Predicting Sales Allocation to Third-Party Merchants

Leon Musolff
Princeton University
lmusolff@princeton.edu

Abstract

As an increasing proportion of marketplace transactions take place online, the power that online platforms such as Amazon or eBay wield when deciding which seller to match to a buyer is increasing. This increase in power may be concerning from a regulators' point-of-view if the power is being abused to favour the platforms' own listings. We examines what determines whether or not a merchant is assigned to the coveted 'Buybox' spot on Amazon and find that the relative price is the most important factor. However, Amazon also takes into account measures of seller quality as well as shipping time. We also find some evidence of Amazon giving itself or merchants that employ its in-house fulfilment service (FBA) an advantage.

1 Introduction

As the economy digitizes, consumers make an increasing share of their purchases on online marketplaces. As there are often multiple suppliers for the same product, owners of platforms like Amazon or eBay have a large amount of power in deciding how to match a consumer with a supplier. As consumers are only boundedly rational, this is true even when they are (theoretically) given the ability to order from whichever supplier they prefer: e.g. as we discuss below, consumers rarely buy outside the 'Buybox' on Amazon even though doing so is (literally) just one click away.

This paper investigates this matching process between consumers and suppliers on one of the largest e-commerce websites worldwide using extensive proprietary data from a repricing company. We find that Amazon takes into account both relative prices and measures of seller quality and shipping time when assigning merchants' to the coveted 'Buybox' spot. Crucially, we also find results that could suggest Amazon is giving itself or merchants that employ its in-house fulfilment service (FBA) an advantage. However, this result needs to be taken with a grain of salt as we presumably do not observe all variables that Amazon observes.

1.1 Setting

With a total revenue of $177 billion in 2017, Amazon is one of the largest e-commerce platforms worldwide [1]. Its success is built partly on a strategy of allowing third-party sellers to list offers right alongside Amazon's own offers on their website: in 2017, more than half of units sold were from third-party sellers [1].

Amazon's web presence is organized around the concept of a (narrowly-defined) product: sellers do not create their own separate listings for the same product (as they would on, e.g. eBay) but instead their offers for a product are pooled together on a product page. This pooling is heavily enforced by Amazon by requiring all products sold to be listed under their UPC code; listing a product without a UPC code is not allowed.

As far as customers are concerned, the fact that there exist multiple offers for most products often goes ignored: about 80% of purchases go through the so-called *Buybox*, the framed section of

Figure 1: How Offers Are Depicted on Product Page

(a) The Buybox

(b) The Featured Merchants Section

the product page depicted in *Figure 1a* which prominently displays 'Buy Now' and 'Add to Cart' buttons. The remaining 20% of sales click the link to the full offer listing page displayed in *Figure 1b* or choose directly from the featured merchants section displayed in the same image.

Amazon selects which merchant 'owns' the Buybox on the basis of a proprietary algorithm. This paper will attempt to (partially) reverse engineer this proprietary algorithm. The question of how Amazon allocates sales using this algorithm is of interest in the empirical market design literature in economics: would a social planner design the algorithm differently to maximize welfare? Or are Amazon's incentives sufficiently aligned with both those of merchants and consumers that its design choices are a reasonable approximation to the first best?

1.2 Prior Work

To the best knowledge of the authors, the only prior work that has attempted to reverse-engineer the Amazon Buybox algorithm from an academic perspective is [2], which employs data scraped from best-selling products on Amazon. We perform a detailed comparison of our results with those of [2] below, where we find the main takeaway that the quality of our data (which contains more covariates, is less or at least differently selected, and contains many more observations) allows us to easily outperform [2] in terms of classification performance. Furthermore, while our results are consistent with [2] in that we find that it seems to matter whether or not an offer is sold by Amazon itself, if we trust the random forest feature importances it matters substantially less than [2] find once we control for shipping speed.

2 Discussion of Data

This analysis will employ proprietary data from a repricing company that offers its services to Amazon third-party sellers. The repricing company manages the offer listings on behalf of its customers, which allows it to register with Amazon to receive so-called *AnyOfferChanged* notifications: these notifications are sent to each company administering an offer listing on a product within seconds if any of the offer listings on this product changes price. They contain the price as well as some other information about the twenty lowest-priced offers on the product (ties are broken randomly).

Crucially, the notifications contain (i) an indicator for which offer (if any) has been allocated to the Buybox and (ii) several variables that we suspect Amazon uses in allocating the Buybox (e.g. price, shipping cost, shipping time and merchant ratings). To be exact, we have access to the following covariates at the offer level:

- the time at which the notification was sent,
- merchant, product, and notification identifiers (so that e.g. we can group offers by product),
- the condition of the product (new or used),

- the total feedback count for merchants as well as the percentage of positive feedback,
- shipping time variables providing information on how soon the product will ship,
- the country and/or state that the offer will ship from,
- whether or not the offer is fulfilled by Amazon
- the listing price and shipping price for the offer,
- and whether or not the offer is in the Buybox.

Of these variables, [2] do not observe any of the shipping variables and in particular are also unable or unwilling to distinguish between shipping costs and the listing price of an item. However, they do employ an additional measure of seller quality (the average seller rating) which we do not observe in this dataset; as this feature has an importance of only 0.03 in their analysis we do not think that we will bias our results by omitting it here.

In order to limit computational time, most of the results in this paper will employ only the 2 437 835 (offer-level) observations from April 1st, 2019; however, we have access to approximately 250x this amount of data. Thus, the total amount of data available to us is approximately 25x as much as that available to [2].

3 Preprocessing & Feature Engineering

While our data are already 'clean' in the sense that they are the output of an algorithm that many repricing companies rely on to have a fixed format (and e.g. we can hence expect that we will see nothing akin to data entry errors), we nevertheless need to preprocess our data with our goal of predicting Buybox ownership in mind. In particular, we discard the 16.92% offers that are not Buybox eligible (as eligibility is determined using a publicly known algorithm) and we restrict attention to offers in the 'new' condition.

Finally, there are two situations in which there may not be exactly one offer in the Buybox. Firstly, when no offer is deemed competitive by Amazon it may 'suppress' the Buybox, i.e. not assign it to anyone – this happens in 10.73% of our observations. While the question when this occurs is an interesting one for future research, answering it requires data on the prices of these products at other stores and hence is beyond the scope of this paper. Secondly, Amazon can put different offers in the Buybox for Prime customers; thus, in 6.26% of cases there are two Buybox offers. We sidestep both of these issues by discarding any such observations. All in all, we discard 41.71% of the data available to us in preprocessing.

Moving on to feature engineering, it seems abundantly clear that as for each product (in our sample) exactly one offer will be assigned to the Buybox, the relevant features are not so much the characteristics of the individual offers but rather how these offers compare to other offers on the same product. Our feature engineering is designed to allow our classifiers to pick up on these relative features.

To begin with, for each of the continuous features available to us we find (on a product-by-product basis) the absolute and relative distances on this feature from (i) the offer with the minimum value of this feature and (ii) the offer with the maximum value of this feature. We also find the rank of the offer on the feature (as a fraction of the total number of offers) and create additional indicators for whether this offer is the maximum or minimum rank. Finally, and most importantly, we find the distance on this feature from the offer with the k-th lowest landed price for $k \in \{1, 2, 3\}$. We perform these transformations for (i) the landed price itself, (ii) the listing price, (iii) the shipping price, (iv) the feedback count, (v) the fraction of positive feedback, (vi) the lower end of the estimated time to ship, (vii) the upper end of the estimated time to ship as well as (viii) for a newly constructed proxy for the number of positive reviews (the product of (v) and (iv)).

Furthermore, we also create features that compare the offers' boolean features to those of the k-lowest offers: e.g. one feature measures whether both this and the lowest-priced offer are fulfilled by Amazon. We perform this comparison for (i) whether an offer is sold by Amazon, (ii) whether an offer is fulfilled by Amazon, (iii) whether an offer is back-ordered and (iv) whether an offer ships from a domestic origin.

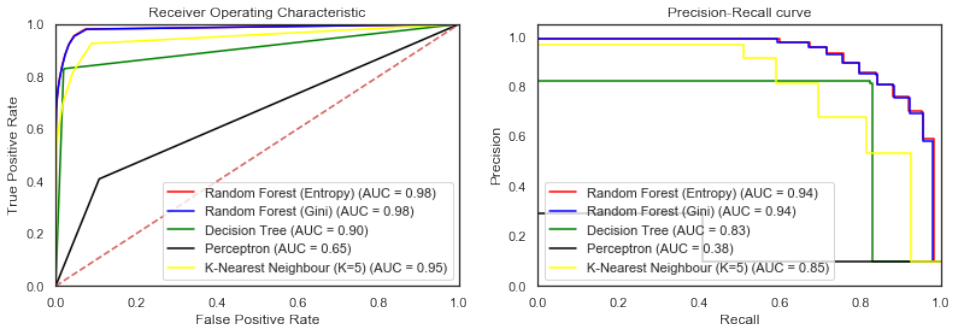

Figure 2: The random forest easily beats other available classifiers on a range of metrics.

Finally, we create an indicator for whether an offer is sold by Amazon or one of its subsidiaries (and in exchange drop the more general merchant identifier) as well as a variable that measures the number of offers. This brings our final feature count to

4 Classification

4.1 Classifier Choice

To begin with, we briefly compare the performance[1] of various classifiers on a small 10% sample of one day of data[2]. We plot ROC and Precision-Recall curves in Figure 2. As we can see, the random forest outperforms all other classifiers across all reported metrics. Hence, we will focus the rest of our analysis on this classifier. This has the additional benefit of (i) comparability with past literature and (ii) interpretability (as the RF has well-defined feature importances).

4.2 Comparison with Chen

The only prior work on this topic known to the author is [2] by Chen et al, which employs data scraped from best-selling products on Amazon. Due to this, Chen et al are limited in both the variables they observe (most importantly they do not observe Shipping Time) as well as the amount of their data. However, even using this limited data, they find that they are still able to obtain a good fit with accuracies around 75% to 85%.

We examine how our classifier performance compares to both a dummy classifier that simply assigns the Buybox to the lowest-priced offer and the classifier limited to the features of [2]. To this purpose, we replicate Figure 10 in [2] as Figure 3(a) and provide the same graph for the F1-Score in Figure 3(b). As we can see, both Chen and we outperform the dummy classifier at a large margin, but the additional covariates we have access to allow us to outperform Chen substantially.

We report a comparison of feature importances[3] found in our data to those reported by [2] in Table 1 where we find broadly similar patterns: overwhelmingly the most important factor in gaining the Buybox is the relative price. However, we also find that in our dataset Feedback Count is far more important than in [2]: this is probably because Chen et al chose to sample best-selling products, which are often sold by quite experienced and established merchants. Our sample, on the other hand, contains much more exposure to small merchants that Amazon may be wary to assign to the Buybox.

[1] All scores are calculated employing a 20% held-out sample.

[2] All classifiers employed in this paper used the implementations in sci-kit learn [3].

[3] We do not observe the Average Seller Rating, which [2] do observe but find to be unimportant; hence we renormalize the Chen feature importances so that they sum to one with this feature removed.

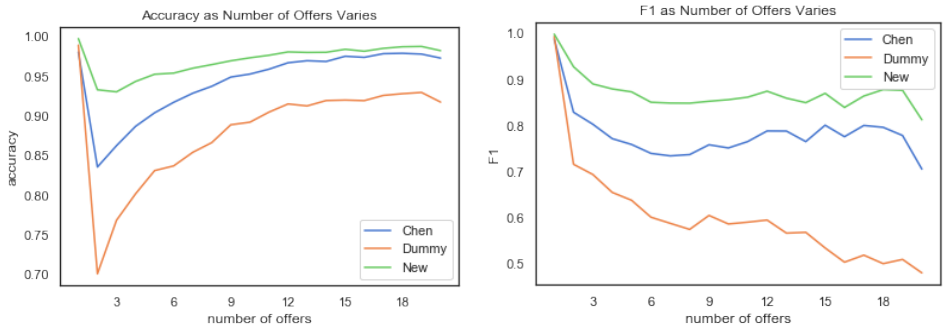

Figure 3: By employing additional covariates, we substantially outperform [2] .

Feature	**Weight** (Original Data)	**Weight** (Our Data)
Price Ratio to Lowest	0.340206	0.264880
Price Difference to Lowest	0.371134	0.261884
Feedback Count	0.103093	0.234800
Is Amazon the Seller?	0.103093	0.102706
Is the Product FBA?	0.061856	0.072339
Positive Feedback	0.020619	0.063391

Table 1: The main difference to [2] in our data is the importance of Feedback Count.

4.3 Big Data

For computational reasons, up unto this point we have utilised only one day of data in our analysis. This is especially problematic insofar as it ensures that the classifier is fit using mostly cross-sectional variation: it is not very common to see products change prices multiple times in a short period of time. This could imply poor performance as the classifier does not observe the same sellers competing for the Buybox for the same product at different prices. Theoretically, it is also possible that we overfit to a single day of data if there are any time-varying components of the Buybox algorithm.

To ameliorate these issues, we now move on to examining the performance of the random forest as we progressively give it access to more data. This causes several logistical problems mostly related to the fact that our data will not fit into memory. However, as the random forest relies on estimators trained on separate subsets of the data this is not as much of an issue for it as it is for other classifiers. We employ a cluster to estimate one decision tree for each day of data in parallel and then merge these trees into a Random Forest. Note that optimally we would want to reshuffle the data so that instead of one estimator per day we have one estimator for a day-sized random chunk of the data; this, however, provides logistical challenges beyond the scope of this paper.

We test the performance of our classifier on a fixed 0.25% random sample of the overall data[4] after every day of training data is added and present the results in Figure 4: on the LHS, we plot the performance of a classifier using a given number of days of data relative to one employing just one day of data (on a fixed test set). On the RHS, we plot the F1 score. As we can see from these plots, adding more data substantially increases classifier performance[5].

[4]This may seem small, but here the memory constraints really do kick in.

[5]We also see an interesting periodicity that we suspect comes from the fact that the random forest performance depends crucially on whether there is an even or an odd number of decision tress; in support of this hypothesis, we ran the model for the first couple of days with multiple decision trees per day and saw the pattern eliminated.

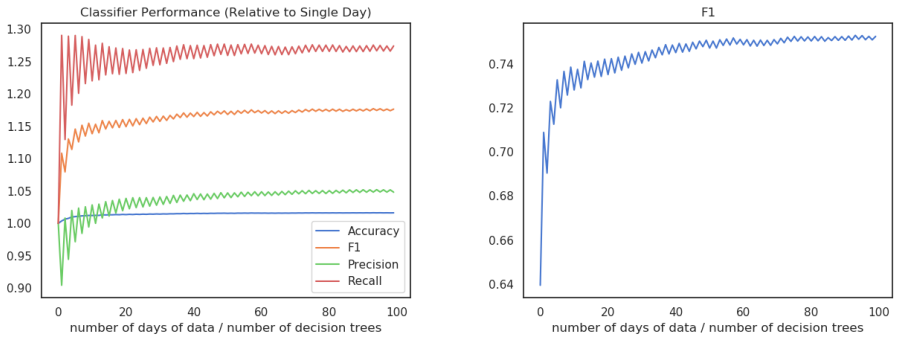

Figure 4: As we increase the amount of data employed, classification performance increases.

Group of Features	**Weight**
Landed Price	0.28
Listing Price	0.21
Shipping Time	0.13
Feedback Count	0.09
Shipping Availability	0.07
Positive Feedback (Fraction)	0.06
Positive Feedback (Count)	0.06
IsAmazon	0.03
Is Fulfilled By Amazon?	0.03
Shipping Origin	0.02
Number of Offers	0.01
Shipping Cost	0.00

Table 2: Relative Price is most important, but Shipping Time and Seller Quality also matters.

4.4 Feature Importances

We want to know which of the features actually matter the most to determining Buybox status. As we have created too many 'relative' features for readability, we report importances in Table 4 at the level of the original features by summing the feature importance across the derived features. It is immediately clear that – as expected – the most relevant feature by far is the relative price of the listing. However, we further see that seller quality also matters to Amazon: the three feedback variables have weights that sum to 21%. In a similar vein, the shipping variables and in particular the shipping time also seem to influence Buybox allocation.

Finally, we report the feature importances on the indicators for whether an offer is fulfilled by Amazon and whether the seller itself is Amazon. Interestingly, these features seem to matter only very little when we look at the feature importances; but we will see in the interpretation section that this may not be the full story. This is consistent with the argument in [4] that the default sci-kit learn feature importances may underestimate the importance of low-cardinality variables.

5 Interpretation

We are particularly interested in whether Amazon takes into account factors other than price. While we have some suggestive evidence in this direction above (e.g. our classifier performed much better than a dummy classifier that just looks at the cheapest price), we now explore this issue in more depth. In particular, we now employ our fitted model to examine predicted probabilities of being in the Buybox in a situation where there are exactly two offers. We define the *Buybox Advantage* of the lower offer as

$$b := \mathbb{P}(\text{lower offer is in Buybox}) - \mathbb{P}(\text{higher offer is in Buybox}).$$

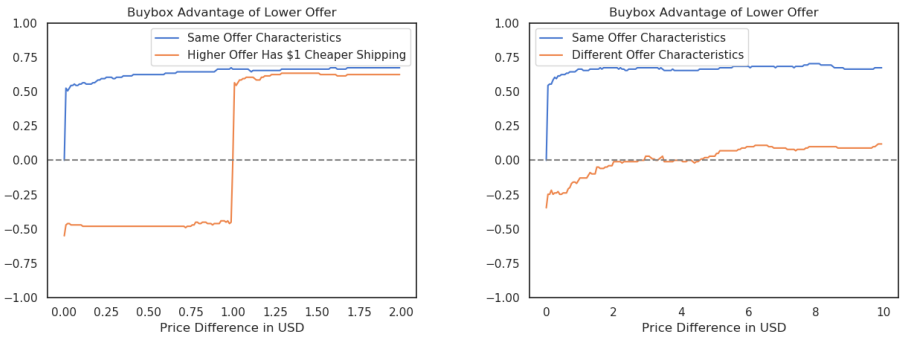

Figure 5: Sellers with More or Better Feedback Are Punished Less for Higher Prices.

We now examine how various factors impact b. To do so, we let our 'big data' model repeatedly predict b for the case of two identical offers as we vary their absolute price difference (the offers have all characteristics other than price set at their median values[6]). We then modify one offer in some way and again let our model repeatedly predict b.

Our baseline expectation is that (i) $b > 0$ as the lower offer should be more likely to win the Buybox, (ii) b should be weakly increasing in the price difference. Given this expectation, we can now vary the lower and higher offer characteristics and see how this affects b: does it shift b down? up? In particular, we are interested whether some factors shift b down for low price differences but Amazon ignores them for high price differences. For instance, one could imagine Amazon assigning the Buybox to the higher quality seller if the two offers have very similar prices but switching to the cheaper offer if the price difference is sufficiently pronounced.

As a first validation of our results, we analyze on the LHS of Figure 5 how b compares between the case of (i) two identical offers and the case (ii) in which one offer has a $1 higher shipping cost. Reassuringly, we see that $b > 0$ in the case (i). Interestingly, for price differences below $1, we have $b < 0$ in case (ii): thus, the Buybox is overwhelmingly more likely to be allocated to the product with the cheaper *landed* price (i.e. price including shipping cost).

5.1 Shipping Speed

We now begin our examination of b with a variable that we know is very important to Amazon but which was unavailable to [2]: shipping speed. On the RHS of Figure 5 we plot b for the case where the higher offer ships in just 24h while the lower offer takes 5 days to ship. The careful reader will note that the horizontal axis in this graph spans up to a price difference of $20: nevertheless, even at a price difference of $20 the odds are only about even that the lower offer will now occupy the Buybox. For price differences under $

5.2 Seller Quality: Feedback Count vs Feedback Quality

Moving on to measures of seller quality, on the LHS of Figure 6, we graph b for the case where the higher offer has a 100x higher feedback count. Such a high feedback count could indicate an experienced seller who might be more likely to satisfy the large amount of demand that can suddenly occur if a product becomes popular on Amazon. Whether this is the case or not, Figure 6 certainly indicates that Amazon trusts sellers with a high feedback count to perform better when occupying the Buybox as these sellers have less of a Buybox disadvantage for even large price differences than merchants with less feedback do.

On the RHS of Figure 6 we see that the same effect occurs for merchants that have *better* feedback (as opposed to just more): in the case depicted, one seller has 70% positive feedback and the other 100%. Interestingly, the effect seems about the same size in this specific case. However, caution

[6]It may be of interest to the reader that the median landed price in the data is $30.73.

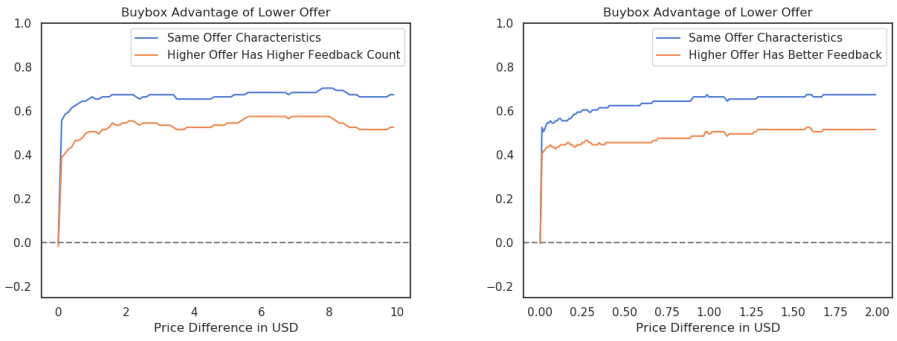

Figure 6: Sellers with More or Better Feedback Are Punished Less for Higher Prices.

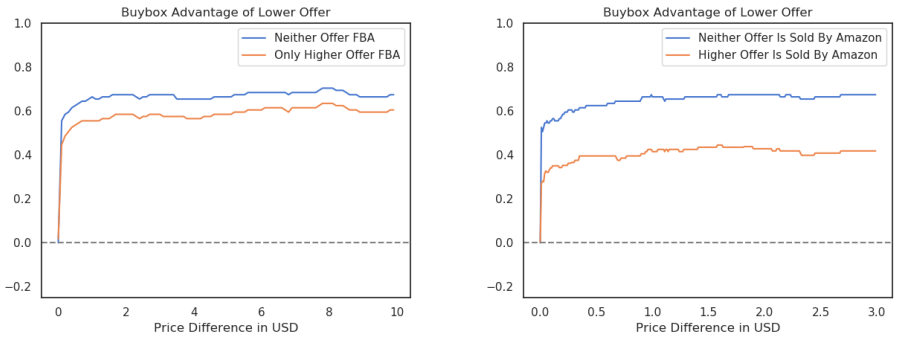

Figure 7: Sellers employing FBA are advantaged, but Amazon is even more so.

should be taken not to extrapolate from these two graphs to statements about the relative importance of more versus better feedback as it is not clear that the scales for these variables are comparable.

5.3 Is Amazon Playing Fair?

It is a peculiar feature of Amazon that it understands itself not only as a platform that matches sellers and merchants but also as one of these merchants itself. This naturally makes one worry as to whether Amazon has the correct incentives to always assign the Buybox to the seller that is best from the consumers' perspective: if it assigns the Buybox to itself, after all, it gets to make the profit on the product sold! Similarly, some sellers employ Amazon's fulfillment service (FBA) and thus pay higher fees to Amazon so that Amazon may have a larger incentive for assigning these sellers to the Buybox.

In Figure 7 we show that these worries may be justified. In the left panel, we see that compared to a situation were neither seller is FBA, Amazon is slightly more likely to give the Buybox to an FBA seller even when his offer is not the cheapest. In the right panel, we see that similarly Amazon seems to be giving itself an advantage in terms of Buybox allocation. Note that neither of these effects can be related to the fact that Amazon reserves the right to let a different merchant occupy the 'Prime Buybox' which is shown only to Prime customers: in those cases, Amazon explicitly assigns two merchants to the Buybox and hence these cases where dropped in preprocessing above.

At this point, like [2], we have to caution that Amazon no doubt observes many metrics unobservable to us and on which Amazon and/or FBA sellers likely perform better than the average merchant: for instance, Amazon has access to the number of products returned by the customer. Nevertheless, the most concerning variable that [2] did not observe and on which we would expect Amazon to outperform other merchants – shipping speed – is controlled for here.

6 Conclusion

This paper has employed extensive proprietary data on Amazon Buybox assignment to train a random forest model to predict how Amazon allocates sales to merchants. We find that the relative price is by far the most important feature in Buybox allocation. However, other varibales play an important secondary role: faster shipping, higher feedback count and better feedback all make it more likely that a sellers' offer finds its way into the Buybox. Finally, even when controlling for all of these covariates, Amazon seems to give itself as well as FBA sellers a slight edge when allocating the Buybox.

References

[1] Amazon. Annual Report: 2017. Technical report, April 2018.

[2] Le Chen, Alan Mislove, and Christo Wilson. An Empirical Analysis of Algorithmic Pricing on Amazon Marketplace. In *the 25th International Conference*, pages 1339–1349, New York, New York, USA, 2016. ACM Press.

[3] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[4] Carolin Strobl, Anne-Laure Boulesteix, Achim Zeileis, and Torsten Hothorn. Bias in random forest variable importance measures: Illustrations, sources and a solution. *BMC Bioinformatics*, 8(1):25, Jan 2007.