal component analysis, the decision tree won’t be stable. Even small variations of data will affect the results. Multiple classification models are evaluated with this training dataset. and the one which gives the best result is chosen as shown in Figure 2.

Considering Naïve Bayes, Support vector machine and Random Forest [52] algorithms, all these fit very well for a smaller amount of training data (between 75K-150K records) [53]. While Naïve Bayes is faster compared to the random forest, the random forest has built-in dimensionality reduction capabilities and can reduce over-fitting. Support vector machine on the other hand works well with high dimensional spaces compared to both naïve Bayes and the random forest. Support vector machine is a representation of the training data as points in space separated into categories by a clear gap that is as wide as possible. It’s effective in high dimensional spaces and uses a subset of training points in the decision function so it is also memory efficient [54].

To validate the above assumptions and theories, a random sample dataset of five thousand records from the master dataset is extracted and used to evaluate all three classification algorithms. The extracted dataset is marked as the training dataset, this is used for training and to fit the model. The model learns from this dataset. A subset of this training dataset is utilized to compute the model performance. This is marked as the test dataset. Another subset of the training dataset is utilized as the Validation dataset. This gives an estimate of model performance while fine-tuning the model's hyperparameters. One thousand records are marked for the test dataset from five thousand records in the training dataset. A low-code platform open-source tool called Orange [55] was used for faster and easier ML testing. With Orange, machine learning methods and functions are transformed as simple drag-and-drop elements and the same is used to train, test, validate and evaluate the model.

The workflow created in orange using drag and drop features. A sample data of five thousand random data is taken from the master dataset and post data pre-processing (Cleaning and transformation) it has been fed to three models for evaluation. Accuracy, precision, and recall are evaluated for each model.

Table 1. Orange Workflow Evaluation Results

Model	Accuracy	Precision	Recall
RF	0.846	0.846	0.846
SVM	0.812	0.869	0.812
NB	0.735	0.804	0.763

There were challenges in implementing principal component analysis due to the increased number of categorical variables and exponential one-hot encoded attributes. Even though dimensions were reduced by plotting the distribution of distinct values and manually eliminating the variables, it is noted that another dimensionality reduction technique that works well with categorical variables needs to be implemented. As seen in Table 1, the support vector machine and the random forest have similar performance scores, the advantage of using the support vector machine is the capability of providing better results with more dimensions. On the other hand, the Random Forest algorithm is more efficient with reducing dimensions and works well with high-dimensional datasets.

Another major challenge in the dataset is the skewness. Out of the whole dataset, there are 23% of the dataset represents class 1 (swap) and 77% of the dataset represents class 2 (non-swap). This creates bias and will result in predicting wrong results. To avoid this, the best possible method is oversampling. There are various methods to perform oversampling, Random oversampling, SMOTE (Synthetic Minority Over-sampling Technique), ADASYN (Adaptive Synthetic Sampling), Borderline-SMOTE, SMOTE-ENN. SMOTE is used for the research as it SMOTE creates synthetic samples by interpolating between minority class which is swapped data. It generates new samples by identifying nearest neighbours in the feature space and creating synthetic examples along the line segments connecting them, thus increasing the ratio between class 1 and class 2. Cross validator is used with random forest, this will help in testing the model’s ability to predict new datasets which were not used in estimating. This will also help mitigate challenges related to selection bias and overfitting. The whole dataset is divided into five folds (k=5) where five models are represented, and each model is built on four parts and tested in the fifth one.

4. Results and Discussion

To evaluate the above assumption in choosing the Random Forest model, the model is compared with the support vector machine and the Naïve Bayes. Seventy-five thousand records are considered for evaluating the final model without excluding any dimensions for testing. Complete data cleaning and dimensionality reduction techniques are incorporated and executed. Apache Spark is used for conducting the evaluation and the results are reported in Table 2.

Table 2. Model Evaluation

Model	Accuracy	Precision	Recall
RF	0.832	0.872	0.846
SVM	0.825	0.864	0.823
NB	0.752	0.804	0.763

Fine-tuning the Random Forest model by selecting the right features contributed to the model achieving the best results. Understanding which features contributes to swap is another important goal of the research. Feature importance functionality in the random forest algorithm is used and features are ranked based on the F1 score. This maps attribute to the product swap request.

To ensure that our model is not a random guesser but indeed classifying the data as expected, statistical tests are conducted. A combination of categorical and numerical features is ranked by the model, these are validated together or separately.

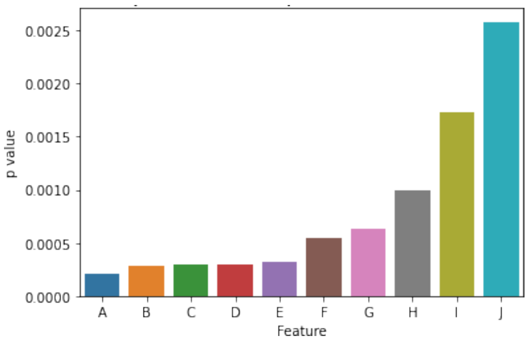

Figure 3. Sample features with p-value

The Chi-square test is a common test which is used to validate the relation between categorical features. Kruskal- Wally's test [56], G-test are a few other options. The goal is to compare the model against the results generated from the Chi-Square test [57]. Features recommended by both the model and those derived from the Chi-square test are recorded. The features were ranked using feature engineering in the random forest and compared the features with the results from the Chi-Square test. Sample features with corresponding p-values to check for association is shown in Figure 3. There was a similarity in both evaluations and hence, the null hypothesis was rejected (the null hypothesis states that the model is a random guesser). This completes the validation of the model.

5. Conclusion and Future Work

The research was designed to identify the factors affecting the swap (limited to scope) and to predict the probability of new swaps. The research work was kicked off to explore the possibility, identify the datasets, transform, and clean the dataset and use the dataset to arrive conclusion. As a part of the research, three algorithms were evaluated. The support vector machine, naive Bayes, and the Random Forest algorithm. Both the support vector machine and the random forest algorithm gave better results. Considering the nature of the dataset and leveraging the feature importance functionality, the random forest was fine-tuned to cater for the needs of the dataset. Factors affecting the swap are identified from the dataset (for products in scope) and have evaluated the performance of the model. These features were validated using the Chi-Square test to verify if the association exists. The model performed at an accuracy of 0.83, and an F1 score of 0.86.

In this research study, the focus was on analysing product and customer attributes, while support cases were not included in the analysis. To enhance the analysis, customer sentiments can be analysed using chatter comments and case logs, which can be incorporated into the random forest algorithm for re-training. The dataset encompassed pre-covid, covid, and post-covid data points, capturing significant changes in various business areas. However, it is important to note that these changes were only considered through the nature of features, and in future research, it would be beneficial to address them separately in the post-covid era. Given the limited scope of the dataset, restricted to a specific firm and product suite, it is crucial to recognize that the nature of customer and product attributes may differ significantly across other products or services.

To broaden the scope of future research, it is recommended to incorporate customer sentiments collected through surveys, support cases raised, and data collected during the product onboarding process. This would provide deeper insights into customer/user sentiments and their correlation with product swap requests. Additionally, considering the significant shifts in business dynamics post-covid, conducting a comparison between pre- and post-covid swap requests could reveal valuable insights into any changes in customer behaviour.

6. References

1. Rao, M. M. (2022). Transformation story of a new manager!. India: Amazon Digital Services LLC – kdp.
2. https://www.linkedin.com/business/sales/blog/modern-selling/infographic-great-reshuffle-affect-on-selling. Accessed 22 Nov 2022
3. Sikandar, M. A., & Rahman, P. F. (2021). Edtech Start-ups in the education ecosystem in the post-Covid-19 era in India. Towards Excellence: Journal of Higher Education, UGC-HRDC, Gujarat University, India.
4. Urbanke, P., Kranz, J., & Kolbe, L. (2015). Predicting product returns in e-commerce: the contribution of mahalanobis feature extraction.
5. Parra-Frutos, I. (2009). The behaviour of the modified Levene’s test when data are not normally distributed. Computational Statistics, 24(4), 671-693.
6. Kedia, S., Madan, M., & Borar, S. (2019). Early bird catches the worm: Predicting returns even before purchase in fashion E-commerce. arXiv preprint arXiv:1906.12128.
7. Bonifield, C., Cole, C., & Schultz, R. L. (2010). Product returns on the internet: a case of mixed signals. Journal of Business Research, 63(9- 10), 1058-1065.
8. Harris, L. C. (2010). Fraudulent consumer returns: exploiting retailers' return policies. European Journal of Marketing.
9. Chen, J., & Bell, P. C. (2009). The impact of customer returns on pricing and order decisions. European Journal of Operational Research, 195(1), 280-295.
10. Ma, J., & Kim, H. M. (2016). Predictive model selection for forecasting product returns. Journal of Mechanical Design, 138(5), 054501.
11. Al Imran, A., & Amin, M. N. (2020). Predicting the return of orders in the e-tail industry accompanying with model interpretation. Procedia Computer Science, 176, 1170-1179.
12. Zhu, Y., Li, J., He, J., Quanz, B. L., & Deshpande, A. A. (2018, July). A Local Algorithm for Product Return Prediction in E-Commerce. In IJCAI (pp. 3718-3724).
13. Zhongguo, Y., Hongqi, L., Ali, S., & Yile, A. (2017). Choosing classification algorithms and its optimum parameters based on data set characteristics. Journal of Computers, 28(5), 26-38.
14. Wang, L. (Ed.). (2005). Support vector machines: theory and applications (Vol. 177). Springer Science & Business Media.
15. Liao, J., & Bai, R. (2008, December). Study on the performance support vector machine by parameter optimized. In International Conference on Advanced Software Engineering and Its Applications (pp. 79-92). Springer, Berlin, Heidelberg.
16. Bartlett, P., & Shawe-Taylor, J. (1999). Generalization performance of support vector machines and other pattern classifiers. Advances in Kernel methods—support vector learning, 43-54.
17. Gündüz, Y., & Uhrig-Homburg, M. (2011). Predicting credit default swap prices with financial and pure data-driven approaches. Quantitative Finance, 11(12), 1709-1727.
18. Do, T. N., Lenca, P., Lallich, S., & Pham, N. K. (2010). Classifying very-high-dimensional data with random forests of oblique decision trees. In Advances in knowledge discovery and management (pp. 39- 55). Springer, Berlin, Heidelberg.
19. Te Beest, D. E., Mes, S. W., Wilting, S. M., Brakenhoff, R. H., & van de Wiel, M. A. (2017). Improved high-dimensional prediction with random forests by the use of co-data. BMC bioinformatics, 18(1), 1- 11.
20. Kursa, M. B., & Rudnicki, W. R. (2011). The all relevant feature selection using random forest. arXiv preprint arXiv:1106.5112.
21. Shrivastava, V.K., Shrivastava, A., Sharma, N., Mohanty, S.N., & Pattanaik, C.R. (2022). Deep learning model for temperature prediction: an empirical study. Model. Earth Syst. Environ.
22. Shrivastava, V. K., Kumar, A., Shrivastava, A., Tiwari, A., Thiru, K., & Batra, R. (2021, August). Study and trend prediction of Covid-19 cases in India using deep learning techniques. In Journal of Physics: Conference Series (Vol. 1950, No. 1, p. 012084). IOP Publishing.
23. Batra, R., Mahajan, M., Shrivastava, V. K., & Goel, A. K. (2021). Detection of COVID-19 Using Textual Clinical Data: A Machine Learning Approach. In Impact of AI and Data Science in Response to Coronavirus Pandemic (pp. 97-109). Springer, Singapore.
24. Saini, V., Rai, N., Sharma, N., & Shrivastava, V. K. (2022, December). A Convolutional Neural Network Based Prediction Model for Classification of Skin Cancer Images. In International Conference on Intelligent Systems and Machine Learning (pp. 92-102). Cham: Springer Nature Switzerland.
25. Batra, R., Shrivastava, V. K., & Goel, A. K. (2021). Anomaly Detection over SDN Using Machine Learning and Deep Learning for Securing Smart City. In Green Internet of Things for Smart Cities (pp. 191-204). CRC Press.
26. Saini, V., Rai, N., Sharma, N., & Shrivastava, V. K. (2022, December). A Convolutional Neural Network Based Prediction Model for Classification of Skin Cancer Images. In International Conference on Intelligent Systems and Machine Learning (pp. 92-102). Cham: Springer Nature Switzerland.
27. Singhal, A., Phogat, M., Kumar, D., Kumar, A., Dahiya, M., & Shrivastava, V. K. (2022). Study of deep learning techniques for medical image analysis: A review. Materials Today: Proceedings, 56, 209-214.
28. Lalli, K., Shrivastava, V. K., & Shekhar, R. (2023, April). Detecting Copy Move Image Forgery using a Deep Learning Model: A Review. In 2023 International Conference on Artificial Intelligence and Applications (ICAIA) Alliance Technology Conference (ATCON-1) (pp. 1-7). IEEE.
29. Streiner, D. L., & Norman, G. R. (2006). “Precision” and “accuracy”: two terms that are neither. Journal of clinical epidemiology, 59(4), 327- 330.
30. Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society: Series B (Methodological), 39(1), 1-22.
31. Rykov, V. V., Balakrishnan, N., & Nikulin, M. S. (Eds.). (2010). Mathematical and statistical models and methods in reliability: applications to medicine, finance, and quality control. Springer Science & Business Media.
32. Yates, F. (1934). Contingency tables involving small numbers and the χ 2 test. Supplement to the Journal of the Royal Statistical Society, 1(2), 217-235.
33. Rana, R., & Singhal, R. (2015). Chi-square test and its

application in hypothesis testing. Journal of the Practice of Cardiovascular Sciences, 1(1), 69.

34. Hoey, J. (2012). The two-way likelihood ratio (G) test and comparison to two-way chi squared test. arXiv preprint arXiv:1206.4881.

35. Dey, A., Mondal, S., & Pal, T. (2019). Robust and minimum spanning tree in fuzzy environment. International Journal of Computing Science and Mathematics, 10(5), 513-524.

36. Mohanta, K., Dey, A., Pal, A., Long, H. V., & Son, L. H. (2020). A study of m-polar neutrosophic graph with applications. Journal of Intelligent & Fuzzy Systems, 38(4), 4809-4828.

37. Mohanta, K., Dey, A., & Pal, A. (2021). A note on different types of product of neutrosophic graphs. Complex & Intelligent Systems, 7, 857-871.

38. Deli, I., Long, H. V., Son, L. H., Kumar, R., & Dey, A. (2020). New expected impact functions and algorithms for modeling games under soft sets. Journal of Intelligent & Fuzzy Systems, 39(3), 4463-4472.

39. Dey, A., Agarwal, A., Dixit, P., Long, H. V., Werner, F., Pal, T., & Son, L. H. (2019). A genetic algorithm for total graph coloring. Journal of Intelligent & Fuzzy Systems, 37(6), 7831-7838.

40. Khatri, I., & Shrivastava, V. K. (2016). A survey of big data in healthcare industry. In Advanced Computing and Communication Technologies (pp. 245-257). Springer, Singapore.

41. Sethi, R., Traverso, M., Sundstrom, D., Phillips, D., Xie, W., Sun, Y., Berner, C. (2019, April). Presto: SQL on everything. In 2019 IEEE 35th International Conference on Data Engineering (ICDE) (pp. 1802- 1813). IEEE.

42. Shaikh, E., Mohiuddin, I., Alufaisan, Y., & Nahvi, I. (2019, November). Apache spark: A big data processing engine. In 2019 2nd IEEE Middle East and North Africa COMMunications Conference (MENACOMM) (pp. 1-6). IEEE.

43. Swarna, C., & Ansari, Z. (2017). Apache Pig-a data flow framework based on Hadoop Map Reduce. International Journal of Engineering Trends and Technology (IJETT), 50(5), 271-275.

44. Jankatti, S., Raghavendra, B. K., Raghavendra, S., & Meenakshi, M. (2020). Performance evaluation of Map-reduce jar pig hive and spark with machine learning using big data. International Journal of Electrical and Computer Engineering, 10(4), 3811.

45. Martinez, A. M., & Kak, A. C. (2001). Pca versus lda. IEEE transactions on pattern analysis and machine intelligence, 23(2), 228- 233.

46. Shereena, V. B., & David, J. M. (2015). Comparative Study of Dimensionality Reduction Techniques Using PCA and LDA for Content Based Image Retrieval. Computer Science & Information Technology, 41.

47. Chavent, M., Kuentz-Simonet, V., Labenne, A., & Saracco, J. (2014). Multivariate analysis of mixed data: The R package PCAmixdata. arXiv preprint arXiv:1411.4911.

48. Hryhorzhevska, A., Wiewiórka, M., Okoniewski, M., & Gambin, T. (2017). Scalable framework for the analysis of population structure using the next generation sequencing data. In Foundations of Intelligent Systems: 23rd International Symposium, ISMIS 2017, Warsaw, Poland, June 26-29, 2017, Proceedings 23 (pp. 471-480). Springer International Publishing.

49. Batra, R., Shrivastava, V. K., & Goel, A. K. (2021). Anomaly Detection over SDN Using Machine Learning and Deep Learning for Securing Smart City. In Green Internet of Things for Smart Cities (pp. 191-204). CRC Press.

50. Nagalla, R., Pothuganti, P., & Pawar, D. S. (2017). Analyzing gap acceptance behavior at unsignalized intersections using support vector machines, decision tree and random forests. Procedia Computer Science, 109, 474-481.

51. Ketkar, N. (2017). Stochastic gradient descent. In Deep learning with Python (pp. 113-132). Apress, Berkeley, CA.

52. Ye, Y., Wu, Q., Huang, J. Z., Ng, M. K., & Li, X. (2013). Stratified sampling for feature subspace selection in random forests for high dimensional data. Pattern Recognition, 46(3), 769-787

53. Singh, A., Halgamuge, M. N., & Lakshmiganthan, R. (2017). Impact of different data types on classifier performance of random forest, naive bayes, and k-nearest neighbors algorithms. International Journal of Advanced Computer Science and Applications, 8(12).

54. Ralaivola, L., & d'Alché-Buc, F. (2001, August). Incremental support vector machine learning: A local approach. In International conference on artificial neural networks (pp. 322-330). Springer, Berlin, Heidelberg.

55. Demšar, J., Curk, T., Erjavec, A., Gorup, Č., Hočevar, T., Milutinovič, M., ... & Zupan, B. (2013). Orange: data mining toolbox in Python. the Journal of machine Learning research, 14(1), 2349-2353.

56. Ostertagova, E., Ostertag, O., & Kováč, J. (2014). Methodology and application of the Kruskal-Wallis test. In Applied Mechanics and Materials (Vol. 611, pp. 115-120). Trans Tech Publications Ltd.

57. Plackett, R. L. (1983). Karl Pearson and the chi-squared test. International statistical review/revue internationale de statistique, 59-72.