A MACHINE LEARNING APPROACH TO UNDERSTAND THE AMAZON BUY BOX MECHANISM

by
EMRE ERYILMAZ

Submitted to the Sabancı Graduate Business School
in partial fulfilment of
the requirements for the degree of Master of Science in Business Analytics

Sabancı University
July 2022

A MACHINE LEARNING APPROACH TO UNDERSTAND THE AMAZON BUY BOX MECHANISM

Approved by:

Assoc. Prof. Dr. Ayse Kocabıyıkoglu .....
(Thesis Advisor)

Asst. Prof. Dr. Fatma Tevhide Altekin .....

Asst. Prof. Dr. Celile Itır Göğüş .....

Date of Approval: July 6, 2022

ABSTRACT

A MACHINE LEARNING APPROACH TO UNDERSTAND THE AMAZON BUY BOX MECHANISM

EMRE ERYILMAZ

Business Analytics M.Sc. Thesis, July 2022

Thesis Advisor: Assoc. Prof. Dr. Ayşe Kocabıyıkoğlu

Thesis Co-Advisor: Asst. Prof. Dr. Burak Gökgür

Keywords: Amazon, buy box, machine learning, classification, Random Forest, XGBoost, LightGBM, hyperparameter tuning, subset selection

Amazon marketplace is the leading e-commerce company globally. One of the most important features of the marketplace is a product can be offered to the customers by more than one seller. One of these sellers is selected by Amazon as the buy box winner on the product details page. Winning the buy box position is very important to a seller because more than 80% of the sales occur by buy box sellers. In this thesis, we developed a machine learning approach to understand the Amazon Buy Box mechanism. We have gathered the data set via Amazon AnyOfferChangedNotification API. The data set consists of the lowest twenty offers of a product and features of the sellers with the gathering time of the data set which is publicly available. We have developed supervised machine learning classification models which are Random Forest, XGBoost, and LightGBM to predict buy box winners. We have applied hyperparameter tuning and several subset selection techniques. These models reflected over 97% of accuracy for selected products. XGBoost model performed slightly higher than other models in terms of accuracy, precision, recall, and f1 score.

ÖZET

AMAZON BUY BOX MEKANİZMASINI ANLAMAK İÇİN BİR MAKİNE ÖĞRENMESİ YAKLAŞIMI

EMRE ERYILMAZ

İş Analitiği Yüksek Lisans Tezi, Temmuz 2022

Tez Danışmanı: Doç. Dr. Ayşe Kocabıyıkoğlu

İkinci Tez Danışmanı: Yrd. Doç. Dr. Burak Gökgür

Anahtar Kelimeler: Amazon, buy box, makine öğrenimi, sınıflandırma, Random Forest, XGBoost, LightGBM, hiperparametre ayarlama, alt küme seçimi

Amazon marketplace, dünyanın önde gelen e-ticaret şirketidir. Pazar yerinin en önemli özelliklerinden biri, bir ürünün birden fazla satıcı tarafından müşterilere sunulabilmesidir. Bu satıcılardan biri, ürün ayrıntıları sayfasında satın alma kutusu kazananı (buy box) olarak Amazon tarafından seçilir. Bir satıcı için buy box pozisyonunu kazanmak çok önemlidir çünkü satışların %80'inden fazlası buy box satıcıları tarafından yapılır. Bu tezde, Amazon Buy Box mekanizmasını anlamak için bir makine öğrenimi yaklaşımı geliştirdik. Veri kümesini Amazon AnyOfferChanged-Notification API aracılığıyla topladık. Veri seti, bir ürünün en düşük yirmi teklifi ve satıcıların özelliklerini tarih bilgisi ile birlikte içermektedir. Buy box kazananlarını tahmin etmek için Random Forest, XGBoost ve LightGBM olan denetimli makine öğrenimi sınıflandırma modelleri geliştirdik. Ayrıca, hiperparametre ayarlama ve çeşitli alt küme seçim teknikleri uyguladık. Bu modeller, seçilen ürünler için %97'den fazla doğruluk yansıtmaktadır. XGBoost modeli, doğruluk, kesinlik, geri çağırma ve f1 puanı açısından diğer modellerden biraz daha yüksek performans göstermiştir.

ACKNOWLEDGEMENTS

This work was completed with the great support of many people. First of all, I would like to thank my thesis advisor Assoc. Prof. Ayse Kocabıyıkoğlu and co-advisor Asst. Prof. Burak Gökgür for their support and valuable guidance during all phases of the thesis.

Secondly, I would like to thank my dear friend ZHA for his help with data collection.

Thirdly, I would like to thank my friends Afşin Sancaktaroğlu, Osman Göktepe and Özgün Göl for the brainstorming during model development and friendship during the master's study.

Finally, I would like to thank my beloved wife Eda. Without her support, I will not be able to have had enough motivation and time to complete this study.

July 2022

Emre Eryılmaz

dedicated to my family:
my wife Eda,
my son Mete Ali,
my daughter Defne

July 2022

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AUC: Area Under Curve

GB: Gradient Boosting Classifier

LightGBM: Light Gradient Boosting Machine Classifier

LR: Logistic Regression

ML: Machine Learning

NB: Naive Bayesian

NN: Neural Network

NPV: Negative Predictive Value

P1: Product 1

P2: Product 2

P3: Product 3

P4: Product 4

RF: Random Forest Classifier

SVM: Support Vector Machine Classifier

XGBoost: eXtreme Gradient Boosting Classifier

1. INTRODUCTION

Amazon has been founded in 1994 as a website that only sold books (Hartmans, 2021). The company launched a marketplace in 2000 where third-party sellers could sell items from selected categories such as books, DVDs, video games, electronics, etc. on Amazon (Allen, 2001).Over the years, the available categories on Amazon have increased as the share of online retailing increased. The share of e-commerce in retail is less than 1% in 2000 in the USA. The share of online retailing dramatically increased and reached more than 15% in 2022 in the USA (census.gov, 2020). Amazon is one of the most successful companies in the world which benefit from the increase in the share of e-commerce over the retail market. Amazon.com is leading the global e-commerce market, with a revenue of US$ 120,968 million in 2020 (Statista, 2022).

Besides allowing sellers to sell products on the Amazon marketplace, Amazon provides services to sellers such as management of inventories, advertisement, and Fulfilled by Amazon (FBA) program where Amazon manages logistics of the products. Amazon applies a customer-centric business model. Customer centricity is the ability of people in an organization to understand customers' situations, perceptions, and expectations. Customer centricity demands that the customer is the focal point of all decisions related to delivering products, services, and experiences to create customer satisfaction, loyalty, and advocacy (Gartner, 2022). Jeff Bezos, the founder of Amazon, stated that "If a third party could offer a better price or better availability on a particular item, then we wanted our customer to get easy access to that offer." (sec.gov, 2005). Parallel to this approach, the platform allows different sellers to sell the same products even with Amazon.

Hence, when a customer wants to purchase a product on Amazon, he could make a choice between different sellers with different prices while investigating the features of a seller such as feedback rating, feedback count, and shipment details. If multiple sellers offer the same product, the Amazon algorithm selects one of the sellers as the "buy box". The buy box is the box on a product detail page where customers can begin the purchasing process by adding items to their shopping carts

(sellercentral.amazon.com, 2022).

The details of a product details page are presented in Figure 1.1. As shown in Figure 1.1, box number 1 illustrates the buy box. A customer can add the product directly to the buy box. Box number 2 shows the buy box winner and if the seller is using the FBA program to deliver the product. A customer can click on the link at box number 3 to view all the sellers of the product. By clicking add to cart button next to a seller he can add the product to his shopping cart. Some of the sellers other than the buy box winner are shown in box number 4. A customer can add the product to his basket by clicking add to cart button next to a seller in this box. Buying a product directly from the buy box is the easiest way to purchase a product.

Figure 1.1 Product Detail Page on Amazon

Winning the buy box position is very important for a seller because it boosts the sales of the product. More than 80% of the sales on Amazon go through the buy box position (Vanaman, 2022). One may argue that the seller with the cheapest product price will win the buy box. However, this is not the case most of the time. Chen, Mislove & Wilson (2016) has shown that price is not the sole feature used by the Buy Box algorithm. The algorithm behind the selection of the buy box winner is not disclosed by Amazon. There are some publicly available features of the sellers such as listing price, shipment price, feedback count, etc. that may affect the selection of the buy box winner. However, there are some publicly not available features such as the product return ratio of a seller, number of negative comments, buy box ratio

of a seller on all products, etc. may have an effect on the determination of the buy box winner.

In this study, we develop supervised classification models to predict buy box winners of Amazon sellers. We have collected the data of twenty products from ten different categories for a month from Amazon AnyOfferChangedNotification API. This API provides the information of the twenty sellers who offers the lowest prices of a product. If there is a price change for any of these sellers, the API provides new data set with current information. The new data set is gathered with the time stamp of price change time which is called publish time. We focused on the four products where the change of the buy box winner occurred most in the data set. The collected data sets include sellers' characteristics such as feedback rate, feedback count, sending the product domestically, shipment price, listing price, etc. for each product, and publish time. This data is publicly available and can be gathered via web crawling.

We first started working on the data set with feature building. By using existing features, we have built new numerical and binary features which will help us to relate a seller with other sellers within the publish time and previous publish time. The new numerical features mostly measure the difference between mean, minimum, and maximum values of the listing price, shipping price, feedback count, and feedback rate at publish time and relate these features with the buy box winner of the previous publish time. Binary features include several conditions of the sellers such as offering the lowest price, being the buy box winner at the previous publish time, offering a lower price than the previous buy box winner, etc. Next, we have scaled the numerical features within their publish time. This strategy allowed us using of a row independent from the publish time.

We have split the data set into three parts which are train, validation, and test sets. The train set covers 60% of the data while the validation set covers 20%, and the test set covers 20%. We used the train data set to fit the selected machine learning algorithms and to determine hyperparameters of machine learning algorithms. We have used the validation set to select a subset of important features based on a feature's contribution to accuracy. We have used the test set to measure model prediction performances.

Before fitting supervised machine learning classifier algorithms, we filtered features to decrease high dimensionality and mitigate potential overfitting problems on the train data set. We have used Pearson's correlation coefficient to eliminate features that have an absolute value of the correlation of more than 0.8. We have used Random Forest (RF), eXtreme Gradient Boosting (XGBoost), and Light Gradient Boosting Machine (LightGBM) algorithms on the remaining feature set. We have

determined hyperparameters of each algorithm on the train data of each product. Afterward, we benefited from the validation set for subset selection of important features of the trained models. We have calculated accuracy, precision, recall, and f1 scores based on the prediction of the test data set for each algorithm and product. All of the three machine learning algorithms have provided accuracy results higher than 97% for all products. XGBoost algorithm has provided slightly better accuracy results for all products with an average of 98.2%. RF has provided 98.06% of accuracy while LightGBM resulted in 98.02%. Predicting with high accuracy rates for different products confirms the robustness of the approach.

We have applied a different feature set, subset selection, and algorithms from other research on the buy box mechanism and achieved promising results. Our key managerial contribution is that our approach to creating a predictive model can be applied by sellers to develop their strategies related to pricing, feedback count, rate, etc. They can make a simulation that predicts the buy box seller while creating what-if scenarios for different cases such as price levels. Additional to the sellers, price advisory sites such as sellics.com may benefit from the approach.

The remainder of this study is organized as follows. In Section 2, related studies and their findings have been explained in detail. The data set, empirical analysis, and data preprocessing have been provided in Section 3. Section 4 covers the predictive model creation process and discussion the results of each model. Finally, concluding remarks and potential future studies were discussed in Section 5.

2. LITERATURE REVIEW

In this section, we are going to explain related literature under two headings. First, we will examine feature selection and ensembled machine learning models literature. Secondly, we will look at the literature on buy box prediction research.

2.1 Feature Selection Literature

Dimension reduction via feature selection is a very significant step of machine learning to increase computational time and mitigate an overfitting problem on the train data. Lazar, Taminau, Meganck, Steenhoff, Coletta, Molter, de Schaetzen, Duque, Bersini & Nowe (2012) state that there are four types of feature selection methods which are filters, wrappers, embedded, and ensembled. Filter methods evaluate the discriminative power of features based only on intrinsic properties of the data via determining a threshold or relevance score such as Pearson's correlation coefficient, variance threshold, and chi-square score. Wrapper methods select the subset of features by minimizing error based on a specific classifier such as forward feature selection, backward feature elimination, and exhaustive feature selection. This technique does not guarantee optimal subset selection for another classifier and requires high computational power. Embedded methods benefit both filter and wrapper techniques where a machine learning algorithm is used while applying a penalty to prevent overfitting such as Lasso, and Ridge regression. Ensemble methods generate multiple diverse feature selectors and combine their outputs such as RF, XGBoost, LightGBM, and Gradient Boosting (GB). The common disadvantage of the first three techniques is the dependence on the training data set where a change in the training set may change the selected features (Meinshausen & Bühlmann, 2010). The ensemble approach is superior to conventional feature selection methods in many aspects such as the ability to handle stability issues (Guan, Yuan, Lee, Na-

jeebullah & Rasel, 2014). It is also common that apply more than one method to reach the optimal feature selection. Lee & Cha (2002) states that applying the filter method initially and the wrapper method next provides a better feature subset while benefiting less computational power.

2.2 Model Selection Literature

From the methodological point of view, our work is in management science and machine learning research areas. Tree-based algorithms for prediction problems are widely applied by marketing scholars (Sikdar, Kadiyali & Hooker, 2019). In this section, we have focused on research related to e-commerce.

Niu, Li & Yu (2017) explored Walmart's online customer search and purchase behavior. They have adopted RF and LR machine learning models. RF model has provided a high accuracy rate of 76% while the LR model provided 61%. The RF model suggested that page and session dwell time, user type, click entropy, and click position are among the most important features of the conversion factor.

Song & Liu (2020) have developed an XGBoost algorithm for predicting purchasing behavior on an e-commerce platform. For comparative reasons, they have also applied RF to the same data set. The data set contains ten numerical and four categorical features. They have applied one-hot encoding for categorical features because the algorithm requires numerical inputs. They have achieved a 90.15% prediction accuracy score by applying the XGBoost algorithm while RF provided 89,58%. Additionally, the XGBoost algorithm has provided slightly better scores than RF with positive precision of 59%, the positive recall of 73%, and the positive f-1 score of 65%.

Hambarde, Silahtaroğlu, Khamitkar, Bhalchandra, Shaikh, Kulkarni, Tamsekar & Samale (2020) has made a comparative analysis of several supervised machine learning algorithms by predicting customer purchase behavior of an online retailing company. They have applied RF, LR, SVM, GB, XGBoost, K-Neighbors, Decision Tree, and Naïve Bayes algorithms. The top three algorithms that provided the highest accuracy are RF with 94.81%, XGBoost with 94.78%, and GB with 94.66%. These three algorithms are tree-based ensemble models.

Li, Gu, Zhou & Sun (2015) worked on an m-commerce recommendation system for Alibaba's competition. The dataset consists of about 6 billion operation logs made by 5 million Taobao users towards over 150 million items spanning one month. They started their research with data preprocessing which includes outlier removal, and feature engineering. Then they applied GBDT as a training model and combined the outputs of the model with LR to get final predictions. They have achieved an 8.66% of f1 score and ranked third in the competition.

Vanderveld, Pandey, Han & Parekh (2016) has modeled a customer lifetime value system (CLTV) by applying the RF algorithm for Groupon which is a global e-commerce company. The CLTV predicts the future value of a user. The future value of a customer is the prediction of the net dollar value attributed to each individual customer. The feature set consists of almost every aspect of each customer's relationship with the platform. Initially, they grouped the customers into six clusters. Secondly, they have applied the RF algorithm to all groups. On average, they have achieved 93% accuracy, 50% precision, and 63% recall scores.

2.3 Buy Box Prediction Literature

Although the e-commerce market has grown dramatically over the past decades and Amazon is the leading company globally, there are limited numbers of research that investigates Amazon's buy box algorithm. Chen et al. (2016) focused on the algorithmic pricing strategies of sellers and the impact of these strategies on the dynamics of the Amazon marketplace. As a starting point for their research, they tried to understand the buy box mechanism. They have collected 1641 best-seller products for four months and the top 20 sellers' features of these products such as listing price, and feedback rate via web crawling. They have used seven features which are price difference to lowest, price ratio to lowest, average rating, positive feedback, feedback count, is product FBA, and is Amazon seller. In this research, only the ensembled feature selection method has been used. They have applied a Random Forest (RF) classifier to predict the buy box winner. The prediction results indicate that price difference to lowest, price ratio to lowest, positive feedback, and is Amazon the seller features have the highest importance. Their model has achieved 75% - 85% accuracy. Additionally, their research reflected that having the lowest price does not guarantee the buy box position.

Gómez-Losada & Duch-Brown (2019) tried to understand the dynamics of the Amazon buy box algorithm via a classification-based predictive approach. They collected the data of new products from 26 categories on the Amazon Italy web page via web crawling. The data consist of price, product rating, feedback count, FBA condition, stock availability, and Amazon's choice condition features of products, seller information, and time of crawling. Additional to these features, they have created more than 20 candidate features such as weekday, the ratio of previous price and current price, the difference between the lowest price and current price, and ratio and difference of current price and 4, 8, 16, 32 rolling mean of prices, etc. They have applied a filter-based feature selection method to decrease the high dimensionality of the data. Features with near-zero variance and Pearson's correlation coefficient greater than ± 0.8 were removed from the data set. They have created three predictive models which are Support Vector Machine (SVM), Neural Network (NN), and RF classifiers. RF provided the highest accuracy with 94% while SVM is 91% and NN is 92%. The most important features were consecutive prices in products and feedback count in the RF model.

Our study aims to predict Amazon buy box winners and understand important features of the mechanism similar to the work of Chen et al. (2016) and Gómez-Losada & Duch-Brown (2019). There are several contributions of our work to the literature on the Amazon buy box mechanism. Firstly, we have used XGBoost and LightGBM algorithms in addition to RF algorithms which are not benefited in previous works. Secondly, we have applied a scaling strategy that takes into account only the data set at publish time. Thirdly, we have applied hyperparameter optimization for each algorithm. In addition to these, we have used validation data set to make an additional subset selection of important features that are determined.

3. DATA SET ANALYSIS AND PREPROCESSING

In this chapter, we are going to explain steps from data collection to the preparation of the data set we are going to use in the predictive models. Firstly, we are going to explain how the data set is collected and its structure. Secondly, we will explore the data set via descriptive statistics. Finally, we are going to explain feature building, data preprocessing, and feature elimination.

3.1 Data Collection and Data Structure

The data is obtained via Amazon AnyOfferChangedNotification API for the cheapest 20 products from 10 categories that cover one month starting from 14 February 2022. The AnyOfferChanged notification is sent whenever there is a listing change for any of the cheapest 20 offers, or if the external price (the price from other retailers) changes for an item that you sell (Amazon MWS, 2022). The data set is publicly available that can be gathered via web crawling such as seller list, listing price, feedback count, buy box winner, etc. of the top sellers of a product. The data set consists of snapshots of sellers and their features at times of price change of a product. The features and explanations are detailed in Table 3.1.

Each snapshot includes one buy box seller and information on sellers' and products' characteristics and which seller wins the buy box. Therefore, there is only one seller who won the buy box which is positive class, and the remaining is in the negative class for each snapshot. The classes are unbalanced due to the number of the buy box winner and the rest are not equally distributed. These snapshots have been aggregated which constitutes a longitudinal data set for a product indexed by the publish time containing sellers' characteristics and product information. We aim to understand the importance of the features that enable a seller to win the buy box while predicting the buy box seller. To have a better understanding, we have

Table 3.1 Data Set Feature List

Feature	**Data Type**	**Description**
publish_time	Date and Time	A time snapshot of the data has been created. A new record has been created when a seller updates the product's price.
ASIN	Text	Unique product code
seller_id	Text	Unique seller id
is_prime	Binary	The product is fulfilled from the Amazon warehouse (1: True, 0: False)
is_amazon	Binary	The seller is Amazon itself (1: True, 0 False)
is_domestic	Binary	Is product shipment domestic (Canada based) (1: True, 0: False)
feedback_count	Integer	Number of feedbacks from the seller
feedback_rating	Integer	Rating of the feedback of the seller
availability	Binary	Availability condition of the product for a seller (1: True, 0: False)
minimum_hours	Integer	Minimum hours of dispatch to the customer
maximum_hours	Integer	Maximum hours of dispatch to the customer
listing_price	Float	The listing price of the product
shipping_price	Float	Shipment price of the product
is_buybox_winner	Binary	The condition of a seller is the buy box winner (1: True, 0: False)
category	Text - (Categorical)	The category of the product

selected 4 products from 2 categories that contain the highest number of distinct sellers who won the buy box.

3.2 Descriptive Analysis

In this section, we are going to analyze the descriptive properties of the features for selected four products from two different categories which are Toys&Games and Home&Kitchen. Product 1 (P1) and Product 2 (P2) belong to the Toys&Games category while Product 3 (P3) and Product 4 (P4) belong to the Home&Kitchen category. We will make the analysis product by product and contains all publish_times as a longitudinal data set.

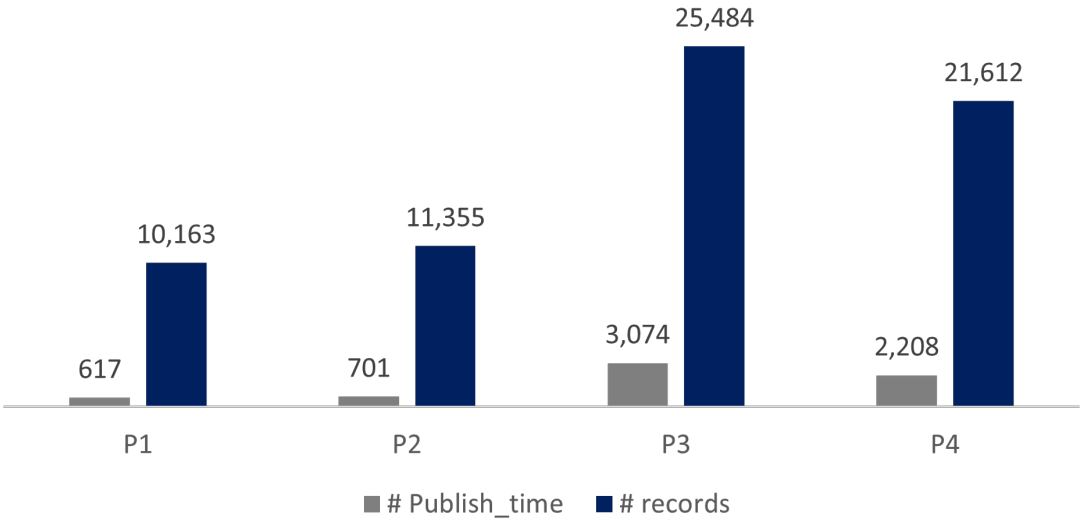

Figure 3.1 # of publish_time and records

3.2.1 Numerical Features

The descriptive statistics of the numerical features will be examined in this part. listing_price, feedback_count, feedback_rating, minimum_hours, maximum_hours, and shipping_price features have numerical values. The number of publish_time and records for each product are provided in Table 3.1

Table 3.2 Descriptive Statistics of P1 (Numerical Features)

P1	**listing _price**	**feedback _count**	**feedback _rating**	**minimum _hours**	**maximum _hours**	**shipping _price**
mean	56.2	128.3	82.0	35.5	54.6	0.7
std	5.8	251.4	27.7	50.1	66.3	2.7
min	40.0	0.0	0.0	0.0	0.0	0.0
25%	48.5	4.0	83.0	0.0	0.0	0.0
50%	58.3	22.0	93.0	24.0	48.0	0.0
75%	60.8	96.0	100.0	24.0	48.0	0.0
max	65.9	1016.0	100.0	264.0	360.0	29.9

Table 3.3 Descriptive Statistics of P2 (Numerical Features)

P2	**listing _price**	**feedback _count**	**feedback _rating**	**minimum _hours**	**maximum _hours**	**shipping _price**
mean	62.6	5150.3	83.5	49.1	69.8	0.4
std	4.7	10697.6	20.9	43.5	63.4	1.8
min	43.9	0.0	0.0	24.0	24.0	0.0
25%	61.7	14.0	80.0	24.0	24.0	0.0
50%	62.9	195.0	89.0	24.0	48.0	0.0
75%	64.6	960.0	92.0	72.0	96.0	0.0
max	79.4	31408.0	100.0	264.0	360.0	9.4

There is a significant difference between the minimum and the maximum listing_price of P1. The maximum listing_price is more than 50% higher than the minimum listing_price. There are sellers which do not have any feedback_count and feedback_rating while some have almost 8 times higher than average feedback count. Some sellers offer minimum and maximum hours of shipment that can be measured on a weekly scale while the majority of the sellers offer to make shipments between a minimum of one and a maximum of two days. Most of the sellers do not ask for any price for shipment while some asks significant amount of money in comparison to the P1 mean listing price. The details are provided in Table 3.2.

Similar to the P1, there is a significant amount of difference between the minimum and the maximum listing price of P2. Almost 50% of the listing_prices are lower than the average listing price. Some sellers do not have any feedback_count while some have 6 times higher than the mean feedback count. As minimum_hours and maximum hours features reflect, one of the sellers can ship the product within a day. The shipment varies between one day and fifteen days. Most of the sellers do not require shipment_price to deliver the product. The details are provided in Table 3.3.

Like P1 and P2, maximum listing prices are significantly higher than the minimum

Table 3.4 Descriptive Statistics of P3 (Numerical Features)

P3	**listing _price**	**feedback _count**	**feedback _rating**	**minimum _hours**	**maximum _hours**	**shipping _price**
mean	80.5	4448.3	84.7	51.2	85.9	1.4
std	13.9	9831.0	8.1	45.6	68.8	3.8
min	56.0	6.0	50.0	24.0	24.0	0.0
25%	73.9	51.0	80.0	24.0	48.0	0.0
50%	75.5	183.0	83.0	24.0	48.0	0.0
75%	83.1	1081.0	91.0	96.0	120.0	0.0
max	114.3	30871.0	100.0	144.0	240.0	11.8

Table 3.5 Descriptive Statistics of P4 (Numerical Features)

P4	**listing _price**	**feedback _count**	**feedback _rating**	**minimum _hours**	**maximum _hours**	**shipping _price**
mean	108.8	5844.4	77.0	126.3	189.3	0.9
std	17.2	10398.8	27.2	194.5	286.7	4.1
min	56.6	0.0	0.0	0.0	0.0	0.0
25%	107.8	22.0	80.0	24.0	48.0	0.0
50%	109.4	196.0	80.0	24.0	48.0	0.0
75%	120.4	3253.0	92.0	96.0	120.0	0.0
max	139.4	30871.0	100.0	672.0	1008.0	35.0

where the maximum listing price is more than double the minimum listing price. However, the majority of the sellers have lower listing_prices than mean listing prices. All sellers have feedback, but some sellers have a very high number in comparison to the rest of the sellers. 75% of the sellers have a lower feedback_count than average that reflects some of the sellers have very high feedback counts in compared to the rest. The shipment time varies sellers dramatically. Most of the sellers are making shipments between one and two days while it takes more than 5 days to 10 days for some of the sellers. Similar to the other products, the majority of the sellers do not request shipment_price. The details are provided in Table 3.4.

Similar to the previous products, there is an important difference between the minimum and maximum listing_price. 50% of the sellers' prices are very close to the average listing price. Some sellers do not have any feedback_count and feedback_rating while some have more than 30 thousand feedback_counts. Minimum and maximum hours to shipment vary between 0 hours to 42 days. 50% of the sellers are ready for shipment within two days. Like P1, P2, and P3 shipment_price is not required for the majority of the sellers. The details are provided in Table 3.5.

3.2.2 Binary Features

is_prime, is_amazon, is_domestic, and availability features are the binary features we are going to examine the descriptive statistics. 1 means True and 0 means False for binary variables. Around 30% of the sellers of P1 are using Amazon prime services while none of them is Amazon itself. 13.3% of the sellers are making deliveries from Canada and all sellers have available products to fulfill the request of the customers. The details of binary variables for P1 are provided in Figure 3.2.

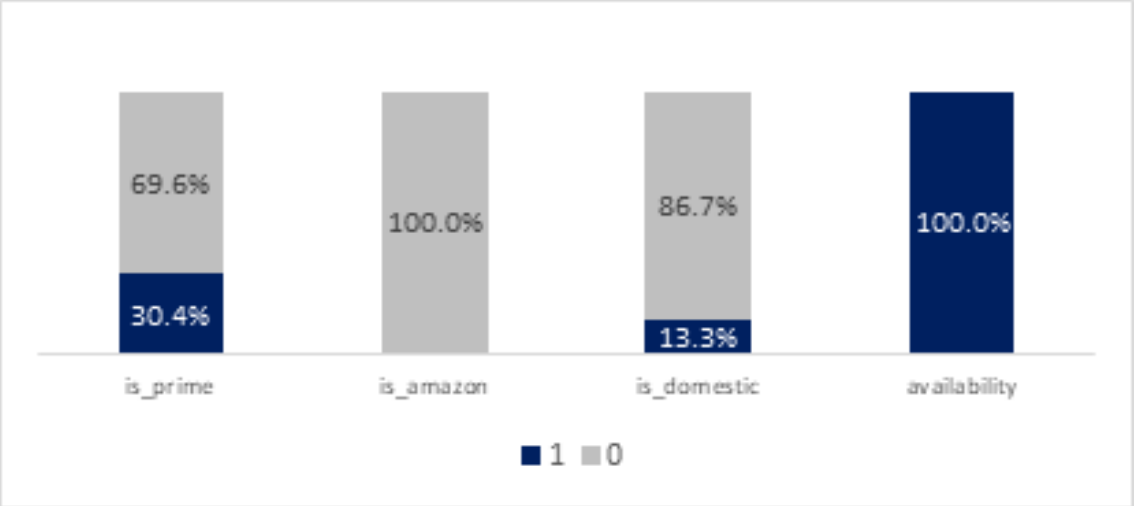

Figure 3.2 Descriptive Statistics of P1 (Binary Features)

None of the sellers of P2 is using Amazon prime services and Amazon is not selling P2. 33.7% of the sellers are delivering products from Canada and all of them have available products to fulfill customer orders. The details of binary variables for P2 are provided in Figure 3.3.

None of the sellers of P3 is using Amazon prime services and Amazon is not selling P3 itself. 36.7% of the sellers are delivering products from Canada and all of them have available products to fulfill customer orders. The details of binary variables for P3 are provided in Figure 3.4.

Only 10.2% of the sellers of P4 are using prime services and these seller records belong to Amazon itself. 40.9% of the sellers are fulfilling demand domestically and all of them available stocks of P4. The details of binary variables for P4 are provided in Figure 3.5.

For P2 and P3, none of the sellers is prime or amazon. P1 seller list is not including amazon. Additionally, all sellers have available products for all products. These features for the specified products do not provide any extra information to predict which seller will become the buy box seller.

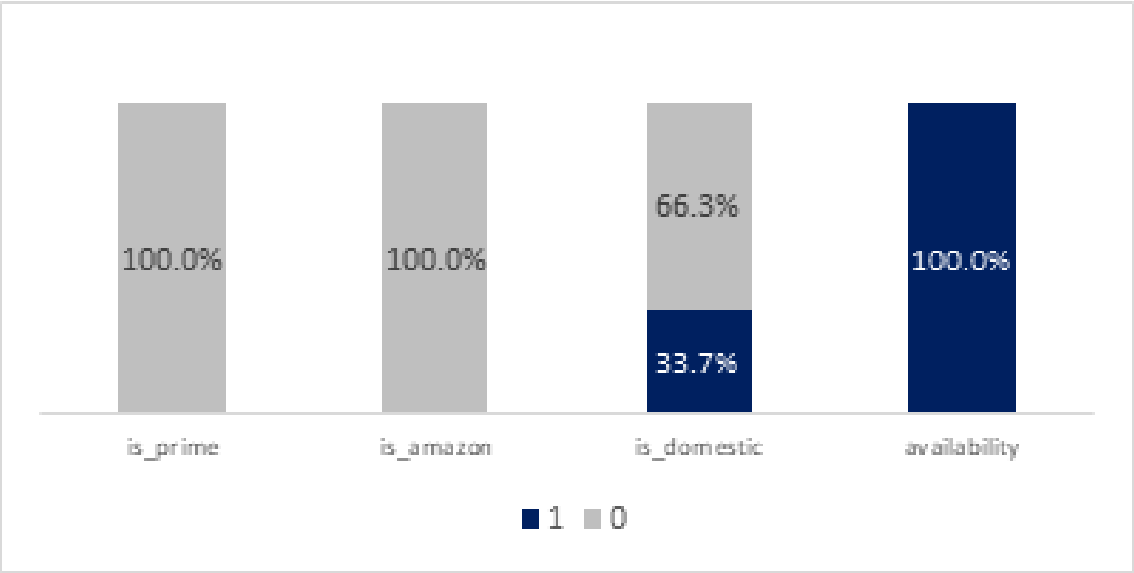

Figure 3.3 Descriptive Statistics of P2 (Binary Features)

3.3 Feature Building and Data Preprocessing

In this section, we are going to describe the creation and selection of features used for building predictive models. Secondly, we are going to explain the data preprocessing and feature elimination processes.

3.3.1 Feature Building

There are six numerical and four binary variables that we can use in a machine learning model and some of the binary features are not explanatory to predict the final class of the sellers as detailed in Section 3.2.2. Building additional features from existing ones may help to increase modeling performance. Additionally, as explained in the Data Collection part, when there is a price change of a seller, in other words for a publish_time, there is only one buy box winner class out of a maximum of 20 sellers. This situation creates two complications. Firstly, the size of data is not enough to create a machine learning model for a publish_time. However, we want to use all data that includes all publish_time values for a product. To achieve this, we are going to build features that position a seller's features with other sellers' feature values within a publish_time. Secondly, available features are not explaining a seller's characteristics in comparison to buy box winning seller's characteristics

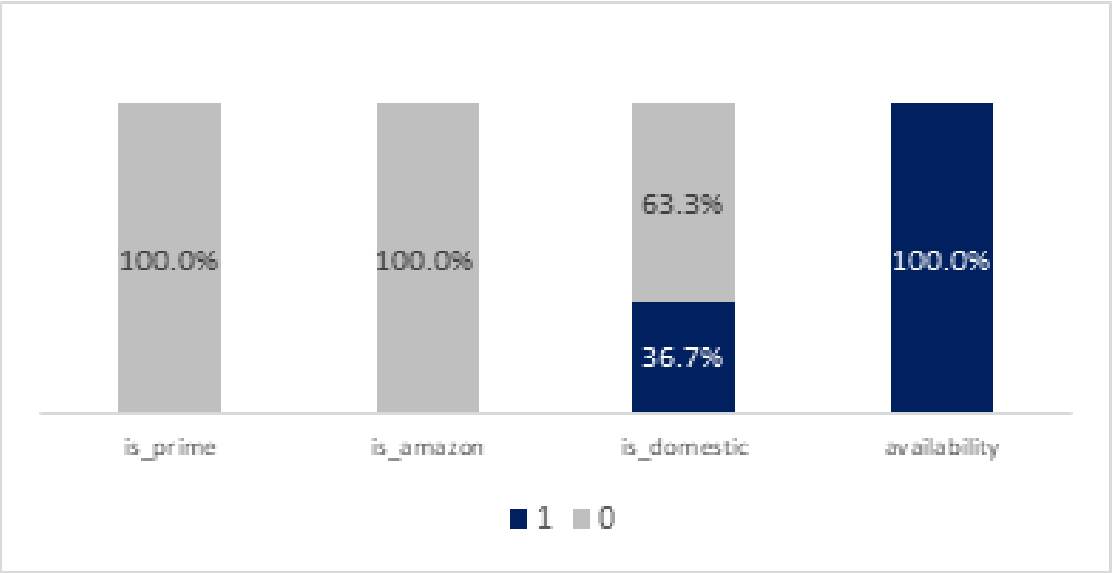

Figure 3.4 Descriptive Statistics of P3 (Binary Features)

at previous publish_time. Therefore, we are going to build several features that relate to the characteristics of a seller at publish_time and buy box winner seller's characteristics at previous publish_time. By creating these new features, we will be able to use the whole dataset of a product without using publish_time and depending on a time series model. Initially, we have created two new features which are feedback points and average hours by using existing features. The details are provided in Table 3.6.

Finally, we have created a new set of numerical and binary features that relates a seller's features at publish_time with the buy box winner seller's features at previous publish_time. The details of this feature set are provided in Table 3.7.

3.3.2 Data Preprocessing

Amazon is selecting a buy box winner within available sellers at a publish_time based on the features reachable via web crawling and features that only Amazon has. At each listing_price change based on the sellers' features, Amazon decides which seller will be the buy box winner. We think that a seller's position against other sellers within a publish time is the key element to estimate the buy box winner for that publish_time. For that reason, we have applied standard scaling to all numerical features and all products within a publish_time for all publish_time values. This scaling strategy enabled us to use each row independently for train and test purposes while not depending on the time series property of the longitudinal

Table 3.6 New Set of Features (Current publish_time)

Feature	**Data Type**	**Description**
diff_listing_ price_mean	Float	The difference between a seller's listing price and the mean listing price in the publish_time
diff_listing_ price_lowest	Float	The difference between a seller's listing price and the lowest listing price in the publish_time
diff_average_ hours_mean	Float	The difference between a seller's average hours and the mean average hours in the publish_time
diff_average_ hours_lowest	Float	The difference between a seller's average hours and the lowest average hours in the publish_time
diff_fpt_ cnt_mean	Float	The difference between a seller's feedback count and the mean feedback count in the publish_time
diff_fpt_ cnt_highest	Float	The difference between a seller's feedback count and the highest feedback count in the publish_time
diff_fpt_ rate_mean	Float	The difference between a seller's feedback rate and the mean feedback rate in the publish_time
diff_fpt_ rate_highest	Float	The difference between a seller's feedback rate and the highest feedback rate in the publish_time
diff_fpt_ point_mean	Float	The difference between a seller's feedback point and the mean feedback point in the publish_time
diff_fpt_ point_highest	Float	The difference between a seller's feedback point and the highest feedback point in the publish_time
is_lowest	Binary	The condition of a seller has the lowest listing price in the publish time. (1 True, 0 False)

Table 3.7 New Set of Features (Related to the Buy Box Winner of the Previous publish_time)

Feature	**Data Type**	**Description**
diff_pr_bb_ listing_price	Float	The difference between a seller's listing price at publish_time and the listing price of the buy box winner at the previous publish_time
diff_pr_bb _avg_hours	Float	The difference between a seller's average hours at publish time and the average hours of the buy box winner at the previous publish_time
diff_pr_bb_ fpt_point	Float	The difference between a seller's feedback point at the publish time and the feedback point of the buy box winner at the previous publish_time
diff_pr_bb_ fpt_count	Float	The difference between a seller's feedback count at the publish time and the feedback count of the buy box winner at the previous publish_time
diff_pr_bb_ fpt_rate	Float	The difference between a seller's feedback rate at the publish time and the feedback rate of the buy box winner at the previous publish_time
check_pr_bb _winner	Binary	The condition of a seller has is the boy box winner at the previous publish time (1 True, 0 False)
check_pr_bb _isprime	Binary	The condition of a seller has the same is_prime value with the buy box winner of the previous publish time (1 True, 0 False)
check_pr_bb _isdomestic	Binary	The condition of a seller has the same is_domestic value with the buy box winner of the previous publish time (1 True, 0 False)
check_pr_bb _price	Binary	The condition of a seller has a lower listing price than the buy box winner of the previous publish time (1 True, 0 False)

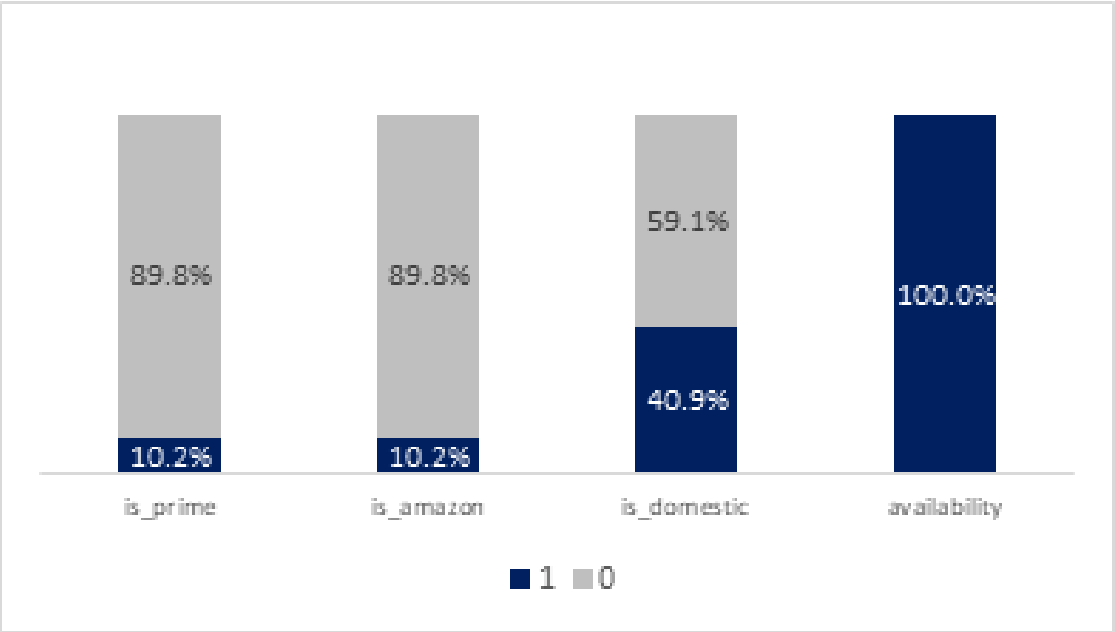

Figure 3.5 Descriptive Statistics of P4 (Binary Features)

data set structure. We have used the scikit-learn StandardScaler library for the scaling process. StandardScaler standardizes features by removing the mean and scaling to unit variance (scikit learn.org, 2022b). This approach is compatible with real-life conditions. An Amazon seller can collect its own and competitor sellers' features. By using the data set, a seller make a simulation by updating some features such as listing_price, feedback count, etc., and can predict which seller will win the buy box. Additionally, first publish_time has been extracted from the data set for all products due to some of the new features checking the buy box winner's features of the previous publish_time.

12 features are collected via Amazon API and 22 new features had been created that make 31 features for each seller. Variables with an absolute value of Pearson correlation coefficient higher than 0.8 were excluded from the data set to avoid high dimensionality. Pearson correlation coefficient is a used to measure the linear association between two variables and is denoted by r. r can be equal to a maximum of 1 which reflects the perfect positive correlation and can be a minimum of -1 which reflects the perfect negative correlation. r equal to 0 represents that there is no correlation. The figures of correlation matrixes for each product provided in Figure 3.5, Figure 3.6, Figure 3.7, and Figure 3.8. Additionally, the remaining features for each product listed in Table 3.8.

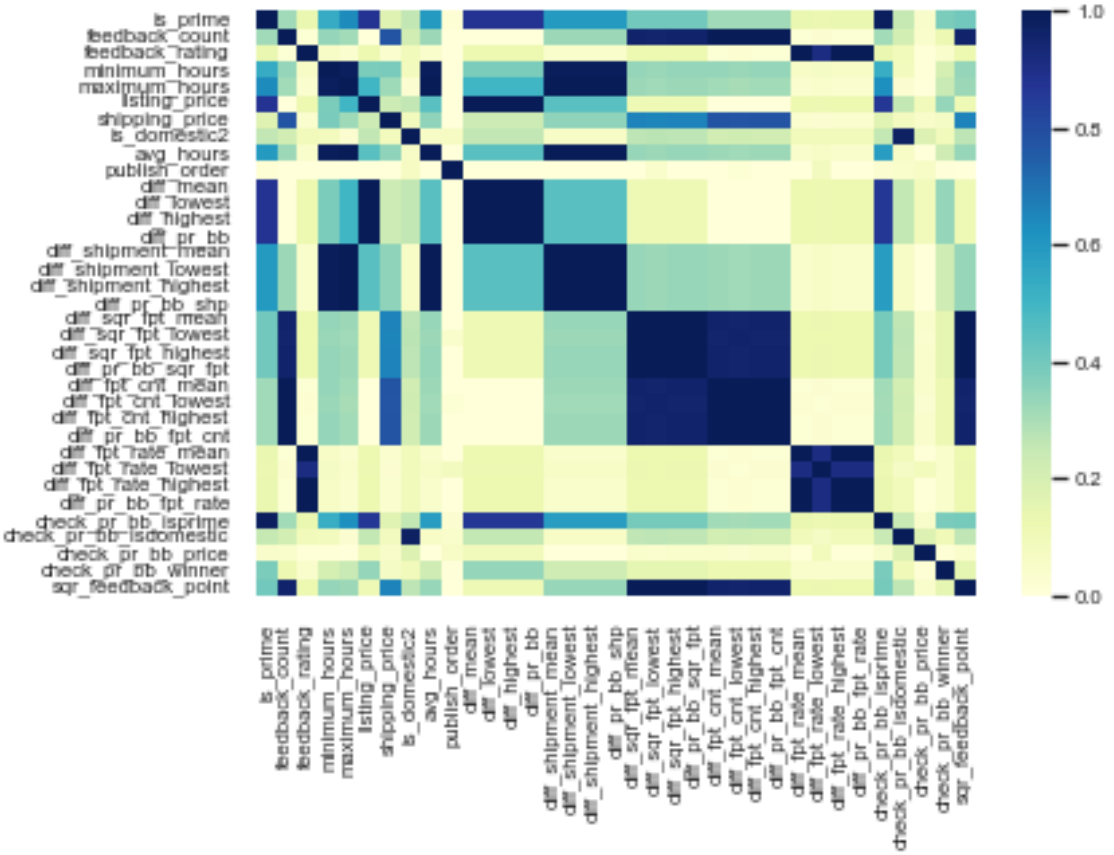

Figure 3.6 Correlation Matrix of P1

Table 3.8 Remaining Feature Set

Product	**Feature Set**
P1	is_prime, feedback_count, feedback_rating, minimum_hours, listing_price, shipping_price, seller_id2, is_domestic, check_pr_bb_price, check_pr_bb_winner
P2	feedback_count, feedback_rating, minimum_hours, listing_price, shipping_price, is_domestic, diff_lowest, is_lowest, check_pr_bb_isdomestic, check_pr_bb_price, check_pr_bb_winner
P3	feedback_count, feedback_rating, minimum_hours, listing_price, shipping_price, is_domestic, diff_mean, diff_fpt_rate_lowest, check_pr_bb_price, check_pr_bb_winner
P4	is_prime, feedback_count, listing_price, shipping_price, is_domestic, check_pr_bb_price, check_pr_bb_winner

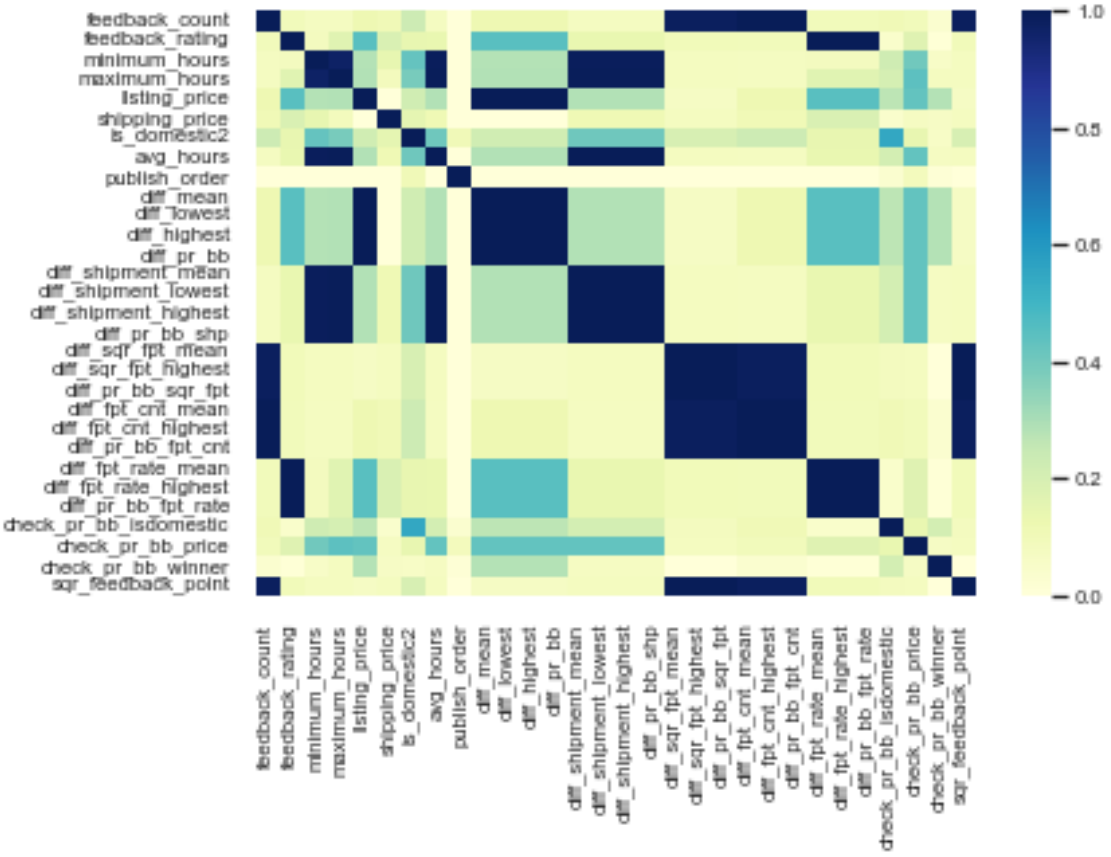

Figure 3.7 Correlation Matrix of P2

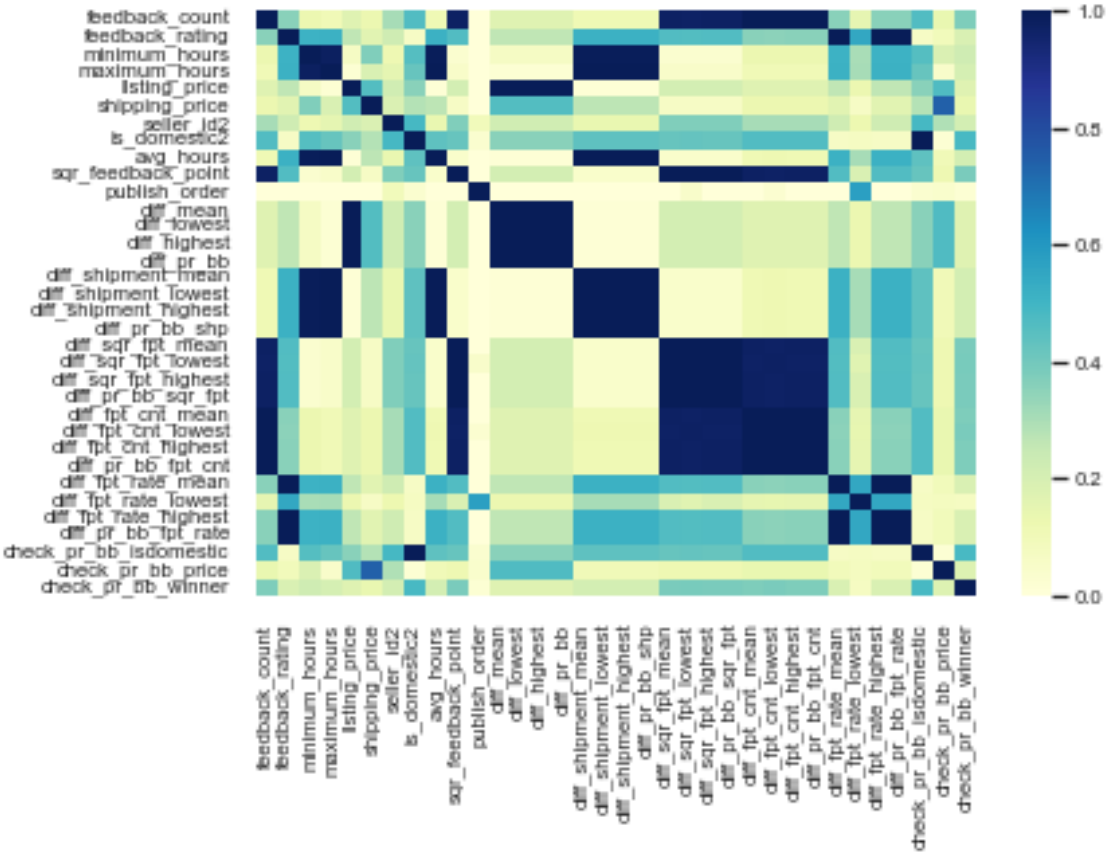

Figure 3.8 Correlation Matrix of P3

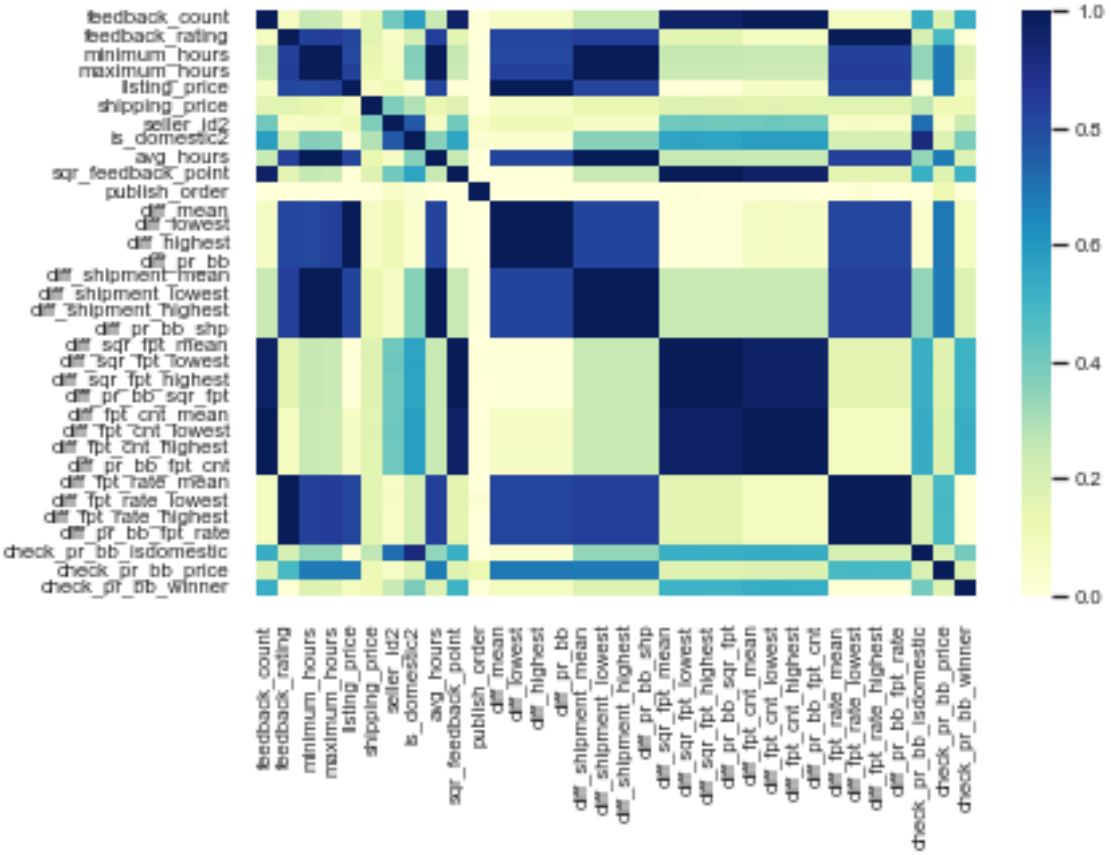

Figure 3.9 Correlation Matrix of P4

4. ANALYSIS AND PERFORMANCE EVALUATION

In this chapter, we are going to explain the details of the data analysis and discuss the results of the data analysis. Initially, we will explore the data partitioning process. Secondly, we are going to mention the machine learning algorithms that are selected. Thirdly, we will examine the hyperparameter tuning of the selected models. Furthermore, we will visit the best feature subset selection. Finally, we are going to evaluate the prediction performances of the selected models for each product.

4.1 Data Partitioning

Partitioning the data into different sets is a commonly used technique in machine learning. In general, the data is split into three sets which are the train, validation, and test data sets. Train data set is used to build the machine learning model where model parameters fit on the data set. The validation set is a separate data set from train data used to evaluate the model performance. Harrington (2018) proved that having only training and validation sets could also give a wrong estimation of model performance. The test set which is withheld from model training, allows us to evaluate model performance.

Theoretical and numerical investigations on the optimality of the data splitting ratio so far have not led to any consensus (Joseph, 2022). Picard & Berk (1990) have recommended 25%–50% for the testing set. Dobbin & Simon (2011), and Nguyen, Ly, Lanh, Al-Ansari, Le, Van Quan, Prakash & Pham (2021) have suggested that around a 30% test set ratio is a reasonable choice. Joseph (2022) states that the commonly used test ratio is 20%, which means 20% of data is split for testing while other ratios such as 30% to 50% of test ratio are also used in practice. The 20% split draws its justification from the well-known Pareto principle. Similar to the

common practice, we have also applied a 20% of test ratio for each product. In order to have the same size of test data, we have decided to create a 20% of the validation set. The remaining 60% of the data set has been used as train data to fit the models and determine hyperparameters of the models that have been selected. We have used validation data to mitigate a potential overfitting problem. Overfitting is a phenomenon often seen when a trained model performs extremely well on the samples used for training but performs poorly on new unknown samples; that is to say, the model does not generalize well (Xu & Goodacre, 2018). We have decided on the features which will be used to make predictions on the test data set by evaluating their contribution to the accuracy performance of the validation data set.

4.2 Machine Learning Algorithms

Our main goal is the prediction of the buy box winner among different sellers by using several numerical and binary features that characterize sellers as explained in the data section. In other words, we want to make a binary classification of the sellers where the buy box winner is in the positive class and the rest are in the negative class. Supervised machine learning is the construction of algorithms that can produce general patterns and hypotheses by using externally supplied instances to predict the fate of future instances. Supervised machine learning classification algorithms aim at categorizing data from prior information (Singh, Thakur & Sharma, 2016). To achieve our goal, we have decided to apply supervised machine learning classification algorithms

There are popular algorithms used for supervised classification problems in different research areas such as SVM, NN, LR, Decision Trees, RF, XGBoost, Naïve Bayesian, and K-neighbors (Hambarde et al., 2020). The first algorithm we have decided to use is RF for several reasons. Firstly, Chen et al. (2016), and Gómez-Losada & Duch-Brown (2019) both applied RF classifier on buy box winner prediction and received satisfactory results as explained in Section 2.2. Additionally, in the latter study, RF performed better than SVM and NN. Using the RF algorithm will provide us to compare the performance results and feature importance with these studies. Furthermore, RF provides some advantages such as avoiding the problem of overfitting (Pallathadka, Ramirez-Asis, Loli-Poma, Kaliyaperumal, Ventayen & Naved, 2021) and providing feature importance. XGBoost is the second algorithm

we have decided to apply. This algorithm has recently gained immense popularity especially due to its exceptional performance in Kaggle competitions (Kavzoglu & Teke, 2022). Additionally, this algorithm performed better or very close to the best performing algorithm in e-commerce studies such as Song & Liu (2020) Song and Liu (2020) and Hambarde et al. (2020). Similar to the RF, XGBoost also prevents overfitting issues (Kavzoglu & Teke, 2022) and provides feature importance. LightGBM is the third and the last algorithm we have used in this study. This algorithm is a relatively new algorithm with few reading resources, mostly used in online machine learning competitions for its good performance (Effrosynidis & Arampatzis, 2021). Like RF and XGBoost, LightGBM mitigates potential overfitting problems and provides feature importance. We wanted to include this algorithm to compare prediction performance with widely used RF and XGBoost algorithms and feature importance sets.

4.3 Hyperparameter Tuning

Building an effective machine learning model is a complex and time-consuming process that involves determining the appropriate algorithm and obtaining an optimal model architecture by tuning its hyperparameters (Shawi, Maher & Sakr, 2019). Two types of parameters exist in machine learning models: one that can be initialized and updated through the data learning process, named model parameters; while the other, named hyperparameters, cannot be directly estimated from data learning and must be set before training an ML model because they define the model architecture (Yang & Shami, 2020). Tuning hyper-parameters is considered a key component of building an effective ML model, especially for tree-based ML models which have many hyper-parameters (Feurer & Hutter, 2019)). Grid search, Random search, and Bayesian optimization are the 3 main hyperparameter tuning approaches. The user specifies a finite set of values for each hyperparameter, and grid search evaluates the Cartesian product of these sets. This approach suffers from the curse of dimensionality since the required number of function evaluations grows exponentially (Feurer & Hutter, 2019). Random search is a good alternative to grid search that searches samples from parameter sets randomly which works better than grid search when some hyperparameters are much more important than others (Bergstra & Bengio, 2012). Random search is a useful baseline because it makes no assumptions about the machine learning algorithm being optimized, and,

given enough resources, will, in expectation, achieves performance arbitrarily close to the optimum (Harrington, 2018). Lastly, Bayesian optimization is an iterative algorithm with two key ingredients: a probabilistic surrogate model and an acquisition function to decide which point to evaluate next (Joseph, 2022). We have decided to apply random search to tune hyperparameters for all three algorithms due to it is close to the optimum parameters and it provides results faster than other search algorithms.

4.3.1 Hyperparameter Tuning and Feature Importance

In this subsection, we are going to explore the hyperparameter selection of the models and analyze the feature importance of each model. We are going to start with RF hyperparameter tuning and feature importance and continue with XGBoost and LightGBM algorithms.

4.3.1.1 Random Forest Hyperparameter Tuning

RF is an ensemble model that uses multiple decision trees and is able to provide solutions to complex problems. Max_depth, min_sample_split, min_samples_leaf, n_estimators, and max_features are among the mostly used hyperparameters (Yang & Shami, 2020). Default values are used for other hyperparameters. The hyperparameter grid of RF has been provided in Table 4.1. There are no default best values of a hyperparameter grid set for an algorithm. The values in the grid are determined to provide enough range for randomized search while keeping the number of the values limited due to computational requirements. The definitions of the hyperparameters were collected from the documentation of the library of scikit-learn (scikit learn.org, 2022a) and interpreted.

- **max_depth:** It describes how can a tree in the forest can grow. The deeper the tree fits better the data set and provides more accurate the results.
- **min_samples_split:** It describes the minimum amount of samples that an internal node has to split further nodes. Increasing min_samples_split decrease the total number of split that prevents overfitting on the data set.
- **min_samples_leaf:** It describes the number of samples that a node must include after getting a split. Similar to the min_samples_split, increasing this

Table 4.1 RF Hyperparameter Grid

Hyperparameters	**Values**
max_depth	5, 10, 25, 50, 100
min_samples_split	2, 4, 6, 8, 10
min_samples_leaf	1, 2, 3, 4, 5, 6
n_estimators	100, 150, 200, 250, 300
max_features	2, 3, 4, 5, 6, 7, 8

Table 4.2 RF Best Estimators

	Category 1		**Category 2**		
Hyperparameter	**P1**	**P2**	**P3**	**P4**	**Avg**
max_depth	5	10	5	10	8
in_samples_split	6	2	6	6	5
in_samples_leaf	1	4	5	4	4
n_estimators	250	200	100	150	163
max_features	7	7	6	4	6

hyperparameter reduces the risk of overfitting the data set.

• **n_estimators:** It defines the number of trees in the random forest. Increasing this hyperparameter may result in more generalized models.

• **max_features:** It defines the subset of the features while searching for the best split. The maximum number of this hyperparameter is the number of features of the data set.

As a result of random search stratified 3-folds cross-validation, the best estimator values for each product are provided in Table 4.2. Some hyperparameter values are the same for different products such as max_depth is equal to 5 for P1 and P3. However, the set of hyperparameters for a product is different from another product. The different values of the hyperparameters reflect that each data set has its unique structure and should be evaluated separately.

4.3.1.2 Random Forest Feature Importance

In this subsection, the best estimators that have been derived via hyperparameter tuning fit to the train data set and feature importance of the algorithms have been provided. The feature importance for each product is provided in Table 4.3. check_pr_bb_winner is the most important feature for all data sets with an average of 0.74. listing_price is the second important feature for P1 and P2 while feedback_count is the second most important feature for P4 and is_domestic is

Table 4.3 RF Feature Importance

	Category 1		Category 2		
Feature Name	**P1**	**P2**	**P3**	**P4**	**Avg**
check_pr_bb_winner	0.64	0.86	0.83	0.64	0.74
feedback_count	0.09	0.02	0.04	0.19	0.08
listing_price	0.10	0.07	0.00	0.09	0.07
is_domestic	0.00	0.00	0.09	0.04	0.03
minimum_hours	0.07	0.01	0.00	0.00	0.02
feedback_rating	0.06	0.01	0.01	0.00	0.02
check_pr_bb_price	0.00	0.02	0.02	0.03	0.01
is_prime	0.05	0.00	0.00	0.00	0.01
shipping_price	0.01	0.00	0.00	0.01	0.01
diff_mean	0.00	0.00	0.01	0.00	0.00
is_lowest	0.00	0.00	0.00	0.00	0.00
check_pr_bb_is_domestic	0.00	0.00	0.00	0.00	0.00
diff_fpt_rate_lowest	0.00	0.00	0.00	0.00	0.00

the second important feature for P3. RF feature importance results reflect that a buy box winner is most likely to win the buy box again. Let's remember that at each price change, a new data set is collected. However, several features such as listing_price, feedback_count, and is_domestic are significant and affect the buy box winner. Sellers can compete to win the buy box by listing price and they can focus on customer feedback which is related to customer satisfaction.

4.3.1.3 XGBoost Hyperparameter Tuning

XGBoost is a highly efficient gradient boosting library that is able to solve many complex data science problems in a fast and accurate way. Similar to Random Forest, XGBoost uses decision trees and instead of fitting on a big decision tree, it fits on many small decision trees. learning_rate, gamma, max_depth, n_estimators, reg_alpha, and reg_lambda are mostly used XGBoost hyperparameters among many others (Yang & Shami, 2020). In addition to these hyperparameters, the model objective is set to binary due to the model predicting binary classes. Gamma, learning_rate, reg_alpha, and reg_lambda hyperparameters are different from random forest classifier hyperparameters and are explained below. Similar to RF, there is no best grid for XGBoost hyperparameters. We have provided a broad range of values for randomized search while limiting the number of different values due to computational costs. The hyperparameter grid of XGBooost is provided in Table 4.4. The definitions of the hyperparameters were collected from the documentation

Table 4.4 XGBoost Hyperparameter Grid

Hyperparameter	**Values**
gamma	0 ,0.1, 0.2, 0.4, 0.8, 1.6, 3.2, 6.4, 12.8, 25.6, 51.2, 102.4, 200
learning_rate	0.01, 0.03, 0.06, 0.1, 0.15, 0.2, 0.25, 0.30, 0.4, 0.5, 0.6, 0.7
max_depth	5, 6, 7, 8, 9, 10, 11, 12, 13, 14,15
n_estimators	50, 65, 80, 100, 115, 130, 150
reg_alpha	0, 0.1, 0.2, 0.4, 0.8, 1.6, 3.2, 6.4, 12.8, 25.6, 51.2, 102.4, 200
reg_lambda	0, 0.1, 0.2, 0.4, 0.8, 1.6, 3.2, 6.4, 12.8, 25.6, 51.2, 102.4, 200

Table 4.5 XGBoost Best Estimators

	Category 1		**Category 2**		
Hyperparameter	**P1**	**P2**	**P3**	**P4**	**Avg**
gamma	3.2	0.8	0.8	25.6	7.6
learning_rate	0.20	0.10	0.60	0.06	0.24
max_depth	14	10	14	10	12
n_estimators	115	65	115	115	103
reg_alpha	3.2	12.8	12.8	0.8	7.4
reg_lambda	200.0	0.8	51.2	1.6	63.4

of the XGBoost website (readthedocs.io, 2022b).

• **gamma**: It defines the minimum loss reduction to make a further split on a node of the tree. The smaller gamma reduces the overfitting problem and produces more generalized results.
• **learning_rate:** It is used to shrink the weights of the new features to make the boosting process more conservative. Using smaller learning_rate produce better results but it requires more boosting rounds.
• **reg_alpha:** It defines the L1 - Lasso regularization on weights of the model. The model becomes more conservative when this hyperparameter increased.
• **reg_lambda:** It defines the L2 - Ridge regularization on weights of the model. The model becomes more conservative when this hyperparameter increased.

As a result of random search stratified 3-folds cross-validation, the best estimator values of the XGBoost algorithm for each product are provided in Table 4.5. XGBoost algorithm has used different hyperparameter values for each product that allows fitting better on the training data set. Some hyperparameters are very distinct for different products. For instance, the learning_rate value of P3 is 0.6 while 0.06 for P4 although P3 and P4 are in the same category. Similarly, the reg_lambda hyperparameter value for P1 is 200 while it is 0.8 for P2. This situation confirms that each product should be evaluated individually.

Table 4.6 XGBoost Feature Importance

	Category 1		Category 2		
Feature Name	**P1**	**P2**	**P3**	**P4**	**Avg**
check_pr_bb_winner	0.67	0.93	0.85	0.81	0.81
is_prime	0.23	0.00	0.00	0.00	0.06
is_domestic	0.00	0.00	0.11	0.05	0.04
listing_price	0.06	0.02	0.00	0.02	0.02
feedback_count	0.00	0.01	0.02	0.06	0.02
check_pr_bb_price	0.00	0.01	0.01	0.06	0.02
minimum_hours	0.04	0.01	0.00	0.00	0.01
shipping_price	0.00	0.01	0.00	0.01	0.00
feedback_rating	0.00	0.01	0.00	0.00	0.00
is_lowest	0.00	0.01	0.00	0.00	0.00
diff_mean	0.00	0.00	0.00	0.00	0.00
diff_fpt_rate_lowest	0.00	0.00	0.00	0.00	0.00

4.3.1.4 XGBoost Feature Importance

The best estimators that have been derived via hyperparameter tuning fit to the train data set and feature importance of the XGBoost algorithms have been provided in Table 4.6. Similar to RF, check_pr_bb_winner is the most important feature among all products with an average of 0.81. is_prime is the second important feature for P1 but does not have any importance for other products that reflect there are Amazon prime sellers of P1 while there are not any prime sellers for other products. The second important feature varies among P2, P3, and P4 which are listing_price, is_domestic, and feedback_count accordingly. The feature important set of XGBoost is very similar to RF where check_pr_bb_winner is the most important feature and feedback_count, and is_domestic are among the other important features. is_domestic is the second important feature of P3 while it has not any importance for P1 and P2. In contrast to RF, listing_price is not among the top 3 important features except P1. Each product data set has a different order of feature importance.

4.3.1.5 LightGBM Hyperparameter Tuning

LightGBM is a gradient boosting framework that uses tree-based learning algorithms that provides faster training speed and higher efficiency and better accuracy. learning_rate, max_depth, n_estimators, and ma_bin hyperparameters are considered

Table 4.7 LightGBM Hyperparameter Grid

Hyperparameter	**Values**
learning_rate	0.01, 0.03, 0.06, 0.1, 0.15, 0.2, 0.25, 0.30, 0.4, 0.5, 0.6, 0.7
max_depth	5, 6, 7, 8, 9, 10, 11, 12, 13, 14
n_estimators	50, 65, 80, 100, 115, 130, 150
max_bin	100, 200, 250, 300, 400, 500
is_unbalance	True

Table 4.8 LightGBM Best Estimators

	Category 1		**Category 2**		
Hyperparameter	**P1**	**P2**	**P3**	**P4**	**Avg**
learning_rate	0.01	0.01	0.01	0.01	0.01
max_depth	13	7	11	5	9
n_estimators	80	80	80	130	93
max_bin	400	200	200	400	300

for tuning. is_unbalance hyperparameter is selected as True due to the there is only one positive class among several negative classes for each publish_time. Like other algorithms, there is no optimal grid for LightGBM. We have tried to provide a reasonable range for the randomized searches. The hyperparameter grid is provided in Table 4.7. The definitions of the hyperparameters were collected from the documentation of the LightGBM website (readthedocs.io, 2022a).

• **max_bin:** It defines the maximum number of bins that feature values will be bucketed. Increasing this number increases the accuracy while risking overfitting problem

• **is_unbalance:** It defines if the data is balanced or unbalanced on a classification model. Due to the data set having an unbalanced class, this hyperparameter was selected as True.

As a result of random search stratified 3-folds cross-validation, the best estimator values of the LightGBM algorithm for each product are provided in Table 4.8. LightGBM hyperparameters for different products are more alike in comparison to RF and XGBoost algorithms. For instance, the learning rate is 0.01 for all products and n_estimators is 80 for P1, P2, and P3. However, hyperparameter sets are different for each product. For instance, P2 and P3 have the same learning_rate, n_estimators, and max_bin values but max_depth values are different.

4.3.1.6 LightGBM Feature Importance

Table 4.9 LightGBM Feature Importance

	Category 1		Category 2		
Feature Name	**P1**	**P2**	**P3**	**P4**	**Avg**
listing_price	835	429	1140	882	822
feedback_count	683	468	800	564	629
feedback_rating	650	360	11	0	255
shipping_price	15	206	87	264	143
minimum_hours	227	308	26	0	140
check_pr_bb_winner	100	80	80	65	81
check_pr_bb_price	0	142	80	80	76
is_domestic	0	78	80	95	63
is_lowest	0	169	0	0	42
check_pr_bb_is domestic	0	160	0	0	40
diff_mean	0	0	96	0	24
is_prime	6	0	0	0	2
diff_fpt_rate_lowest	0	0	0	0	0

The best estimators that have been derived via hyperparameter tuning fit to the train data set and feature importance of the LightGBM algorithms have been provided in Table 4.9. listing_price and feedback_count are the most important two features for LightGBM. feedback_price is the third most important feature for P1 and P2 while it has among the least important features for P3 and P4. In comparison to other algorithms, check_pr_bb_winner is not among the LightGBM top important features for all products. The feature importance of a product is different from another product's feature importance that reflects each product has its unique competitional environment.

4.4 Subset Selection

In section 2.1, we have mentioned feature selection methods which are filters, wrappers, embedded, and ensembled. In subsection 3.3.2, we have applied a filter method which is the Pearson coefficient. Features with an absolute value of Pearson correlation coefficient higher than 0.8 were eliminated. In section 4.3 we have applied RF, XGBoost, and LightGBM algorithms and received feature importance for each product which is an ensemble method. In this section, we are going to add another step for feature selection to be able to get a more generalized algorithm that avoids overfitting. This step is very similar to a wrapper method which is forward

selection. Forward selection typically starts with an empty feature set and then considers adding one or more features to the set (Jović, Brkić & Bogunović, 2015). By applying this step, we have used filter, ensemble, and wrapper methods to select a subset of the features.

We have used the validation set for subset selection of the features based on the feature importance values calculated on the train data set for each model and product. Firstly, the best model is based on cross-validation and its hyperparameters retrieved from the train data set. Secondly, the most important feature is selected as if it is the only feature and accuracy value calculated predictions of the validation data set. The next important feature is added to the feature list, the model is run again to predict buy box winners of the valuation data set. At each step, the accuracy metric has been calculated until accuracy is not increased by an additional feature. The subset of the features that provide the highest accuracy has been used to predict the test data set while keeping hyperparameters. Using the validation data set to determine the best subset decreases the risk of overfitting on the train data set.

4.4.1 Subset selection for P1

The top 5 features in the feature importance list of RF contributed accuracy to the validation set while is_prime and shipping_price did not provide additional accuracy. Therefore, these two features did not include in the prediction feature set. Only 2 features which are check_pr_bb_winner, and is_prime features contributed to the accuracy set positively. Although listing_price and minimum_hours are in the feature importance list, these values are excluded from features to be used in prediction. shipping_price is excluded from the LightGBM algorithm feature set. The list of the features has contributed to the accuracy of the value set prediction listed in Table 4.10

4.4.2 Subset selection for P2

All features which have positive feature importance of RF have increased the prediction performance for the validation set. check_pr_bb_price, minimum_hours, and shipping_price discarded for the XGBoost algorithm feature set although they

Table 4.10 P1 Subset Selection

#	**Random Forest**
1	check_pr_ bb_winner
2	listing_price
3	feedback_count
4	minimum_hours
5	feedback_rating

#	**XGBoost**
1	check_pr_ bb_winner
2	is_prime

#	**LightGBM**
1	listing_price
2	feedback_count
3	feedback_rating
4	minimum_hours
5	check_pr_ bb_winner

Table 4.11 P2 Subset Selection

#	**Random Forest**
1	check_pr_ bb_winner
2	listing_price
3	feedback_count
4	minimum_hours
5	check_pr_ bb_price
6	feedback_rating
7	is_lowest

#	**XGBoost**
1	check_pr_ bb_winner
2	listing_price
3	feedback_count

#	**LightGBM**
1	feedback_count
2	listing_price
3	minimum_hours
4	shipping_price
5	check_pr_ bb_is_domestic
6	check_pr _bb_price
7	check_pr _bb_winner

have importance on the train data set. feedback_rating, is_lowest, and is_domestic features are not used for the prediction of the test set for the LightGBM algorithm which has importance in fitting to train data set. The list of the features has contributed to the accuracy of the value set prediction listed in Table 4.11.

4.4.3 Subset selection for P3

All features which have positive feature importance on the train data set of RF have a positive impact on the prediction performance of the validation set. check_pr_bb_price is excluded from the feature list to be used for the prediction of the test data for the XGBoost algorithm. minimum_hours, feedback_rating, is_domestic, and diff_mean features are not used for the prediction process of the LightGBM. The list of the features has contributed to the accuracy of the value set prediction listed in Table 4.12.

Table 4.12 P3 Subset Selection

#	**Random Forest**
1	check_pr_ bb_winner
2	is_domestic
3	feedback_count
4	check_pr_ bb_price
5	feedback_rating
6	diff_mean

#	**XGBoost**
1	check_pr_ bb_winner
2	is_domestic
3	feedback_count

#	**LightGBM**
1	listing_price
2	feedback_count
3	shipping_price
4	check_pr_ bb_winner

Table 4.13 P4 Subset Selection

#	**Random Forest**
1	check_pr_ bb_winner
2	feedback_count
3	listing_price
4	is_domestic

#	**XGBoost**
1	check_pr_ bb_winner
2	is_domestic
3	feedback_count

#	**LightGBM**
1	listing_price
2	feedback_count
3	shipping_price
4	is_domestic
5	check_pr_ bb_price
6	check_pr_ bb_winner

4.4.4 Subset selection for P4

check_pr_bb_price and shipping_price have importance in fitting to train data while they did not make any contribution to the accuracy of the prediction of valuation data set. listing_price, check_pr_bb_price, and shipping_price are excluded from the XGBoost feature set. All features which have positive feature importance on the train data set of RF have a positive impact on the prediction performance of the validation set. The list of the features has contributed to the accuracy of the value set prediction listed in Table 4.13

4.5 Performance Evaluation

In this section, we are going to explore the prediction performance metrics initially. Secondly, we are going to explore the prediction performance of each algorithm for

each product on the test set by using selected metrics. Additionally, we are going to compare the performance results of these algorithms.

4.5.1 Performance Metrics

Accuracy, precision, recall, f1 score, Area Under Curve (AUC) score, and Negative Predictive Value (NPV) have been considered to evaluate the prediction performance of the algorithms for each of the products. These metrics are widely used to evaluate machine learning classification performance. Chen et al. (2016), and Gómez-Losada & Duch-Brown (2019) used accuracy to test buy box prediction performance. Niu et al. (2017), Vanderveld et al. (2016), Hambarde et al. (2020), Sikdar et al. (2019) have used accuracy to test model prediction performance on test data set. Song & Liu (2020) used accuracy rate, precision rate, recall rate, and f1-score to test XG-Boost algorithm performance on the prediction purchase behavior of the customers on e-commerce. Boz, Günneç, Birbil & Öztürk (2018) used AUC score and NPV to evaluate the performance of loan application assessment. We are going to use accuracy metric to measure model overall performance. Additionally, we are going to use precision, recall, f1 score, and AUC to better understand the positive class prediction performance. Furthermore, we have used NPV to consider negative class performance. In our research, buy box winning condition labeled as positive and not winning condition labeled as negative. All metrics are calculated by using the scikit-learn metrics library except NPV.

Accuracy:
The accuracy metric is calculated as the ratio between the number of correct predictions to the total number of predictions.

$$accuracy = \frac{number\,of\,correct\,predictions}{total\,number\,of\,predictions} \tag{4.1}$$

Precision:
The precision metric is the ratio between the number of positive samples correctly classified to the total number of samples classified as positive. The precision measures the model's accuracy in classifying a sample as positive.

$$precision = \frac{number\,of\,True\,Positive}{number\,of\,(True\,Positive + False\,Positive)} \tag{4.2}$$

Recall:
Recall metric is the ratio between the number of positive samples correctly classified as positive to the total number of Positive samples. The recall measures the model's ability to detect positive samples.

$$recall = \frac{number\,of\,True\,Positive}{number\,of\,(True\,Positive + False\,Negative)} \tag{4.3}$$

F1 score:
F1 score is the harmonic average of the Precision and Recall metrics. Although it is possible to give different weights on precision and recall metrics, the same weight is used to calculate the F1 score.

$$f1\,score = \frac{2 * Precision * Recall}{Precision + Recall} \tag{4.4}$$

AUC score:
AUC score is calculated via the area under the ROC curve which visualizes the tradeoff between true positive rate and false positive rate. The higher AUC score represents better prediction. The maximum score can be 1 and a random prediction score is expected to be 0.5 for a binary classification problem.

NPV score:
NPV is defined as the number of true negative predictions divided by the total number negative predictions.

$$NPV = \frac{number\,of\,True\,Negative}{number\,of\,(True\,Negative + False\,Negative)} \tag{4.5}$$

4.5.2 Performance Results

All three models have been provided with over 97% accuracy for an unbalanced binary classification problem where there is an unequal distribution of buy box winners and other sellers. For P1, all performance metrics results of the algorithms are very close to each other. XGBoost algorithm has provided the highest results for all performance metrics with an accuracy of 97.87%, precision of 82.79%, recall 82.11%, and f1 score of 82.75% while RF has the second-highest value except recall. LightGBM has the second-highest recall with 81.3% whereas RF resulted in 80.34%. The highest accuracy number is provided for all algorithms with P2 data set where the XGBoost algorithm is 99.47%. RF provided the second-highest results for all metrics. XGBoost and LightGBM have provided exact scores for all metrics which is slightly higher than RF where accuracy is 97.37% for P3. For P4, XGBoost provided the highest scores for all metrics while LightGBM has the second place in terms of the highest accuracy. Shortly, the XGBoost algorithm has performed better than RF and LightGBM in terms of accuracy. LightGBM has performed better than RF for P3 while RF has a higher accuracy score for other products.

We had the highest scores for all performance metrics for P2. The range of accuracy varies between 97.37% with LightGBM for P3 and 99.47% with XGBoost for P2. Other metrics have a wider range of values. Precision varies between 80.65% with LightGBM for P1 and 95.71% with XGBoost for P2. Similarly, the recall has the lowest score of 80.34% for P1 with RF and the highest score of 95.71% for P2 with XGBoost. AUC score has the lowest value of 96.53% for P1 and has the highest value of 99.89% for P2. All NPV scores are higher than 98.52% for all products and algorithms. The performance metric results as percentages are provided in Table 4.5.1.

Each algorithm has different hyperparameters and the selected subsets of features are also different from each other regardless of the product and its category. This situation reveals that to predict a buy box seller with high scores, it is very important to focus on a product at a time. The competitive environment to win the buy box of a product is different from another one in terms of the listing price, sellers' properties such as being domestic, feedback count, etc. For example, being a prior buy box is an important feature to remain a buy box seller, but it does not guarantee the buy box position. To maintain it, there are other important features such as listing_price, feedback_count that keep a continuous competitive environment for all sellers. Additionally, we see that is_lowest feature is not among the most important features of any algorithms. This reflects that to win the buy box having the minimum price does not guarantee a seller the buy box position which is parallel to the findings of Gómez-Losada & Duch-Brown (2019), and Chen et al. (2016).

Table 4.14 Performance Results

#P	Algorithm	accuracy	precision	recall	f1 score	AUC	NPV
P1	RF	97.78	82.46	80.34	81.40	96.53	98.96
	XGBoost	97.87	82.79	82.11	82.45	96.63	98.89
	LightGBM	97.69	80.65	81.30	80.97	96.81	98.74
P2	RF	99.43	95.03	95.71	95.37	99.89	99.67
	XGBoost	99.47	95.71	95.71	95.71	99.66	99.72
	LightGBM	99.38	94.36	95.71	95.03	99.88	99.63
P3	RF	97.35	89.16	89.16	89.16	98.70	98.49
	XGBoost	97.37	89.31	89.24	89.24	98.36	98.52
	LightGBM	97.37	89.31	89.16	89.24	98.30	98.52
P4	RF	97.68	88.44	89.24	88.84	98.28	98.66
	XGBoost	98.08	93.00	88.79	90.90	98.79	99.20
	LightGBM	97.71	90.05	87.86	88.95	98.76	98.86

5. CONCLUSION AND DISCUSSION

This research investigates the mechanism of the Amazon buy box selection of a seller. We have collected data from Amazon AnyOfferChangedNotification API that provides seller and price information of the lowest 20 offers. The API provides new data set when a price has been changed by any of the sellers, namely publish time. The data set includes one month period, and we have focused on four products where the highest number of different buy box winners exist. We have used features provided by Amazon and built features based on the provided data set at publish time and the data set at the previous publish time. We have applied filter, ensemble, and wrapper methods to avoid high dimensionality and potential overfitting problem. We have benefited from Random Forest, XGBoost, and LightGBM algorithms to unveil the most important features to win the buy box and compared the prediction performance of these algorithms.

We have discovered that being the previous buy box winner is the most important feature for all products based on Random Forest and XGBoost algorithms while the LightGBM algorithm considers listing price and feedback count as the most important features. Listing price, feedback count, feedback rating, and being a domestic seller are among the other most important features of all algorithms. The algorithms resulted in different feature importance for each product. This situation reflects that there is a unique competition environment that includes different sellers and their characteristics. Therefore, to predict the buy box winner, different products should be modeled differently.

All algorithms provided an accuracy score of more than 97%. XGBoost algorithm has provided slightly higher accuracy than Random Forest and LightGBM for all of the 4 products. On average, the XGBoost algorithm resulted in 98.2% accuracy score while RF provided 98.06%, and LightGBM provided 98.04%. Similar to the accuracy, XGBoost generated slightly better results for precision, recall, and f1-score metrics with 90.2%, 88.96%, and 89.57% on average for the four products. These results suggest that although Amazon does not reveal the buy box mechanism, machine learning models can be used effectively to predict the buy box winners.

Application of the XGBoost and LightGBM algorithms are new to the Amazon buy box dynamics literature. Moreover, we have used data scaling for each publish time which allows us to use each row independent of the publish time. Furthermore, we have applied hyperparameter tuning for all algorithms for all product data sets. In addition to these, we have used validation data set to apply additional subset selection based on incremental prediction accuracy of each important feature for each algorithm. We had higher accuracy results from previous research for all product data sets.

There was some limitation to this research. We have used only one month of data. We think that a data set that covers a longer period may increase the performance scores. A longer data set will enable to train of a model with different conditions. Additionally, our data set retrieved from Amazon AnyOfferChangedNotification API provides the cheapest twenty offers. Therefore, the number of sellers is limited to twenty. For popular products, the number of different sellers may be more than twenty. A combination of Amazon API to determine when to retrieve data and web crawling can be used to get all sellers' data at publish time.

For further studies, a longer period of data that contains and is not limited to twenty sellers can be gathered for more products. Additionally, especially the listing price, and feedback count value optimization to win the buy box within a set of other sellers' data can be studied. This study will allow to buy box winner to understand how much more can the listing price be increased while keeping the buy box winning position to maximize profit. Also, other sellers can determine at which listing price value and feedback count they can win the buy box. In addition to these, additional hyperparameters of RF and XGBoost can be applied to manage the imbalanced data set. class_weight hyperparameter of RF and scale_pos_weight hyperparameter of XGBoost could be used for hyperparameter tuning.

BIBLIOGRAPHY

Allen, L. (2001). Amazon marketplace a winner for customers, sellers and industry; new service grows over 200 percent in first four months.

Bergstra, J. & Bengio, Y. (2012). Random search for hyper-parameter optimization. *The Journal of Machine Learning Research*, *13*, 281–305.

Boz, Z., Günneç, D., Birbil, I., & Öztürk, K. (2018). Reassessment and monitoring of loan applications with machine learning. *Applied Artificial Intelligence*, *32*, 1–17.

census.gov (2020). Quarterly retail e-commerce sales 1st quarter 2022.

Chen, L., Mislove, A., & Wilson, C. (2016). An empirical analysis of algorithmic pricing on amazon marketplace. (pp. 1339–1349).

Dobbin, K. & Simon, R. (2011). Optimally splitting cases for training and testing high dimensional classifiers. *BMC medical genomics*, *4*, 31.

Effrosynidis, D. & Arampatzis, A. (2021). An evaluation of feature selection methods for environmental data. *Ecological Informatics*, *61*, 101224.

Feurer, M. & Hutter, F. (2019). *Hyperparameter Optimization*, (pp. 3–33). Cham: Springer International Publishing.

Gartner (2022). Customer centricity.

Gómez-Losada, Á. & Duch-Brown, N. (2019). Competing for amazon's buy box: A machine-learning approach. In Abramowicz, W. & Corchuelo, R. (Eds.), *Business Information Systems Workshops*, (pp. 445–456). Springer International Publishing.

Guan, D., Yuan, W., Lee, Y.-K., Najeebullah, K., & Rasel, M. K. (2014). A review of ensemble learning based feature selection. *IETE Technical Review*, *31*(3), 190–198.

Hambarde, K., Silahtaroğlu, G., Khamitkar, S., Bhalchandra, P., Shaikh, H., Kulkarni, G., Tamsekar, P., & Samale, P. (2020). *Data Analytics Implemented over E-commerce Data to Evaluate Performance of Supervised Learning Approaches in Relation to Customer Behavior*, (pp. 285–293).

Harrington, P. (2018). Multiple versus single set validation of multivariate models to avoid mistakes. *Critical Reviews in Analytical Chemistry*, *48*(1), 33–46. PMID: 28777019.

Hartmans, A. (2021). Jeff bezos originally wanted to name amazon 'cadabra,' and 14 other little-known facts about the early days of the e-commerce giant.

Joseph, V. R. (2022). Optimal ratio for data splitting. *Statistical Analysis and Data Mining: The ASA Data Science Journal*.

Jović, A., Brkić, K., & Bogunović, N. (2015). A review of feature selection methods with applications. In *2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, (pp. 1200–1205).

Kavzoglu, T. & Teke, A. (2022). Predictive performances of ensemble machine learning algorithms in landslide susceptibility mapping using random forest, extreme gradient boosting (xgboost) and natural gradient boosting (ngboost). *ARABIAN JOURNAL FOR SCIENCE AND ENGINEERING*.

Lazar, C., Taminau, J., Meganck, S., Steenhoff, D., Coletta, A., Molter, C.,

de Schaetzen, V., Duque, R., Bersini, H., & Nowe, A. (2012). A survey on filter techniques for feature selection in gene expression microarray analysis. *IEEE/ACM transactions on computational biology and bioinformatics / IEEE, ACM*, *9*, 1106–19.

Lee, K. & Cha, S. (2002). Combining multiple feature selection methods.

Li, Q., Gu, M., Zhou, K., & Sun, X. (2015). Multi-classes feature engineering with sliding window for purchase prediction in mobile commerce. In *2015 IEEE International Conference on Data Mining Workshop (ICDMW)*, (pp. 1048–1054).

Meinshausen, N. & Bühlmann, P. (2010). Stability selection. *Journal of the Royal Statistical Society Series B*, *72*, 417–473.

Nguyen, Q., Ly, H.-B., Lanh, H., Al-Ansari, N., Le, H., Van Quan, T., Prakash, I., & Pham, B. (2021). Influence of data splitting on performance of machine learning models in prediction of shear strength of soil. *Mathematical Problems in Engineering*, *2021*.

Niu, X., Li, C., & Yu, X. (2017). Predictive analytics of e-commerce search behavior for conversion. In *AMCIS*.

Pallathadka, H., Ramirez-Asis, E. H., Loli-Poma, T. P., Kaliyaperumal, K., Ventayen, R. J. M., & Naved, M. (2021). Applications of artificial intelligence in business management, e-commerce and finance. *Materials Today: Proceedings*.

Picard, R. R. & Berk, K. N. (1990). Data splitting. *The American Statistician*, *44*(2), 140–147.

readthedocs.io (2022a). Lightgbm parameters.

readthedocs.io (2022b). Xgboost documentation.

scikit learn.org (2022a). sklearn.ensemble.randomforestclassifier.

scikit learn.org (2022b). sklearn.preprocessing.standardscaler.

sec.gov (2005). Letter to shareholders.

sellercentral.amazon.com (2022). How the buy box works.

Shawi, R. E., Maher, M., & Sakr, S. (2019). Automated machine learning: State-of-the-art and open challenges. *ArXiv*, *abs/1906.02287*.

Sikdar, S., Kadiyali, V., & Hooker, G. (2019). Price dynamics on amazon marketplace: A multivariate random forest variable selection approach. *ERN: Other Microeconomics: Production*.

Singh, A., Thakur, N., & Sharma, A. (2016). A review of supervised machine learning algorithms. In *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*, (pp. 1310–1315).

Song, P. & Liu, Y. (2020). An xgboost algorithm for predicting purchasing behaviour on e-commerce platforms. *Tehnički vjesnik*, *27*(5), 1467–1471.

Statista (2022). Top online stores worldwide in 2020, by e-commerce net sales.

Vanaman, D. (2022). Win the amazon buy box in 2022.

Vanderveld, A., Pandey, A., Han, A., & Parekh, R. (2016). An engagement-based customer lifetime value system for e-commerce. KDD '16, (pp. 293–302)., New York, NY, USA. Association for Computing Machinery.

Xu, Y. & Goodacre, R. (2018). On splitting training and validation set: A comparative study of cross-validation, bootstrap and systematic sampling for estimating the generalization performance of supervised learning. *Journal of Analysis and Testing*, *2*.

Yang, L. & Shami, A. (2020). On hyperparameter optimization of machine learning

algorithms: Theory and practice. *Neurocomputing*, *415*, 295–316.