

UNIVERSITY OF SOUTHAMPTON

Southampton Business School

MSc Dissertation

MANG6545- Dissertation (Analytics)

**Modelling product returns behaviour at the
level of individual products**

**ERGO Submission ID:
75420**

**Student ID:
32519001**

Word count: 16491

Table of Contents

1	<i>Introduction</i>	10
1.1	Significance of Project Topic	10
1.2	Background	14
1.3	Aim and Objectives	16
2	<i>Literature Review</i>	18
2.1	Overview	18
2.2	Literatures:	20
2.2.1	Prediction using Advanced Analytics:	20
2.2.2	Evaluating Model Performance	30
2.2.3	Predicting product return based on different product characteristics:	34
2.2.4	Other researches related to product returns	37
2.3	Conclusion:	41
3	<i>Model Selection and Overview of Models</i>	43
3.1	Model Selection	43
3.2	Modelling using LightGBM:	45
3.3	Modelling using RandomForest:	49
3.4	Logistic Regression:	51
3.5	Introduction to Data:	54
4	<i>Methodology:</i>	57
4.1	Data Collection	58
4.2	Data Cleaning using PostgreSQL	59
4.3	Establishing connection with IDE	63
4.4	Data Pre-processing using Python	64
4.5	Exploratory Data Analysis	66
4.5.1	Data Exploration	66
4.5.2	Data Transformation	70
4.5.3	Data Cleaning: Outlier Detection, Treatment, and Verification:	71
4.6	Model Creation and Evaluation	79
4.6.1	Splitting data:	79
4.6.2	Model fitting and prediction:	81
4.6.3	Prediction	84
4.6.4	Important Features	84
4.6.5	Cross-validation score	85
4.6.6	Model Accuracy (Training and Testing Accuracy)	86
4.6.7	AUC Score	88
4.6.8	ROC Curve	89

4.6.9	Prediction Probability	89
4.6.10	Confusion Matrix.....	90
4.6.11	Classification Report	92
4.6.12	SHAP values.....	93
5	<i>Result and Analysis.....</i>	99
5.1	Prediction results	99
5.2	Model accuracy score.....	102
5.2.1	Cross-Validation (CV) score	102
5.3	AUC score	105
5.4	ROC Curve.....	106
5.5	Confusion Matrix	107
5.6	Classification Report	109
5.7	Analysis of Model.....	111
5.8	Feature Importance.....	112
5.9	SHAP Values and Plots.....	113
5.9.1	SHAP Values:	113
5.9.2	Summary plot:	114
5.9.3	Dependence plot	115
5.9.4	Force Plot.....	116
5.9.5	Waterfall Plot.....	117
5.9.6	Decision Plot	117
5.10	Insights Using Visualisation:	119
6	<i>Conclusion.....</i>	124
6.1	Managerial Implications	126
7	<i>Discussion</i>	128
7.1	Limitation and Future Research	129
8	<i>References</i>	132
9	<i>Appendix:.....</i>	145

Table of Figures

Figure1.1: Global Ecommerce Sales	11
Figure1.2: Blended Return Rate	11
Figure1.3: Return Order Process- Without Quality Check step	12
Figure1.4: Environmental Impact of Product Returns	13
<i>Figure 2.1: Recent publications and opportunities in Retail Analytics</i>	21
Figure2.2: Overview of Methodology process	22
<i>Figure2.3 : Effect of product category and retailer on returns.....</i>	23
Figure2.4: Predicting returns based on three scores.....	25
<i>Figure2.5: Online vs Offline return rate based on product category ...</i>	26
<i>Figure2.6: Comparison of scores of two algorithms</i>	28
Figure2.7: Performance of TreeSHAP	32
Figure2.8: Time complexity comparison in feature selection	33
<i>Figure2.9: Comparison of product returners and non-keepers.....</i>	35
<i>Figure2.10 :Return reasons of products</i>	36
<i>Figure2.11: Big data and advanced analytics in retail.....</i>	40
<i>Figure3.1: Overview of use of different types of analytics</i>	43
<i>Figure3.2: Flow diagram of ML algorithm selection based on data....</i>	44
Figure3.3 Working of LightGBM algorithm	46
<i>Figure3.4: Generic workflow of LightGBM algorithm.....</i>	46
<i>Figure3.5: Equation of information entropy in LightGBM.....</i>	47
Figure 3.6: RandomForest Classifier	49
<i>Figure3.7: Working of Bagging Process</i>	50
<i>Figure3.8: Representation of Logistic Regression.....</i>	52
<i>Figure3.9: Representation of Sigmoid Function.....</i>	52
Figure4.1: Overview of steps in Model Creation	57
Figure4.2: Data types before label encoding	67
Figure 4.3: Data types after label encoding	67
<i>Figure 4.4: Sample of data before cleaning</i>	67
Figure 4.5: Sample of data after cleaning	68
<i>Figure4.6: Null values count before data cleaning.....</i>	68
Figure4.7: Count of Null values after data cleaning.....	68
Figure4.8: Data Info before transformation	69
<i>Figure4.9: Data Info after transformation.....</i>	69
Figure 4.10: Output of 'describe' function on dataset.....	70
Figure4.11: Output of 'Describe' function after data transformation ...	70
Figure 4.12: Output of 'Describe' function after transformation	71
Figure4.13: Boxplots of independent variables before data cleaning .	73
Figure4.14: Representation of Z-score	74
Figure 4.15: Z-scores of continuous parameters before data cleaning	75

<i>Figure 4.16: Boxplots of independent variables of model after data cleaning.....</i>	<i>77</i>
<i>Figure4.17: Combined plot of all independent variables before cleaning.....</i>	<i>77</i>
Figure4.18: Combined plot of all independent variables after cleaning.....	77
Figure4.19: Z-scores of continuous parameters after data cleaning...	78
<i>Figure4.20: Representation of Training and Testing Accuracy.....</i>	<i>87</i>
Figure4.21: Accuracy Score	88
Figure4.22: Confusion Matrix Diagram	91
Figure4.23: Representation of SHAP (Mage, 2021)	94
Figure5.1: Prediction/Classification using LightGBM	100
Figure5.2: Prediction/Classification using RandomForest.....	100
Figure5.3: Prediction/Classification using Logistic Regression.....	101
Figure5.4: Scores-LightGBM	102
Figure5.5: Scores- RandomForest	102
Figure5.6: Scores- Logistic Regression.....	102
Figure5.7: CV score of 10-folds (LightGBM)	103
Figure5.8: CV score summary of folds (LightGBM)	103
Figure5.9: CV score (RandomForest)	103
Figure5.10: CV score summary (RandomForest)	103
Figure 5.11: CV score (Logistic Regression)	104
Figure 5.12: CV score summary (Logistic Regression).....	104
Figure5.13: AUC score- LightGBM.....	105
Figure5.14 :AUC- RandomForest.....	105
Figure5.15: AUC- Logistic Regression.....	105
Figure5.16: ROC Curve- LightGBM	106
Figure5.17: ROC Curve- RandomForest.....	106
Figure5.18: ROC Curve- Logistic Regression	106
Figure5.19: Confusion Matrix- LightGBM	107
Figure5.20: Confusion Matrix- RandomForest	108
Figure5.21: Confusion Matrix- Logistic Regression	108
Figure5.22: Classification Report- LightGBM	109
Figure5.23: Classification Report- RandomForest.....	110
Figure5.24: Classification Report- Logistic Regression	110
Figure5.25: Plot for Important Feature	112
Figure5.26: Important Features scores.....	112
Figure5.27: SHAP Values	113
Figure5.28: Summary Plot-1	114
Figure5.29: Summary Plot-2.....	115
Figure5.30: Dependence Plots.....	116
Figure5.31: Force Plot	116
Figure5.32: Waterfall Plot.....	117
Figure5.33: Decision Plot.....	118

Figure5.34: Footwears with highest Return Rates.....	119
Figure5.35: Brooks Products with high return.....	120
Figure5.36: VANS Products returns	121
Figure 5.37: Products with highest return rates.....	122
Figure5.38 Products with highest Sales	122

ABSTRACT

Product returns is a huge challenge for retailers as it causes huge financial loss and concerns related to resource and inventory management. Returns cannot be abolished altogether as it will negatively impact the overall sales but certainly using technology can help to mitigate the challenge to certain extent. The purpose of the analysis implemented is to classify whether product will be returned or not based on characteristics of product, identify probability of classification, identifying potential parameters influencing model's outcome, and to develop insights in return data of products. The data used for analysis is provided by Appriss Retail Limited which stores and manages data of its retailer clients, data comprises product description, purchase behaviour, return's data, transaction level data and others over a period of 4-5 years. Relevant data is identified and collected at product-level in a table which will be pre-processed and once ready is fed into machine learning (ML) models. Three supervised learning classification algorithms are used in this dissertation (project) namely, LightGBM, RandomForest, and Logistic Regression. Out of the three algorithms, LightGBM fits data most appropriately and can predict with greater accuracy and with probability of predicted outcome and enables feature selection.

Acknowledgement

I would like to thank my dissertation supervisor Steffen Bayer for guiding, providing assistance, and prompt turnaround time whenever I required his support.

Secondly, I would like to acknowledge the internship opportunity provided to me which was great learning experience and would like to express my gratitude towards Fin Baurer from Appriss for his all-around support. I would like to thank my family and friends for supporting me.

CHAPTER 1

INTRODUCTION

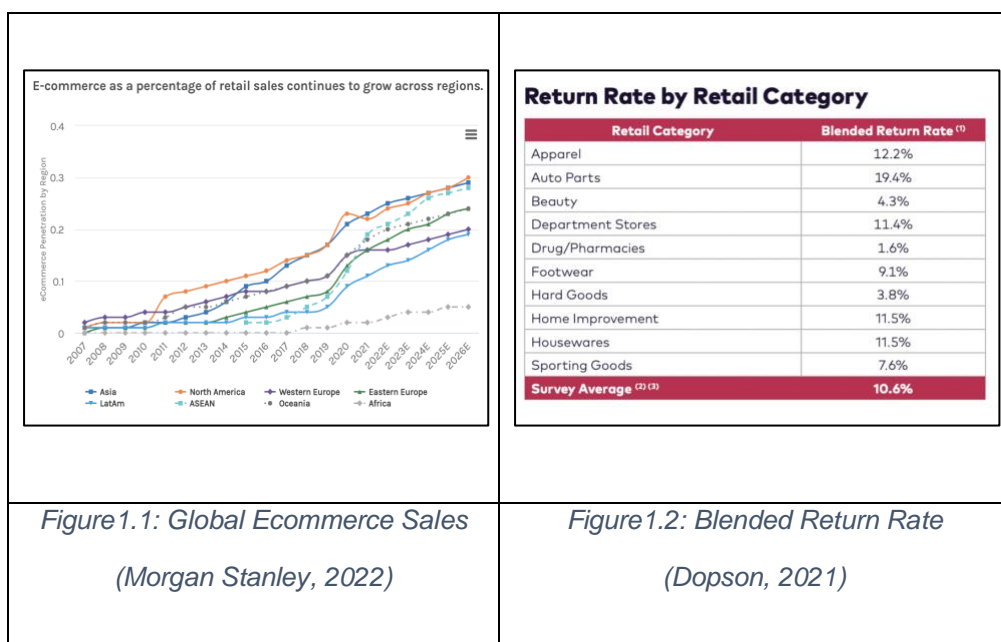
1 Introduction

1.1 Significance of Project Topic

In today's emerging market, retailers and wholesalers are crucial links in a modern economy where people can access goods at competitive prices (Telma, no date). The exponential growth in retail has brought pace in economy, employment growth, and enhanced customer experience across regions, for e.g., UK's retail output in 2020 was 97 billion pounds with 3 million employed in 2019 (Georgina, 2021). As per the article published in 2021, online sales alone is expected to be worth \$6.5 trillion by 2022 (Frei and Jack, 2021). The recent pandemic event has changed purchasing behaviour and preference for online channels is higher than before the pandemic (Sides and Lupine, 2022). The digital revolution has given modern consumers access to unparalleled convenience and tailored engagements which will bring exponential growth in online retail (dataclarity, 2021).

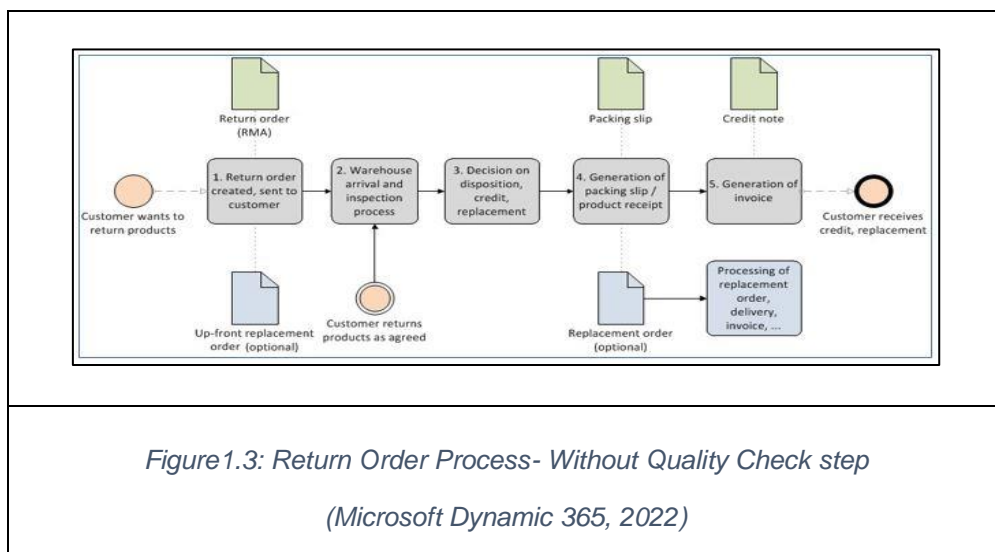
With growing multichannel retail sector there is a huge competition among retailers to benefit from this market, attract and retain consumers and one such attempt is 'product return policy'. Product return is not a new concept, in-store retailers had to deal with these returns earlier too, which in return developed customer satisfaction and loyalty. But with emergence of online stores (e-tailers) predicted to be 95% by 2040 (Frei and Jack, 2021), the huge competition between omnichannel retailers dominated by online retailers has led to lenient return policies in order to expand customer base and promote customer loyalty and satisfaction. These generous policies resulted in extraordinary financial losses for retailers as mentioned by Miao Sun and colleagues in one of the journal article (Sun *et al.*, 2021). In an

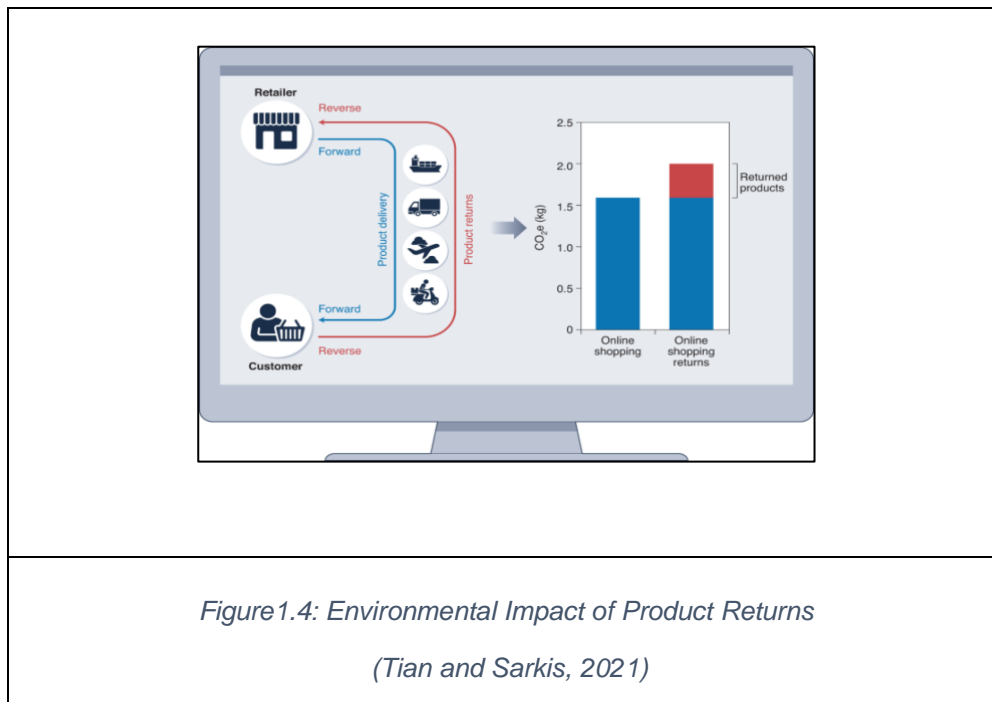
article Guangzhi Shang and colleagues recorded that retailers offered full refund return policy even after couple of months, which is not optimal way of operating as suggested by multiple analytical models and theoretical studies which went unheard by decision-drivers (Shang *et al.*, 2017). This led to eruption in product returns and outburst of financial losses for retailers and environmental damage due to the product returns.



The dissertation focuses on measures to curb the product returns by predicting returns, identifying product characteristics causing returns, and developing insights and suggest measures to mitigate the challenge of abusive or fraudulent product returns. To perform aforementioned tasks, real-world industrial data of retailer is being analysed in dissertation. Access to data has been provided by Apriss Retail Limited, which manages data of different retailers which are Apriss’s clients.

The high motivation factor behind this analysis is to curtail undue financial losses of retailers, contribute to identify fraudulent returns, also reduce environmental damage. Retailers deal with buying, selling, inventory, supply chain, and returns management (Fig 1.3) for every return which involves huge finances and other significant resources. The inefficient and greater volume of returns can equate to more than 30% of carbon emission of initial deliveries (Tian and Sarkis, 2021) as shown in Fig 1.4. Protecting environment with financial benefits would give the required nudging force to initiate efforts and impetus to sustainable policy designing. A small reduction in rate of returns and additional costs can improve profitability, as per a research five percent improvement in return rates has the potential to improve 200 basis points in net margin (Regina Frei and Lisa Jack, no date).





Identifying characteristics of products influencing these returns would empower to take corrective actions and enable to protect genuine returns. The findings and outcomes of analysis will help to understand what type of products are being returned frequently and parameters that are causing these returns. The project output can be used to minimise the returns of products under certain categories which could shrink the losses being incurred by retailers and enable to curb the environmental damage at large.

1.2 Background

This project is carried out in association with Appriss Retail Limited, an organisation working with advanced technologies like artificial intelligence to detect and minimise product returns. One of the survey showed that 52% of consumers won't buy product again from same retailer if it charges for returns (Popkin, no date). However, the policy has led to increase in fraud returns, Appriss has developed four modules to identify and curb fraud returns and increase sales and customer retention with returns. Brief description of the modules are as follows:

- Verify uses statistical and machine learning algorithms to predict real-time returns of product by customers, therefore allowing smooth buying experience without any hassle to customers. (Appriss Retail, no date c)
- Secure module uses artificial intelligence to identify in-store frauds, tracks employees, creates reports, etc. based on requirements of clients. (Appriss Retail, no date d).
- Engage uses artificial intelligence to identify pattern and recognise consumer's preferences and history and recommend enhancements based for customer's journey, provides enhanced method to deal with returns and refunds (Appriss Retail, no date a).
- Incent aims to increase revenue of retailers, increase loyalty of customers, and bring customers in-store by providing intelligent incentives to customers. Appriss claims retailers using this solution has noticed an increase in revenue of 34% after a return (Appriss Retail, no date b).

These modules developed by Appriss is used by its retail clients to avoid in-store employee frauds related to cashier, coupons, cashbacks, optimising returns and refunds, increased revenue by providing incentives to customers on return of selected products and curb customer frauds related to product returns and other fraudulent behaviours.

The outcome of this project related to product returns will be aligned with the verify module of Appriss as it would help retailer to identify the products which are vulnerable for returns. Specifications of products that are influencing returns. It would provide them the probability of return of the product therefore, retailers can design customised policies for products with current high return rate and high probability of getting returned in future. It would also promote sensible purchase by customers.

Retailers can design policy or amend the existing return policies to manage the risks associated with certain products. The outcome can also be used with modules designed by Appriss to curb returns. Corrective actions at product level for respective products with high return history and/or high predicted returns policies would reduce returns and enhance customer experience and loyalty.

1.3 Aim and Objectives

The project's aim is to classify product return using product characteristics and identify probability of classification and characteristics influencing returns. The pivotal idea is related to study on returns linked to product specifications like description, colour, size, online purchase, receipted returns, return rate, quantity, and price. The results can help to detect characteristics of products influencing returns. This would enable to identify product specific details causing financial losses and reduce the negative environmental impacts of product returns, thus enabling to achieve sustainable solution.

Objectives:

Build a classification model to classify product returns:

- Model must be able to predict whether respective product will be returned (return rate equal to or more than median value) or not (return rate less than median) along with probability of prediction.
- The model must be analysed further to find key parameters causing returns.
- Model must calculate probability of classified outcome.
- Developed model must be validated and verified with available data, accuracy scores, and related plots to measure model's performance enabling to decide which model suits data best.
- Some insights using visualisations for better understanding of insights would be created. Advanced visualisations cannot be created as implementation of analysis is done in virtual environment of Appriss.

CHAPTER 2

LITERATURE REVIEW

2 Literature Review

2.1 Overview

With emerging online retailers, product returns increased and other mode of retailers had to adopt it to sustain in the market i.e., around 49% retailers started to offer product return service to customers (Saleh Khalid, May 16). However, one report in 2016 mentioned that around 30% merchandise and 40% of clothing products purchased online were returned, in USA alone in 2015 customers returned products worth whopping \$260 billion as per the National Retail Federation, USA (Courtney Reagan, 2016). American consumers returned a whopping \$428 billions of goods in 2020, a return rate of 10.6 percent, with e-commerce returns accounting for nearly a quarter of returns volume but still many retailers believe they need a generous returns policy to grow financially (Ader *et al.*, 2021). The environmental impact of product returns is that 4.7 million metric tonnes of CO₂ is emitted annually as per the report published in 2019 (Khusainova Gulnaz, 2019). US alone create 5 billion pounds of landfill waste and 15 million tonnes of carbon emissions annually (Regina Frei and Lisa Jack, no date). Product returns is a real problem and highly underestimated by retailers, if managed appropriately by retailers, changes in product returns management can significantly reduce financial burden and negative environmental effects.

Rise in omnichannel retail, specifically with dominance of online retail in recent times has led to many research and studies are being carried out related to product returns- how returns volume can be reduced, how returns can be predicted before purchase, enhanced or customised

returned policies based on customer's past purchase behaviour and others. These ongoing studies are important with exploding online retail sales in 2019 and 2020 (Bauer, Downs and Speights, 2021b) and returns along with it supplemented by the environmental concerns linked to returns.

2.2 Literatures:

Definition:

Predictive Analytics is branch of advanced analytics which as name suggests is used to make predictions based on available or past data. It uses statistical modelling including ML algorithms and sophisticated predictive modelling to study historical data and predict likelihood of event in future (SAP, no date).

Machine Learning (ML) is a branch of artificial intelligence which is used to build a model so that when applied to a dataset the model learns the data on its own and can be used to make predictions on unseen data. It is a method to automate model building for analytics which can learn from data, identify patterns, and make predictions or decisions on its own (SAS, no date).

2.2.1 Prediction using Advanced Analytics:

Role of ML as Predictive Analytics in Returns Handling:

Rooderkerk and colleagues observed that traditionally prescriptive and diagnostic analytics have been playing significant role in retail analytics, however from 2014 there has been growth in usage of advanced analytics. Predictive analytics has been widely used to forecast demands; however recent efforts have also contributed to returns handling in retail analytics. Retailers are adopting advanced analytics but authors also noted that to harness the power of analytics retail industry needs to invest in data management systems, analytics

solutions, and expertise (Rooderkerk, DeHoratius and Musalem, 2022). Fig 2.1 explains recent publications and studies related to retail analytics and areas where still there are gaps:

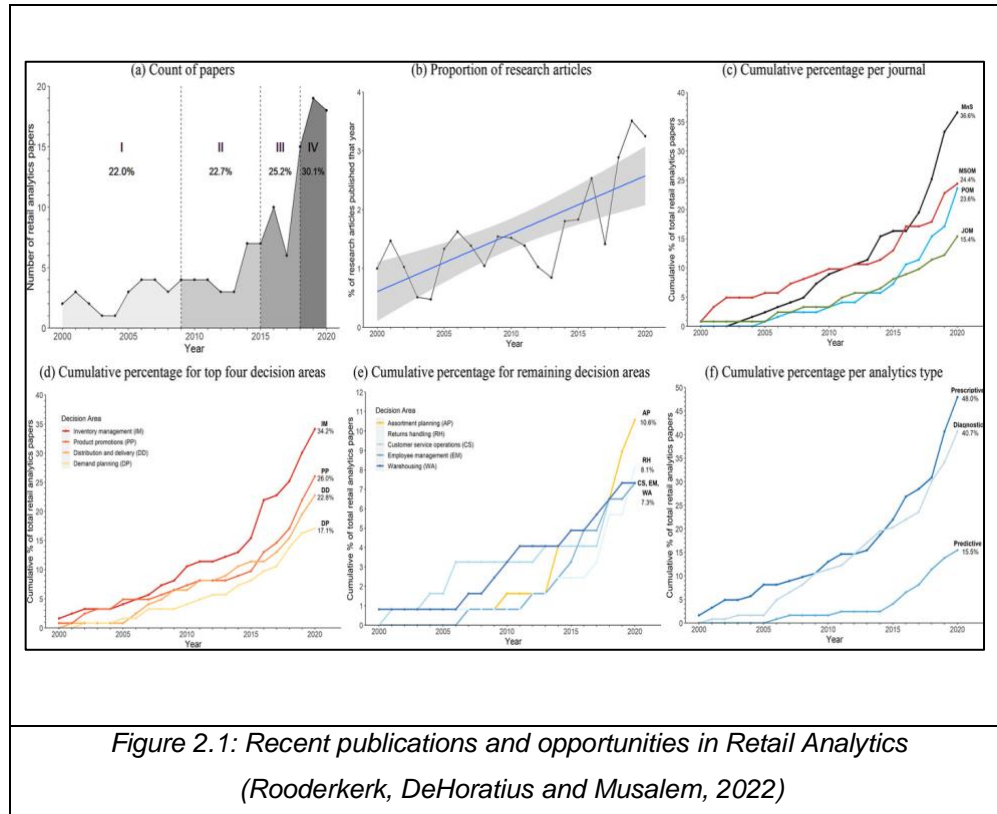


Figure 2.1: Recent publications and opportunities in Retail Analytics
(Rooderkerk, DeHoratius and Musalem, 2022)

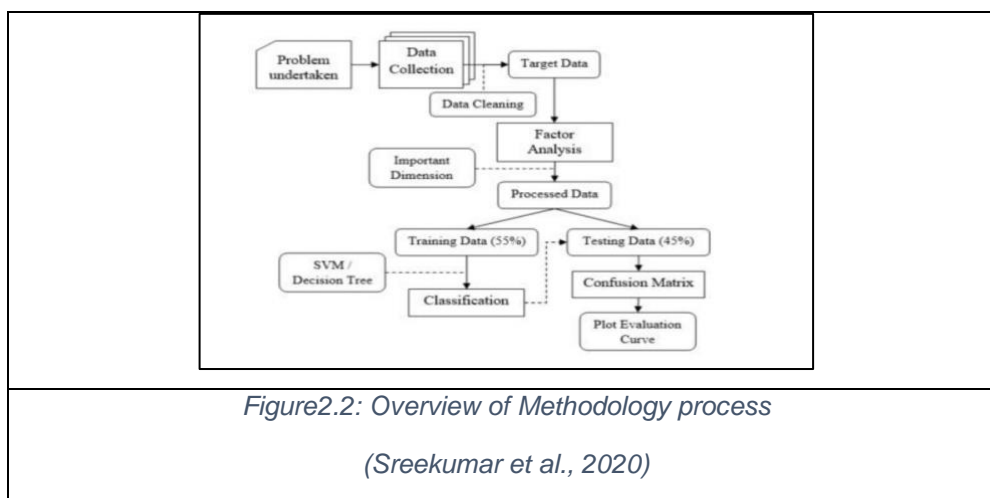
The author highlights benefits and barriers to adopt advanced analytics in retail industry and opportunities that usage of technology can bring in retail. In dissertation, the focus is on using advanced analytics in return handling and classifying returns based on product's characteristics which is in line with the author's idea.

Customer classification using ML Classification algorithm:

Authors have analysed data from retail firm to classify customers as satisfied or dissatisfied using classification algorithm namely, Support

Vector Machines (SVM) and decision tree classification technique. The author claims that identifying customers is important to design customised marketing policies for different group of customers and how it can help to scale and increase relevance. On comparing both the classification techniques it was deduced that SVM performed better in classification (Sreekumar *et al.*, 2020).

In dissertation, methodology is similar of using ML Classification algorithm, but the focus is on returns and analysis is based upon characteristics of products but the however insights developed in paper by authors can be helpful to study classification algorithms. Fig 2.2 highlights steps involved in literature similar to the implementation in dissertation is same except the ML algorithms used.

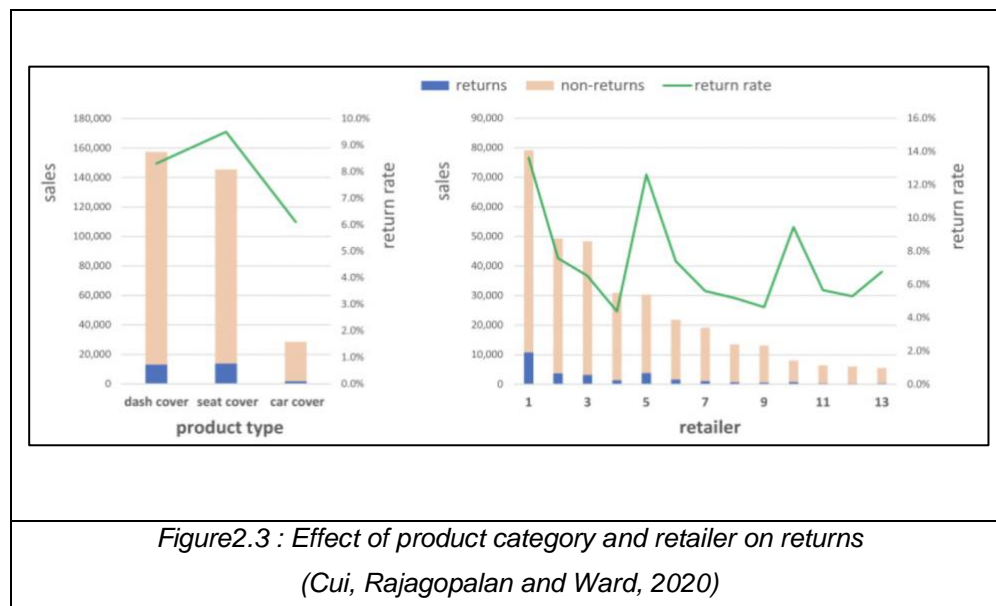


Predict returns using ML and importance of product category:

In this literature authors used data of a company selling car accessories to predict volume of goods that were returned and their conditions, they considered four factors- sales volume, time, product category, retailer, and calculated return rate based on return numbers and sales data.

They used predictive machine learning algorithm like LASSO, SCAD, LARSOLS and Elastic Net and to capture non-linear structure and reduce bias, RandomForest and GBM has been used to improve predictive accuracy and develop model. (Cui, Rajagopalan and Ward, 2020).

The literature focuses on returns of products based on parameters related to retailers, however, it highlights significance of product category in returns. Product category is one of the parameters in models implemented in dissertation, the literature helps to understand implementation approach of ML algorithms. The below image highlights effect of product category and retailers on returns of product.



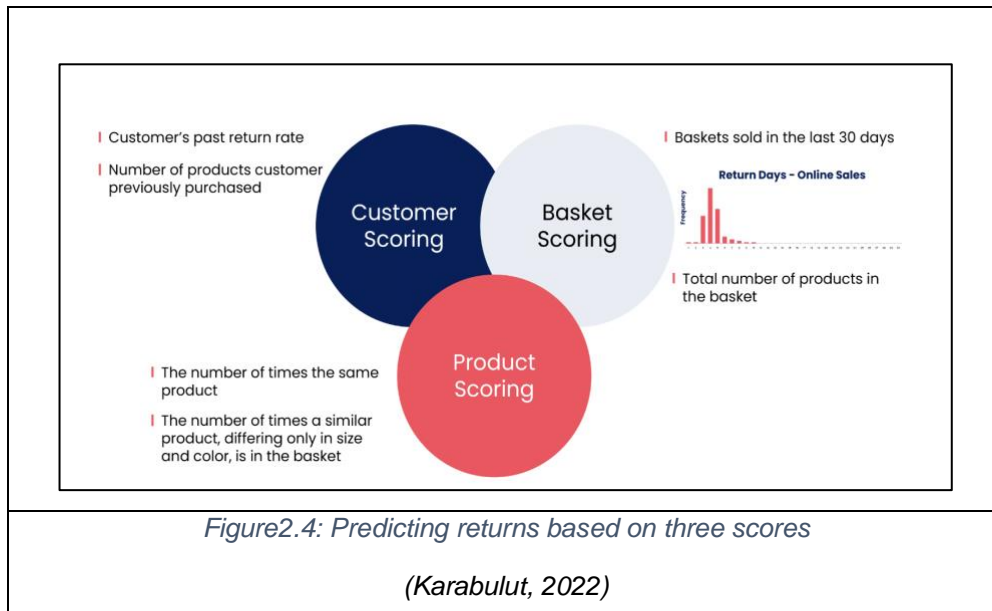
Proposed generic framework to predict returns

The authors propose a generic framework to predict product returns before customers make any purchase- HyperGo. The returns will be predicted based on the basket and product level where the historical basket composition and products will be analysed like same product in multiple colours and sizes in a basket. The HyperGo framework uses Hypergraphs, Local cut hypergraph, JacWght, JacNorm and k-NN techniques to analyse basket composition and products in basket. The methods make pairwise similarities and then make prediction of returns at product level (Li, He and Zhu, 2018).

Literature makes prediction using product and basket parameters both. In dissertation, only product characteristics are considered in model to classify products.

AI and Analytics for forecasting returns:

Karabulut in his blog explains how advanced forecasting capabilities using AI and analytics will be able to predict returns based on past behaviours and insights focusing on planning to optimise inventory based on returns, as returns were sometimes damaged, need repairing and some can be put on shelf for sell again. The forecasting of product return was calculated based on three parameters: customer score basket score, and product score. Author emphasises on adopting return handling through advanced analytics (Karabulut, 2022).



Author's holistic approach can be used in future research as many factors are included in model. In dissertation, similar real industrial data has been collected however, focus is only on product and its characteristics in classifying returns using advanced analytics.

Product returns comparison online and offline using ML:

In their literature Dzyabura and others, conducted analysis on a large retailer having online and offline presence and compared rate of online and offline sales and returns of goods. They suggested product's return behaviour must be studied initially in offline sales and they put on online market to understand return behaviour. The study involved Gradient Boosting Machine (GBM) regressor to predict returns and highlighted how usage of image of products in online sales reduces returns and image parameter improved prediction accuracy of model (Dzyabura, El Kihal and Ibragimov, 2018) .

The analysis done by authors involves parameters such as online and offline sales, product category and predicting return using GBM algorithm which is similar to implementation in dissertation. However, unlike dissertation, the study does not include multiple algorithms in analysis to make comparative study of models. The below figure shows rate of returns differs across product categories and depending on mode of sales:



Predicting product returns based on product's characteristics:

Eroglu in a literature found out that due to mistakes in production, packaging, transportation, forecasting, stock planning, etc. and when identified by customers it leads to initiation of reverse logistics and this

consumes resources, energy, and capital. The study claims that machine learning algorithms can make faster and accurate predictions of return rate in case of such complex datasets. Author used Linear Regression (LR), Support Vector Regression (SVR) and Artificial Neural Networks (ANN) from functional algorithms, M5Rules, M5P, REPTree, Random Tree, and many others but M5P and M5Rules to provide better performance and thus used as model to make predictions for data under study (Eroglu, 2019).

The analysis done by authors focuses on predicting product returns using ML algorithms, however, in dissertation, classification algorithm is used to classify return and non-return using multiple machine learning algorithms and the one with best verification scores will be considered for outcome of analysis.

Product returns prediction using advanced analytics:

The authors implemented decision support system which would predict the chances of product being returned in real time when put in basket by customer based on previous history of high-risk products. Multiple algorithms were tested, however Mahalanobis Feature Extraction in combination with adaptive boosting algorithm provided highest accuracy and logistic regression can be used as classifiers (Kranz, Urbanke and Kolbe, 2015).

In dissertation multiple machine learning classifier algorithms are compared to provide better accuracy and enabling feature extraction which cause product returns which is similar approach as in paper.

Forecasting returns and analysis of return quantity using Holts and ARIMA:

In a working paper, authors conducted research on group of products from Singapore's multinational company involved in remanufacturing industry. The authors created a forecasting model to predict when (time) and how many (quantity) of products would be returned using Holts and ARIMA algorithm for prediction. The prediction was made based on the past sale data of products. Algorithm with better performance to make forecasts was decided on basis of following-MAE, MAPE, and RSME (Canda, Yuan and Wang, 2015). The below figure shows the scores of both models used for analysis:

Product No.	Holt's Method			ARIMA		
	MAD	MAPE	RMSE	MAD	MAPE	RMSE
1	1.79	6.63	2.74	8.16	30.44	8.29
2	11.85	236.86	12.56	6.13	156.76	7.8
3	2.36	66.03	2.9	9.95	303.37	10.26
4	8.71	126.48	8.93	9.3	163.47	9.61
5	1.98	19.25	2.29	2.81	27.79	3.07
6	2.6	30.76	2.7	2.62	31.22	2.66
7	6.42	85.71	7.64	3.35	72.05	3.75
8	4.56	26.21	4.59	6.24	33.41	6.7
9	20.53	213.83	24.55	20.97	240.02	23.92
10	59.81	74.34	76.24	38.77	88.43	41.56
11	5.6	42.15	6.79	5.41	44.58	6.13
12	11.24	483.68	11.51	1.97	97.46	2.37
13	2.01	79.55	2.14	0.68	25.13	0.82
14	6	74.28	6.01	6.15	73.24	6.17
15	206.34	369.32	212.32	47.46	73.18	51.66
16	101.55	128.23	110.72	30.81	30.03	38.37
17	42.43	140.53	43.45	15.1	38.81	17.45
18	117.64	160.93	125.25	55.62	52.31	60.57
19	21.67	43.12	26.35	16.63	21.34	23.52
20	535.88	182.3	616.58	536.57	168.1	697.24

*Figure2.6: Comparison of scores of two algorithms
(Canda, Yuan and Wang, 2015)*

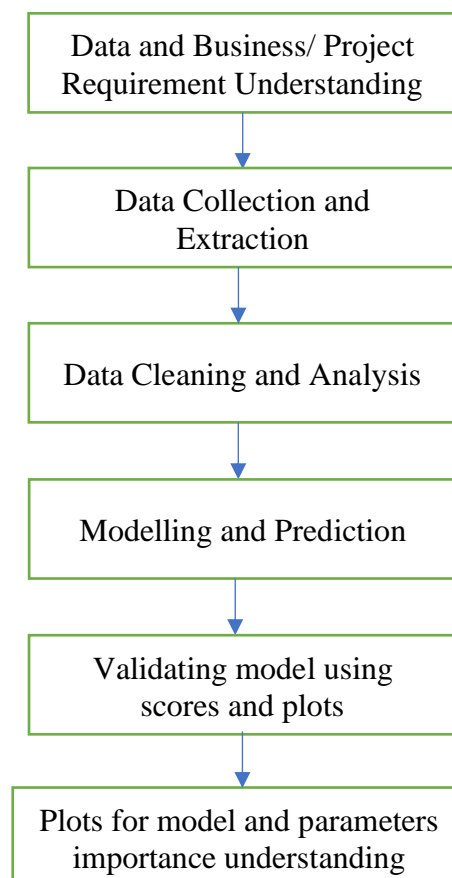
The author concluded that return quantity was intermittent sometimes it is zero and sometimes very large. The gap in the study is it did not identify parameters which are influencing the returns which is implemented in dissertation. Similar approach of using multiple predictive models and deciding final model based on multiple validation scores will be used in dissertation.

Predicting returns using LightGBM and RandomForest:

Authors developed a decision support system using Machine Learning algorithms like LightGBM, RandomForest and DART to predict product returns. The research was not limited to any particular industry like fashion, sports, electronics goods, etc. it was a general system developed for all category of goods (Hofmann *et al.*, 2020).

The same approach as mentioned in literature by authors is followed in dissertation of using same algorithms as in literature on same kind of dataset.

Below is the approach in literature and dissertation:



Predicting product returns using RandomForest and Boosting Algorithms:

The authors implement a Forecasting Support System (FSS) to forecast returns using advanced analytics cloud-based ensemble approach. In this paper, authors state the parameters based on which model can be selected for prediction like high accuracy, scalability, and adaptability. Multiple algorithms were tried for designing FSS like Boosting-AdaBoost, Ensemble Selection, RandomForests, Logistic Regression, SVM, CART, and MLP (Heilig *et al.*, 2016).

In dissertation, prediction of product returns is being implemented using some of the above mentioned algorithms- Logistic Regression and tree-based algorithms like Boosting and RandomForest.

2.2.2 Evaluating Model Performance

Understanding of Confusion Matrix:

Confusion matrix is visualisation of counts of values from actual dataset and the one's predicted using model. Accuracy of a model which is mostly used to measure performance of classification model takes same values as displayed in confusion matrix. Confusion matrix is used to evaluate performance of classification models with binary or multiclass levels in outcome/dependent variable (Kulkarni, Batarseh and Demir, 2020).

As three classification models are implemented in dissertation, the literature highlights importance of confusion matrix in evaluating performance of models.

Understanding of Classification report:

The critical aspect involved with growing implementation of machine learning (ML) models is to measure the performance of model. There are multiple techniques to evaluate performance, however, as per author's research F1-score and area under precision-recall curve are significant in ML applications (Orozco-Arias *et al.*, 2020).

Similar approach has been implemented in dissertation along with other techniques to measure model's performance in dissertation.

Understanding of ROC Graphs:

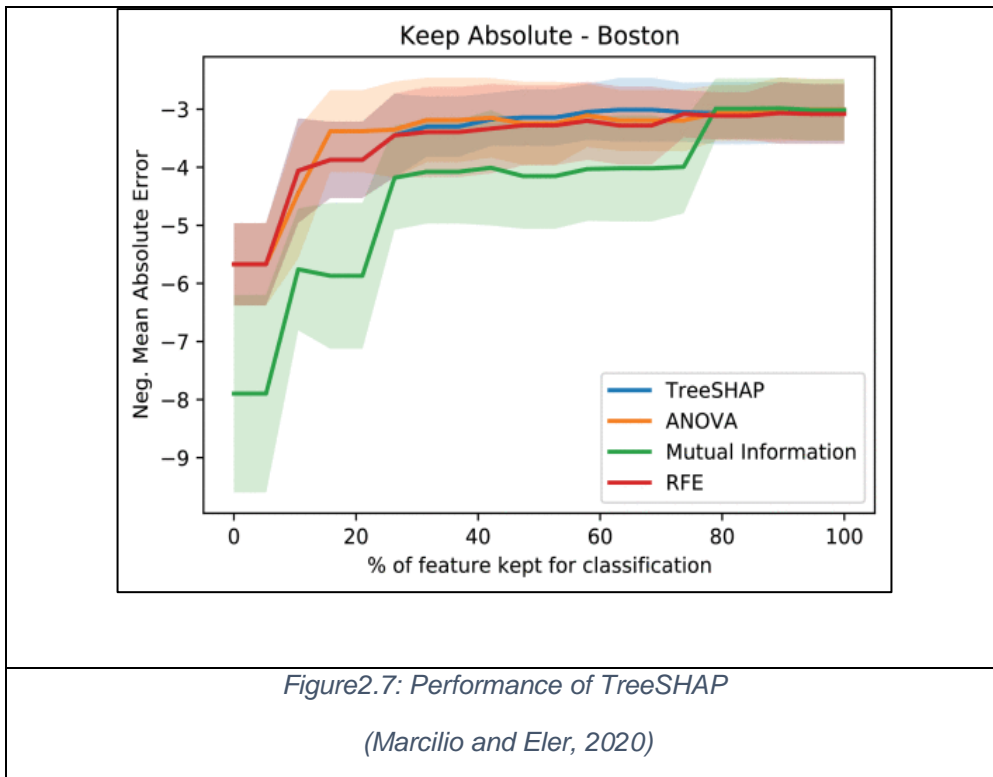
Receiver Operating Characteristics (ROC) graphs are used in measuring and comparing performances of classification machine learning algorithms. It is used to visualise performance of models which are simple to interpret and make comparison. While plotting curves, ROC technique de-couples class skew and error cost which is a great advantage as it is not affected by majority class (Fawcett, 2006).

ROC curve and accuracy score to quantify the outcome both have been used in dissertation to evaluate performance of models.

Use of SHAP values and plot:

Datasets with highly dimensionality is common phenomenon these days which increases its redundancy. Techniques like tSNE and UMAP can be applied to reduce dimensionality, but they make feature interpretation worse, other traditional algorithms raise explainability concerns. Therefore, authors tried the novel approach of SHAP values on multiple datasets with regression and classification models both and after thorough experimentation concluded that SHAP values are the best method to detect feature importance and selection (Marcilio and Eler, 2020).

In dissertation, SHAP values and plots are used to explain feature selection of predictive model. Below figures show comparative analysis of feature selection techniques:



Run-Time Execution (In Milliseconds) For Feature Selection On the Classification			
Technique	Boston	Diabetes	NHANESI
TreeSHAP	10894.861	2565.136	62.809
ANOVA	1.010	0.999	4.959
Mutual Inf.	25.896	19.917	403.947
RFE	251591.223	21728.887	33322.885

Figure2.8: Time complexity comparison in feature selection
(Marcilio and Eler, 2020)

2.2.3 Predicting product return based on different product characteristics:

Effect of price of product on returns:

Thomas conducted statistical equation modelling and conducted analysis to identify causes of product returns and concluded that price of product and customer expectation not met are two primary reasons. Author used two sets of retail customers from Target and Walmart for analysis. To test the statistical validity of data and handling covariation, tests of normality, skewness, and kurtosis were performed. Partial Least Square (PLS) technique was used to test model and hypotheses involved. Thomas also observed that emotional dissonance is significantly greater than the product dissonance (Powers and Jack, 2015).

The literature did not involve identifying parameters that cause returns and was more of descriptive nature. In dissertation, focus will be on product characteristics and identifying parameters under analysis which cause returns and predicting returns.

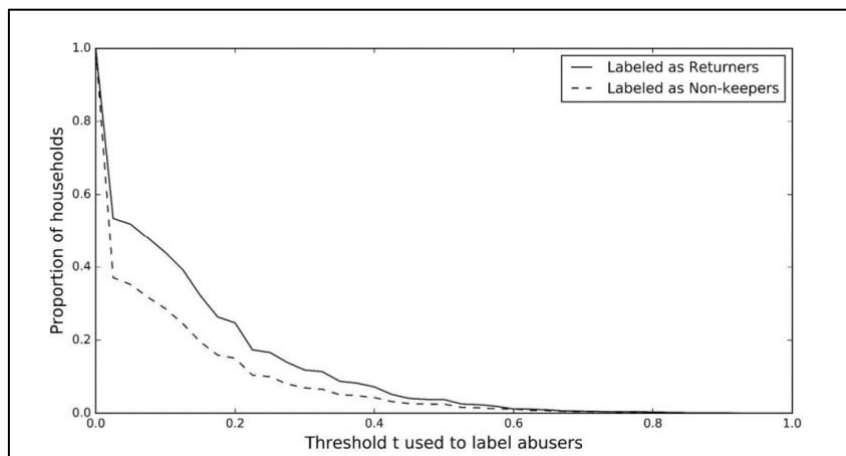
Effect of average price of product on returns:

Michele in his paper has tried to establish relationship between average price of product and its return rate using data analytics on data obtained from electronics retailer. He described concept of *product return episode* and concluded that products with higher price tend to be returned frequently but customer completes the episode and purchases alternative product as customers try to find right product which fits choice and needs (Samorani and Messinger, 2016).

Product Return episode: “An episode begins with an initial purchase in a certain product subcategory, and continues with subsequent replacements (i.e., returns followed by purchases) with other items in the same subcategory. Each episode can end after a purchase or a return transaction” (Samorani and Messinger, 2016).

In research, author analysed and concluded the positive correlation between average price of product and return rate but did not make any predictions on return of product and did not analyse how significant is average price in return of a product. In dissertation, average product price is a parameter used in model to predict returns of product and its importance in return is also analysed.

The below image compares the rate of returners who complete return episode and non-keepers are abusers who return product and opt out of product return episode.

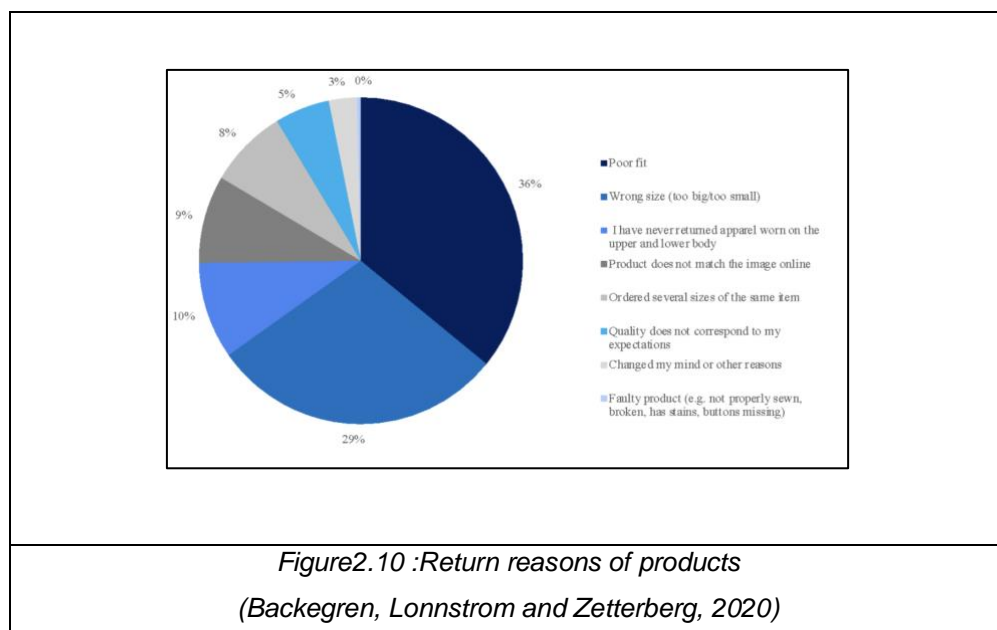


*Figure2.9: Comparison of product returners and non-keepers
(Samorani and Messinger, 2016)*

Effect of size and quality of product on returns:

In this literature data is obtained via customer survey with motive to find reasons for product returns associated with apparel goods. The primary reason for returns was the size and customer dissatisfaction as product does not match the online image displayed which can include quality, colour, manufacturing defects, etc. Authors performed descriptive analysis to develop insights and conducted univariate and bivariate analysis to understand reasons. Chi-square test was performed to detect if any relationship exists and if any, then it is by chance or statistically significant (Backegren, Lonnstrom and Zetterberg, 2020).

Authors used descriptive tests to find characteristics of products (fig 2.10) that influence returns and deduced that size or fit and quality are considerable reasons of product returns. Size is analysed in model created in dissertation and quantitative score is calculated to identify parameters causing returns.



Effect of mode of shopping and product category on returns:

Author highlights the high adoption of online shopping after pandemic hit in 2020 and 2021. Author claims that return rate of a product is around 5-10% when a product is purchased in-store while return rate is around 25% for same product in online sales. The author also highlights rise in sales and return rates of certain product categories for e.g., loungewear, garden, and homeware (Zigzag Global, 2021).

The blog compares the online and offline sales and returns, and underlines increase in return rates in online sales. It also points certain product categories that have increase sales and return rates. In dissertation, both parameters i.e., mode of sale- ecommerce or offline and product category has been used in the model for analysing and predicting returns using advanced analytics.

2.2.4 Other researches related to product returns

Impact of return policies on product returns:

Authors study return policies of retailers and analyse how it affects the purchase and return behaviour. Return policies are measured on certain leniency dimensions i.e., time (deadline to return), effort (forms or other formalities), money. The study is based on analysis of combination of these parameters and how the positively or negatively impact the purchase and return of products. The study involves multivariate analysis using HLM concludes that return policies with some tweak in parameters under study can impact purchases and returns in both ways (Janakiraman, Syrdal and Freling, 2016).

Product Specific Pick-up and Return policy:

In this literature, Appriss tried to focus on difference between rates of online returns and in-store returns based on product description. Exploratory Data Analysis was used in this literature for analysis and the outcome of analysis concluded that certain products are frequently purchased in multiple numbers in single order with intention to try at home and return most of them. Such products must be incentivised and promoted for pick-up of order from store and in-store returns. This would bring down losses to great extent and more flexible and enhanced approach toward such products (Bauer, Downs and Speights, 2021a).

Significance of online reviews on product purchase and returns:

Authors propose a duopoly model where they study influence of online reviews on purchasing and returning products. Study is carried considering products of different qualities and having varied return policies. They have used game theory to conclude their research and the outcome is that quality of product along with reviews- positive and negative do impact purchase and return of product. As online reviews reduce risk of pre-purchase dissatisfaction majority customers in online retail rely on reviews (Sun *et al.*, 2021).

Promote BORIS:

To mitigate challenges linked to reverse logistics, Appriss proposed to promote BORIS i.e., online and return in-store rather than BORO i.e.,

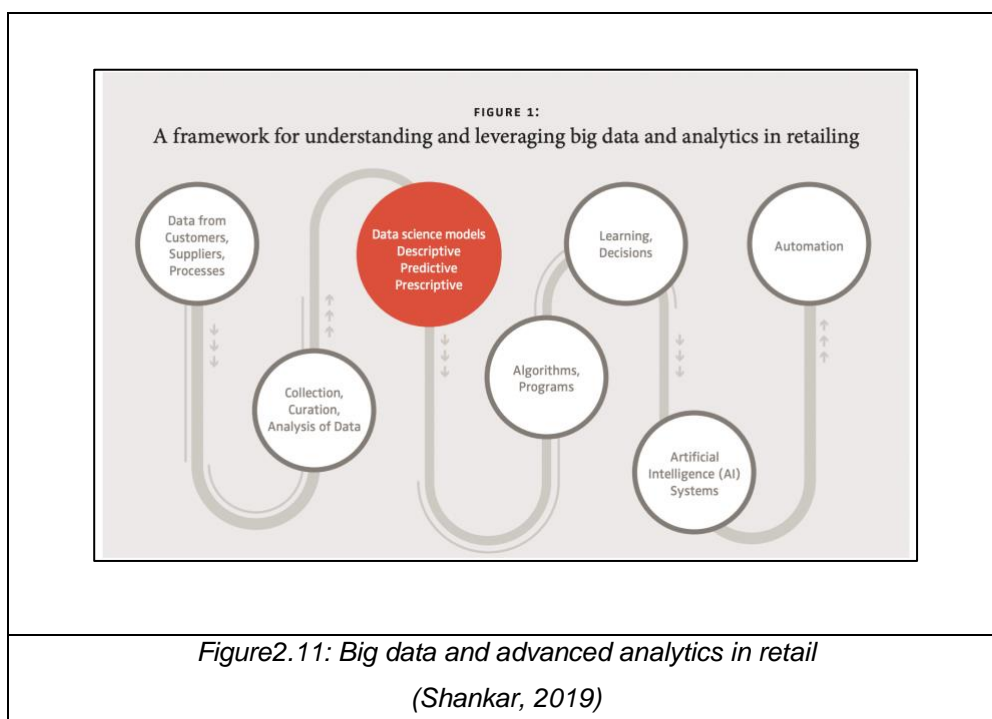
buy online and return online. This can help to reduce the extra burden on supply chain and minimise financial losses drastically (Bauer, Downs and Speights, 2021b). The paper suggested that how financial losses can be minimised by providing incentives to customers who purchase online but return in-store (Bauer, Downs and Speights, 2021b). The analysis used XGBoost algorithm claiming it to predict better than other machine learning algorithms like logistic regression and RandomForest. The outcome was that customers staying in 10-mile radius of in-store retailer will opt for in-store returns looking at incentives.

Product returns challenge:

Product returns in one of the huge challenges being faced by retailers, even after all time high consumption retail shops are getting closed, jobs lost, towns losing their liveliness and with growing online retail, retailers are trying to adopt omnichannel retail (Frei, Jack and Brown, 2020). A newspaper article mentioned explosion of online retail has started the process of reverse logistics, it claims some online retailers face as high 30% return rate, creating extra burden of recycling and refurbishing products (Paul, 2015). Retailers need to build digital infrastructure to improve the situation. Below mentioned literatures highlight use of analytics in forecasting returns and how it can help to reduce returns:

In one paper Venky described use of advanced analytics in retail sector where he defined data science as heart of the framework which includes descriptive, prescriptive, or predictive analytics. Predictive analytics is being used predominantly to forecast in retail analytics. He emphasises on predictive analytics, artificial intelligence, and big data

as future of retail analytics (Shankar, 2019). Fig 2.11 depicts sequence of advanced analytics techniques that can be implemented in retail.



Another author used Bayesian estimation technique to predict product returns and claims to be 50% more accurate than traditional approaches like Moving Averages and Holt's Approach, he suggests that returned products on damage can be recycled or redeveloped (Krapp, Nebel and Sahamie, 2013). In a literature, authors conducted survey of electronic retailers and conducted statistical analysis and developed a measurement model for prediction, it concluded that there is a positive correlation in forecasting product returns and operations performance of reverse logistics (Agrawal and Singh, 2019). In research conducted by Brito and Van, they highlighted method to predict returns if data collected is not accurate and defined four methods to compare and study using mean and variance and analytical techniques and concluded 'return distribution and tracked individual

returns' method is more appropriate for prediction. It tracks individual returns and perform well in predicting future returns (de Brito and van der Laan, 2009).

2.3 Conclusion:

Predicting product returns using advanced analytics has been performed in multiple above-mentioned literatures using algorithms like LightGBM, logistic, SVM, LASSO, Halts, RandomForest, ARIMA, etc. however the most popular ones are not used together in one analysis for classification prediction along with other expected outcomes that are to be implemented in this dissertation. In this dissertation, three popular and efficient algorithms- LightGBM, RandomForest, and Logistic Regression will be implemented and compared to find the most efficient one in terms of model accuracy, prediction accuracy, identifying feature importance and providing probability of predictions.

After analysing multiple literatures to decide on product characteristics, some characteristics from literatures like- online sales rate, price, quantity, size, etc. has been taken into consideration.

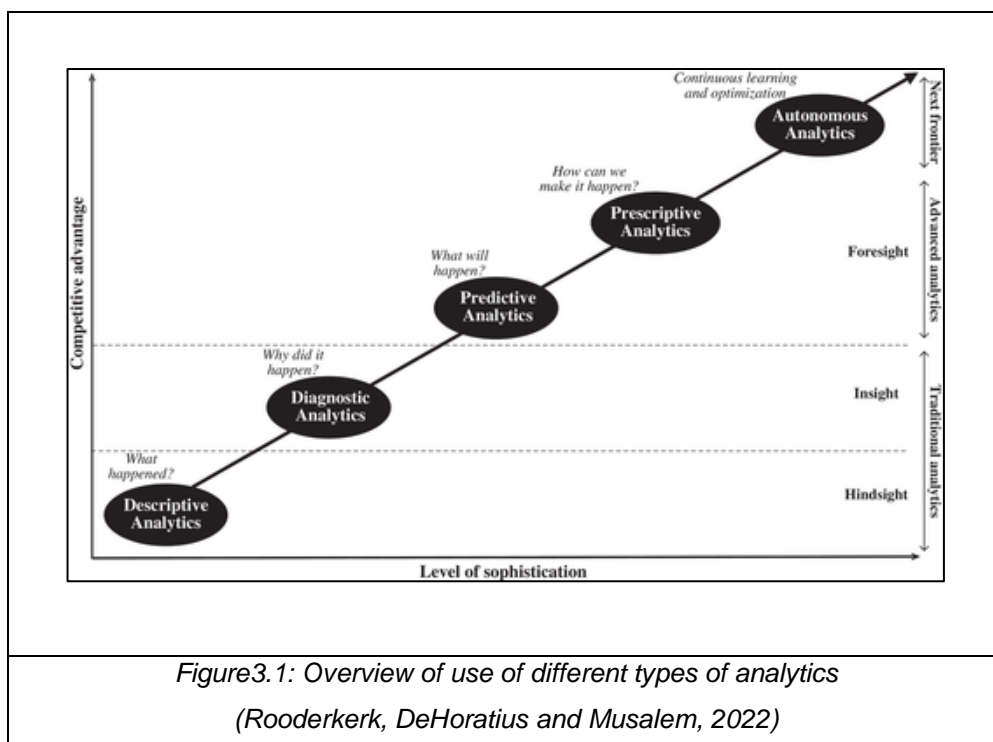
CHAPTER 3

MODEL SELECTION, OVERVIEW OF MODELS AND INTRODUCTION TO DATA

3 Model Selection and Overview of Models

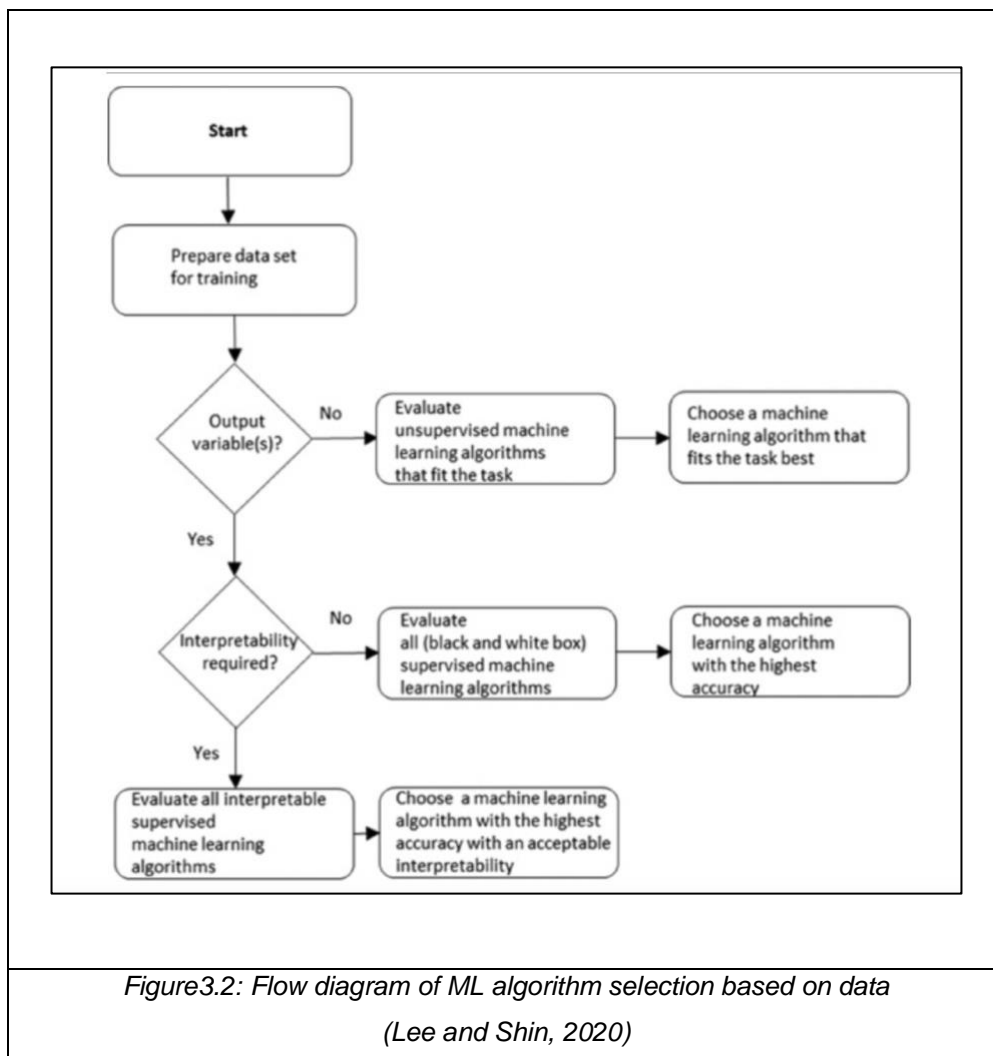
3.1 Model Selection

Selecting algorithm to create model for any project is a challenging task as it cannot be directly deduced which one model is best in terms of fitting the data and providing accurate classifications or predictions. Recently, the retail industry has begun adopting advanced analytics in retail industry and Fig 3.1 highlights use of different types of analytics that has been traditionally adopted and what is being adopted by retail analytics (Rooderkerk, DeHoratius and Musalem, 2022).



To decide on which type of analytics and further which algorithm is to be implemented depends on understanding data, business context, and expected outcomes. Based on data, like if input and output variables

are available then supervised learning can be used, if input data is available but output is not available then unsupervised learning must be used (Lee and Shin, 2020). To classify output variable classification algorithm must be used and to make prediction regression algorithms must be used. Fig 3.2 provides overview of steps involved in modelling:



Considering dataset available and expected outcome in dissertation supervised classification algorithms has been implemented. Out of the implemented algorithms in dissertation, one with better performance will

be selected to predict outcomes. Description of three algorithms implementation are as follows:

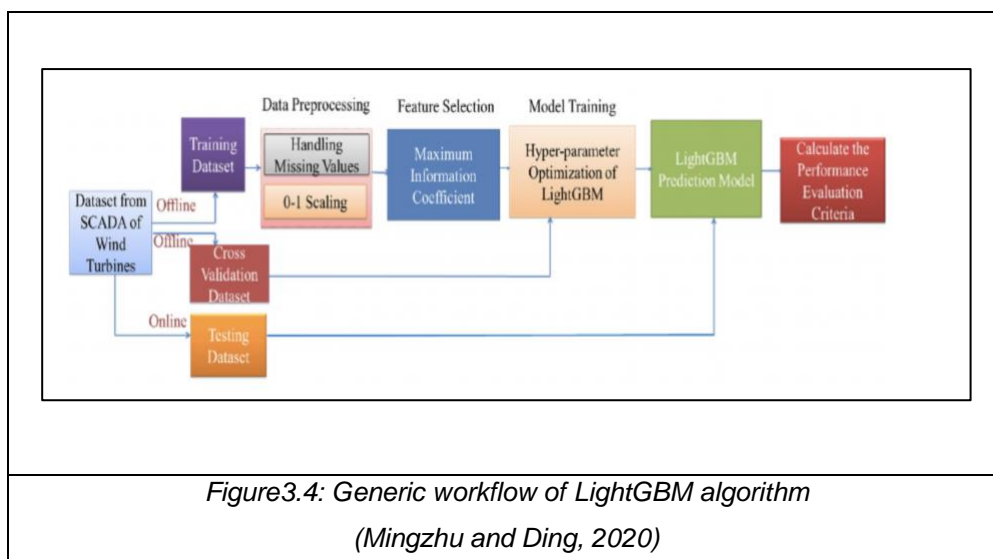
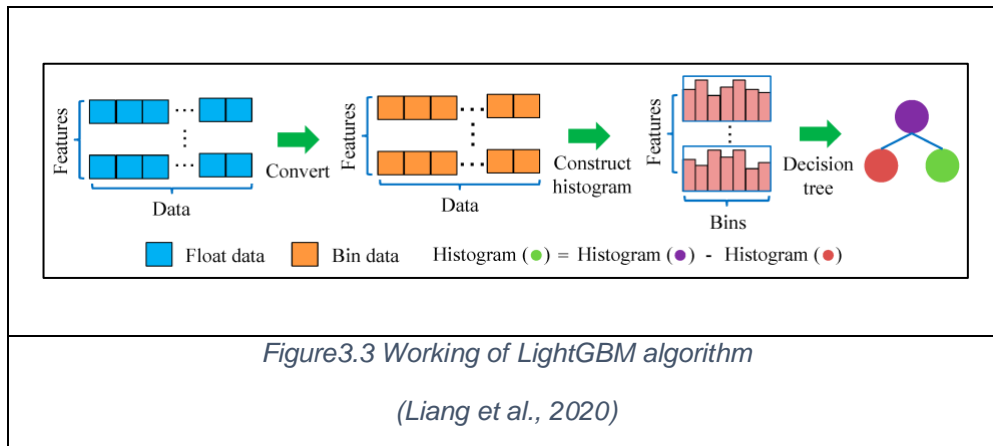
3.2 Modelling using LightGBM:

Gradient boosting machines build sequential decision trees and previous tree's error is used to build further trees, later sum of these trees will be used to make predictions (Serengil, 2018). Gradient Boosting Machine (GBM) is a boosting algorithm which significantly manages both the aspects- variance and bias and is considered to be very efficient.

To overcome this, LightGBM algorithm was introduced, a tree-based learning algorithm. It was designed by Microsoft Research Asia using GBDT framework to improve computational efficiency (Liang *et al.*, 2020). LightGBM can be used on large dataset and uses low memory to work. The name includes 'Light' highlighting the great speed at which algorithm runs. It adopts two novel techniques Gradient-based One-Side Sampling (GOSS) and Exclusive Feature Bundling (EFB), using GOSS, it can train each tree with only a small fraction of the full dataset while with EFB, LightGBM handles high-dimensional sparse features much more efficiently (Yan Liu *et al.*, 2016).

LightGBM grows tree vertically as compared to other tree based algorithms which develop horizontally, leaf with maximum delta loss is chosen to grow (Banerjee, 2020a). It provides lot of flexibility in terms of tuning and adjusting hyperparameters based on requirement. The growing size of big data has led to increase in demand of algorithms that can simultaneously train and process data. LightGBM framework was designed for distributed training, supports large-scale datasets and

training on GPU, the boosting trains models sequentially where the upcoming model learns from errors of previous model (van wyk, 2018).



The equation in below figure is in a literature where author explains how leaf-wise strategy works on same layer. They argue it is favourable and optimal with control mode complexity. Leaves on same layer are managed differently as they have different information gain. Information gain indicates decrease in entropy caused by splitting nodes based on attributes of nodes:

$$IG(B, V) = En(B) - \sum_{v \in Values(V)} \frac{|B_v|}{B} En(B_v)$$

$$En(B) = \sum_{d=1}^D -p_d \log_2 p_d$$

where $En(B)$ is the information entropy of the collection B , p_d is the ratio of B pertaining to category d , D is number of categories, v is the value of attribute V , and B_v is the subset of B for which attribute has value v .

Figure3.5: Equation of information entropy in LightGBM

Advantages of LightGBM:

Faster training and higher efficiency: LightGBM deals with continuous features values as discrete value bins resulting into faster training procedure and comparatively greater efficiency (Khandelwal, 2017).

Low memory consumption: Replacing continuous values with discrete ones enables lower memory usage (Kasturi, 2019).

Better accuracy than other boosting algorithm: It uses complex trees with leaf level split approach compared to level-wise approach which is used to achieve greater accuracy. Chances of overfitting can be controlled by 'max_depth' parameter tuning (Kasturi, 2019).

Compatibility with Large Datasets: It is compatible with large datasets and displays good performance taking significantly lesser time to train compared to XGBOOST (Khandelwal, 2017).

Parallel learning: It supports parallel learning and GPU learning, increasing its usability and takes lesser time comparatively (Saha, 2022).

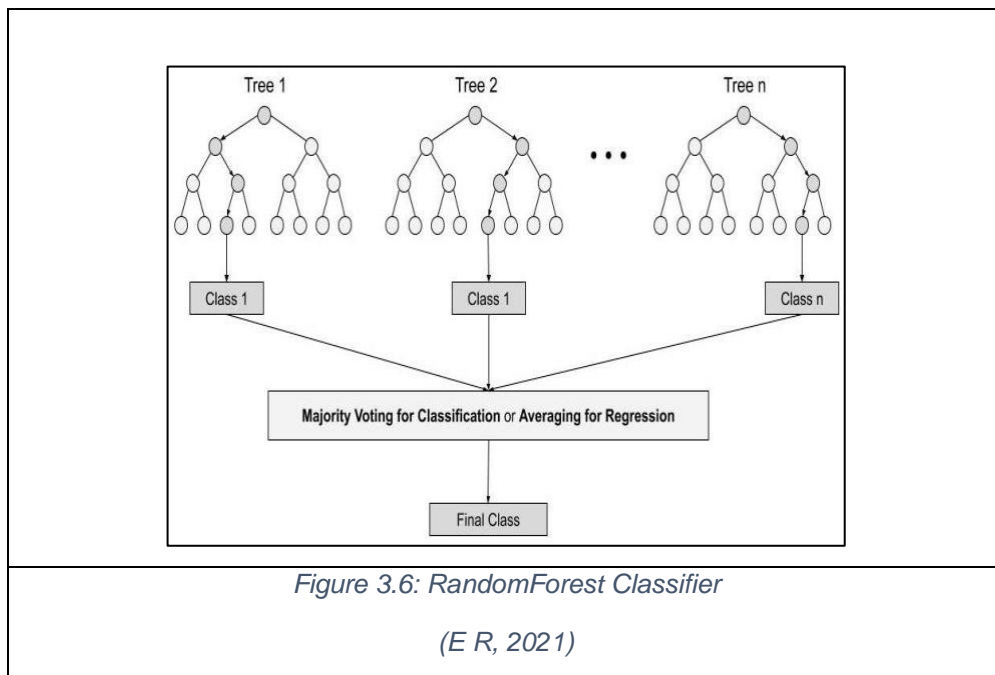
Disadvantages:

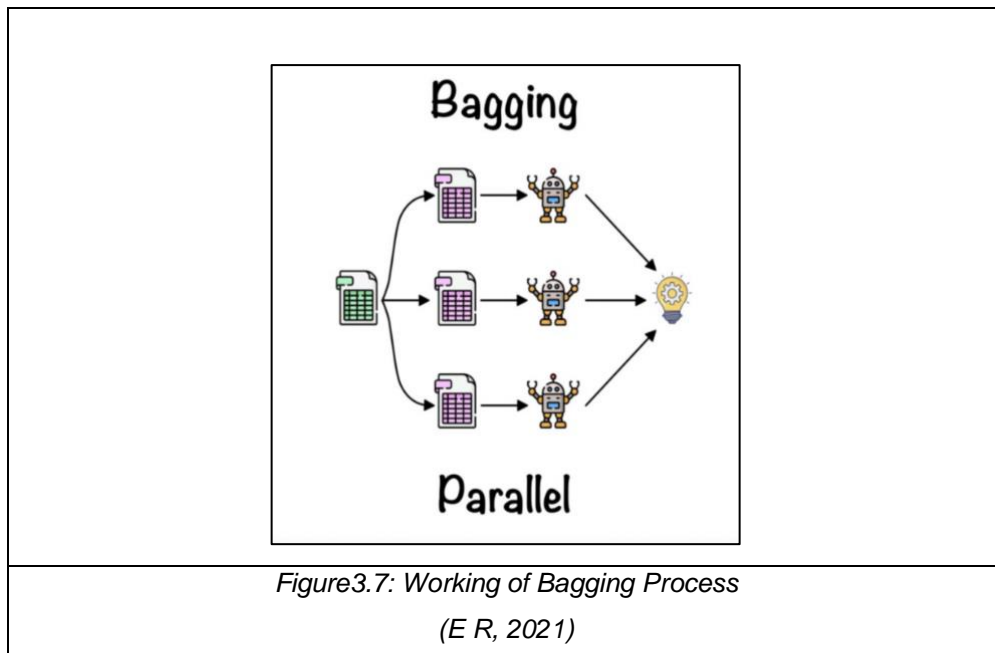
Overfitting: LightGBM supports leaf-wise approach which may lead to overfitting as it produces complex trees (Surana, no date).

Compatibility with datasets: It is sensitive to overfitting and thus can easily overfit small data (Surana, no date).

3.3 Modelling using RandomForest:

RandomForest is a supervised machine learning algorithm and can be used for both regression and classification problems. One of the important part of Machine Learning is Classification, it enables to find the class to which a data record belongs to. RandomForest algorithm is built using collection of decision trees, each tree is comprised of data sample from training set with replacement, called the bootstrap sample(IBM Cloud Education, 2020b). RandomForest algorithm does not consider all attributes/features together, each feature has different tree which reduces risk of dimensionality. It creates different sample subset from training dataset through replacements and then every tree votes for the outcome, this process is known as bagging. RandomForest is an ensemble of decision trees created on random dataset, this group of trees perform bagging process and the class with most votes is considered as final outcome of the model (datacamp, no date).





Advantages of RandomForest:

- In RandomForest algorithm, large number of decision trees participate which provides high accuracy and robustness (datacamp, no date).
- It can handle missing values by replacing missing continuous values by median or calculate proximity weighted average value of missing values (Vadapalli, 2021).
- RandomForest enables to identify relative feature importance which helps in identifying features which influence the outcome (datacamp, no date).

Disadvantages of RandomForest:

- One of the greatest disadvantage is computational complexity, it is very slow in making in predictions because of large number of

decision trees and voting process involved in prediction (Banerjee, 2020b).

- Interpretation of the model is difficult as compared to decision trees where decision can be made following the tree's path (Banerjee, 2020b).

3.4 Logistic Regression:

Logistic Regression is a supervised machine learning classification algorithm. It's outcome is an estimation of probability of an event based on independent variables in a dataset, the outcome being a probability so it always lies between 0 and 1 (IBM, no date). The logistic regression model passes the outcome of a linear function (which is a type of sigmoid) of features through a logistic function to calculate the probability of an occurrence and then maps the probability to binary outcomes (Naeem, no date).

- **Sigmoid:** Sigmoid is a mathematical function that takes any real number and maps it to a probability between 1 and 0 (Naeem, no date).

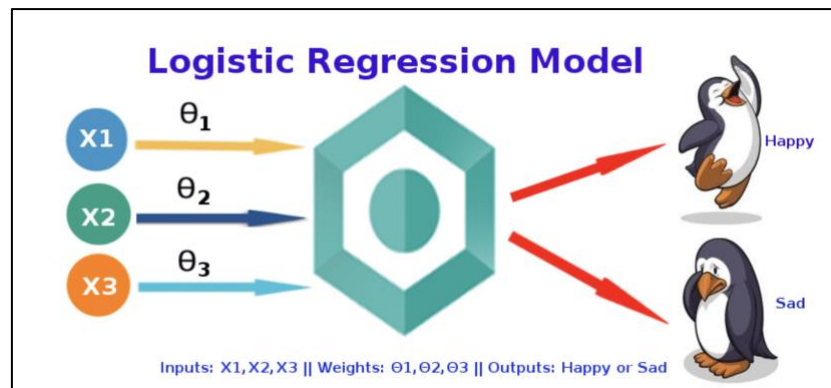


Figure3.8: Representation of Logistic Regression
(Swaminathan, 2018)

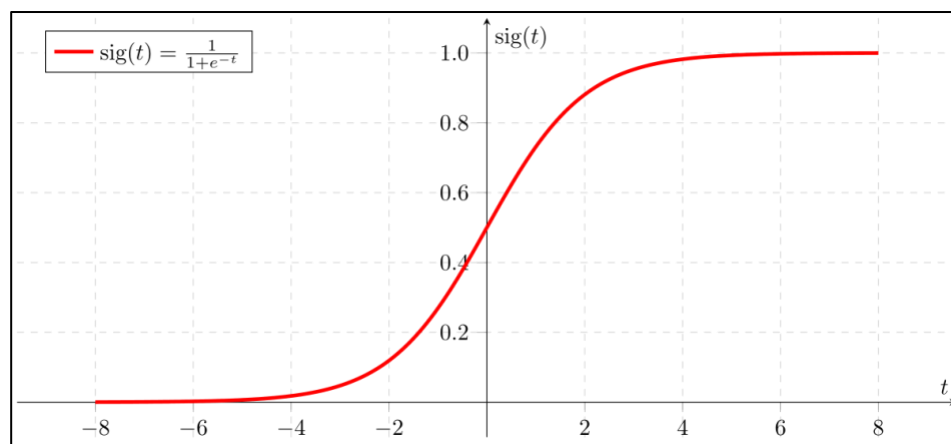


Figure3.9: Representation of Sigmoid Function
(Swaminathan, 2018)

If 'Z' goes to infinity, Y(predicted) will become 1 and if 'Z' goes to negative infinity, Y(predicted) will become 0 (Swaminathan, 2018).

Advantages:

- In a low dimensional dataset having a sufficient number of training examples, logistic regression is less prone to over-fitting (Grover, no date).
- Logistic regression is less prone to over-fitting but it can overfit in high dimensional datasets (i2 tutorial, 2019).
- Easy to implement and does not take long time and high computation power to get trained (Berke and Colakoglu, 2019).

Disadvantages:

- Does not work well when there are correlated attributes (Berke and Colakoglu, 2019).
- Logistic Regression assumes linearity between dependent and independent variables which is very rare in real world scenario (i2 tutorial, 2019).

3.5 Introduction to Data:

Data of multiple retailers which are Appriss's clients is stored and managed by Appriss Retail Limited. Access to data has been provided by Appriss to perform analysis and derive outcomes based on the agreed problem statement i.e., the objective of this dissertation. Raw data from multiple retailers are wrangled and processed by Appriss before loading it in data warehouse. The processed data is stored in well-structured format in data warehouse in different tables and views based on the schema designed by Appriss. There are millions and billions of records stored for each retailer in warehouse which is used for analysis. These data are stored based on type of records like client details, product description, transaction level details, store details and others. Data from different tables which are frequently used is clubbed and stored in a view for better and quick access. The data maintained in data warehouse is pulled to perform analysis.

Data used for analysis in this project is of a retailer dealing in range of sports goods. Goods of different sports like football, golf, lawn bowls, mountaineering, running, and many others which included clothes, footwears, gloves, sunglasses, and other goods linked to respective sports, sports specific goods like golf balls, golf clubs, golf shoes, footballs, cleats, apparels, etc. Goods were for all range of customers like kids, adults, senior citizens and available in different sizes and colours. Goods of multiple major and medium sized brands were available with the retailer.

The table created specifically for this project is created by extracting relevant data from multiple tables and views in data warehouse. The

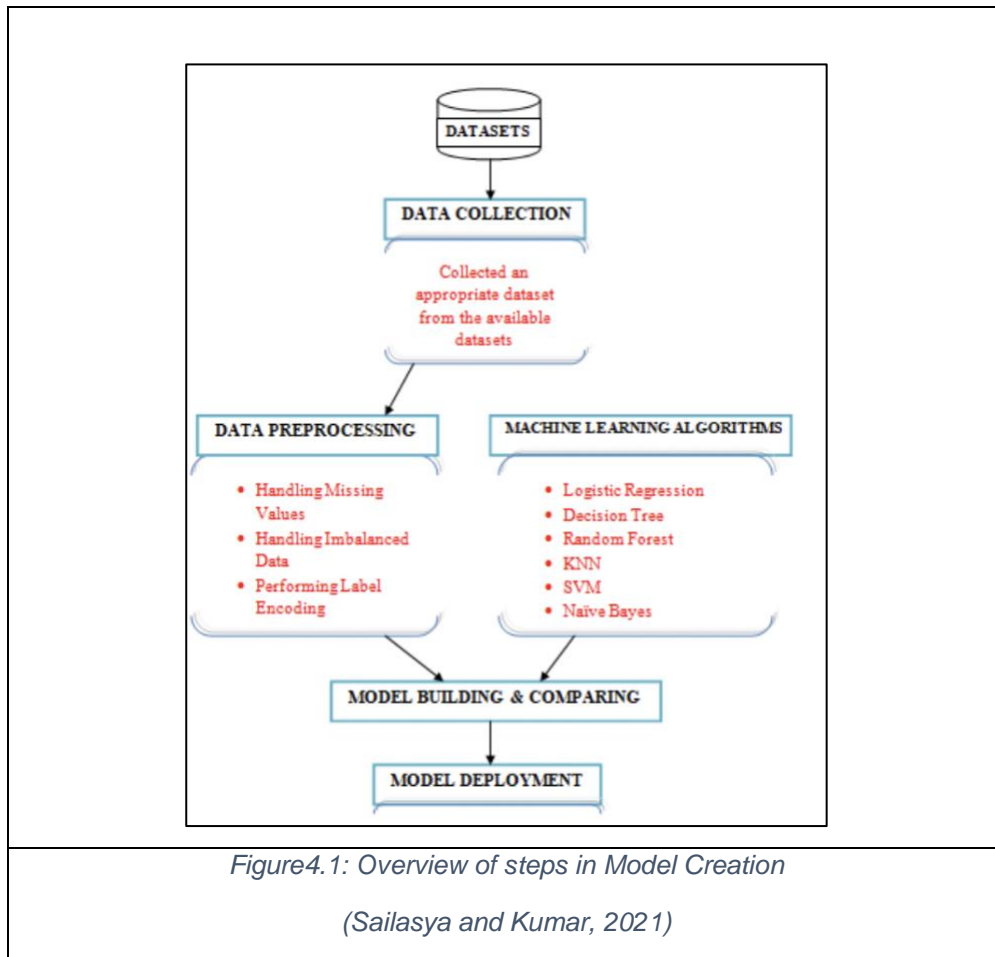
data is processed and stored at level of unique product description i.e., each record in created table contained unique products and its characteristics. Product characteristics included parameters like product description, return rate, online sales rate, average price of product, size, colour, and other relevant parameters. Each unique product was denoted by 'sku_key', if any parameter specific to product is changed like descriptions, colour, size, etc. 'sku_key' gets changed accordingly.

CHAPTER 4

METHODOLOGY

4 Methodology:

The overview of steps involved in from collecting datasets, cleaning, and creating model to make predictions and verifying predictions is implemented in dissertation as shown in below image. Each step is explained in detail further how data has been collected, source of data, details of pre-processing, splitting of dataset, model creation and validating and explaining model and outcome with relevant scores and plots.



4.1 Data Collection

Raw data consisting of details of product, customer, transactions, etc. is provided by retailers which is stored and managed by Appriss. Data is cleaned, formatted, pre-processed, and then stored in data warehouse. As many significant and relevant tables, schemas, views present in warehouse are investigated, to find relevant parameters for analysis considering dissertation objectives. Once data from warehouse is identified, merged, and cleaned, it is put into tables and views created in schema for this project.

- *Data warehouse* is a system that stores structured and semi structured data. Data warehouse provides an enterprise level solution to an organisation where it can store data from multiple departments like marketing, CRM, sales, transaction level structured and semi structured data (Google Cloud, no date).
- *Data Cleaning* is a process to remove or treat the irrelevant data present in dataset. It comprises removing duplicate or irrelevant entries, fix structural errors, filter unwanted outliers, handle missing data, validate and check the quality of data after performing cleaning operations (Tableau, no date).

4.2 Data Cleaning using PostgreSQL

After data collection, data is processed to fit the requirements of model. The analysis is to be performed at product level using machine learning (ML) models therefore, all relevant parameters i.e., characteristics of product are identified and processed to fit the data at product level. It means that every single record in table must contain unique product ('sku_key') and its characteristics like colour, size, price, etc. The processed data is then used as input to model. Below data processing steps are performed in this dissertation:

The below code shows column 'return_rate' is created using data in return column, as rate of positive return (return=Y) to total count (return= Y and N) for respective sku_key. Using, return column, 'return_rate' is calculated which helps to identify products with high return rates (returns with respect to sales). A blog briefly described in section 2.2 highlights importance of return rate in product return prediction using advanced analytics (Karabulut, 2022).

Similarly, columns 'receipted_rate' and 'ecom_rate' are created where 'Y' is replaced by 1 and 'N' by 0 and then average rate is calculated. Importance of 'ecom_rate' i.e., online sales rate in product returns is mentioned in a literature in section 2.2 (Dzyabura, Kihal and Ibragimov, 2018).

Input: Return, receipted, ecom columns

1. Transforms and replaces 'Y' with 1 and 'N' with 0 values

- **CASE WHEN** return = 'Y' **THEN 1**
WHEN return = 'N' **THEN 0 END** **AS** return,
- **CASE WHEN** receipted = 'Y' **THEN 1**
WHEN receipted = 'N' **THEN 0 END** **AS** receipted,
- **CASE WHEN** ecom = 'Y' **THEN 1**
WHEN ecom = 'N' **THEN 0 END** **AS** ecom

2. Calculate rate of return using average function and storing it in column 'return_rate'

- **AVG(return)** **AS** return_rate,
- **AVG(receipted)** **AS** receipted_rate,
- **AVG(ecom)** **AS** ecom_rate

Output: Newly created columns 'return_rate', 'receipted_rate', and 'ecom_rate' for each unique product were created in table mapped against 'sku_key'

'Quantity' is total number of respective products sold when 'sale'=1 i.e., sale was positive. Sum of quantity sold in dataset is calculated against 'sku_key' as explained in below pseudo code:

Input: Quantity column

1. Sum of values in quantity column when sale is positive

- SUM(CASE WHEN** sale = 1 **THEN** quantity **END)** **AS** quantity

Output: Values in column quantity are added when sale is '1'

Authors in two papers have explained how average price of product effects product returns and is therefore, included for analysis in the project (Samorani and Messinger, 2016), (Powers and Jack, 2015). Price (plu_amt) being paid by consumer for a product ('sku_key') across

dataset is averaged using function in PostgreSQL as shown in below pseudo code:

Input: Plu_amt column

1. Sum of values in quantity column when sale is positive

AVG(CASE WHEN sale = 1 THEN plu_amt END) AS avg_price_product

Output: New column named 'avg_price_product' containing average price of each unique product

Authors in a literature highlighted importance of product description as parameter in predicting product returns (Cui, Rajagopalan and Ward, 2020), similarly it has been used in model in dissertation. 'Product_desc' column is created by combining values of two columns 'class_desc' and 'department_desc' as shown in below pseudo code:

Input: 'class_desc' and 'department_desc' columns

1. Both columns are combined to form 'product_desc' column

- class_desc || ' ' || b.department_desc AS product_desc

Output: New column named 'product_desc' containing information related to product

In a literature, authors conducted data analysis and concluded that size and quality of product are primary reasons for returns of product (Backegren, Lonnstrom and Zetterberg, 2020). Below pseudo code shows how colour (sku_color) and size (sku_size) of products are identified from 'sku_desc' column which will be used as parameters in model for analysis.

Input: 'sku_desc' column

1. colour and size of product is detected by using split function on 'sku_desc' column.

- `split_part(b.sku_desc, '/', 2)` **AS** sku_color
- `split_part(b.sku_desc, '/', 3)` **AS** sku_size

Output: Two new column named 'sku_color' and 'sku_size' containing colour and size related information about product

To map all the above processed columns against unique product (sku_key), all columns must be grouped against sku_key. To ensure that data doesn't get skewed or imbalanced which can lead to algorithm predicting majorly the larger class with greater model accuracy (Hofmann *et al.*, 2020) only those products are selected for analysis whose total sales in dataset is more than 100. The final table containing data related to product needs to be processed in certain format so that the records (rows) are as follows:

- Each record must describe unique product (sku_key).
- Each record i.e., every 'sku_key' in table must have been sold more than 100 times.
- Each record must contain unique product with its features like colour, description, average price, etc.

Input: All relevant parameters (after data processing) selected for analysis

1. To group all processed data against 'sku_key' and select products with sales greater than 100:

- **SELECT** sku_key, other parameters as above **FROM**
TABLE_NAME
GROUP BY sku_key **HAVING SUM(sale) > 100**

Output: All data in table is stored at 'sku_key' level with sales not less than 100

4.3 Establishing connection with IDE

Once data is cleaned and processed using PostgreSQL, a database connection is established between the DBeaver (PostgreSQL scripting platform) and PyCharm (Python scripting IDE). Connection between both the platforms is established using python's psycopg2 library and providing relevant credentials like dbname, username, schema, and other details. After setting up the environment, installing the library, importing the library, and providing connection details as per the syntax, the IDE attempts to make a connection with DBeaver and if credentials are validated successfully then connection is established and relevant data from DBeaver is pulled in DataFrame, as in below pseudo code:

Input: Schema, user, database name, table name

- *Establishing connection with DBeaver:*

```
schema = 'schema_name'
user = 'user_name'
gp = greenplum.GPConnect(schema=schema_name,
user=user_name, dbname=dbname)
gp.connect()
```

- *Executing query to fetch data in DataFrame 'df'*

```
df = pd.read_sql_query(f'SELECT * FROM {schema}.table_name',
con=gp)
```

Output: DataFrame 'df'

4.4 Data Pre-processing using Python

'Return' column is added to the dataframe and is binarised based on the median value of return_rate i.e., value in return column would be '1' if 'return_rate' in corresponding record is greater than or equal to median value otherwise, it would be '0'. It is assumed that value '1' as it has return rate greater than median value would represent return and '0' would represent non-return of respective product. The operation was performed as shown in below code:

Input: 'return_rate' column

1. Creating return column based on values in return_rate column:

```
- df['return'] = (df['return_rate'] > median of return_rate).  
  astype(int)
```

Output: New column 'Return' in dataframe

Encoding:

Variables that contain string and categorical values are label transformed which is part of pre-processing data as some ML models cannot handle or performance varies with string values (Garg, 2022). Label encoding has been performed on two such columns- 'sku_color' and 'sku_size' using 'labelEncoder' function as shown below:

Input: 'sku_color' and 'sku_size' columns

1. creating object of the function:

```
- lab_size = LabelEncoder()  
  lab_color = LabelEncoder()
```


2. *transforming the column values:*

- `df['sku_size'] = lab_size.fit_transform(df['sku_size'])`
- `df['sku_color'] = lab_color.fit_transform(df['sku_color'])`

Output: Transformed values of both input columns

'sku_cost' and 'return_rate' columns were dropped from dataframe as two other columns- 'avg_price_product' and 'return' were created based on earlier mentioned two columns.

4.5 Exploratory Data Analysis

Exploratory Data Analysis (EDA) is performed by professionals to understand and get an overview of the dataset which is under analysis. It is used by data scientists to investigate dataset and summarise its characteristics using visualisations, spot anomalies, or check assumptions (IBM Cloud Education, 2020a). Below are the steps performed in EDA

4.5.1 Data Exploration

The 'dtypes' function is used to identify the data type of each parameter stored in dataframe and accordingly corrective actions can be taken as necessary. If any data typecasting is performed the results can be cross verified using 'dtype' function. Below figures show data types of columns before and after transformation:

Cleaning and Transformation is implemented in Section 4.5.2 and 4.5.3

<pre> In [304]: df.dtypes Out[304]: sku_key object return_rate float64 product_desc object sku_color object sku_size object sku_cost float64 avg_price_product float64 quantity float64 ecom_rate float64 receipted_rate float64 return int32 dtype: object </pre>	<pre> In [309]: df.dtypes Out[309]: sku_key object return_rate float64 product_desc object sku_color int32 sku_size int32 sku_cost float64 avg_price_product float64 quantity float64 ecom_rate float64 receipted_rate float64 In [310]: </pre>
Figure 4.2: Data types before label encoding	Figure 4.3: Data types after label encoding

The data sample is checked using 'head' function of Python which displays top 5 records of data. The sample helps to understand the parameters (columns) and records (rows) of data stored in dataframe as shown in Fig 4.2 and Fig 4.3 displays sample of data after cleaning.

	sku_key	return_rate	product_desc	sku_color	sku_size	sku_cost	avg_price_product	quantity	ecom_rate	receipted_rate	return
0	21711161-194957178711	0.872349	N RUNNING WOMENS ATHLETIC APPAREL	PNK6Z	S	12.60	28.774776	1866.0	0.157585	0.979452	1
1	18571604-84984536403	0.825882	GRIPS GOLF COMPONENTS	NCOLR	NSIZE	22.00	38.531111	414.0	0.980800	0.909091	0
2	21207319-194512709930	0.129252	TRAINING BOYS ATHLETIC APPAREL	B	B	12.98	26.987188	384.0	0.481042	0.947368	1
3	21989959-194375642184	0.840146	FLEECE MENS OUTDOOR APPAREL	SPWHT	S	13.10	44.814563	263.0	0.841825	1.000000	0
4	21711146-194957231447	0.882447	N RUNNING WOMENS ATHLETIC APPAREL	BLACK	XXL	14.70	34.268425	1033.0	0.957005	0.989247	1
5	21457220-194375440643	0.884923	WH SHORTS MENS GOLF APPAREL	GYHIR	32	15.62	69.398921	1477.0	0.815467	0.971014	1
6	16625088-53474511381	0.208855	W COMPETITIVE/AQUATIC FIT SWIM	UNRED	30	21.00	49.891056	1494.0	0.881551	0.978723	1
7	17001449-889362320007	0.842382	SPORT BANDS TEAM SPORTS ACCESSORIES	MNYWH	NSIZE	2.55	5.983289	1783.0	0.347899	0.949367	0
8	22297030-195251957309	0.880624	UA FLEECE WOMENS ATHLETIC APPAREL	BKBL	L	28.44	64.816124	705.0	0.845262	0.951613	1
9	21446395-97512444871	0.863291	BAGS DIAMOND SPORTS	BLACK	NSIZE	49.50	89.998080	296.0	0.778270	0.950800	0

Figure 4.4: Sample of data before cleaning

```
In [298]: df.head(10)
Out[298]:
```

	sku_key	return_rate	product_desc	sku_color	sku_size	sku_cost	avg_price_product	quantity	ecom_rate	receipted_rate	return
0	22024915-195239721234	0.084788	N FITNESS WOMENS ATHLETIC APPAREL	10866	1122	14.70	35.800000	367.0	0.816349	0.941176	1
1	20303974-010839035100	0.005245	HARD BAIT5 TACKLE	6306	899	9.11	14.970622	7386.0	0.825014	0.948718	0
2	20800911-194277468349	0.084783	FLEECE GIRLS ATHLETIC APPAREL	2474	1003	23.10	41.575962	419.0	0.311164	0.897436	1
3	21760504-194514023171	0.032974	TRAINING BOYS ATHLETIC APPAREL	7924	811	12.98	28.620273	1609.0	0.873776	0.945455	0
4	11653132-47708716390	0.004547	FISHING ACCESSORIES TACKLE	6306	899	0.73	2.486880	11369.0	0.808785	0.942398	0
5	13207990-43193111771	0.004579	SALTWATER LURES TACKLE	6306	899	3.82	7.486689	453.0	0.800000	1.000000	0
6	17110500-190005440726	0.104072	UA TOPS MENS GOLF APPAREL	8059	811	19.49	39.924550	1220.0	0.337152	0.917241	1
7	20414613-015196029963	0.167488	INLINE SKATES WHEEL GOODS	9064	937	44.76	79.990000	169.0	0.976331	0.970588	1
8	22320505-194476699551	0.035503	WOMANS GLOVES OUTERWEAR ACCESSORIES	9388	777	15.98	39.950000	163.0	0.282209	1.000000	0
9	20914930-194389002547	0.141962	NEW BALANCE W05 ATHLETIC FOOTWEAR	1203	420	36.80	80.172482	411.0	0.231144	0.955082	1

Figure 4.5: Sample of data after cleaning

The 'isnull' function is used to identify any parameter/column contains null values and it is summed up to calculate the total null values count in that respective column. Below figures show null values count before and after cleaning.

```
In [313]: df.isnull().sum()
Out[313]:
```

sku_key	0
return_rate	0
product_desc	0
sku_color	0
sku_size	0
sku_cost	267
avg_price_product	0
quantity	0
ecom_rate	0
receipted_rate	4115
return	0
dtype:	int64

Figure4.6: Null values count before data cleaning

```
In [372]: df[predictors].isnull().sum()
Out[372]:
```

receipted_rate	0
sku_color	0
sku_size	0
avg_price_product	0
quantity	0
ecom_rate	0
dtype:	int64

Figure4.7: Count of Null values after data cleaning

'Info' function is used to identify counts of records with data type of each parameter as many pre-processing is performed over parameters. The components of info function include the number of columns, column labels, column data types, memory usage, range index, and the number

of cells in each column (non-null values) (W3schools, no date, p. 3). Below figures show data type of columns before and after transformations:

```
In [311]: print(df.info(show_counts=True'))
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 219585 entries, 0 to 219584
Data columns (total 11 columns):
#   Column          Non-Null Count  Dtype
---  ---
0   sku_key          219585 non-null  object
1   return_rate      219585 non-null  float64
2   product_desc     219585 non-null  object
3   sku_color        219585 non-null  int32
4   sku_size         219585 non-null  int32
5   sku_cost         219318 non-null  float64
6   avg_price_product 219585 non-null  float64
7   quantity         219585 non-null  float64
8   ecom_rate        219585 non-null  float64
9   receipted_rate   215470 non-null  float64
10  return           219585 non-null  int32
dtypes: float64(6), int32(3), object(2)
memory usage: 15.9+ MB
None
```

Figure4.8: Data Info before transformation

```
Data columns (total 6 columns):
#   Column          Non-Null Count  Dtype
---  ---
0   receipted_rate   187510 non-null  float64
1   sku_color        187510 non-null  int32
2   sku_size         187510 non-null  int32
3   avg_price_product 187510 non-null  float64
4   quantity         187510 non-null  float64
5   ecom_rate        187510 non-null  float64
dtypes: float64(4), int32(2)
memory usage: 8.6 MB
None
```

Figure4.9: Data Info after transformation

Python's 'describe' function is one of the useful functions which provides complete descriptive statistics summary with significant details just by calling the function. It helps to understand the distribution and variation in parameters of dataframe. The components in describe function include mean, count, standard deviation, percentile, and min-max features of all the parameters (pandas, no date). It helped to identify parameters with large number of outliers, distribution of parameters and count of records in each parameter in this project.

```
[In [314]: print(df.describe().T)]
```

	count	mean	std	min	25%	50%	75%	max
return_rate	219585.0	0.079740	0.057970	0.00000	0.041002	0.070175	0.106557	9.473463e-01
sku_color	219585.0	5287.329358	3110.729315	0.00000	2075.000000	6267.000000	7602.000000	1.066700e+04
sku_size	219585.0	776.239037	310.248725	0.00000	741.000000	852.000000	1003.000000	1.147000e+03
sku_cost	219318.0	27.341860	44.317829	0.00000	9.100000	16.000000	31.050000	1.889370e+03
avg_price_product	219585.0	54.855311	71.043823	0.00984	21.222101	35.000000	64.943576	2.919990e+03
quantity	219585.0	1638.760243	22643.940805	53.00000	184.000000	404.000000	1116.000000	7.747367e+06
ecom_rate	219585.0	0.242487	0.237718	0.00000	0.060718	0.160714	0.356004	1.000000e+00
receipted_rate	215470.0	0.953601	0.071840	0.00000	0.938356	0.969072	1.000000	1.000000e+00
return	219585.0	0.499920	0.500001	0.00000	0.000000	0.000000	1.000000	1.000000e+00

Figure 4.10: Output of 'describe' function on dataset

```
[In [309]: print(df[predictors].describe().T)]
```

	count	mean	std	min	25%	50%	75%	max
receipted_rate	187510.0	0.954105	0.071126	0.00000	0.941176	0.968254	1.000000	1.000000e+00
sku_color	187510.0	5302.231438	3066.452873	0.00000	2163.000000	6267.000000	7588.000000	1.066700e+04
sku_size	187510.0	792.616932	302.617332	0.00000	777.000000	899.000000	1003.000000	1.147000e+03
avg_price_product	187510.0	51.846448	68.194785	0.00984	19.990000	34.850694	60.397954	2.919990e+03
quantity	187510.0	1828.848355	20606.851199	53.00000	207.000000	488.000000	1325.000000	7.747367e+06
ecom_rate	187510.0	0.162487	0.134890	0.00000	0.050314	0.125000	0.252033	5.036364e-01

Figure4.11: Output of 'Describe' function after data transformation

4.5.2 Data Transformation

- 'Describe' function output after cleaning shows mean and 50% of quantity (sku_color is discrete variable so it is not considered) value has significant difference, therefore needs to be transformed. Log transformation is arguably the most popular among different types of transformations (Feng *et al.*, 2014). Different transformations like log, standard scaler, cube root, and boxcox were tried on but log transformation showed better results and thus it is used for transforming the column as in below pseudo code:

Input: Column which need to be transformed

1. 'quantity' column is Log transformed

○ `df['quantity']= np.log(df['quantity'])`

Output: Transformed 'quantity' column

```
In [391]: print(df[predictors].describe().T)
```

	count	mean	std	min	25%	50%	75%	max
receipted_rate	187510.0	0.954105	0.071126	0.000000	0.941176	0.968254	1.000000	1.000000
sku_color	187510.0	5302.231438	3066.452873	0.000000	2163.000000	6267.000000	7588.000000	10667.000000
sku_size	187510.0	792.616932	302.617332	0.000000	777.000000	899.000000	1003.000000	1147.000000
avg_price_product	187510.0	51.846448	68.194785	0.009840	19.990000	34.850694	60.397954	2919.990267
quantity	187510.0	6.376669	1.267567	3.970292	5.332719	6.190315	7.189168	15.862864
ecom_rate	187510.0	0.162487	0.134890	0.000000	0.050314	0.125000	0.252033	0.503636

Figure 4.12: Output of 'Describe' function after transformation

4.5.3 Data Cleaning: Outlier Detection, Treatment, and Verification:

Outliers are data points which are significantly different from the rest of the data points in dataset, outliers are often abnormal observations that skew the data distribution, and are present in data due to inconsistent data entry, or erroneous observations (Bala, 2022). Outliers are abnormal values which have negative impact on the results of model or trying to run statistical tests and therefore, must be removed so that model's performance is not impacted. Methods used to detect and treat outliers in this dissertation are as follows:

Outlier Detection:

- **Boxplot:** Boxplot enables to visualise distribution of data by plotting data into quartiles which helps to identify data points near mean and detect outliers (Microsoft, no date). Boxplots in this project enables to identify the parameters/columns that contains outliers. Fig 4.13 shows the boxplots of each column in dataset before removing outliers. Post removal, boxplot is plotted again to verify the outcome of outlier treatment.

User-defined function 'data_cleaning' has multiple steps, first step in function is to boxplot each of the independent variables which is implemented using 'for loop' and 'boxplot' function as shown below:

Input: Dataset for which boxplot is to be plotted

1. User-defined function 'data_cleaning' with parameter

- `def data_cleaning():`

2. Box plot for each continuous variable is plotted

- `for col_names in predictors:`
 `df.boxplot(col_names) #plot for each predictor`

Output: Boxplot of each predictor variable

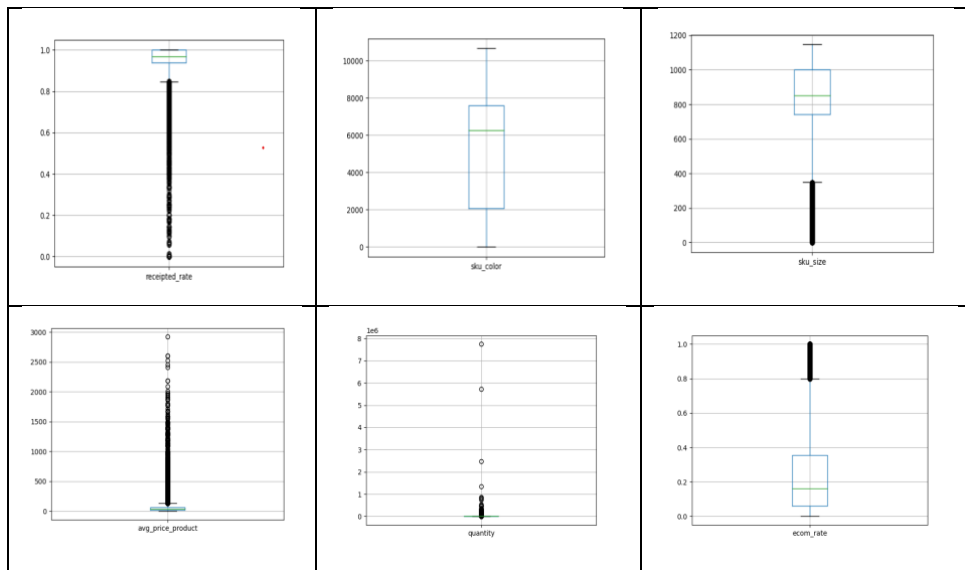
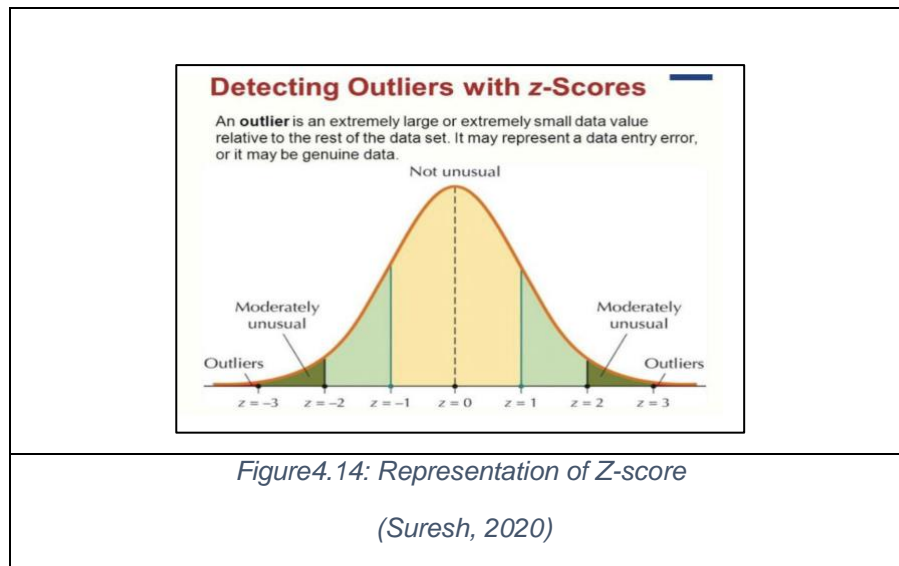


Figure4.13: Boxplots of independent variables before data cleaning

- **Z-score:** Z-score of '1' for a data point indicates that particular data point is one standard deviation away from the mean. In most datasets, 99% of values have z-score between 3 and -3 (Hayes, 2022). Positive and negative scores indicate that points lie on either side of the mean. In this project, Z-score is used to detect records which are outliers based on their Z-scores. Z-score is calculated for each record in the dataframe, and a threshold of Z-score 3 and -3 is set, records with beyond the threshold values are considered as outliers and must be treated.



After plotting boxplot, z-scores of predictor continuous variables are calculated (second step in 'data_cleaning' function) as shown below:

Input: Dataset for which Z-score is to be plotted

1. After boxplot, z-score is calculated for independent continuous variables
 - `for col_names in z_predictors:`
`np.abs(stats.zscore(col_names)). #z-score calculation`

Output: Z-score for each record of continuous predictor variables



Figure 4.15: Z-scores of continuous parameters before data cleaning

Outlier Treatment using IQR:

- Inter Quartile Range (IQR) is used to measure variability by dividing dataset into quartiles, complete data is sorted in ascending order and split into four quartiles- Q1 representing 25th percentile of data , Q2- 50th percentile of data, Q3- 75th percentile of data, and Q4 (Maini, 2020). In the project, a function is created to calculate IQR which is the difference between Q3 and Q1. Data points which lie below $Q1 - 1.5 \times IQR$ or above $Q3 + 1.5 \times IQR$ are considered as outliers and same has been implemented to

detect and remove outliers (third step in 'data_cleaning' function)
in this project as below:

Input: columns which need to be cleaned

1. Calculate 25th(Q1), 75th(Q3) percentiles. IQR (Q3-Q1), upper and lower bounds are calculated.

for col_name in predictors:

- Q1 = np.percentile(col_name, 25) # similarly Q3 with parameter as 75
- upper_bound = values in column_name greater than (Q3 + 0.5 * IQR)
- lower_bound = values in column_name less than (Q1 - 0.5 * IQR)

2. Records beyond upper and Lower bounds are dropped

- df.drop(indexes in upper_bound, inplace=True)
#similarly for lower bound

Output: Records (indexes) which contained outliers are removed from dataset.

Treatment Verification:

Boxplot is used to identify columns containing outliers and then Z-score is used to identify the values which are outliers in respective columns and IQR is performed to remove the respective outliers based on calculated IQR values and transformation is done to normalise the data points. The outlier detection and treatment are verified again using the boxplot and Z-scores on processed dataset after outlier removal and data transformation, z-scores between +3 and -3 are acceptable.

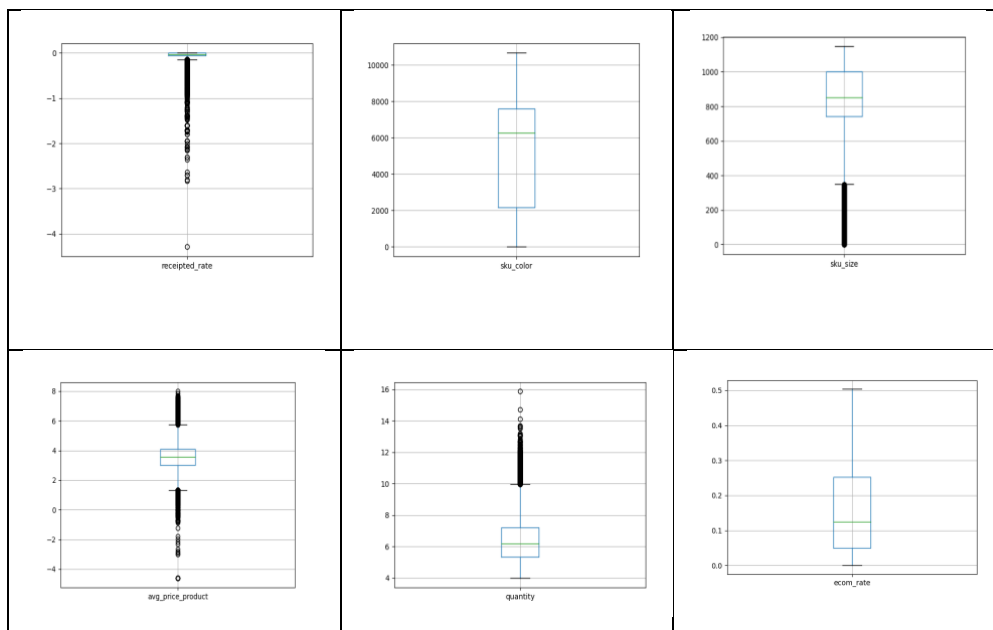


Figure 4.16: Boxplots of independent variables of model after data cleaning

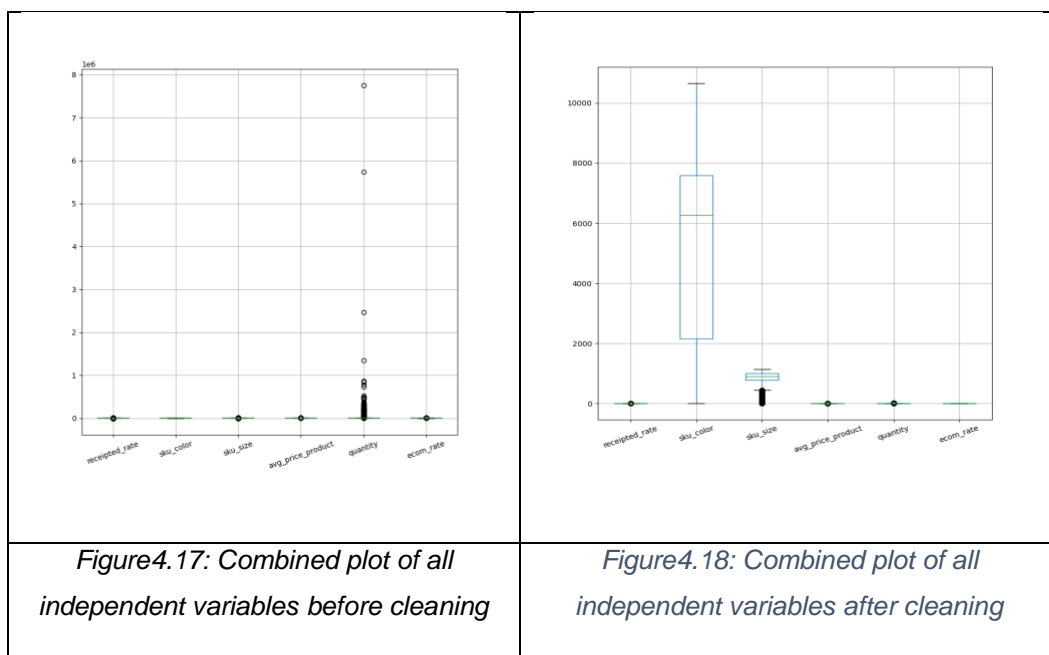


Figure4.17: Combined plot of all independent variables before cleaning

Figure4.18: Combined plot of all independent variables after cleaning



Figure4.19: Z-scores of continuous parameters after data cleaning

4.6 Model Creation and Evaluation

Post outlier detection and treatment processed data can be used to fit in the model. In this project, one function has been created to evaluate three models on same dataset namely, Light Gradient Boosting Machine (LightGBM), RandomForest Classifier, and Logistic Regression. The function involves multiple steps which are described as below: splitting of data into training and testing sets, fitting data in models, making predictions, identifying important features influencing outputs, and calculating different scores to validate model like cross validation score, testing and training accuracy, model accuracy, roc score and curve plot, plotting confusion matrix, and classification report.

4.6.1 Splitting data:

Data is split into training and testing sets to prevent overfitting and to accurately evaluate the model's prediction and efficiency. Train and test split is simple and quick method to train algorithm and compare prediction results of algorithm with the testing dataset (Isitapol, 2022). Data splitting is used in data modelling where models are created to make predictions using advanced analytics (Gillis, no date). The data has been split in ratio of 70:30, where 70% data is used as training set and remaining 30% as testing set, so that both datasets created are of significant size. Split has been performed using 'train_test_split' function.

- Overfitting:

Overfitting is a concept in data science, which occurs when a statistical model fits exactly against its training data and the algorithm unfortunately cannot perform

accurately against unseen data, defeating its purpose (IBM Cloud Education, 2021).

Function 'model_eval' is defined with parameter as 'ml_model' which splits data into training and testing dataset:

Input: Processed final dataset which is used to train model and make predictions.

1. Splitting data in training and testing dataset

```
- def model_eval(ml_model):  
    x_train, x_test, y_train, y_test= train_test_split(df[predictors],  
df[dependent/outcome], test_size=0.3, random_state=10)
```

Output: Training dataset with 70% size and testing dataset with 30% size are created

Other predictor variables after transformation can be added in predictors list to carry analysis specific to any product e.g., footwear or specific brands as it has been implemented in dissertation. Below pseudo code explains same:

Input: Transformed column with values and then add to predictors

1. Transforming values in column before adding to dataframe:

```
- if(pred== 'Footwear'):  
    df['footwear'] = df['col_name'].contain('shoes|cleats|footwear')  
- if(pred== 'Brand'):  
    df['brand'] = df['col_name'].contains('NIKE| ADIDAS')
```

2. Defining predictors List based on new transformed columns

- predictors= ['receipted_rate', 'sku_color', 'sku_size', 'avg_price_product', 'quantity', 'ecom_rate', 'footwear']
- predictors = ['receipted_rate', 'sku_color', 'sku_size', 'avg_price_product', 'quantity', 'ecom_rate', 'brand']

Output: New predictor list which can be used in analysis for specific product or brands and any other parameters.

4.6.2 Model fitting and prediction:

After splitting data, training data is fed into model to train it. In this project, three models have been fed same training dataset to make better comparison among the model's performance. Training data is fed into the model using 'fit' function. Once model fitting is done successfully, the model can learn about the data and make predictions.

Fitting data to model:

Once model is initialised 'fit' method trains the algorithm on the training data (Ebner, 2022). Models have different hyperparameters to tune the model, an improper fitted model doesn't provide expected outputs and can be less helpful in deriving insights and expected outcomes. A properly fitted model must capture complex relationship between independent and dependent variables.

- Models have different data requirements like LightGBM model can learn and handle null values in training dataset itself. In LightGBM, three hyperparameters are used in fit function, namely, learning rate, max depth, and random state (second step in 'model_eval' function) as shown in below code:

Input: Training dataset and defining model with hyperparameters

1. Defining model:

```
- if (ml_model == 'LGBM'):  
    mod = LGBMClassifier(learning_rate=0.09, max_depth=3,  
                        random_state=42)
```

2. Fitting the model using training data and hyperparameters:

```
- mod.fit(training dataset, (testing dataset)),  
        categorical_feature=[col_names])
```

Output: LightGBM model is defined and is trained with training dataset

- While RandomForest and Logistic Regression cannot handle null values which makes it necessary to treat the null values and has been treated using 'SimpleImputer' class. 'SimpleImputer' provides basic strategies for imputing missing values like imputing with a provided constant value, or using the statistics (mean, median or most frequent) of each column in which the missing values are located (scikit learn, no date). After replacing null values, both the algorithms are defined and trained using training dataset. RandomForest has been tuned using two hyperparameters in 'fit' function- number of estimators and maximum depth as shown in below pseudo code:

Input: Replacing null values in columns, training dataset, and defining model with hyperparameters

1. Replacing null values using Imputation technique:

```
- elif (ml_model == 'RF'):  
  
    imp = SimpleImputer(replace 'NaN' with mean)  
  
    imp_train = imp_train = imp.fit(df['col_name'])
```

2. Defining model:

- `mod = RandomForestClassifier(n_estimators=100, max_depth=3)`

3. Fitting the model using training data and hyperparameters:

- `mod.fit(predictor, independent- train set)`

Output: RandomForest model is defined and is trained with training dataset

Input: Replacing null values in columns, training dataset, and defining model with hyperparameters

1. Replacing null values using Imputation technique:

- o `elif (ml_model=='LR'):`

- `imp = SimpleImputer(replace 'NaN' with mean)`

- `imp_train = imp.fit(df['col_name'])`

2. Defining model:

- o `mod= LogisticRegression()`

3. Fitting the model using training data and hyperparameters:

- `mod.fit(predictor, independent- train set)`

Output: Logistic Regression model is defined and is trained with training dataset

4.6.3 Prediction

After training data has been fit into the model successfully then the model can be used to make predictions on unseen data and the predictions can also be validated using the model on testing dataset. Predictive analytics comprises many statistical techniques including machine learning, predictive modelling and data mining and uses statistics (both historical and current) to estimate, or 'predict', future outcomes (Wakefield, no date). Predictions on existing data is performed on testing dataset using 'predict' function and of respective models and same can also be used to predict new instances (Brownlee, 2020).

Model can make predictions on testing dataset of predictor variables (third step in 'model_eval' function) as below:

Input: Testing dataset

1. *Prediction using model:*

○ `mod.predict(testing dataset)`

Output: Prediction data is available which is made using predictors in testing dataset.

4.6.4 Important Features

One of the objectives of project is to identify features or dependent variables which influence the outcome of model based on scores. Models have attribute 'feature_importances_' which assigns score to features of machine learning model that defines how "important" is a feature i.e., enabling feature selection (Bonaros, 2021). It measures

contribution of each feature on the outcome of the classifier regardless of shape, direction of effect, or relationship (Saarela and Jauhiainen, 2021).

The below grid explains how important features are identified (fourth step in 'model_eval' function):

Input: Calling inbuilt function of model

1. Identifying important features using inbuilt function of model:

- `feat_importances = pd.Series(mod.feature_importances_, index=training dataset columns)`

Output: Series with predictor name and importance score.

4.6.5 Cross-validation score

Cross-validation (CV) is used to quantify generalisation ability of predictive models and restricts overfitting using data resampling method, CV belongs to family of Monte Carlo methods (Berrar, 2018). Cross-Validation is a powerful tool and understand data and provides information about algorithm's performance (Shulga, 2018). It partitions data into defined number of partitions or folds, predictive analytics is implemented on all the folds and the average of error estimate is considered as CV score.

'Kfold' function is used to define number of folds and other relevant parameters which are used to calculate CV score. Average, minimum, maximum and standard deviation (summary) among the CV scores across folds have also been calculated to get better understanding of overall CV score.

- Kfold: In this method, dataset is divided into 'k' number of folds, for each model build model on 'k-1' folds and test model's performance on kth fold. This process needs to be repeated until each fold in 'k' folds becomes test set. Finally average of accuracies of model is called as CV score and serve as performance metric for the model (Goyal, 2021).
- CV score was calculated (fifth step in 'model_eval' function) in project as shown in below pseudo code:

Input: Calling inbuilt function of model

1. Defining method to perform cross-validation:

- o `cv1 = KFold(number of splits, random_state=1, shuffle=True)`

2. Calculating CV score:

- o `cv_score= cross_val_score(mod, predictor_vars, dependent_var , cv=cv1, scoring='accuracy')`

3. Summary of CV score:

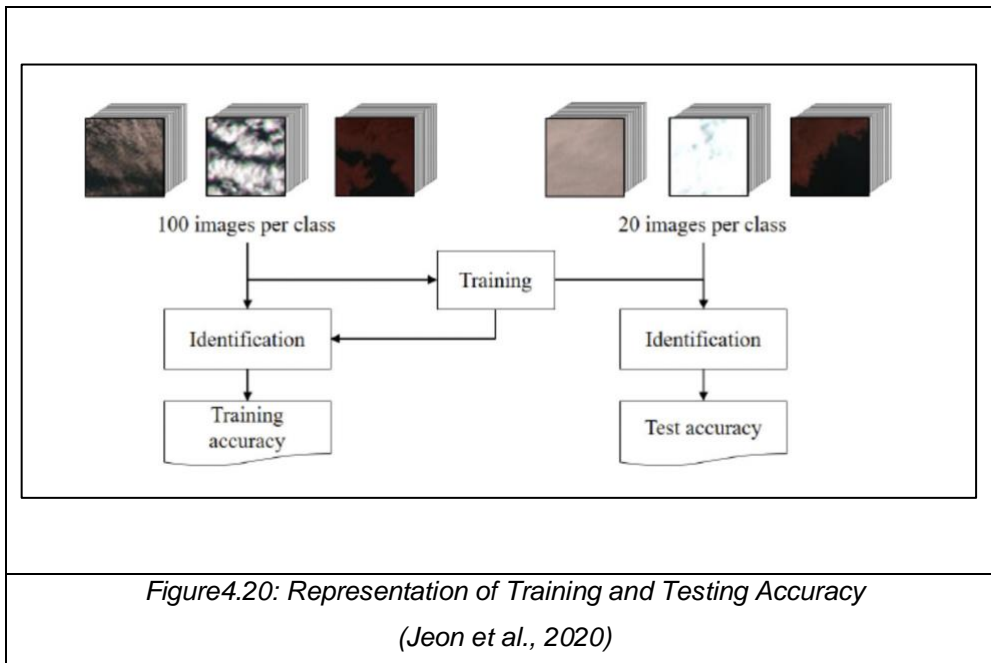
- o `np.mean(cv_score), np.std(cv_score), np.min(cv_score), np.max(cv_score)`

Output: CV scores of datasets with its summary

4.6.6 Model Accuracy (Training and Testing Accuracy)

A model's accuracy is based on correct predictions that are made based on the columns in training dataset, training files are bundled and then verified against algorithms to predict accuracy (IBM, 2022). Training accuracy means how much of identical data are used for both training and testing and training, whereas test accuracy explains how

trained model is able to identify data which were not included in training dataset (Jeon *et al.*, 2020).



‘model_name.score’ function is used calculate model score with inputs as independent and dependent variables as shown in below pseudo code:

Input: Predictor and Independent Variables

1. Calculating model score:

○ `mod.score(predictor_vars, dependent_var)`

Output: Overall score of respective model

4.6.7 AUC Score

The 'metrics.accuracy_score' function accepts two arguments- predicted values using model and testing dataset, each pair of testing and predict labels is compared in iterations to record correct predictions then the number of correct predictions is divided by total number of labels to calculate accuracy score (Java Point, no date). Accuracy is the number of correct predictions from all predictions made (IBM, 2022). In section 2.3, a literature highlights importance of model accuracy score for classification models (Kulkarni, Batarseh and Demir, 2020). Accuracy is one of the widely used metrics to measure the performance of the model.

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

Fraction predicted correctly

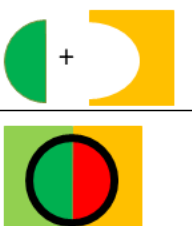


Figure4.21: Accuracy Score

(Long, 2018)

The below pseudo code shows general syntax to calculate AUC score using 'roc_acc_score' function with parameters as dependent variable's testing set and values predicted using model:

Input: Dependent Variable of testing dataset and predicted values

1. Calculating AUC score:

○ `roc_auc_score(y_testing_datset, predicted_values))`

Output: AUC score of prediction

4.6.8 ROC Curve

A Receiver Operating Characteristic (ROC) curve is a graphical plot used to visualise the performance of a binary classifier system as its discrimination threshold is varied (Scikit Learn, no date). In many applications, ROC shows how a predictor compares to the true outcome with a great advantage of estimating performance without pre-defined threshold it gives criteria to choose an optimal threshold based on certain cost function or objective (Muschelli, 2019).

The 'roc_curve' function takes two arguments as testing dataset of dependent variable and the calculated probability of predicted outcomes and plots roc curve as output as displayed in below pseudo code:

Input: Testing dataset and probability of predicted outcomes

1. Plotting ROC curve:

- `metrics.roc_curve(y_testing_dataset, probability of outcomes)`

Output: ROC curve of predicted outcomes

4.6.9 Prediction Probability

One of the objectives of this project is also to determine the probability of predicted returns. The model classifies whether product will be returned or not and 'predict_proba' function enables to identify the probability of event. The 'sklearn' estimators implement the 'predict_proba' method that returns the class probabilities for each data point in an array of lists containing probabilities for each data

point (Myrianthous, 2021). 'predict_proba' function uses the line equations passed through a sigmoid activation function to calculate the prediction probabilities (Radecic, 2021).

'model_name.predict_proba' function is used to calculate probability of the outcomes with argument as testing set of independent variables as shown in below pseudo code:

Input: Testing dataset of predictors

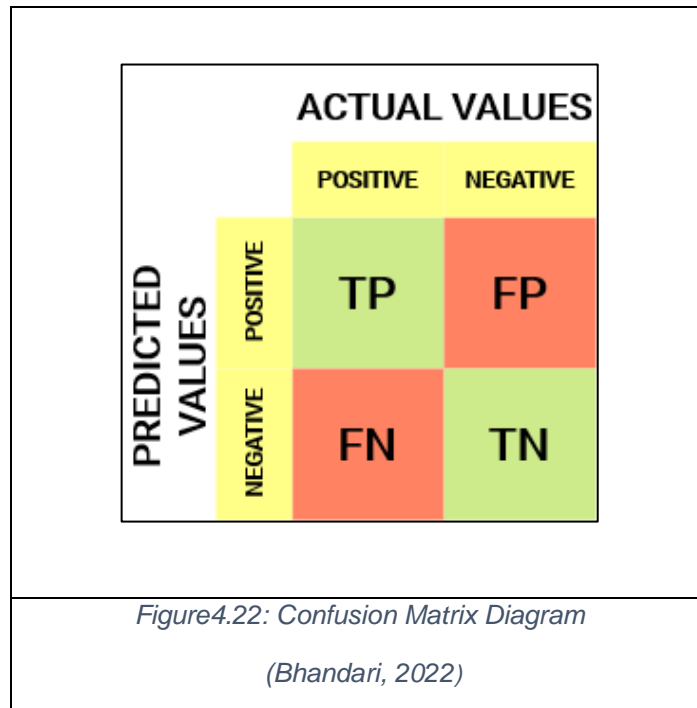
1. Calculating probability of predictions:

○ mod.`predict_proba`(x_test)

Output: Probability of each predicted outcome

4.6.10 Confusion Matrix

Performance of a classification model can be measured using Confusion matrix, an $N \times N$ matrix, where N is the number of target classes (Bhandari, 2022). Confusion matrix is a matrix of counts of True Positive (TP) values, True Negative (TN), False Positive (FP), and False Negative (FN) values. These values are comparison of actual target variables and those which are predicted using model.



The below pseudo code shows general syntax to plot confusion matrix using 'plot.confusion_matrix' function with model, independent and dependent variables testing set as arguments of function:

Input: Testing dataset and probability of predicted outcomes

1. Plotting Confusion matrix:

- `metrics.plot_confusion_matrix(mod, x_testing_set, y_testing_set)`

Output: Confusion matrix

4.6.11 Classification Report

It is report of the classification which summarises majority of details about the classification, the report includes summary of the classification performed by the models in the project. The importance of scores in classification report has been highlighted briefly in literature review section (Orozco-Arias *et al.*, 2020). The summary report is created for each model. It comprises of mainly 5 columns and (N+3) rows, where columns include class label's name followed by Precision, Recall, F1-score, and Support, N rows are for class labels and other three rows are for accuracy, macro average, and weighted average (Saini, 2022).

Using values from confusion matrix below parameters are calculated:

$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$	$Recall = \frac{TP}{TP + FN}$
$Precision = \frac{TP}{TP + FP}$	$F1\ Score = 2 * \frac{Precision * Recall}{Precision + Recall}$

Classification Report Values Formula (Kanstren, 2020)

The below pseudo code shows syntax to print classification report using 'metrics.classification_report' function with dependent variable testing set and predicted values as parameters of function:

Input: Testing dataset and probability of predicted outcomes

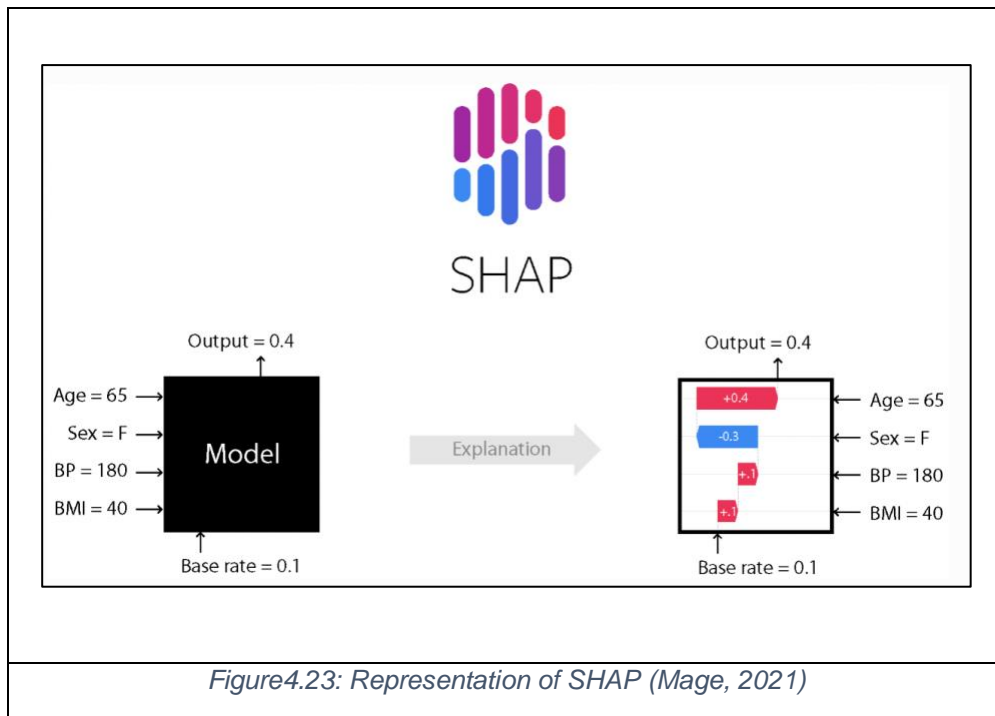
1. *Display classification report:*

- `metrics.classification_report(y_testing_set,
mod.predict(x_testing_set))`

Output: Classification Report

4.6.12 SHAP values

The models implemented provide insights, prediction, different validation scores, etc. whereas, SHAP values help to understand the contribution of features of model in prediction/classification. SHAP is an acronym for Shapley Additive exPlanations. SHAP values interpret the contribution of a feature for having respective score in comparison to predicted values using baseline values (Danb and Cook, no date). SHAP values calculation uses cooperative game theory to measure quantify positive or negative contributions towards final outcome for a particular record in dataset (Mage, 2021).



Below are the plots which elucidate the contribution of features and working of model:

4.6.12.1 SHAP Summary Plot

The plot displays the contribution of each feature on the classes of outcome, outcome can be binary or multi class (Cohen, 2021). For each feature, different colours can be used to symbolise different classes of outcomes.

'shap.summary_plot' function is used to plot summary plot with SHAP values, independent variables values and other optional parameters as inputs as shown in below pseudo code:

Input: SHAP values, predictor variable values, and labels on axes

1. Function to plot SHAP summary:

- `shap.summary_plot(shap_values, predictors.values, plot_type=bar, labels on X-axis, labels on Y-axis)`

Output: SHAP Summary plot

4.6.12.2 SHAP Dependence Plot

In Dependence plot, x-axis mentions true values of feature and y-axis mentions SHAP values of same features and the dispersion in graph represents the interaction effects between the two values (SHAP, no date b). The plot helps to analyse the complexity of relation between feature's values and it's SHAP values.

'shap.dependence_plot' function is used to plot dependence plot with number of iterations, SHAP values, lables, and independent variable's values as arguments as shown in below pseudo code:

Input: SHAP values, predictor variable values, and Labels on axes

1. Function to plot SHAP Dependence:

- `for i in (lenth of predictors):`
`shap.dependence_plot(i, shap_values, df[predictors].values, feature_names=predictors)`

Output: SHAP Dependence plot

4.6.12.3 SHAP Force Plot

This plot can be used to analyse the case of a particular observation how features influenced the prediction made by model, it is used for error analysis or for detailed understanding of an observation (Cohen, 2021).

The below pseudo code shows syntax to plot force using 'shap.force_plot' function with expected values, SHAP values and independent variable's values as parameters of function:

Input: Expected values, SHAP vales, predictor variable values, labels on axes, and plot method

1. Function to plot SHAP Dependence:

- `shap.force_plot(expected_value, shap_values, df[predictors].values, feature_names=predictors, matplotlib=True)`

Output: SHAP Dependence plot

4.6.12.4 SHAP Waterfall Plot

Waterfall plot is next local level understanding of a particular record and its outcome calculated by model, the parameters of waterfall plot function need specific record as input to create the plot (Cohen, 2021). The values indicate effect of each feature on the predicted outcome.

The below pseudo code shows syntax to plot waterfall using 'shap.waterfall_plot' function with expected values, SHAP values and independent variable's test set as parameters of function:

Input: SHAP vales, expected values, data, Labels on axes, and plot method

2. Function to plot SHAP Waterfall:

- `shap.waterfall_plot(shap.Explanation(values=shap_values, expected_value, data=x_test_rows, feature_names=x_test.columns`

Output: SHAP Waterfall plot

4.6.12.5 SHAP Decision Plot

The plot describes the overall effect and interaction of observations on particular outcome unlike dependence plot where it shows effect of single feature on prediction. It accounts both main and interaction effects during plot (SHAP, no date a).

‘shap.decision_plot’ function with arguments as expected values, SHAP values, and data to be plotted is for decision plotting as shown in below pseudo code:

Input: SHAP vales, expected values, data subset, Labels on axes, and plot method

1. Function to plot SHAP Decision:

- `shap.decision_plot(explainer.expected_value, shap_values, data_subset, ignore_warnings=True)`

Output: SHAP Decision plot

CHAPTER 5

RESULT AND ANALYSIS

5 Result and Analysis

The objective of this dissertation is to predict if a product will be returned or not based on the product's characteristics. To do so, we train three different machine learning algorithms: LightGBM, RandomForest, and Logistic Regression. We first study and compare the predictive performance of the three algorithms. Using the best-fitting model (LightGBM), we then analyze how different product characteristics affect the likelihood of a product being returned.

5.1 Prediction results

The three grids below display the predicted classification of each algorithm along with the probability of predicted outcome. In grid '1' represents prediction that product will be returned (return rate greater than or equal to median value) and '0' represents prediction that product will not be returned (return rate less than median value). Below are the following figures of prediction of three models:

Predicited result where 1 stands for predicted return and 0 for predicted non-return along with its probability:

	Prediction	Prediciton Porbability
0	0	0.064118
1	1	0.637030
2	1	0.963275
3	0	0.254149
4	0	0.360399
5	0	0.079816
6	0	0.471998
7	1	0.563971
8	1	0.767581
9	1	0.910305
10	0	0.012415
11	0	0.423156
12	0	0.080436
13	1	0.606466
14	0	0.053358

In [282]:

Figure5.1: Prediction/Classification using LightGBM

Predicited result where 1 stands for predicted return and 0 for predicted non-return along with its probability:

	Prediction	Prediciton Porbability
0	0	0.181683
1	1	0.601522
2	0	0.516390
3	0	0.235049
4	0	0.367071
5	0	0.249815
6	1	0.519281
7	1	0.543302
8	1	0.725642
9	1	0.679337
10	0	0.151379
11	0	0.409450
12	0	0.331257
13	1	0.616105
14	0	0.278131
15	1	0.570309
16	1	0.669808
17	0	0.191551
18	0	0.348944

Figure5.2: Prediction/Classification using RandomForest

Predicted result where 1 stands for predicted return and 0 for predicted non-return along with its probability:

	Prediction	Prediction Probability
0	0	0.457798
1	1	0.687456
2	1	0.507082
3	0	0.388680
4	0	0.440969
5	0	0.193666
6	0	0.452922
7	1	0.932109
8	1	0.606093
9	1	0.577372
10	0	0.252530
11	0	0.338433
12	0	0.443399
13	0	0.450077
14	0	0.360207
15	1	0.601886
16	1	0.610566

Figure 5.3: Prediction/Classification using Logistic Regression

5.2 Model accuracy score

The next verification step to measure model's performance is model score. Model score helps to measure correct predictions/classifications to total predictions (Developers Google, no date). Below figures shows model scores based on dataset passed of all three models:

<pre>...: Training Accuracy: 0.7857105 Testing Accuracy: 0.776723 Model Accuracy: 0.7830142 In [272]:</pre>	<pre>Training Accuracy: 0.7200759 Testing Accuracy: 0.7199083 Model Accuracy: 0.7200256</pre>	<pre>Training Accuracy: 0.6606352 Testing Accuracy: 0.6592182 Model Accuracy: 0.6602101</pre>
<i>Figure5.4: Scores-LightGBM</i>	<i>Figure5.5: Scores-RandomForest</i>	<i>Figure5.6: Scores-Logistic Regression</i>

The above grid shows that the performance of LightGBM in terms of accuracy is better than the other two models on all three data partitions.

5.2.1 Cross-Validation (CV) score

Without appropriately chosen hyperparameters, machine learning models may exhibit poor predictive performance due to over-or-underfitting. To assess how the models perform out-of-sample, we conduct 10-fold cross-validation. 'Cross_val_score' runs cross validation on dataset to understand whether model's generalisation ability over whole dataset, it returns one score per split and average of those scores is the output (Allwright, 2022) as CV score in dissertation. The accuracy score of test and train set is not by chance thereby we verify it with CV score. Below figures show CV score of each fold and

summary of CV scores of all folds for the models implemented in dissertation:

```
CV score of dataset is:  
[0.76055158 0.75818985 0.76078013 0.76314186 0.76504647 0.76580832  
0.76039921 0.75619048 0.76426667 0.76167619]
```

Figure5.7: CV score of 10-folds (LightGBM)

```
Summary of CV score for dataset is:  
Average: 0.7616051 | Std: 0.002884223 | Min: 0.7561905 | Max - 0.7658083
```

Figure5.8: CV score summary of folds (LightGBM)

```
CV score of dataset is:  
[0.70447966 0.71636447 0.71727868 0.72718269 0.71438367 0.72306872  
0.72786835 0.71428571 0.71565714 0.72350476]
```

Figure5.9: CV score (RandomForest)

```
Summary of CV score for dataset is:  
Average: 0.7184074 | Std: 0.006740909 | Min: 0.7044797 | Max - 0.7278684
```

Figure5.10: CV score summary (RandomForest)

```
CV score of dataset is:  
[0.66295901 0.65526436 0.65290264 0.65960689 0.66417797 0.66128295  
0.66448271 0.65942857 0.66598095 0.65988571]
```

Figure 5.11: CV score (Logistic Regression)

```
Summary of CV score for dataset is:  
Average: 0.6605972 | Std: 0.003917274 | Min: 0.6529026 | Max - 0.665981
```

Figure 5.12: CV score summary (Logistic Regression)

The summary of scores of all three models shows that LightGBM best fits the data and underscores the better performance of LightGBM.

5.3 AUC score

The accuracy metric can be uninformative especially for imbalanced datasets. In classification problems, the AUC score can be used as an alternative metric to evaluate predictions (Zvornicanin, 2021). Contrary to the accuracy metric, the AUC metric can better handle imbalanced data. Below figures show the AUC scores of all three models:

```
In [272]: print('\n Roc accuracy score of prediction is: %.7g' % roc_auc_score(y_test, prediction))  
  
Roc accuracy score of prediction is: 0.7779127
```

Figure5.13: AUC score- LightGBM

```
Roc accuracy score of prediction is:  
0.7111761
```

Figure5.14 :AUC- RandomForest

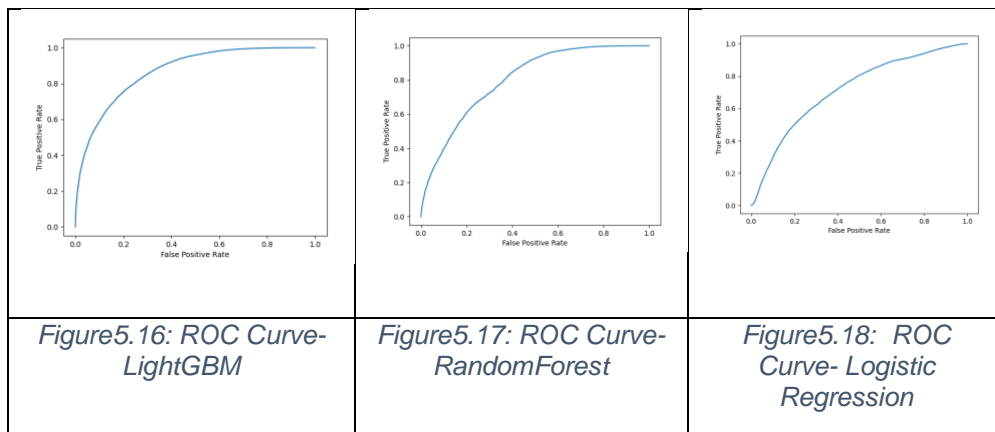
```
Roc accuracy score of prediction is:  
0.6454809
```

Figure5.15: AUC- Logistic Regression

The AUC score of LightGBM model is best among the three model underlining the performance in making predictions by model.

5.4 ROC Curve

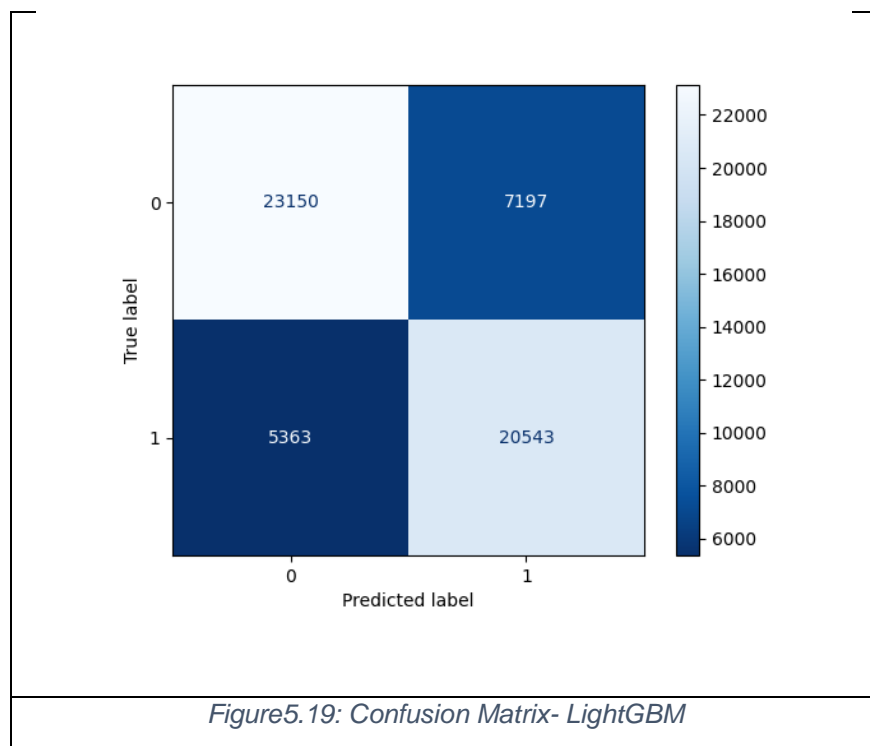
The AUC score is a summary measure of the ROC curve. Accordingly, the ROC curve is another suitable technique to evaluate model performance. The ROC curve shows how the False Positive Rate (FPR) and True Positive Rate (TPR) vary with different classification threshold values (Naveenkumar, 2021). The x-axis is the FPR and the y-axis is the TPR. Just like the AUC, the ROC curve is a good tool to visualize the performance of binary classification algorithms even for imbalanced datasets. Below are the ROC curve plots for all three models:

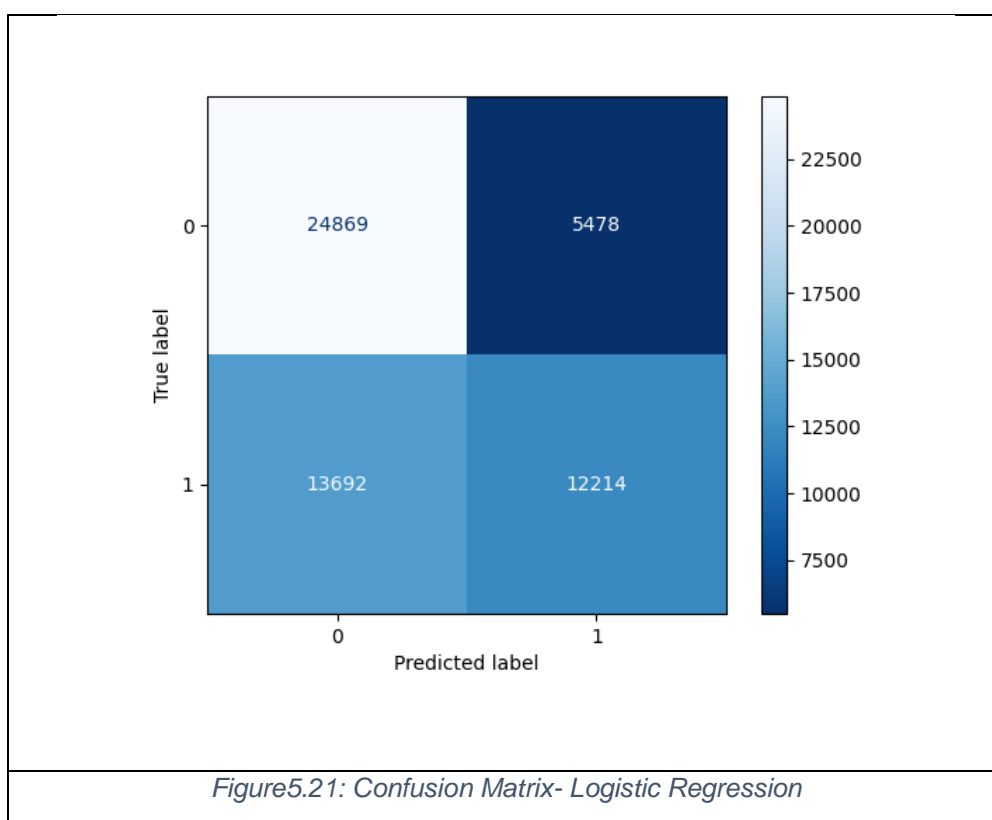
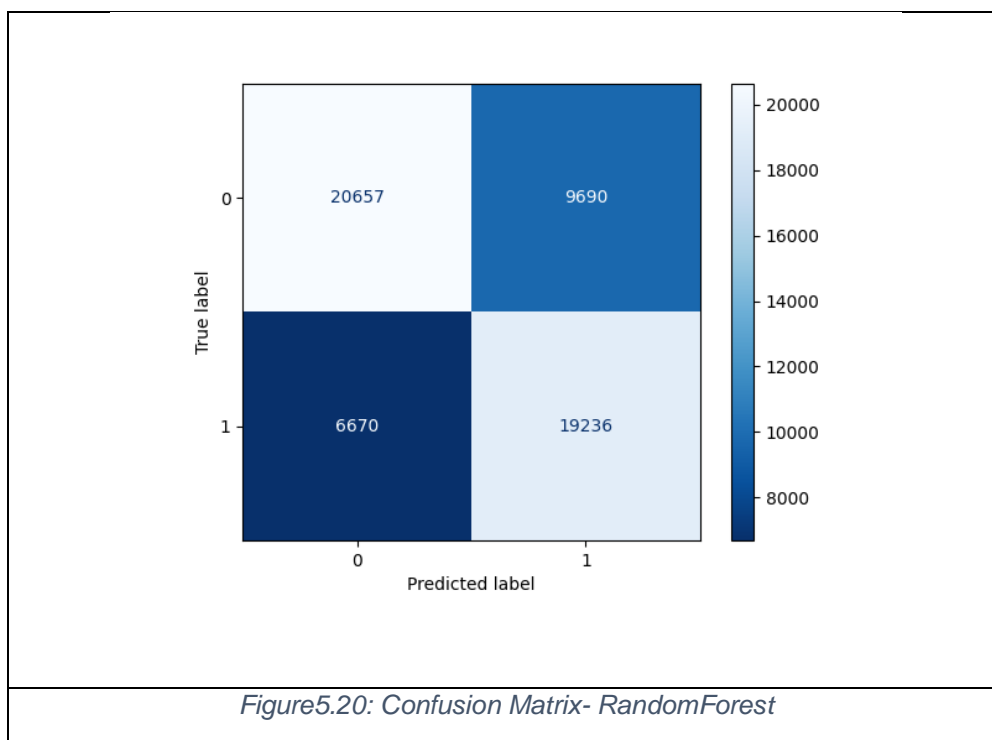


The ROC curve for LightGBM is generally above the ROC curve of the other two models, further substantiating that LightGBM outperforms the other two models.

5.5 Confusion Matrix

The confusion matrix is another common technique to assess the performance of classification algorithms. It can also be used to calculate recall, F1, precision, and accuracy score (Narkhede, 2018) to be displayed in a classification report. The confusion matrix shows True Positives, False Positives, False Negatives, and True Negatives values as predicted by models. Below figures show the confusion matrix for each of the three models:





5.6 Classification Report

The classification report is a commonly used format to summarize the performance of classification algorithms using several key metrics.

The values in classification report can be derived from the values in the confusion matrix (Chouinard, 2022). Precision is the ratio of correct positive predictions to total positive predictions, recall is the ratio of correct positive predictions to actual positives, F1 score is the harmonic mean of precision and recall, and support is the number of values in each class (Zach, 2022). The higher the accuracy, precision, and F1 score the better is the model (Exsilio Solutions, 2016).

Metrics Classification report is as follows:				
	precision	recall	f1-score	support
0	0.81	0.76	0.79	30347
1	0.74	0.79	0.77	25906
accuracy			0.78	56253
macro avg	0.78	0.78	0.78	56253
weighted avg	0.78	0.78	0.78	56253

Figure5.22: Classification Report- LightGBM

```

Metrics Classification report is as follows:
              precision    recall  f1-score   support

         0       0.77       0.69       0.73     30347
         1       0.68       0.75       0.71     25906

    accuracy                0.72     56253
   macro avg       0.72       0.72       0.72     56253
  weighted avg       0.72       0.72       0.72     56253

```

Figure5.23: Classification Report- RandomForest

```

Metrics Classification report is as follows:
              precision    recall  f1-score   support

         0       0.64       0.82       0.72     30347
         1       0.69       0.47       0.56     25906

    accuracy                0.66     56253
   macro avg       0.67       0.65       0.64     56253
  weighted avg       0.67       0.66       0.65     56253

```

Figure5.24: Classification Report- Logistic Regression

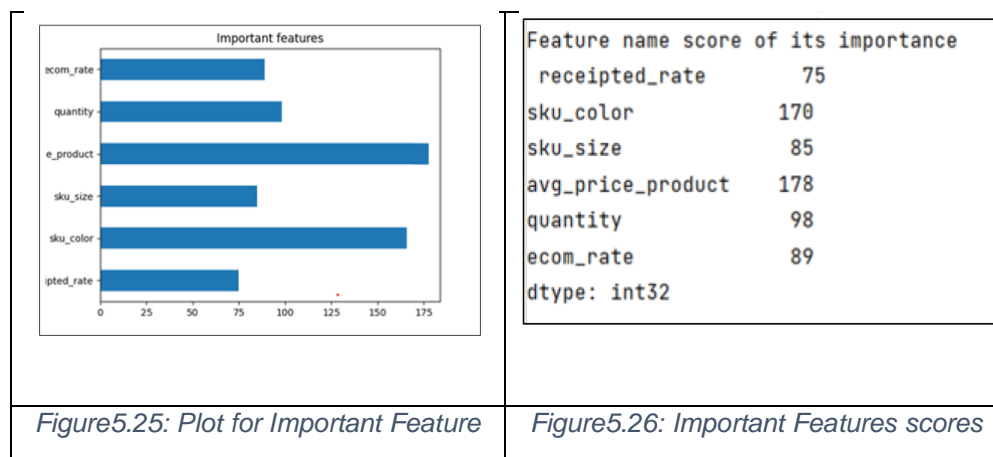
The output of the classification report further underscores the superior performance of LightGBM compared to the other two models based on all three scores in the report.

5.7 Analysis of Model

Across all metrics and evaluation techniques – CV score, accuracy score, AUC score, classification report LightGBM performs better than other two models. Therefore, LightGBM will be used as the basis of feature selection, i.e. the identification of important features that influence the return rate of a product and classification prediction. LightGBM with its additional property to handle null values, categorical features, missing values, and speed of processing becomes the most preferred algorithm to be used for prediction.

5.8 Feature Importance

The importance of features can be assessed with regard to different metrics. A standard approach is to measure a feature's importance by the number of times it appears in a split of the tree/forest. In LightGBM, this metric can be accessed using 'feature_importances_' attribute. It provides the number of times each feature was included in a split. Below figures show features and their importance score value and plot:



Out of all features, the average price of the product followed by its colour and quantity affect product's model's output the most. The importance of the colour feature may signify that the product image or quality doesn't match a customer's requirements.

5.9 SHAP Values and Plots

A more novel approach to assess feature importance are SHAP values. The purpose of SHAP values is to explain the contribution of each feature to the predicted outcome (Christophm, no date). There are different SHAP plots which can help to understand the importance of different features and their role in a model's prediction. Below are SHAP values and plots with LightGBM as input model:

5.9.1 SHAP Values:

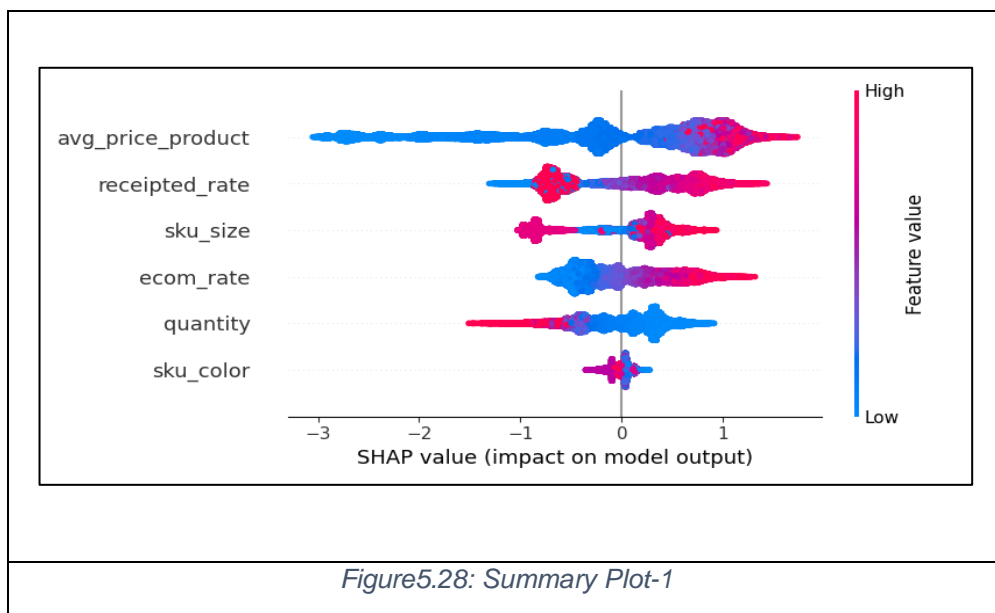
Below figure shows the SHAP value for each feature in each predicted outcome:

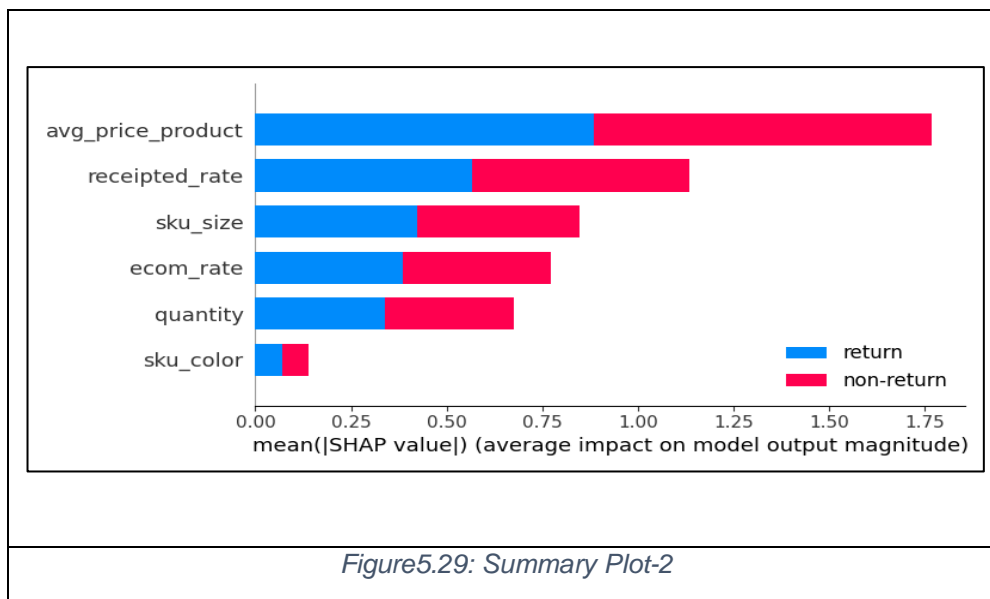
```
In [116]: print('Shap values are: %s' % shap_values)
Shap values are: [array([[ -0.42251554,  0.06180958, -0.2120815 , -0.93690342, -0.70048003,
    -0.28276418],
    [ 0.7443013 ,  0.22532282,  0.0341576 , -1.05517288, -0.40810221,
    -0.14276518],
    [ 0.66871997, -0.03939559, -0.44100538,  0.25874015, -0.03523935,
    0.20766879],
    ...,
    [-0.65063923, -0.02602513, -0.30113621,  1.39387454,  0.37119491,
    0.53581226],
    [ 0.20246519, -0.06290237,  0.11210866, -0.99249122, -0.53560983,
    -0.45698393],
    [-0.48669732, -0.04907095, -0.41404113, -0.37405368,  0.51358707,
    -0.04210272]]), array([[ 0.42251554, -0.06180958,  0.2120815 ,  0.93690342,  0.70048003,
    0.28276418],
    [-0.7443013 , -0.22532282, -0.0341576 ,  1.05517288,  0.40810221,
    0.14276518],
    [-0.66871997,  0.03939559,  0.44100538, -0.25874015,  0.03523935,
    -0.20766879],
    ...,
    [ 0.65063923,  0.02602513,  0.30113621, -1.39387454, -0.37119491,
    -0.53581226],
    [-0.20246519,  0.06290237, -0.11210866,  0.99249122,  0.53560983,
    0.45698393],
```

Figure5.27: SHAP Values

5.9.2 Summary plot:

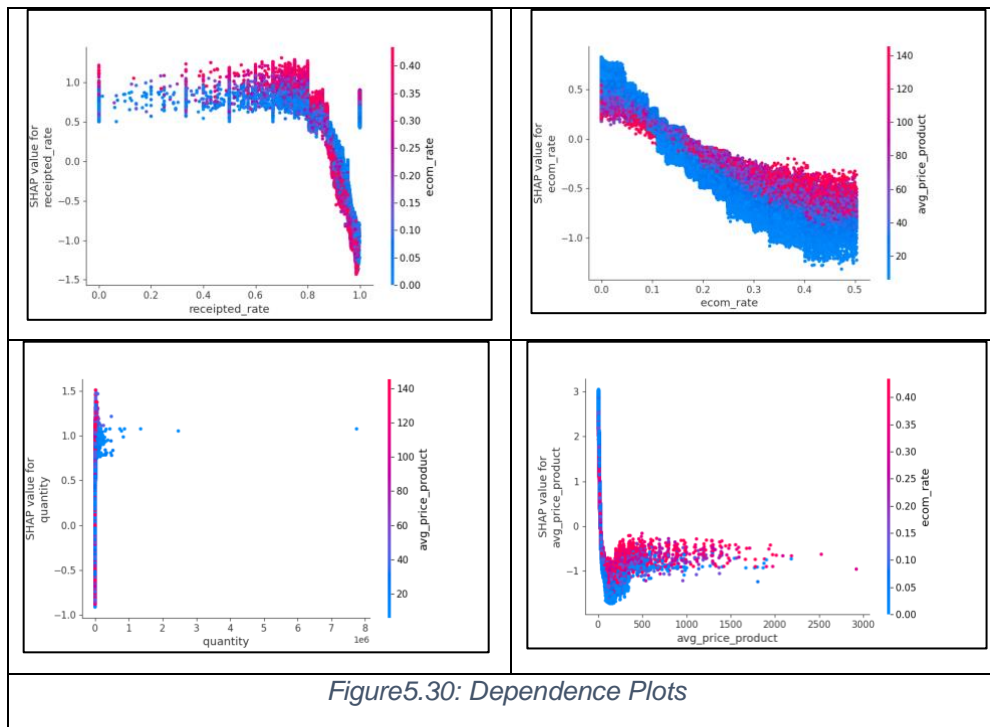
A summary plot provides a visual overview of the contribution of each feature to a model's prediction across all observations. Each point in the plot describes the SHAP value of feature for a given record. The y-axis indicates the feature and the x-axis indicates the SHAP value. 'avg_price_product' is the most important feature according to the summary plot, and plays a significant role for predicted outcomes, followed by the 'receipted_rate' and 'sku_size' features. The first image below shows SHAP values across individual instances and the second image shows mean SHAP values of each feature in order of importance for both class of prediction:





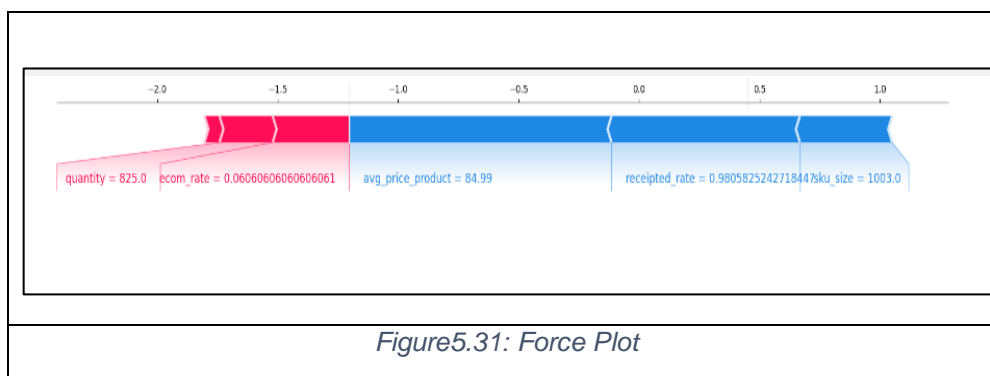
5.9.3 Dependence plot

Multiple features may jointly affect the performance of the model. The dependence plot is one way to assess such feature interactions. It plots feature values on the x-axis, SHAP values on the y-axis and interaction effects with another variable via the color dispersion in the plot. The plot helps to understand interaction effects between two features of the model. Below, we can see that there is a considerable interaction effect between 'receipted_rate' and 'ecom_rate', and between 'avg_price_product' and 'sku_color'.



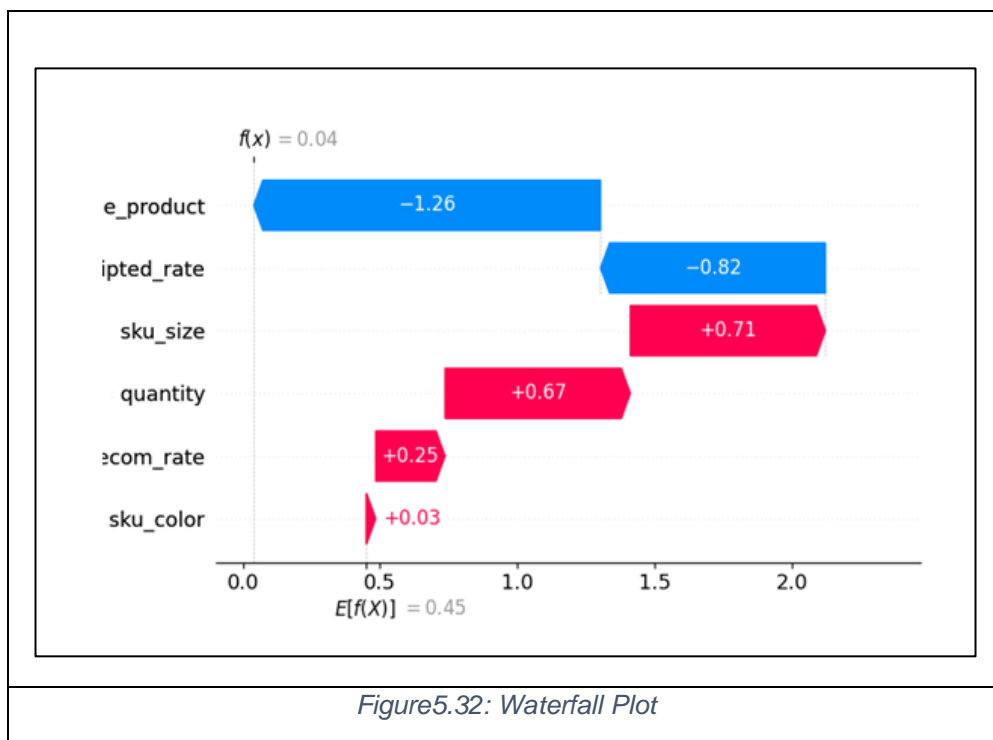
5.9.4 Force Plot

Force plot explains how each feature contributes towards pushing the model from baseline values to the predictions it makes. The extent of each feature's contribution towards the model prediction is shown in the plot. For the observation displayed below, the features 'quantity' and 'ecom_rate' significantly push the predicted likelihood of return up. Below is the force plot for the LightGBM model in this dissertation:



5.9.5 Waterfall Plot

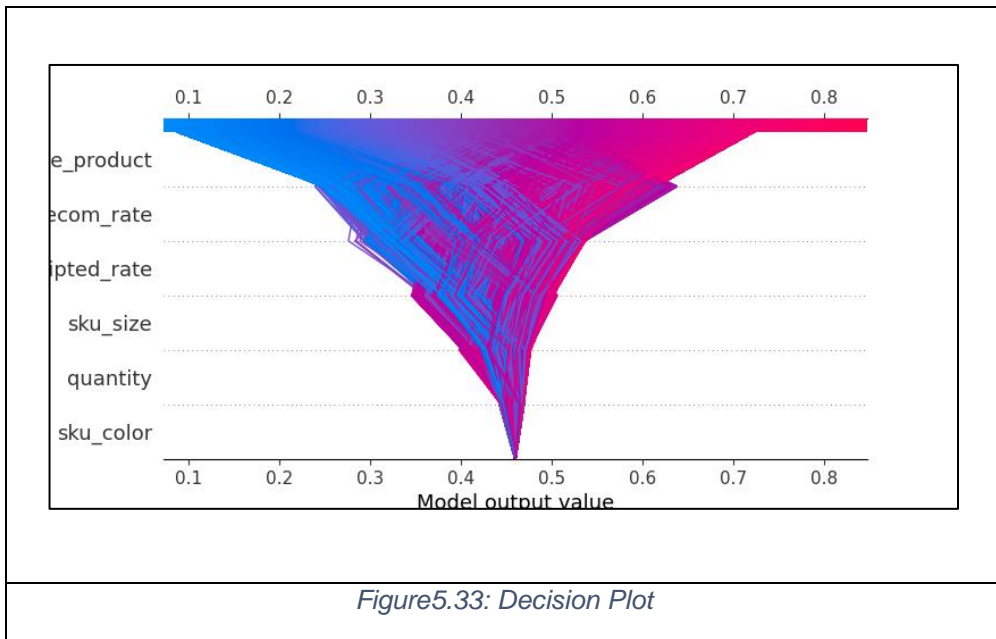
An alternative to a force plot is a waterfall plot. This plot shows how every feature helps model to move towards the predicted value from the expected baseline value. It considers a single record to make the plot which can be analysed in detail. For the record considered in below image, the predicted return likelihood is negatively impacted by 'avg_price_product' and positively impacted by 'sku_size'. Below is the waterfall plot for an instance in the data:



5.9.6 Decision Plot

The y-axis represents features and x-axis represents model's output, by default the features are ordered in descending order of importance in decision plot (SHAP, no data a). It shows how the predicted outcome changes across features. Feature 'avg_price_product' contributes most

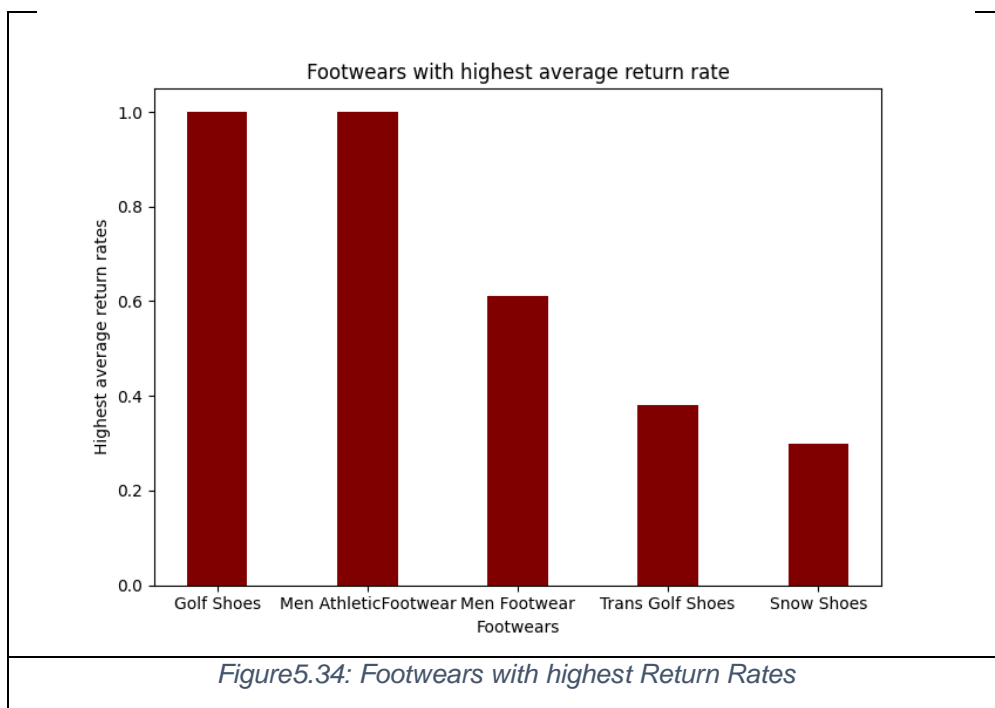
to the predicted outcome by model. In below plot, features are getting overlapped and therefore we are getting such plot. It is meant to show contribution of features in predicted outcome:



5.10 Insights Using Visualisation:

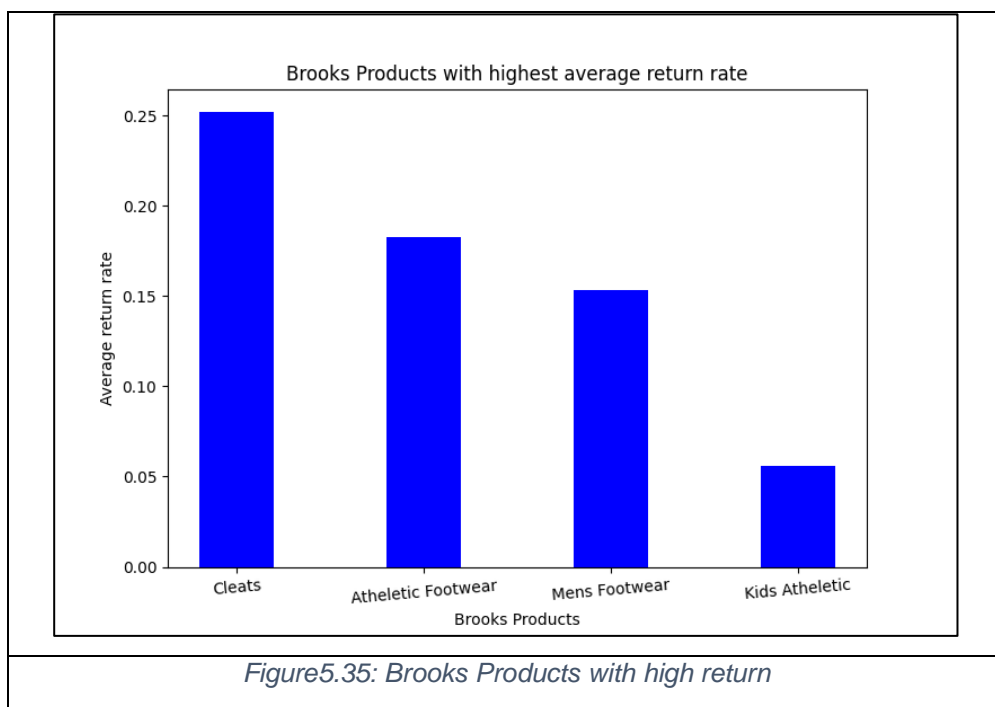
To find insights from data through visualization, 'matplotlib.pyplot' package in python is used. Processed data at product level used to create model is further used to develop insights.

1. Footwears: Below figure shows, top five products in footwears highest return rate. Highest return rate is recorded for athletic footwear followed by Golf shoes.

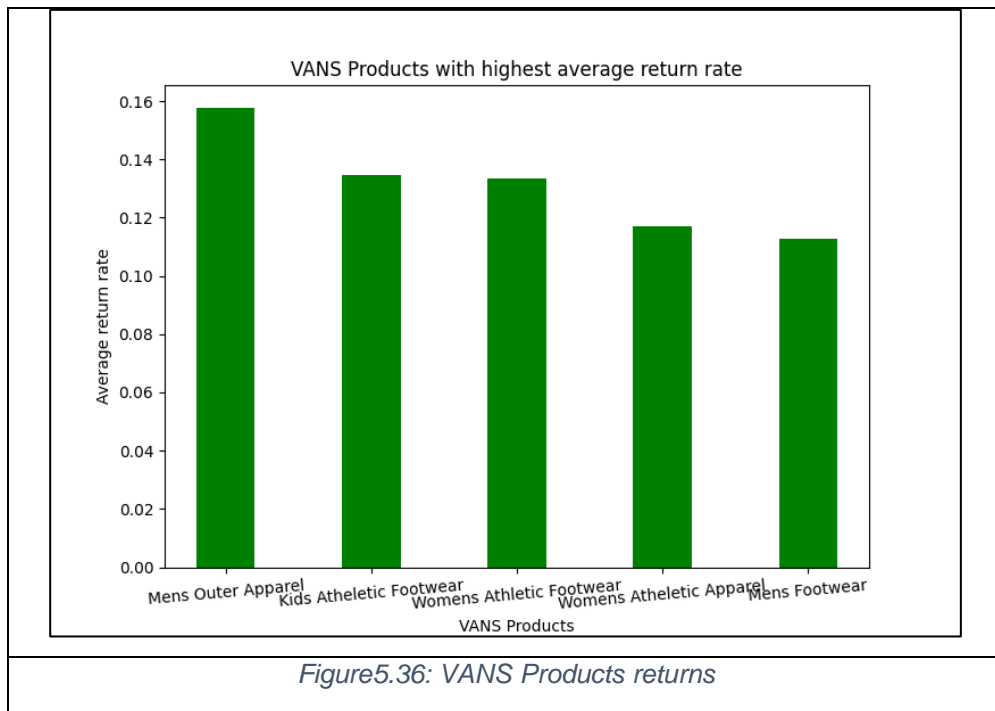


2. Brand Analysis:

- Two popular brands 'Brooks and 'Vans are compared based on return rate and product which are returned highest. The visualisation helps to deduce that highest return rate in Brooks is recorded 'Cleats' which is close to 0.25.

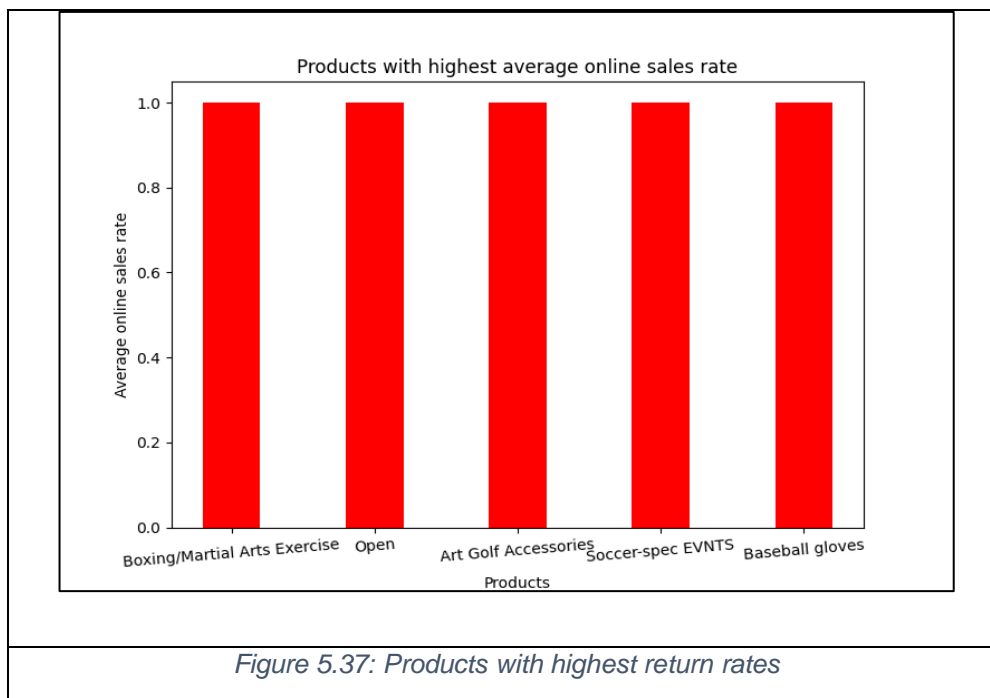


- In case of 'Vans, highest highest return rate is noted for 'Mens Outer Apparels' which is around 0.15. The highest average return rates recorded of Vans is comparatively less than Brooks

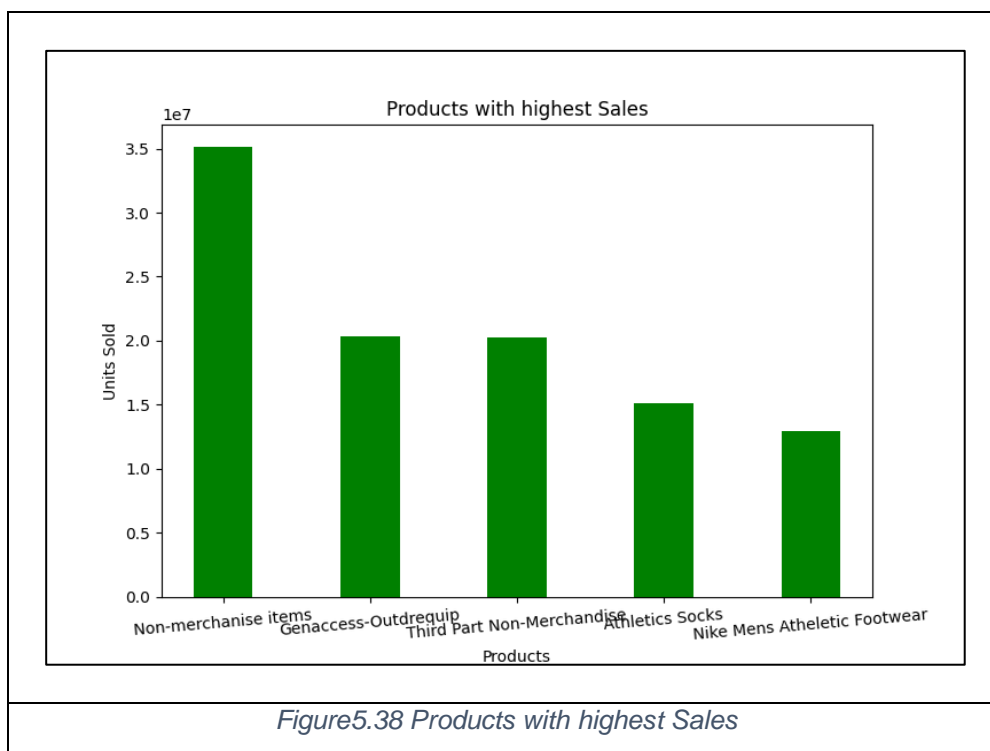


3. Product Category Analysis:

- Out of all the product categories, product with highest return rate is all the five mentioned in below figure. These products have recorded return every time they are ordered which needs to be investigated further based on sales, mode of sale, department, and other characteristics.



- The below figure depicts the product with highest sales in data, it is recorded that 'Non- merchandised' items are sold the most.



CHAPTER 6

CONCLUSION

6 Conclusion

Based on different verification and validation techniques like model score, ROC curve and score, cross-validation score, confusion matrix, classification report output and others it can be concluded that LightGBM is the most appropriate among three models to make classification based on the available data. Classification report provides complete summary of performance of LightGBM model and enables it to compare with other two models. Along with 'feature_importance_' function of LightGBM SHAP values and plots promoted better understanding of contribution of features. Among all features significant contribution towards predicted classification outcome is 'avg_price_product' i.e., average price of product.

Chapter 2 enables in learning different models and approaches of predictive analytics in retail, some algorithms are learned in-depth to fit the problem statement and enable to achieve dissertation objectives. Importance of different features on model's performance was discussed in multiple literatures. After literature review, there was great understanding of different models, retail analytics, identifying features, and approach towards implementation.

In Chapter 3, how to select a model based on problem statement and data available was discussed which is a great challenge in implementing machine learning algorithms. There are multiple types of algorithms, multiple ways to implement it, different requirements in data for algorithms which are important to understand before approaching implementation. Three models were identified considering problem statement and analysed further based on data available and feasibility

with the problem statement. The chapter also included overview and understanding of data which highlighted storage,

Chapter 4 explains complete implementation steps starting from data collection from data warehouse, data cleaning in PostgreSQL followed by pre-processing in Python, implementation of models- fitting, prediction, feature selection, etc. and verification of working of models involved model score, ROC score, confusion matrix, CV score, classification report. This chapter is body of dissertation involving complete execution of algorithms to accomplish project objectives.

In Chapter 5, results of implementation of all three models were critically verified and analysed as only one model had to be selected for making predictions. The most appropriate model is LightGBM considering all verification scores like model score (0.783), ROC accuracy score (0.779), F1 score (0.79 and 0.77), precision (0.81 and 0.74) and other scores and plots and feasibility towards project objectives. 'Feature Importances' identified using LightGBM were further analysed using SHAP values and plots which helped in better understanding of how model was able to make predictions and contribution of each feature in predictions. Average price of product was identified as most contributing feature towards the prediction/classification outcomes with importance score of 178 obtained using LightGBM and SHAP plots like summary plot and decision plot back the output of LightGBM.

6.1 Managerial Implications

Retail sector is on boom across regions along with high rate of acceptability of advanced analytics in retail. However, further investment in technology, data storage, human resources, and infrastructure is required. Technology has proven to help retailers increase their revenues, customer loyalty, and enhanced customer experience. Multiple services offered by retailers can be explored, analysed, and enhanced using sophisticated technologies which can play significant role in day to day and long-term business.

The intent of dissertation is to curb product returns which would reduce capital and other resources losses and environmental damage caused due to reverse logistics. The result of this dissertation enables to predict product returns based on characteristics of product with the probability of prediction. Similarly, an advanced module implemented by Apriss namely, 'Verify' predicts product return in real time, therefore, result of dissertation can be aligned with solutions like 'Verify' to enhance prediction results with more parameters (product characteristics) in prediction model. The result also enables to identify characteristics of products that cause returns; corrective actions or enhanced return policy regarding respective products can be implemented. The study highlights characteristics of product which are going unnoticed by retailers as cause of returns. Results can be used by retailers to forecast returns, identify causes of returns of respective products, fix the responsibility as returns can be because of quality (manufacturer or supply chain), price (retailer), or mode of purchase (online or offline). It would result in lesser returns, lower financial losses, and satisfied customers.

CHAPTER 7

DISCUSSION

7 Discussion

The model in dissertation is successfully able to make predictions and highlights importance of advanced analytics in retail. Same has been reflected in literatures in chapter 2 which mention high adoption of advanced analytics in retail but do also mention challenges associated with it (Roodekerk, DeHoratius and Musalem, 2022), (Karabulut, 2022). In a literature, author implements multiple Machine Learning (ML) Classification algorithm and compares them for better results (Sreekumar *et al.*, 2020), similar approach and process has been implemented in dissertation. With similar approach and models- LightGBM and RandomForest to predict product returns was implemented in a literature (Hofmann *et al.*, 2020). These literatures helped to shape the analysis workflow and implementation in dissertation. However, using advanced ML algorithms probability of outcomes were also identified in dissertation which has not been implemented in any of the literatures mentioned in chapter 2.

The working of LightGBM algorithm in classifying returns has been mentioned by authors in their literature (Dzyabura, Kihal and Ibragimov, 2018). Confusion matrix helped to evaluate performance and understanding of correct and incorrect outcomes of all three models in dissertation as mentioned by Kulkarni and others in a literature (Kulkarni, Batarseh and Demir, 2020). Classification report in dissertation made it easy to compare models with relevant scores in the report, report is created based on values in confusion matrix and provide great insights. (Orozco-Arias *et al.*, 2020). ROC curve plotted for all three models are easily interpretable and reliable plot to assess and compare performance of classification models as highlighted in a literature (Fawcett, 2006). SHAP values and plot in dissertation enabled

to understand contribution of each feature towards predicted outcome, plots showed average price as most important feature in LightGBM model. SHAP plots are one of the best methods to detect important features in a classification model (Marcilio and Eler, 2020).

7.1 Limitation and Future Research

- Other classification algorithms like kNN, Naïve Bayes, etc. can also be implemented to increase scope of analysis.
- Although method to set predictor for specific category of products were created but analysis was not performed to specific product categories like footwear, sports goods, etc.; all product categories were included in analysis.
- Data of more retailers can be combined and analysed so as to broaden analysis for more product categories. Further, products can be segregated sector-wise and analysed.
- As analysis was carried out in Appriss's virtual environment, visualisation tools cannot be installed and it restricted to perform in-depth analysis and identify patterns using visualisations and create advanced visualisations in dissertation.

As there are many classification algorithms which can be implemented, considering time constraint, three algorithms were implemented. Identifying product category was difficult as data was not stored in appropriate format to identify categories. Data from multiple retailers could not be combined as data in warehouse did not mention its description appropriately for all records, so to combine data it was necessary to understand the product to conduct further analysis.

Future research can possibly overcome the limitations and perform analysis with wide range of algorithms and specific product categories. More classification algorithms can be learned and implemented for better comparative study of model's performance. As in dissertation colour and size of products were identified, similarly once data from multiple retailers can be combined and categories can be identified by further cleaning and processing data.

CHAPTER 8

REFERENCES

8 References

Ader, J. et al. (2021) *Returning to order: Improving returns management for apparel companies*. Available at:
<https://www.mckinsey.com/industries/retail/our-insights/returning-to-order-improving-returns-management-for-apparel-companies>.

Agrawal, S. and Singh, R.K. (2019) 'Forecasting product returns and reverse logistics performance: structural equation modelling', *Management of Environmental Quality: An International Journal*, 31(5), pp. 1223–1237. Available at: <https://doi.org/10.1108/MEQ-05-2019-0109>.

Allwright, S. (2022) 'Using cross_val_score in sklearn, simply explained', 22 June. Available at: https://stephenallwright.com/cross_val_score-sklearn/#:~:text=Cross_val_score%20is%20a%20method%20which,metric%20value%20for%20the%20dataset. (Accessed: 4 September 2022).

Appriss Retail (no date a) *Appriss Engage Omnichannel Optimization*. Available at: <https://apprissretail.com/solutions/engage/>.

Appriss Retail (no date b) *Incent Targeted Incentives*. Available at: <https://apprissretail.com/solutions/incent/>.

Appriss Retail (no date c) 'Optimize Omnichannel Returns', *Verify Return Authorization*. Available at: <https://apprissretail.com/solutions/verify/>.

Appriss Retail (no date d) 'Secure', *Cut Shrink and Protect Margin*. Available at: <https://apprissretail.com/solutions/secure-store/>.

Backegren, F., Lonnstrom, E. and Zetterberg, M. (2020) 'CUSTOMERS' RETURN REASONS AND PREFERENCES ABOUT PRODUCT-ORIENTED TOOLS'. The Swedish School of Textiles, University of Borås. Available at: <https://www.diva-portal.org/smash/get/diva2:1476986/FULLTEXT01.pdf> (Accessed: 22 August 2022).

Bala, P.C. (2022) *How to Detect Outliers in Machine Learning – 4 Methods for Outlier Detection*. Available at: <https://www.freecodecamp.org/news/how-to-detect-outliers-in-machine-learning/#:~:text=Outliers%20are%20those%20data%20points,data%20entry%2C%20or%20erroneous%20observations>.

Banerjee, P. (2020a) 'LightGBM Classifier in Python'. Available at: <https://www.kaggle.com/code/prashant111/lightgbm-classifier-in-python/notebook>.

Banerjee, P. (2020b) *Random Forest Classifier Tutorial*, kaggle. Available at: <https://www.kaggle.com/code/prashant111/random-forest-classifier-tutorial> (Accessed: 20 August 2022).

Bauer, F., Downs, D. and Speights, D. (2021a) 'Analytics That Can Help You Save on Ecommerce Returns', in. Appriss Retail Limited. Available at: <https://info.apprissretail.com/form-engage-expert-content-4025>.

Bauer, F., Downs, D. and Speights, D. (2021b) 'Retailers Can Save Millions Redirecting Consumers to Return In-Store: A Machine Learning Approach', in. Appriss Retail Limited. Available at: <https://info.apprissretail.com/form-engage-expert-content-4025>.

Berke, A. and Colakoglu, N. (2019) 'Comparison of Multi-class Classification Algorithms on Early Diagnosis of Heart Diseases', in *SBIS Young Business and Industrial Statisticians Workshop on Recent Advances in Data Science and Business Analytics. y-BIS 2019 Conference*, Istanbul, Turkey. Available at: https://www.researchgate.net/publication/338950098_Comparison_of_Multi-class_Classification_Algorithms_on_Early_Diagnosis_of_Heart_Diseases (Accessed: 25 August 2022).

Berrar, D. (2018) 'Cross-Validation', in. Tokyo, Japan, pp. 542–545. Available at: <https://doi.org/DOI:10.1016/B978-0-12-809633-8.20349-X>.

Bhandari, A. (2022) *Everything you Should Know about Confusion Matrix for Machine Learning*, Analytics Vidhya. Available at: <https://www.analyticsvidhya.com/blog/2020/04/confusion-matrix-machine-learning/>.

Bonaros, B. (2021) *The Ultimate Guide Of Feature Importance In Python, Predictive ['hacks']*. Available at: <https://predictivehacks.com/feature-importance-in-python/#:~:text=Feature%20Importance%20is%20a%20score,useful%20insights%20about%20our%20data>.

de Brito, M.P. and van der Laan, E.A. (2009) 'Inventory control with product returns: The impact of imperfect information', *European Journal of Operational Research*, 194(1), pp. 85–101. Available at: <https://doi.org/10.1016/j.ejor.2007.11.063>.

Brownlee, J. (2020) *How to Make Predictions with scikit-learn, Machine Learning Mastery*. Available at: <https://machinelearningmastery.com/make-predictions-scikit-learn/>.

Canda, A., Yuan, X.-M. and Wang, F.-Y. (2015) 'Modeling and forecasting product returns: An industry case study', in *2015 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*. *2015 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, Singapore, Singapore: IEEE, pp. 871–875. Available at: <https://doi.org/10.1109/IEEM.2015.7385772>.

Cohen, I. (2021) 'Explainable AI (XAI) with SHAP -Multi-Class Classification Problem', *Towards Data Science*, 12 July. Available at: <https://towardsdatascience.com/explainable-ai-xai-with-shap-multi-class-classification-problem-64dd30f97cea> (Accessed: 30 August 2022).

Courtney Reagan (2016) 'A \$260 billion "ticking time bomb": The costly business of retail returns', *CNBC*, 16 December. Available at: <https://www.cnn.com/2016/12/16/a-260-billion-ticking-time-bomb-the-costly-business-of-retail-returns.html>.

Cui, H., Rajagopalan, S. and Ward, A.R. (2020) 'Predicting product return volume using machine learning methods', *European Journal of Operational Research*, 281(3), pp. 612–627. Available at: <https://doi.org/10.1016/j.ejor.2019.05.046>.

Danb and Cook, A. (no date) 'SHAP Values', *kaggle*. Available at: <https://www.kaggle.com/code/dansbecker/shap-values/tutorial> (Accessed: 30 August 2022).

datacamp (no date) *Understanding Random Forests Classifiers in Python Tutorial*. Available at: <https://www.datacamp.com/tutorial/random-forests-classifier-python> (Accessed: 2 September 2022).

dataclarity (2021) *ONLINE VS OFFLINE: HOW RETAILERS CAN BRIDGE THE GAP AND DELIVER THE BEST EXPERIENCES TO SHOPPERS*. dataclarity. Available at: <https://www.dataclarity.uk.com/2021/02/23/online-vs-offline-how-retailers-can-bridge-the-gap-and-deliver-the-best-experiences-to-shoppers/>.

Developers Google (no date) *Classification: Accuracy, Machine Learning*. Available at: <https://developers.google.com/machine-learning/crash-course/classification/accuracy> (Accessed: 1 September 2022).

- Dopson, E. (2021) *The Plague of Ecommerce Return Rates and How to Maintain Profitability, Shopifyplus*. Available at: <https://www.shopify.co.uk/enterprise/ecommerce-returns> (Accessed: 3 September 2022).
- Dzyabura, D., El Kihal, S. and Ibragimov, M. (2018) 'Leveraging the Power of Images in Predicting Product Return Rates', in. Available at: https://pages.stern.nyu.edu/~ddzyabur/index_files/DzyaburaElKihalIbragimov.pdf (Accessed: 20 August 2022).
- Dzyabura, D., Kihal, S.E. and Ibragimov, M. (2018) *Leveraging the Power of Images in Predicting Product Return Rates*. 18–135. Marketing Science Institute, pp. 1, 19, 20. Available at: https://www.msi.org/wp-content/uploads/2020/06/MSI_Report_18-135-1.pdf (Accessed: 9 July 2022).
- E R, S. (2021) 'Understanding Random Forest', *Analytics Vidhya*, 17 June. Available at: <https://www.analyticsvidhya.com/blog/2021/06/understanding-random-forest/> (Accessed: 2 September 2022).
- Ebner, J. (2022) *A Quick Introduction to the Sklearn Fit Method, Sharp Sight*. Available at: <https://www.sharpsightlabs.com/blog/sklearn-fit/>.
- Eroglu, E. (2019) 'Using Machine Learning Algorithms For Forecasting Rate of Return Product In Reverse Logistics Process', *Alphanumeric Journal*, pp. 143–156. Available at: <https://doi.org/10.17093/alphanumeric.541307>.
- Fawcett, T. (2006) 'An introduction to ROC analysis', *Pattern Recognition Letters*, 27(8), pp. 861–874. Available at: <https://doi.org/10.1016/j.patrec.2005.10.010>.
- Feng, C. *et al.* (2014) 'Log-transformation and its implications for data analysis', in, pp. 105–109. Available at: <https://doi.org/10.3969/j.issn.1002-0829.2014.02.009>.
- Frei, R. and Jack, L. (2021) 'Future of high streets: how to prevent our city centres from turning into ghost towns', 29 January. Available at: <https://theconversation.com/future-of-high-streets-how-to-prevent-our-city-centres-from-turning-into-ghost-towns-154108> (Accessed: 7 July 2022).
- Frei, R., Jack, L. and Brown, S. (2020) 'Product returns: a growing problem for business, society and environment', *International Journal of Operations*

& *Production Management*, 40(10), pp. 1613–1621. Available at:
<https://doi.org/10.1108/IJOPM-02-2020-0083>.

Garg, S. (2022) 'How to Deal with Categorical Data for Machine Learning', *KDnuggets*, 4 August. Available at:
<https://www.kdnuggets.com/2021/05/deal-with-categorical-data-machine-learning.html> (Accessed: 30 August 2022).

Georgina, H. (2021) *Retail sector in the UK*. SN06186. House of Commons Library, p. 3. Available at:
<https://researchbriefings.files.parliament.uk/documents/SN06186/SN06186.pdf>.

Gillis, A.S. (no date) 'data splitting'. Available at:
<https://www.techtarget.com/searchenterpriseai/definition/data-splitting>.

Google Cloud (no date) *What is a data warehouse?*, *Google Cloud*. Available at: <https://cloud.google.com/learn/what-is-a-data-warehouse#:~:text=A%20data%20warehouse%20is%20an,analysis%20as%20well%20custom%20reporting>. (Accessed: 15 August 2022).

Goyal, C. (2021) *Importance of Cross Validation: Are Evaluation Metrics enough?*, *Analytics Vidhya*. Available at:
<https://www.analyticsvidhya.com/blog/2021/05/importance-of-cross-validation-are-evaluation-metrics-enough/>.

Grover, K. (no date) *Advantages and Disadvantages of Logistic Regression*, *OpenGenus*. Available at: <https://iq.opengenus.org/advantages-and-disadvantages-of-logistic-regression/> (Accessed: 25 August 2022).

Hayes, A. (2022) *Z-Score*, *Investopedia*. Available at:
<https://www.investopedia.com/terms/z/zscore.asp>.

Heilig, L. *et al.* (2016) 'DATA-DRIVEN PRODUCT RETURNS PREDICTION: A CLOUD-BASED ENSEMBLE SELECTION APPROACH', *Association for Information Systems AIS Electronic Library (AISeL)* [Preprint]. Available at:
<https://doi.org/10.1016/j.jclepro.2017.06.242>.

Hofmann, A. *et al.* (2020) 'An Industry-Agnostic Approach for the Prediction of Return Shipments', in *Prediction of Return Shipments. Americas Conference on Information Systems*. Available at:
https://www.researchgate.net/profile/Fabian-Gwinner/publication/342820571_An_Industry-

Agnostic_Approach_for_the_Prediction_of_Return_Shipments/links/612d0a1438818c2eaf700ca0/An-Industry-Agnostic-Approach-for-the-Prediction-of-Return-Shipments.pdf (Accessed: 30 August 2022).

i2 tutorial (2019) 'What are the advantages and Disadvantages of Logistic Regression?', *i2 tutorials*, 14 November. Available at: <https://www.i2tutorials.com/what-are-the-advantages-and-disadvantages-of-logistic-regression/> (Accessed: 25 August 2022).

IBM (2022) *IBM Cloud Pak for Business Automation*, *IBM Documentation*. Available at: <https://www.ibm.com/docs/en/cloud-paks/cp-biz-automation/21.0.x?topic=project-understanding-model-accuracy>.

IBM (no date) *What is logistic regression?* Available at: <https://www.ibm.com/uk-en/topics/logistic-regression#:~:text=Logistic%20regression%20estimates%20the%20probability,bounded%20between%200%20and%201>. (Accessed: 26 August 2022).

IBM Cloud Education (2020a) *Exploratory Data Analysis*, *IBM*. Available at: <https://www.ibm.com/uk-en/cloud/learn/exploratory-data-analysis> (Accessed: 15 August 2022).

IBM Cloud Education (2020b) *Random Forest*. Available at: <https://www.ibm.com/cloud/learn/random-forest> (Accessed: 2 September 2022).

IBM Cloud Education (2021) *Overfitting*. Available at: <https://www.ibm.com/cloud/learn/overfitting>.

Isitapol (2022) 'How to split a Dataset into Train and Test Sets using Python', 25 May. Available at: <https://www.geeksforgeeks.org/how-to-split-a-dataset-into-train-and-test-sets-using-python/#:~:text=The%20train%20test%20split%20is,model%20results%20to%20machine%20results>.

Janakiraman, N., Syrdal, H.A. and Freling, R. (2016) 'The Effect of Return Policy Leniency on Consumer Purchase and Return Decisions: A Meta-analytic Review', *Journal of Retailing*, 92(2), pp. 226–235. Available at: <https://doi.org/10.1016/j.jretai.2015.11.002>.

Java Point (no date) *Accuracy_Score in Sklearn, java T point*. Available at: https://www.javatpoint.com/accuracy_score-in-sklearn.

- Jeon, H.-K. *et al.* (2020) 'Sea Fog Identification from GOCI Images Using CNN Transfer Learning Models', in. Available at: <https://doi.org/10.3390/electronics9020311>.
- Kanstren, T. (2020) 'A Look at Precision, Recall, and F1-Score', *Towards Data Science*, 11 September. Available at: <https://towardsdatascience.com/a-look-at-precision-recall-and-f1-score-36b5fd0dd3ec> (Accessed: 3 September 2022).
- Karabulut, O. (2022) 'Returns Forecasting: A Must-Have for Retailers', 12 May. Available at: <https://www.inventanalytics.com/blog/product-returns-forecasting-a-must-have-for-retailers/> (Accessed: 8 July 2022).
- Kasturi, S.N. (2019) 'XGBOOST vs LightGBM: Which algorithm wins the race !!!', 11 July. Available at: <https://towardsdatascience.com/lightgbm-vs-xgboost-which-algorithm-win-the-race-1ff7dd4917d>.
- Khandelwal, E. (2017) *Which algorithm takes the crown: Light GBM vs XGBOOST?* Available at: <https://www.analyticsvidhya.com/blog/2017/06/which-algorithm-takes-the-crown-light-gbm-vs-xgboost/>.
- Khusainova Gulnaz (2019) 'There Is No Such Thing As A Free Return', *Forbes*, 28 May. Available at: <https://www.forbes.com/sites/gulnazkhusainova/2019/03/28/there-is-no-such-thing-as-a-free-return/?sh=2aa7caf27135>.
- Kranz, J., Urbanke, P. and Kolbe, L. (2015) 'Predicting Product Returns in E-Commerce: The Contribution of Mahalanobis Feature Extraction', in. *Proceedings of the 36th International Conference on Information Systems (ICIS)*. Available at: https://www.researchgate.net/publication/283270887_Predicting_Product_Returns_in_E-Commerce_The_Contribution_of_Mahalanobis_Feature_Extraction (Accessed: 7 July 2022).
- Krapp, M., Nebel, J. and Sahamie, R. (2013) 'Forecasting product returns in closed-loop supply chains', *International Journal of Physical Distribution & Logistics Management*, 43(8), pp. 614–637. Available at: <https://doi.org/10.1108/IJPDLM-03-2012-0078>.
- Kulkarni, A., Batarseh, F.A. and Demir, F. (2020) *Data Democracy: At the Nexus of Artificial Intelligence, Software Development, and Knowledge*

Engineering provides a manifesto to data democracy. Available at: <https://www.sciencedirect.com/topics/engineering/confusion-matrix>.

Lee, I. and Shin, Y.J. (2020) 'Machine learning for enterprises: Applications, algorithm selection, and challenges', 63(2), pp. 157–170. Available at: <https://doi.org/10.1016/j.bushor.2019.10.005>.

Li, J., He, J. and Zhu, Y. (2018) 'E-tail Product Return Prediction via Hypergraph-based Local Graph Cut', in. *Applied Data Science Track Paper*, London, UK: KDD. Available at: <https://doi.org/doi/pdf/10.1145/3219819.3219829>.

Liang, W. *et al.* (2020) 'Predicting Hard Rock Pillar Stability Using GBDT, XGBoost, and LightGBM Algorithms', *Mathematics*, 8(5), p. 765. Available at: <https://doi.org/10.3390/math8050765>.

Long, A. (2018) 'Understanding Data Science Classification Metrics in Scikit-Learn in Python', *Towards DataScience*, 5 August. Available at: <https://towardsdatascience.com/understanding-data-science-classification-metrics-in-scikit-learn-in-python-3bc336865019>.

Mage (2021) 'How to interpret machine learning models with SHAP values', *Dev*, 23 November. Available at: https://dev.to/mage_ai/how-to-interpret-machine-learning-models-with-shap-values-54jf (Accessed: 30 August 2022).

Maini, E. (2020) *Interquartile Range to Detect Outliers in Data*. Available at: <https://www.geeksforgeeks.org/interquartile-range-to-detect-outliers-in-data/>.

Marcilio, W.E. and Eler, D.M. (2020) 'From explanations to feature selection: assessing SHAP values as feature selection mechanism', in *2020 33rd SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*. *2020 33rd SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, Recife/Porto de Galinhas, Brazil: IEEE, pp. 340–347. Available at: <https://doi.org/10.1109/SIBGRAPI51738.2020.00053>.

Microsoft (no date) 'Create a box and whisker chart', *Microsoft Support*. Available at: <https://support.microsoft.com/en-us/office/create-a-box-and-whisker-chart-62f4219f-db4b-4754-aca8-4743f6190f0d> (Accessed: 30 August 2022).

Microsoft Dynamic 365 (2022) *Sales Returns*, *Microsoft*. Available at: <https://docs.microsoft.com/en-us/dynamics365/supply-chain/warehousing/sales-returns> (Accessed: 2 September 2022).

Mingzhu, T. and Ding, S.X. (2020) 'An Improved LightGBM Algorithm for Online Fault Detection of Wind Turbine Gearboxes', in *Energies*. Available at: <https://doi.org/10.3390/en13040807>.

Morgan Stanley (2022) *Here's Why E-commerce Growth Can Stay Stronger for Longer*, Morgan Stanley. Available at: <https://www.morganstanley.com/ideas/global-ecommerce-growth-forecast-2022> (Accessed: 3 September 2022).

Muschelli, J. (2019) 'ROC and AUC with a Binary Predictor: a Potentially Misleading Metric', in. *J Classif*. Available at: <https://doi.org/10.1007/s00357-019-09345-1>.

Myriantous, G. (2021) 'What Is The Difference Between predict() and predict_proba() in scikit-learn?', 16 September. Available at: [https://towardsdatascience.com/predict-vs-predict-proba-scikit-learn-bdc45daa5972#:~:text=The%20predict_proba\(\)%20method&text=each%20data%20point.-,The%20method%20accepts%20a%20single%20argument%20that%20corresponds%20to%20the,for%20the%20input%20data%20points](https://towardsdatascience.com/predict-vs-predict-proba-scikit-learn-bdc45daa5972#:~:text=The%20predict_proba()%20method&text=each%20data%20point.-,The%20method%20accepts%20a%20single%20argument%20that%20corresponds%20to%20the,for%20the%20input%20data%20points).

Naeem, A. (no date) 'What is sigmoid and its role in logistic regression?', *educative*. Available at: <https://www.educative.io/answers/what-is-sigmoid-and-its-role-in-logistic-regression> (Accessed: 26 August 2022).

Orozco-Arias, S. *et al.* (2020) 'Measuring Performance Metrics of Machine Learning Algorithms for Detecting and Classifying Transposable Elements', *Processes*, 8(6), p. 638. Available at: <https://doi.org/10.3390/pr8060638>.

pandas (no date) *pandas.DataFrame.describe*. Available at: <https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.describe.html>.

Paul, B. (2015) 'Headaches mount when consumers return products: With the rise of global trade pacts and e-commerce, dealing with returns, recalls and defective products has become far more complex', *The Globe and Mail*, 10 November. Available at: <https://www.proquest.com/docview/1731947662/677A2C76A96F49AFPQ/24?accountid=13963> (Accessed: 28 August 2022).

Popkin, B. (no date) 'Offering Free Returns Can Boost Online Purchases 357%', *Royal Mail*. Available at:

<https://www.royalmail.com/business/guides-and-insights/can-free-returns-drive-sales>.

Powers, T.L. and Jack, E.P. (2015) 'Understanding the causes of retail product returns', *International Journal of Retail & Distribution Management*, 43(12), pp. 1182–1202. Available at: <https://doi.org/10.1108/IJRDM-02-2014-0023>.

Radecic, D. (2021) 'Master Machine Learning: Logistic Regression From Scratch With Python', *Python-Bloggers*, 11 March. Available at: <https://python-bloggers.com/2021/03/master-machine-learning-logistic-regression-from-scratch-with-python/>.

Regina Frei and Lisa Jack (no date) 'Product returns: a growing problem for business, society and environment'.

Rooderkerk, R.P., DeHoratius, N. and Musalem, A. (2022) 'The past, present, and future of retail analytics: Insights from a survey of academic research and interviews with practitioners'. Available at: <https://doi.org/10.1111/poms.13811>.

Saarela, M. and Jauhiainen, S. (2021) *Comparison of feature importance measures as explanations for classification models*. 272 (2021). University of Jyväskylä. Available at: <https://doi.org/10.1007/s42452-021-04148-9>.

Saha, S. (2022) 'XGBoost vs LightGBM: How Are They Different', 22 July. Available at: <https://neptune.ai/blog/xgboost-vs-lightgbm>.

Sailasya, G. and Kumar, G.L.A. (2021) 'Analyzing the Performance of Stroke Prediction using ML Classification Algorithms', *International Journal of Advanced Computer Science and Applications*, 12(6). Available at: <https://pdfs.semanticscholar.org/df5c/7d1bd7a59009dc51b9db903aa7f144241879.pdf>.

Saini, P. (2022) *Compute Classification Report and Confusion Matrix in Python*, *GeeksforGeeks*. Available at: <https://www.geeksforgeeks.org/compute-classification-report-and-confusion-matrix-in-python/>.

Saleh Khalid (May 16) 'E-commerce Product Return Rate – Statistics and Trends [Infographic]', *Ecommerce*. Available at: <https://www.invespcro.com/blog/ecommerce-product-return-rate-statistics/>.

Samorani, M. and Messinger, P.R. (2016) *The Analytics of Product Return Episodes in Retailing*. Available at: <https://www.researchgate.net/requests/r105019424> (Accessed: 27 August 2022).

SAP (no date) *What is predictive analytics?*, *SAP Insights*. Available at: <https://www.sap.com/uk/insights/what-is-predictive-analytics.html> (Accessed: 16 August 2022).

SAS (no date) *Machine Learning What it is and why it matters*, *SAS Insights*. Available at: https://www.sas.com/en_gb/insights/analytics/machine-learning.html (Accessed: 16 August 2022).

scikit learn (no date) *Imputation of missing values*, *Scikit-learn*. Available at: <https://scikit-learn.org/stable/modules/impute.html#:~:text=The%20SimpleImputer%20class%20provides%20basic,for%20different%20missing%20values%20encodings.>

Scikit Learn (no date) *Metrics and scoring: quantifying the quality of predictions*, *Scikit Learn*.

Serengil, S. Ilkin (2018) 'A Gentle Introduction to LightGBM for Applied Machine Learning', 13 October. Available at: <https://sefiks.com/2018/10/13/a-gentle-introduction-to-lightgbm-for-applied-machine-learning/>.

Shang, G. *et al.* (2017) 'How much do online consumers really value free product returns? Evidence from eBay', *Journal of Operations Management*, 53–56(1), pp. 45–62. Available at: <https://doi.org/10.1016/j.jom.2017.07.001>.

Shankar, V. (2019) 'Big Data and Analytics in Retailing', *Sciendoo*, 11(1). Available at: <https://doi.org/10.2478/nimmir-2019-0006>.

SHAP (no date a) 'Decision Plot', *SHAP*. Available at: https://shap.readthedocs.io/en/latest/example_notebooks/api_examples/plots/decision_plot.html#:~:text=Decision%20plots%20support%20SHAP%20interaction,for%20one%20or%20more%20observations. (Accessed: 31 August 2022).

SHAP (no date b) 'shap.dependence_plotSHAP', *SHAP*. Available at: <https://shap->

lrjball.readthedocs.io/en/latest/generated/shap.dependence_plot.html
(Accessed: 30 August 2022).

SHAP (no date c) *Welcome to the SHAP documentation*, SHAP. Available at:
<https://shap.readthedocs.io/en/latest/index.html> (Accessed: 30 August 2022).

Shulga, D. (2018) *5 Reasons why you should use Cross-Validation in your Data Science Projects, Towards Data Science*. Available at:
<https://towardsdatascience.com/5-reasons-why-you-should-use-cross-validation-in-your-data-science-project-8163311a1e79>.

Sides, R. and Lupine, S. (2022) *2022 retail industry outlook*. Deloitte, p. 11.
Available at:
<https://www2.deloitte.com/content/dam/Deloitte/us/Documents/consumer-business/2022-retail-industry-outlook.pdf>.

Sreekumar *et al.* (2020) 'Customer Classification in Indian Retail SectorA Comparative Analysis of Various Machine Learning Approaches', 26(1), pp. 1–28. Available at: <https://doi.org/10.46970/2020.26.1.1>.

Sun, M. *et al.* (2021) 'The impact of online reviews in the presence of customer returns', *International Journal of Production Economics*, 232, p. 107929. Available at: <https://doi.org/10.1016/j.ijpe.2020.107929>.

Surana, S. (no date) *What is Light GBM? Advantages & Disadvantages? Light GBM vs XGBoost?*, *kaggle*. Available at:
<https://www.kaggle.com/general/264327>.

Suresh, A. (2020) 'How to Remove Outliers for Machine Learning?', *Analytics Vidhya*, 30 November. Available at: <https://medium.com/analytics-vidhya/how-to-remove-outliers-for-machine-learning-24620c4657e8>
(Accessed: 30 August 2022).

Swaminathan, S. (2018) *Logistic Regression — Detailed Overview, Towards Data Science*. Available at: <https://towardsdatascience.com/logistic-regression-detailed-overview-46c4da4303bc> (Accessed: 26 August 2022).

Tableau (no date) *Guide To Data Cleaning: Definition, Benefits, Components, And How To Clean Your Data, Tableau*. Available at:
<https://www.tableau.com/learn/articles/what-is-data-cleaning#definition>
(Accessed: 15 August 2022).

- Telma, T. (no date) 'The Power of Retail: Delivering Development Impact in Emerging Markets'. Available at: https://www.ifc.org/wps/wcm/connect/news_ext_content/ifc_external_corporate_site/news+and+events/news/the+power+of+retail-eca.
- Tian, X. and Sarkis, J. (2021) 'Emission burden concerns for online shopping returns', *Nature*, 12. Available at: <https://doi.org/10.1038/s41558-021-01246-9>.
- Vadapalli, P. (2021) *Random Forest Classifier: Overview, How Does it Work, Pros & Cons*, *upGrad*. Available at: <https://www.upgrad.com/blog/random-forest-classifier/> (Accessed: 20 August 2022).
- W3schools (no date) *Pandas DataFrame info() Method*. Available at: [https://www.w3schools.com/python/pandas/ref_df_info.asp#:~:text=The%20info\(\)%20method%20prints,\(non%2Dnull%20values\)](https://www.w3schools.com/python/pandas/ref_df_info.asp#:~:text=The%20info()%20method%20prints,(non%2Dnull%20values)).
- Wakefield, K. (no date) *Predictive Modeling Analytics and Machine Learning*. Available at: https://www.sas.com/en_gb/insights/articles/analytics/a-guide-to-predictive-analytics-and-machine-learning.html.
- van wyk, A. (2018) 'An Overview of LightGBM', 16 May.
- Yan Liu, T. *et al.* (2016) 'LightGBM', 1 October. Available at: <https://www.microsoft.com/en-us/research/project/lightgbm/>.
- Zigzag Global (2021) 'Why retailers should be prepared for a rise in returns', 10 June. Available at: <https://www.zigzag.global/why-retailers-should-be-prepared-for-a-rise-in-returns/> (Accessed: 26 August 2022).

9 Appendix:

Below GitHub link is to access code done in PostgreSQL and Python for analysis in dissertation:

https://github.com/Spd8097/Dissertation_Code