Technical Report

MACHINE LEARNING MODELS FOR DYNAMIC PRODUCT REPRICING ON THE AMAZON MARKETPLACE

March 30, 2018

Thanh-Binh Le, Georgiana Ifrim
Insight Centre for Data Analytics, University College Dublin, Ireland
thanh.binh@insight-centre.org
georgiana.ifrim@insight-centre.org

Abstract

E-commerce has grown hugely and has opened opportunities for retailers big and small to sell their products on on-line shopping platforms such as the Amazon Marketplace. The Amazon platform hosts thousands of retailers and uses an algorithmic decision process called the BuyBox, to decide which product offer to show first in return to a user query.

Every time the price of a product changes, an auction among all the competing offers for that product takes place, and the BuyBox winner is re-assigned by Amazon. The BuyBox winner typically sells more products and achieves a higher revenue. The price of a product plays a large role, but it is not the only indicator of an offer winning the BuyBox. If we can understand and accurately approximate the BuyBox assignment algorithm, we can advise sellers about how to best customize their offers. We can also employ the same algorithm for dynamic repricing of products, to increase the likelihood of an offer winning the BuyBox, in the context of continuous change in competing offers.

Most existing solutions for predicting the BuyBox and for product repricing are closed commercial solutions. In this study we analyse historical data describing thousands of Amazon BuyBox auctions and build machine learning models that aim to approximate the Amazon algorithm for selecting winners. We also investigate approaches for algorithmic repricing based on (1) our BuyBox prediction algorithm and (2) efficient price point search strategies. Our evaluation shows that our learning models can accurately predict the BuyBox winner and can recommend effective repricing strategies.

Contents

Chapter 1

Introduction

1.1 Dynamic Repricing

Nowadays, the rise of e-commerce has opened many opportunities and challenges for merchants to sell their products and re-act their selling immediately. The on-line marketplace such as Amazon, provide many supporting tools to sellers that they can supervise their products at any given point of time. On the one hand, Amazon helps the merchants to adapt their product's price effectively in order to gain their profit. On the other hand, it also increases much pressures to the retailers, who have limited experience with such highly competitive markets and their long-term effects. To deal with that concern, the sellers use dynamic pricing tools to adjust product prices and manage inventories in real-time.

Dynamic repricing, also called dynamic pricing, is an approach to determine an optimal selling price for a product or service that is highly flexible. The aim of repricing algorithm is to allow a customer to settle their prices effectively to gain their revenue or to achieve a special goal (e.g. win the Amazon BuyBox) in a very competing marketplace. Nowadays, dynamic pricing has been used widely in various on-line platforms and in some cases it has considered to be an essential part of pricing policies.

At the core of a dynamic repricing technique is a machine learning model, which is built from the past data of retailer's selling behavior. This data may contain important insights on how the seller respond to different selling prices. Exploiting the knowledge contained inside the data and applying that to a repricing model may have a competitive advantage. This consideration is a main driver of research on dynamic repricing: the study of machine learning model, based on the past behavior of selling data, to find an optimized prices for customer's items in order to increase their profit.

1.2 Dynamic Product Repricing on the Amazon Marketplace

The Amazon is the largest and fastest growing retailer marketplace, with more than 80 million worldwide members in 2017 selling products on this platform (refered to *"Global Powers of Retailing 2017"* report[1]). Amazon constantly ranks the sellers based on different attributes, such as product price, customer satisfaction, amount of transactions completed, etc., and presents the best ranked seller/offer in the BuyBox. Pricing is the the most influential factor short-term to rank at the top, but the other attributes describing the seller and the offer are also important (e.g., shipping time, stock available). Existing repricing solutions use seller-provided rules for updating the product price when a repricing event is triggered (e.g., update price by 1% when a competitor changes the price of a product). These fixed rules are set manually and changed infrequently based on a schedule decided by the seller.

A Machine Learning solution for dynamic repricing can take full advantage of past detailed historical data about competing offers and the outcome of auctions, and should remove the need for slow to update and potentially non-optimal manual-rules. This can result in optimised pricing decisions for the sellers which should rank them top on the sales platform consistently, therefore laying the foundation for increased sales and better competitiveness. Additionally, the insights derived from analysing the historical data and the resulting predictive models should allow sellers to adapt to ever changing market conditions beyond pricing.

In this work we have access to large amounts of auction data for products sold over the course of 9 months on the Amazon Marketplace. Our aim is to develop machine learning algorithms to predict the BuyBox winner, as well as study dynamic repricing strategies for individual sellers.

Our key contributions are as follows:

- **Data Preparation Techniques:** We present data pre-processing and preparation techniques suitable for product auction data collected from the Amazon Marketplace, with a view to build effective BuyBox prediction algorithms.
- **BuyBox Predictor Algorithm:** We propose a machine learning algorithm that can be trained on historical product auction data from the Amazon Marketplace,

[1] `https://www2.deloitte.com/content/dam/Deloitte/global/Documents/consumer-industrial-products/gx-cip-2017-global-powers-of-retailing.pdf`

and can be used to predict the winner of a new auction on the Amazon Marketplace (i.e., BuyBox winner predictor). Our algorithm is based on a RandomForest classifier that uses carefully engineered features. We also discuss the importance of different features for predicting the BuyBox.

- **Repricer Algorithm:** We implement and evaluate a dynamic repricing algorithm that can take in an auction for a given product and seller, and recommend a repricing strategy to maximise the probability of winning the next auction for that seller and product. Our algorithm is based on the BuyBox predictor model and on a strategy for selecting candidate price points for recommendation.
- **Evaluation/Deployment:** All our algorithms are tested on research benchmarks and are deployed on commercial platforms. We discuss the results and what we learn from deploying our algorithms.

Chapter 2

BuyBox Prediction

This section provides a machine learning algorithm which can take as input any dataset describing the offers made by sellers for a set of products, for a specific market (e.g., US, UK, France). It then trains a machine learning model that, when presented with an auction, it can predict the offer that will win the auction (BuyBox). The algorithm also outputs a ranked list of the feature importance discovered from the training data. For example, the algorithm can discover data-specific feature importance for each input dataset. This means that the algorithm can be targeted to the data of a single seller or product, and it delivers a list of feature importance for that customized data. So besides predicting the auction winner, the algorithm can be used to advise sellers on how best to update their offer profiles to increase their chance of winning the BuyBox.

2.1 Business Understanding

Amazon hosts thousands of sellers on their marketplace. They use an algorithmic decision process to decide the winner of every auction. An auction as mentioned above is a set of merchants that compete to sell a product. Amazon selects a winner for the auction, and place that seller's product into the BuyBox. The BuyBox winner typically sells more products, and ideally achieves a higher profit margin. This means that the winner is highly competed by many retailers in auction.

Motivated by that, our target is clearly declared as: can we use historical data about auctions and knowledge about the winners and losers, to learn the rules by which Amazon decides the winners? If we can approximate Amazon's algorithm, we can advise sellers about the best strategy to use for winning the BuyBox. For example, we can recommend a new price for the product, or give advice about shipping time and user feedback profile

to improve the offer and increase the chance of winning the BuyBox.

2.2 Data Understanding

As mentioned above, every changing for a seller's offer will provide an auction, in Amazon's cloud service. It includes aggregated information about the 20 lowest prices offered for a product (or less, if there are less than 20 sellers). Each auction is represented as a XML, the total size of raw-data folder, which is a cross-market database, is about 100.000 files.

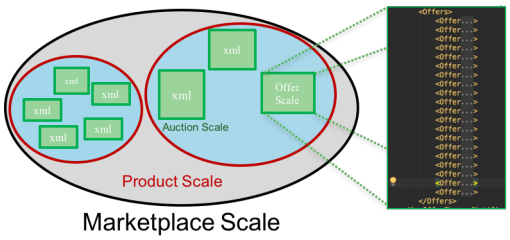


Figure 2.1: The scales of data in one Marketplace.

Illustrating from **Fig.2.1**,which is the scaling scheme of data in one market, there are four basic layer for the Amazon data:

1. **Marketplace layer**: The Amazon has many marketplaces for the United States, Australia, Brazil, Canada, China, France, Germany, India, Italy, Japan, Mexico, Netherlands, Spain, and the United Kingdom. One marketplace is an separated environment with its own characteristics. e.g. in India market, there is one seller who wins almost product's competing, while in U.S. market, the auction is normally with two or three competitors.

2. **Product layer**: From the market place, sellers can sell many products. It could be very different in price, shipping time, conditional note for two different products. The category list of product can be found in the Amazon Web Service.

3. **Auction layer**: In product's layer, the auction can be recorded when sellers update their product's offers. For example, one seller changes their shipping time from 0 hour to 24 hours, the new auction is saved as one XML file.

4. **Offers layer**: There are many sellers give their offer for a product in marketplace. However, only the 20 lowest price offers are saved in XML when an auction is happened.

Fig.2.2 illustrates small vision of one CSV file which has four sellers, each in its own row. Observing from this figure, there are some very important features such

as *IsBuyBoxWinner, ListingPrice, ShippingPrice, ShippingTime_maxHours, ShippingTime_minHours, IsFulfilledByAmazon, ...*

IsBuyBoxWinner	MarketplaceId	ConditionNotes	IsFeaturedMerchant	IsFulfilledByAmazon	ListingPrice	ShippingPrice	ShippingTime_maxHours	ShippingTime_minHours	ShippingTime_availtype	ShipsDomestically	SellerFeedbackRating	SellerFeedbackCount
1	FRANCE	0	1	1	11.94	0	0	0	NOW	1	92	38
-1	FRANCE	0	1	0	27.72	2.08	72	48	NOW	1	100	1
-1	FRANCE	0	0	0	5.99	7.99	48	24	NOW	1	67	3
-1	FRANCE	1	0	0	9.99	6.99	48	24	NOW	1	100	1
-1	FRANCE	0	1	1	16.98	0	0	0	NOW	1	93	14
-1	FRANCE	0	1	0	6.64	0	48	24	NOW	1	90	231
1	FRANCE	0	1	1	15.98	0	0	0	NOW	1	94	47

Figure 2.2: An example of Amazon's data with 7 samples.

2.3 Data Preparation

2.3.1 Raw Data

To facilitate the analysis, we model the BuyBox as a prediction problem. Specifically, for a product offered by n sellers, each of which is characterized by a feature vector, our goal is to predict which seller will be chosen to get in the BuyBox.

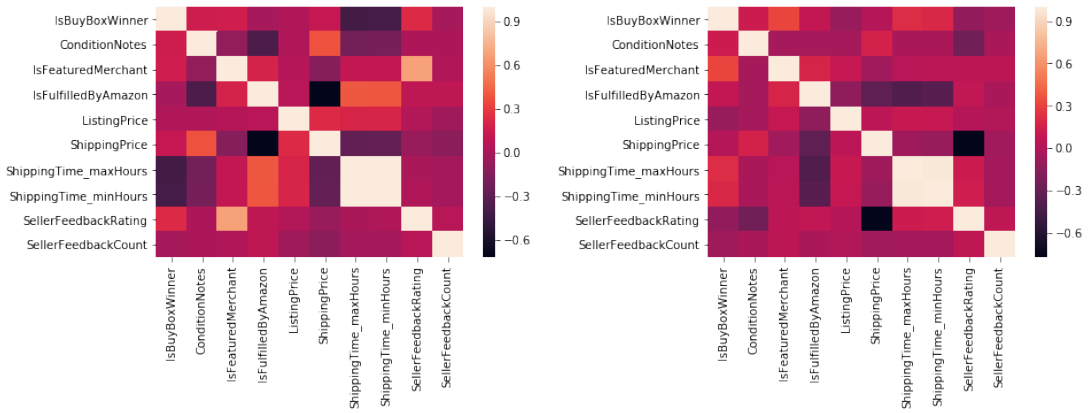
First step is to parse the XML raw data files into a single CSV dataset that can be use for further analysis. In order to do this, the parsing function is provided to read every single offer in a XML and transform it into a row in CSV file. Each child-tag under the tag <**Offer**> in the XML file is converted as a feature in dataset. Finally, the target class feature is parsed through the tag with name <**IsBuyBoxWinner**> (which is 1 if the offer wins BuyBox, -1 otherwise). Since each XML file has up to 20 offers, this means that for each auction we generate up to 20 rows in the CSV file.

The feature vector is described into four categories as follows :

A. Prices: The price's features are related to the price of products, which customers have to pay for buying a product. They are the *ListingPrice* and *ShippingPrice*. In addition, the new feature $LanddedPrice = ListingPrice + ShippingPrice$ is also calculated.

B. Shipping Time Informations: These are the shipping details for one seller's product, including the *ShippingTime_minHours*, *ShippingTime_maxHours* for delivery.

C. Seller Feedback Information: These features describe the detail of seller's feedback, including feedback's counts (*SellerFeedbackCounts*) and feedback's rating (*SellerFeedbackRating*).



(a) The U.S. *"amazon.com"* market **(b)** The U.K. *"amazon.co.uk"* market

Figure 2.3: The correlation between features in (a) U.S. market and (b) U.K. market.

D. Retailers' Details: These features are the basic detail of sellers when they have a cooperate to Amazon. These features denote whether the seller is fulfilled by Amazon (*IsFulfiledByAmazon*) or by merchant (*IsFeaturedMerchant*). The last feature is the product's condition notes *ConditionNotes* from sellers to their buyers.

2.3.2 Data Analysis

After parsing data into CSV format, the features are analyzed to help us having a clear understanding. Our first concern is about whether we can use data from cross-market to learn model. By observation **Fig.2.3**, it clearly shows that we should not train model with cross-market data. The correlations between features of two markets are really different, e.g. the possibility to be in BuyBox is higher if we have smaller shipping times in U.S. market, while it is not a really strong concerned effect in U.K. market. Hence, the separated treatment for each marketplace is necessarily provided here.

In addition, we create some new columns to enrich the information of an auction. These features are described as follows:

1. Difference to Minimum Price in Auction: (*DifftoMinLandedPriceCompetition*)This is the difference from current LandedPrice to the minimum landed price in that auction.

2. Difference to the Minimum Price of Product: (*DifftoMinLandedPriceProduct*) This is the difference from current LandedPrice to the minimum landed price, grouped by product.

3. Difference to the Amazon Seller's Price: (*DifftoAmzLandedPriceCompe-*

tition) We capture who is the Amazon Seller in the auction. Then, we calculate the difference from current LandedPrice to the Amazon Price. If there is not Amazon Seller in the auction, we use the difference to minimum price in the auction instead.

4. Difference to the Ideal Point in Auction: (*IdealPointCompetition*) The ideal point is the combination between the best (i.e., minimum) LandedPrice and the best (i.e., minimum) ShippingTime_maxHours, across all offers, in each auction. This feature captures the difference from this ideal point, for each offer in the auction.

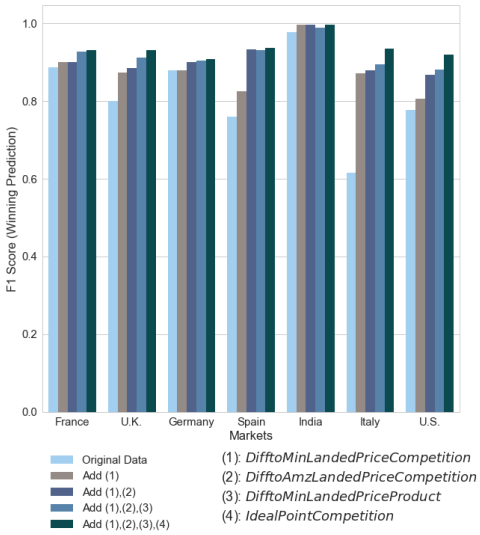


Figure 2.4: A comparison of F1-score when predicting winner BuyBox for 7 marketplaces. The scores are provided by Random Forest classifier.

In order to check the model's improvement capability for the new features, we add those columns one after another and compare them by using F1-scores of Random Forest Classifier. **Fig.2.4** illustrates that the prediction becomes significantly better when adding the extra information. This upgrading also points that the new features can enrich the Amazon data and help to build a better hypothesis.

2.4 Machine Learning Model for Predicting BuyBox Winner

In this section, we introduce the model construction with BuyBox predictor. The goal of this algorithm is to predict the probability to win BuyBox. Firstly, the XMLs are converted into one single featured CSV data, based on the feature importance analysis.

After parsing, we learn a machine learning model, which can help user to estimate the BuyBox winning probability. The **Fig.2.5** shows the flowchart of BuyBox prediction algorithm.

Figure 2.5: The flowchart of BuyBox model for winner prediction using feature importance to rank and select best features.

From the flowchart, it obviously shows that the BuyBox model is constructed by using the past behavior data of customer selling. Hence, if the past data is out-of-date, the model is not appropriate anymore. In order to make the model more accurate and adapted to the current time of business, the model should be retrained based on the updated featured data.

2.5 Experiment for Predicting BuyBox Winner

2.5.1 Comparison between Classification Models

Markets (Sample size)	Class	R.Forest		L.Regression		3-NN		AdaBoost		SVM-RBF		XGBoost	
		mean	std	mean	std	mean	std	mean	std	mean	std	mean	std
France	-1	1.000	0.000	0.990	0.000	0.990	0.000	0.990	0.000	0.990	0.000	0.990	0.000
(475,612)	1	0.960	0.004	0.800	0.000	0.860	0.015	0.870	0.005	0.830	0.011	0.900	0.000
UK	-1	0.980	0.019	0.935	0.013	0.950	0.011	0.905	0.011	0.915	0.031	0.980	0.012
(86)	1	0.970	0.022	0.915	0.019	0.940	0.011	0.880	0.014	0.895	0.045	0.970	0.018
Germany	-1	0.930	0.023	0.940	0.014	0.945	0.024	0.940	0.012	0.930	0.016	0.940	0.012
(113)	1	0.895	0.027	0.910	0.021	0.915	0.031	0.915	0.017	0.905	0.020	0.905	0.017
Spain	-1	0.955	0.028	0.840	0.004	0.925	0.041	0.795	0.021	0.825	0.020	0.960	0.023
(150)	1	0.925	0.037	0.735	0.024	0.890	0.054	0.665	0.025	0.715	0.050	0.935	0.040
India	-1	1.000	0.000	1.000	0.000	1.000	0.000	1.000	0.000	1.000	0.000	1.000	0.000
(36)	1	1.000	0.000	1.000	0.000	1.000	0.000	1.000	0.000	1.000	0.000	1.000	0.000
Italy	-1	0.990	0.000	0.960	0.005	0.970	0.005	0.980	0.000	0.970	0.000	0.990	0.004
(12,506)	1	0.960	0.007	0.830	0.018	0.850	0.013	0.890	0.004	0.850	0.005	0.930	0.007
US	-1	0.950	0.005	0.940	0.007	0.940	0.005	0.940	0.004	0.940	0.005	0.950	0.007
(3,200)	1	0.910	0.010	0.880	0.009	0.890	0.008	0.885	0.007	0.885	0.010	0.900	0.011

Table 2.1: The comparison between 6 classification algorithms for BuyBox prediction through 7 markets.

2.5.2 The Influence of Data Amount to Model

Markets	Class	The Increment of Data Size								
		10%	20%	30%	40%	50%	60%	70%	80%	90%
France	-1	0.99	0.99	0.99	0.99	0.99	1.00	1.00	1.00	1.00
	1	0.82	0.83	0.86	0.88	0.91	0.92	0.94	0.95	0.94
UK	-1	0.97	0.96	0.96	0.96	0.95	0.96	0.96	0.95	0.96
	1	0.96	0.94	0.94	0.94	0.92	0.94	0.94	0.93	0.95
Germany	-1	0.91	0.92	0.95	0.92	0.92	0.93	0.93	0.93	0.93
	1	0.88	0.89	0.93	0.89	0.89	0.91	0.91	0.91	0.9
Spain	-1	0.94	0.89	0.94	0.96	0.95	0.96	0.96	0.96	0.96
	1	0.9	0.83	0.9	0.93	0.92	0.93	0.93	0.94	0.94
India	-1	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	1	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Italy	-1	0.98	0.98	0.99	0.99	0.99	0.99	0.99	0.99	0.99
	1	0.9	0.91	0.94	0.94	0.95	0.96	0.95	0.96	0.96
US	-1	0.94	0.94	0.94	0.95	0.95	0.94	0.95	0.95	0.95
	1	0.88	0.89	0.89	0.90	0.90	0.89	0.90	0.90	0.90

Table 2.2: The xxxx .

Chapter 3

Repricer Algorithm

3.1 Problem Understading

One of the main inputs influencing who becomes the auction winner is the price offered by a seller for the product being auctioned. Product prices change dynamically on the Amazon platform, depending on the real-time competition for selling that product (i.e., the competing offers) and manual repricing rules decided by the seller. We aim to automate the repricing strategy to adapt to observed changes in auction behaviour. The repricing strategy helps to maximise the probability of winning the next auction for the given seller/customer and product.

The problem to be solved is: can we use our BuyBox predictor algorithm to dynamically recommend a product price to a given seller (i.e., a dynamic repricing strategy), for a given auction?

This section provides a dynamic repricing tool that uses BuyBox prediction model in Section 2. The repricing algorithm takes in data describing an auction for a given product and a given seller on the Amazon Marketplace, the BuyBox predictor model trained to predict the winner of any auction (a BuyBox winner predictor algorithm) and the data used to train that model. As output, it produces a few candidate price points for the given customer in the auction, ranks the price candidates and selects the top ranked price as a new price recommendation for that customer.

3.2 Spliting the Price's Bins

The first problem we have to solve is how to generate the recommended candidate prices.

3.3 Finding Recommendation Score

3.4 Proposed Algorithm for Repricer

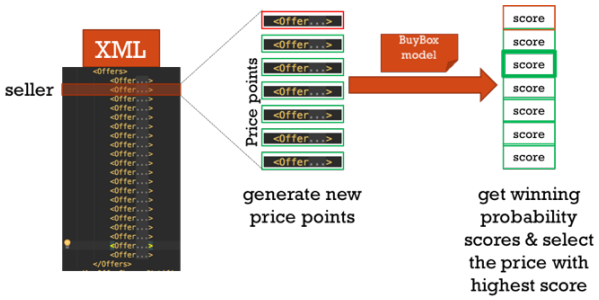


Figure 3.1: The flowchart of repricing strategy for one specific seller.

We provide here the repricing strategy to generate the predictive move of repricing. Our concentration is on the probability of what point of price we can get into buy box with the highest chance we can do. The strategy is defined into three steps:

1) First step is the getting and splitting. We get the point of our customer, who want to make a repricing. After receive the customer's price, we generate many bins of price, which we would like to test the probability.

2) Second step is the applying model and generating prediction score. On this step, we apply the Random Forest model, which is learned in BuyBox Predictor. Then we calculate the final scores of all price's points.

3) Final step is selecting. The top 10 largest scores, which has the highest probability to occupy the buy box, is provided as recommended price points.

3.5 Experiment for Repricing

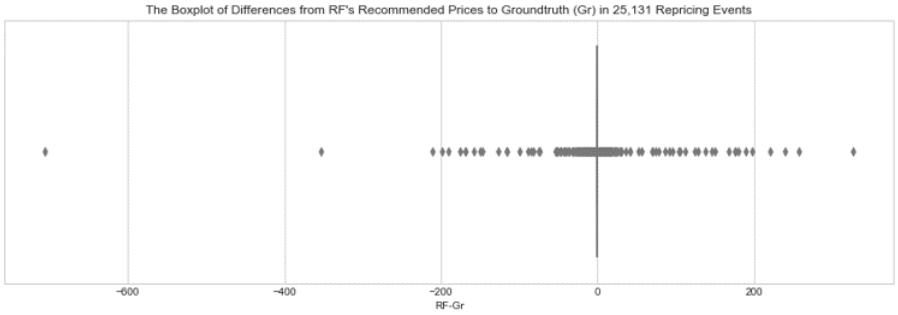


Figure 3.2: The Box plot of Prediction for Winning Cases.

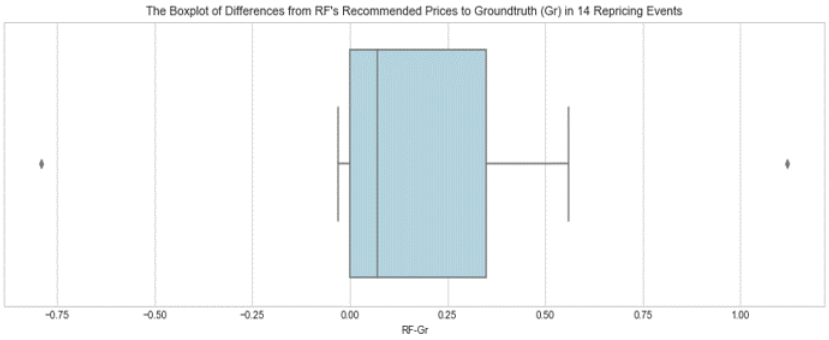


Figure 3.3: The Box plot of Prediction for Losing Cases.

Chapter 4

Conclusion

References

[1] Jodi Kantor and David Streitfeld. Inside amazon: Wrestling big ideas in a bruising workplace. *New York Times*, 15:74–80, 2015.

[2] DK Taft. Amazon buy box: The internet's $80 billion sales button. eweek, October 2014.

[3] Amazon.com. Increase your chances of winning the buy box., 2017. "Online; accessed 10-12-2017".