IE 544
DECISION ANALYSIS

Assignment 3
“How to Win Amazon’s Buy Box?”

By

Group ID: Q
2020702114 Hande Mızrak
2019702126 Burak Şimşek

The main contention of this project is to understand and reveal the factors playing a significant role in winning the Buy Box. At first sight one can easily say that price information has a major influence on Amazon's Buy Box. To see the effect of other features such as seller ratings, shipping costs, product ratings, etc, some analysis on the training data is made and given below.

Question 1.

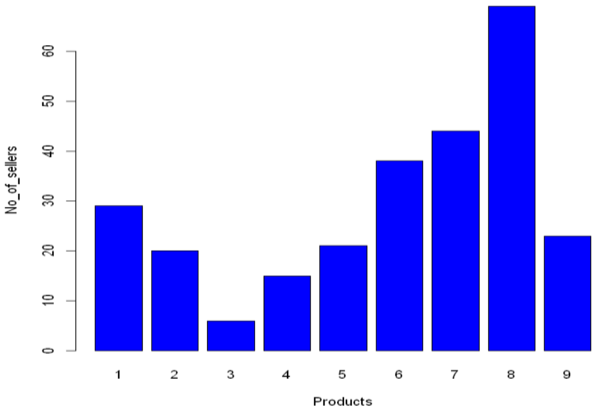
a.

Number of Products (Overall)	9
Number of Sellers (Overall)	184

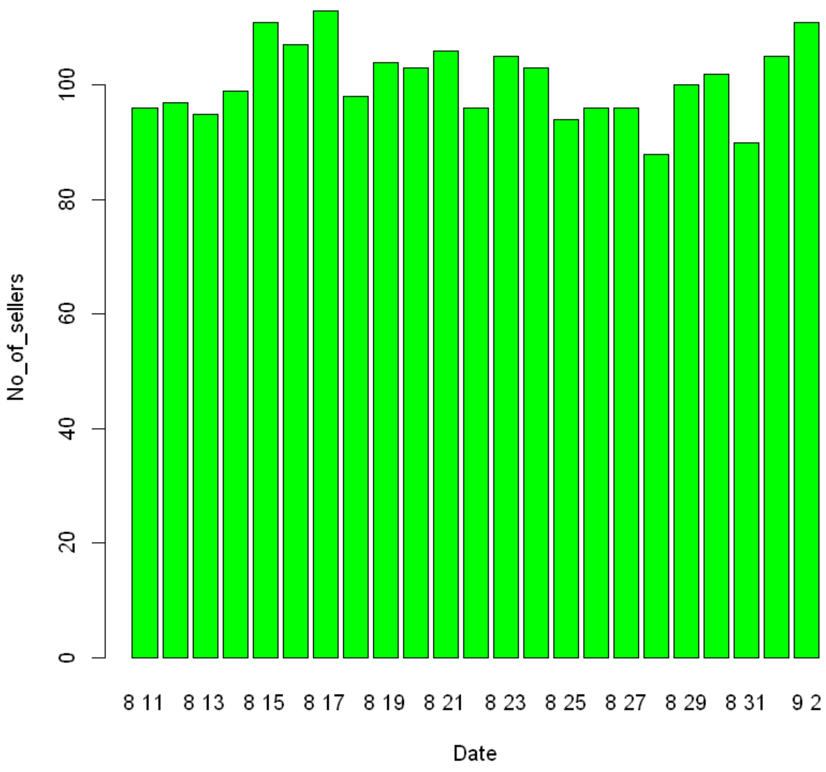
The name of the products and the corresponding seller numbers for these products are the following:

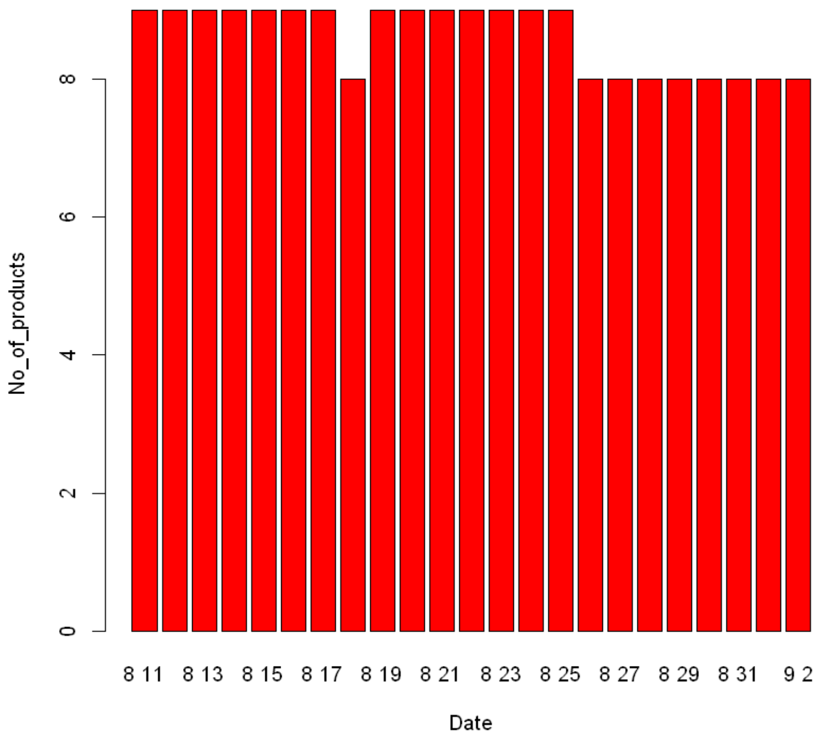
Product Name	**No. of Sellers**
B002ZV0OJO	29
B0083H1INK	20
B00AMFLZLG	6
B00DNSO1OW	15
B00DNSO41M	21
B00MVVI1FC	38
B00VSIT5UE	44
B00VSITBJO	69
B00YR6BMS2	23

The bar plot of the table given above is the following:

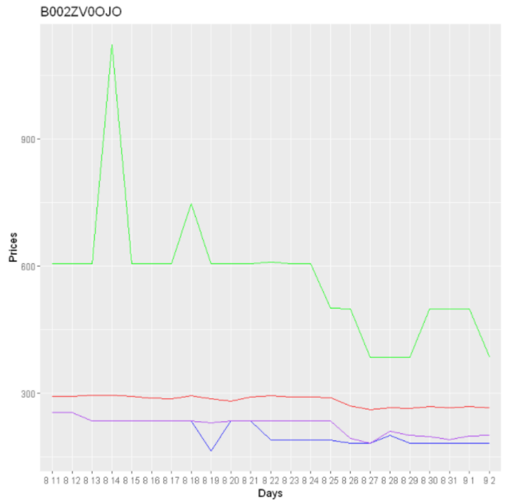


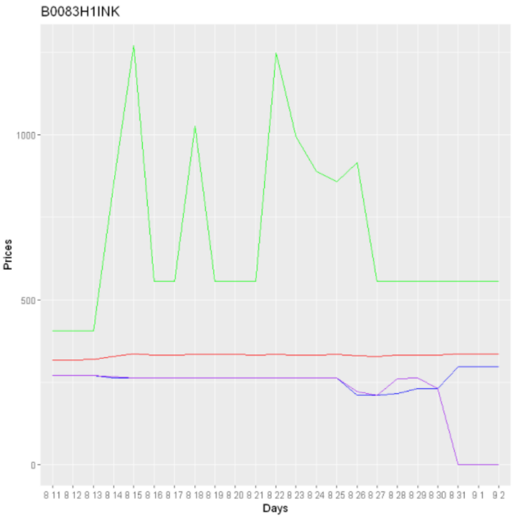
In our training data we have 23 days in total (from 08.11.2015 to 09.02.2015). The bar plots given below represent how the number of sellers and products change with respect to time which is day in our case.

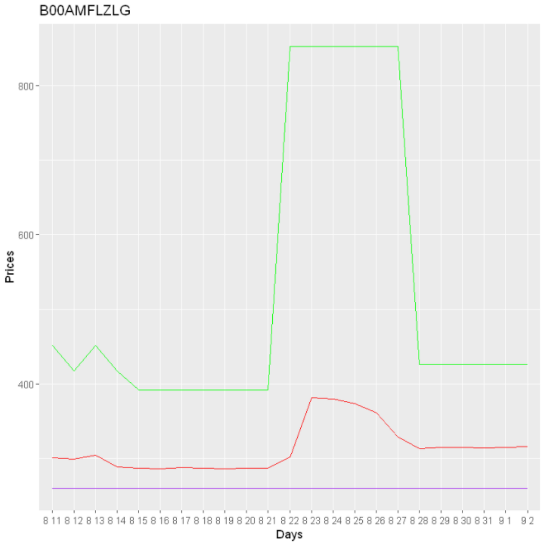


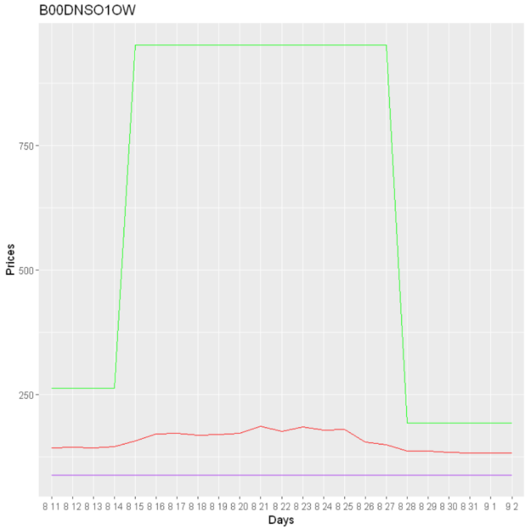


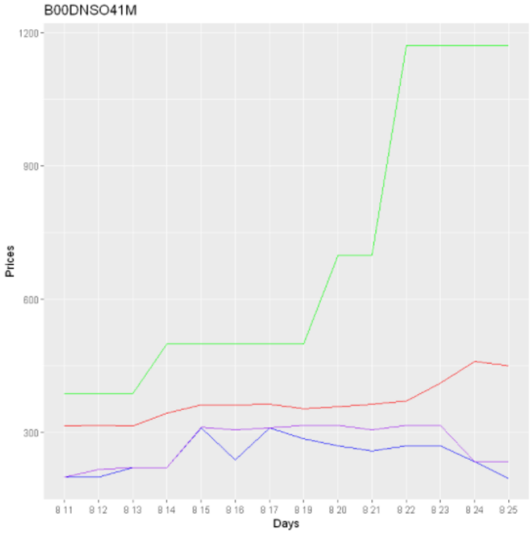
b. Before doing any calculation one can expect that average buy box price should follow min price since an algorithm which lets a seller to win offering a larger price compared to the other sellers for the same product is meaningless. The average, max, min, and average buy-box prices for each product are given below:

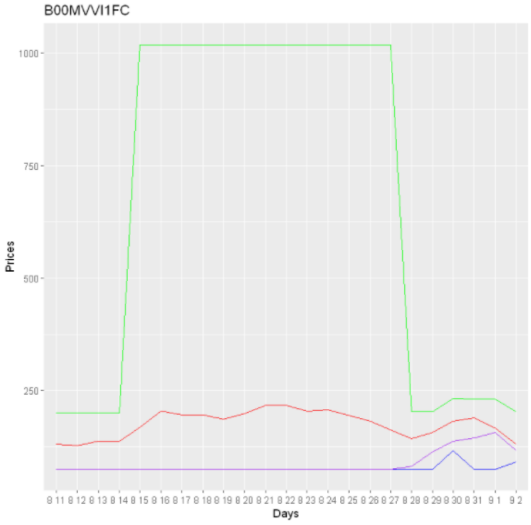


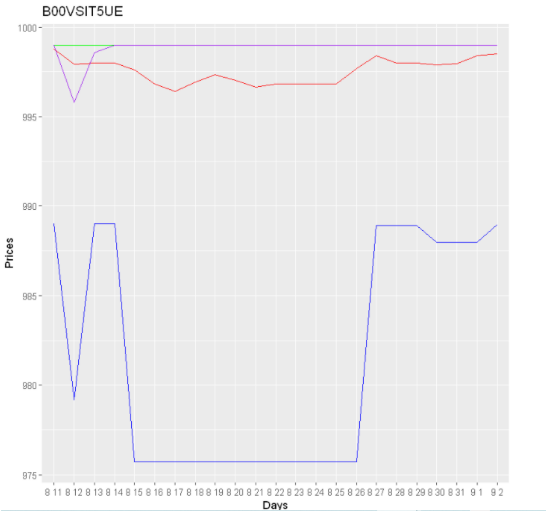


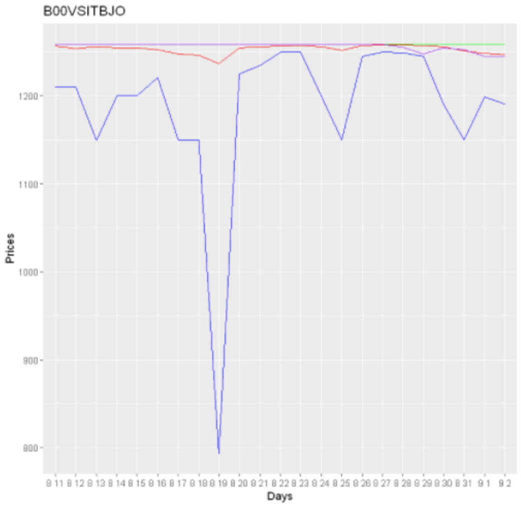


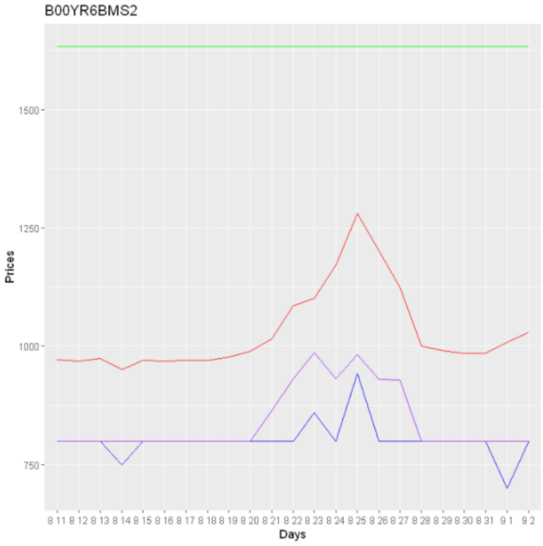






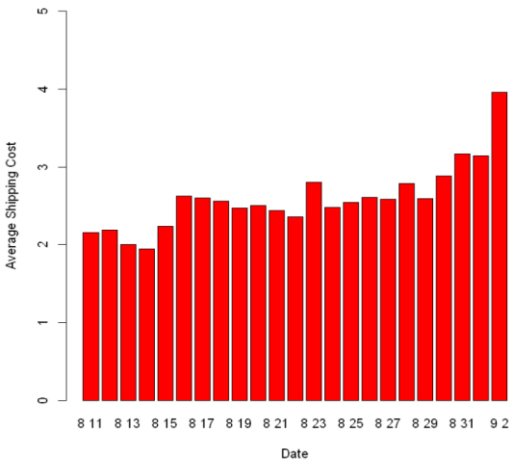


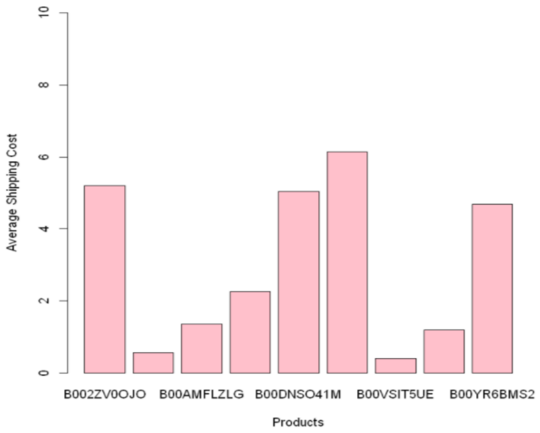




Blue, purple, green and red lines represent min price, average bbox price, max price and average price, respectively. As can be seen that the results agree with our first intuition for most of the products.The exceptions are product “B00VSIT5UE” and product “B00VSITBJO”. For these products bbox price is above the min price.

In addition, average shipping cost over time and by product are calculated as follows:

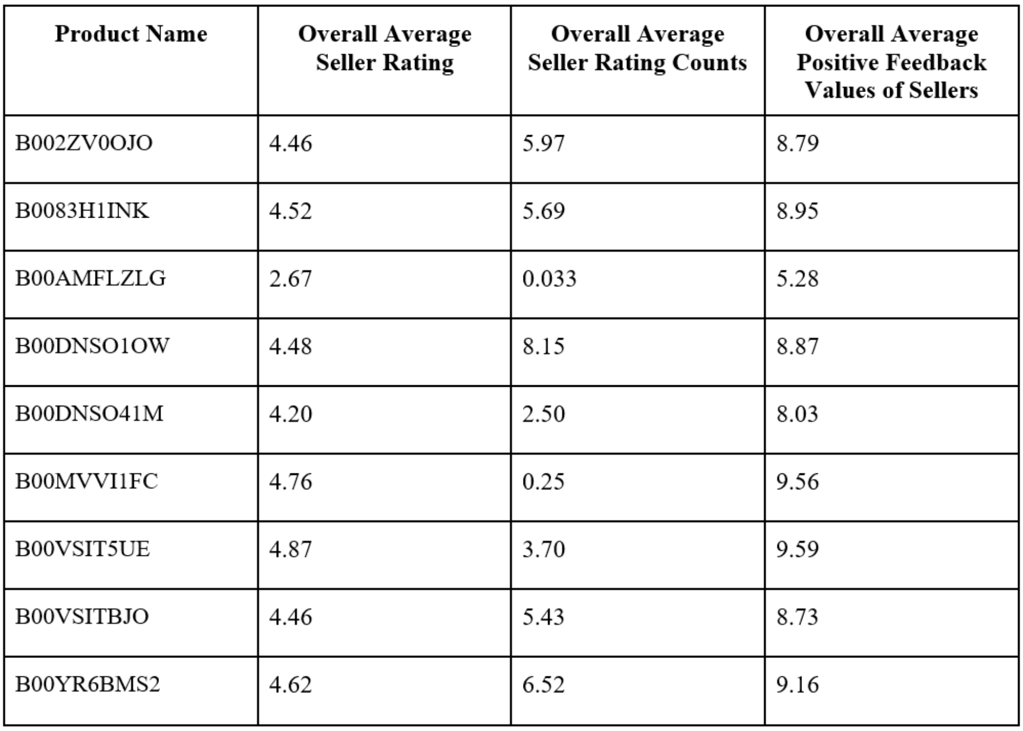




c. Since we do not have access to try the real products physically in the online shopping process, sharing customers' experiences publicly and giving feedback about the seller and product have become more crucial than they used to be. In addition, they play a significant role in promoting the sellers. Therefore, it is a good idea to look at the seller ratings, seller rating counts and positive feedback values of sellers who appear in the buy box for each product. The average results are calculated as follows:

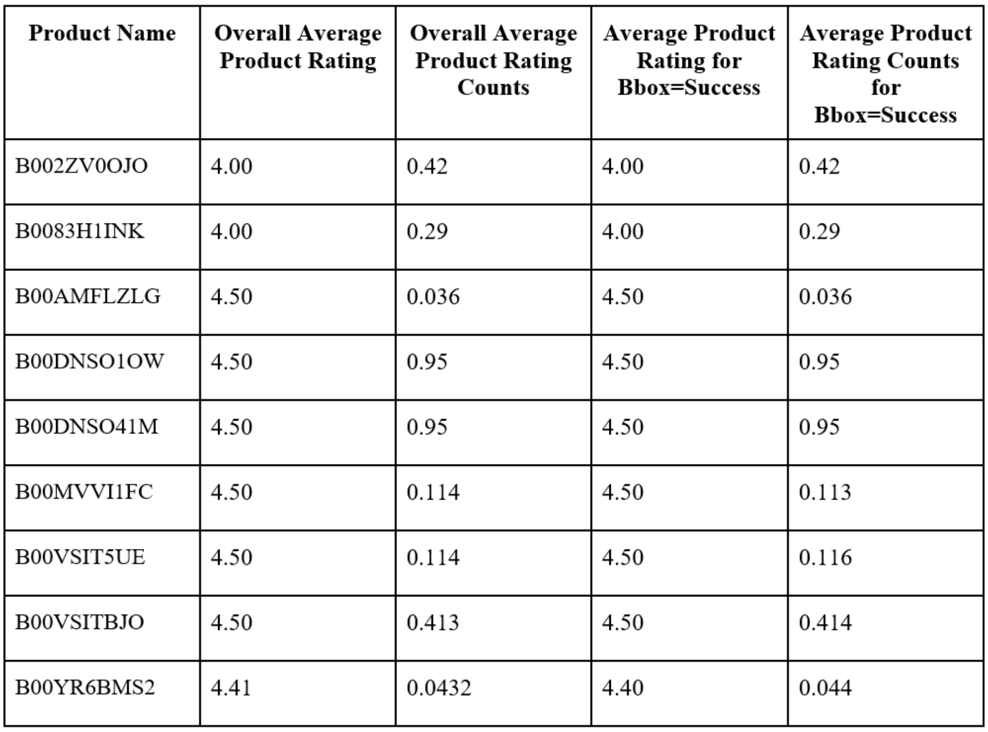
Product Name	**Average Seller Rating for Bbox=Success**	**Average Seller Rating Counts for Bbox=Success**	**Average Positive Feedback Values of Sellers for Bbox=Success**
B002ZV0OJO	5.00	99.72	9.99
B0083H1INK	5.00	100.00	10.00
B00AMFLZLG	5.00	0.14	9.50
B00DNSO1OW	5.00	100.00	10.00
B00DNSO41M	4.51	26.39	8.95
B00MVVI1FC	4.53	0.62	8.94
B00VSIT5UE	4.97	48.53	9.85
B00VSITBJO	4.99	79.11	9.96
B00YR6BMS2	4.94	77.35	9.86

From this table it can be seen that sellers who appear in the buy box have very high average ratings, rating counts and positive feedback values. To convince ourselves about this statement we should take the average using all observations not only the observations who win the buy box. If the values given above are higher than the overall values then we can conclude that these features are important for appearing in the buy box. The table which gives overall average of seller ratings, seller rating counts and positive feedback values of sellers are the following:



Product Name	Overall Average Seller Rating	Overall Average Seller Rating Counts	Overall Average Positive Feedback Values of Sellers
B002ZV0OJO	4.46	5.97	8.79
B0083H1INK	4.52	5.69	8.95
B00AMFLZLG	2.67	0.033	5.28
B00DNSO1OW	4.48	8.15	8.87
B00DNSO41M	4.20	2.50	8.03
B00MVVI1FC	4.76	0.25	9.56
B00VSIT5UE	4.87	3.70	9.59
B00VSITBJO	4.46	5.43	8.73
B00YR6BMS2	4.62	6.52	9.16

Therefore, we can conclude that for most of the products, sellers who appear in the buy box have higher average seller ratings, seller rating counts and positive feedback values compared to other sellers. Let's analyze the average product ratings and product rating counts. Since product ratings are unique properties for each product we do not expect any effect on bbox.



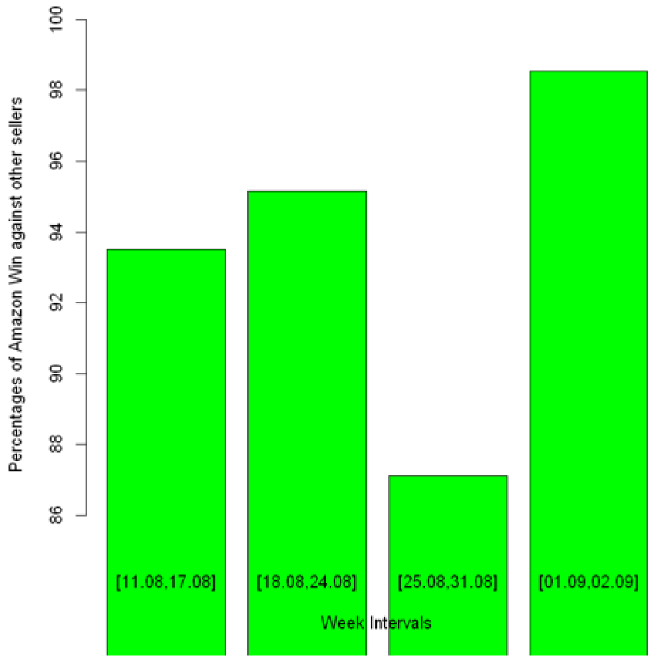
Product Name	Overall Average Product Rating	Overall Average Product Rating Counts	Average Product Rating for Bbox=Success	Average Product Rating Counts for Bbox=Success
B002ZV0OJO	4.00	0.42	4.00	0.42
B0083H1INK	4.00	0.29	4.00	0.29
B00AMFLZLG	4.50	0.036	4.50	0.036
B00DNSO1OW	4.50	0.95	4.50	0.95
B00DNSO41M	4.50	0.95	4.50	0.95
B00MVVI1FC	4.50	0.114	4.50	0.113
B00VSIT5UE	4.50	0.114	4.50	0.116
B00VSITBJO	4.50	0.413	4.50	0.414
B00YR6BMS2	4.41	0.0432	4.40	0.044

As can be seen there is no significant difference between overall average product ratings and average product ratings for Bbox=Success due to the reason explained above.

d. Let's compare for each product how many times amazon is the seller and how many times amazon appears in the buy box when it is seller. The percentages for each product are given below:

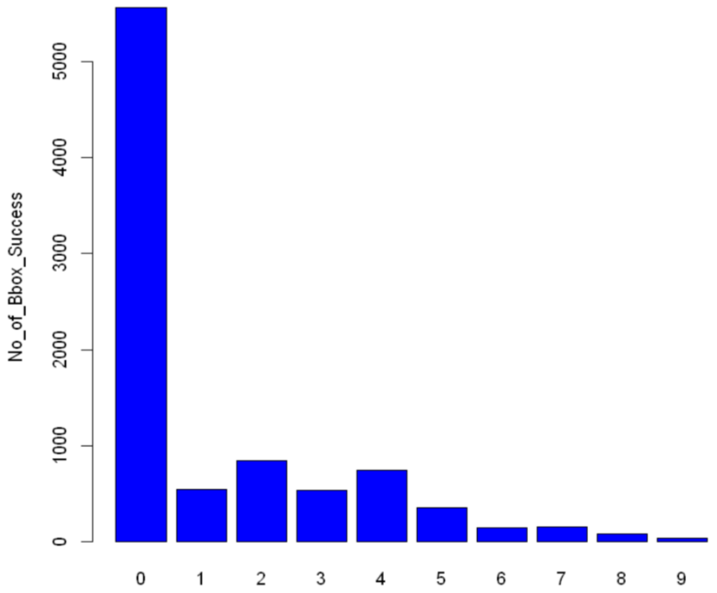
Product Name	**No. of Amazon is Seller**	**No. of Amazon in Buy Box**	**Percentage %**
B002ZV0OJO	1105	1102	99%
B0083H1INK	956	956	100%
B00AMFLZLG	0	0	0%
B00DNSO1OW	1108	1108	100%
B00DNSO41M	123	123	100%
B00MVVI1FC	8	0	0%
B00VSIT5UE	680	490	72%
B00VSITBJO	1072	859	80%
B00YR6BMS2	802	785	97.8%
Total	**5854**	**5423**	**92.6%**

From this table we can conclude that for most of the products the information of the seller is amazon or not plays a key role in appearing in the buy box. The reason why amazon looks unsuccessful for the product "B00AMFLZLG" is that amazon does not sell between the dates from 08.11.2015 to 09.02.2015. When we do the same analysis weekly, the same conclusion is achieved and given below:

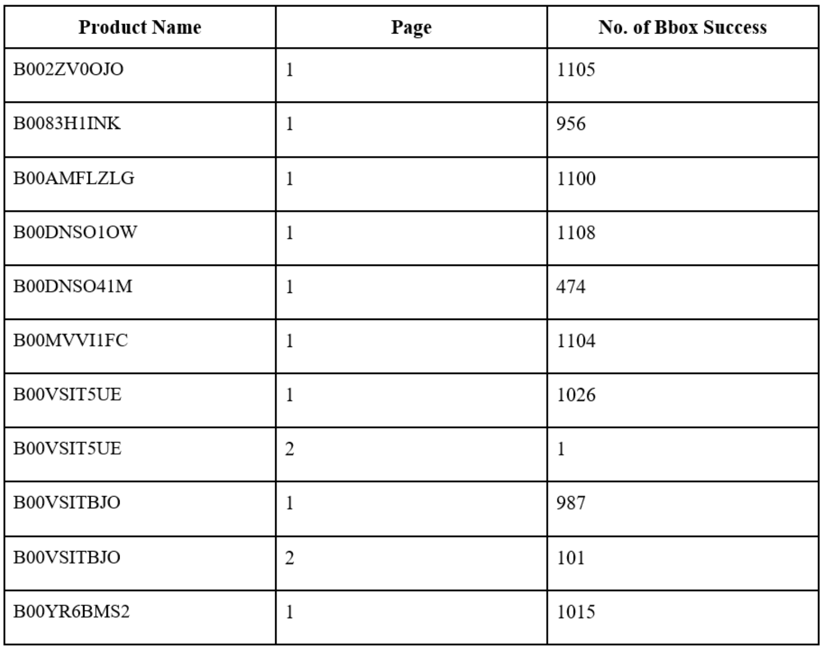


When we divide our total date (23 days) into 4 weeks, we can observe that the percentage of amazon wins against other sellers is very high (they all are above 86%).

e. The rank information in the training data is ranging from 0 to 11. The bar plot of rank vs number of bbox successes are the following:



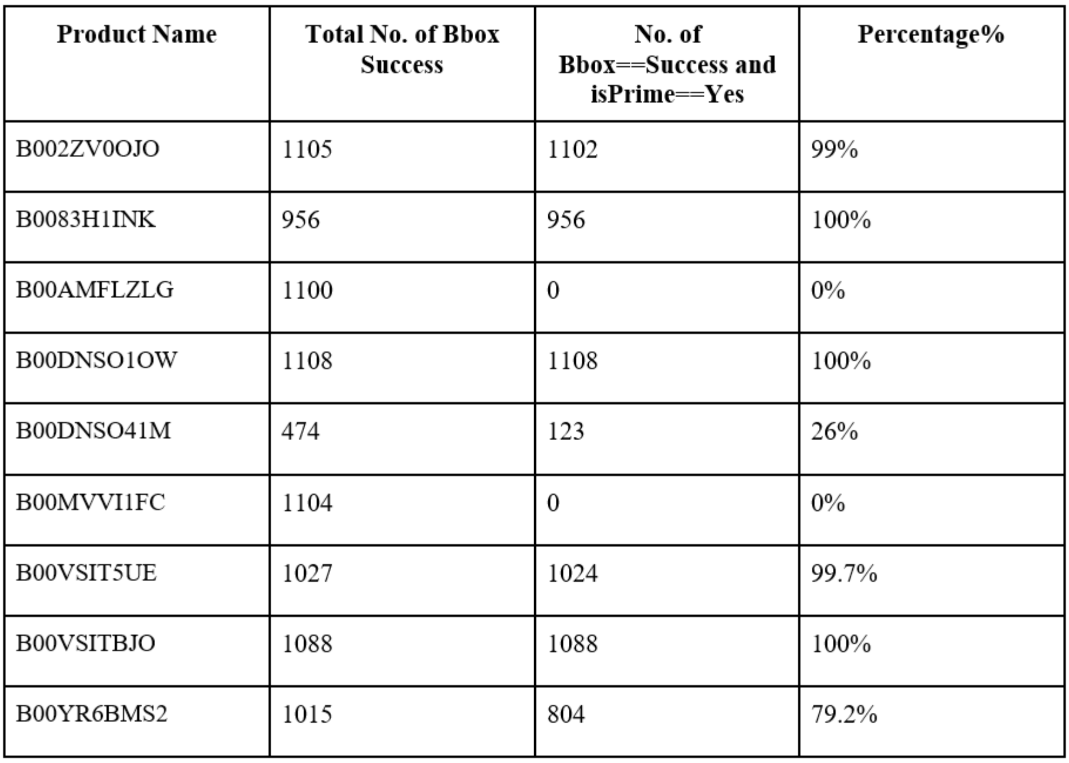
Rank information by seller and by product are given in the "Appendices" section since their tableaus are long. From the plot above, we can say the chance of winning buy box increases as we approach rank 0. When we look at what the rank 0 is we realize that rank 0 occurs when amazon is the seller. This result would not say rank is important in affecting buy box decision if the number of successes had not decreased along x axis.



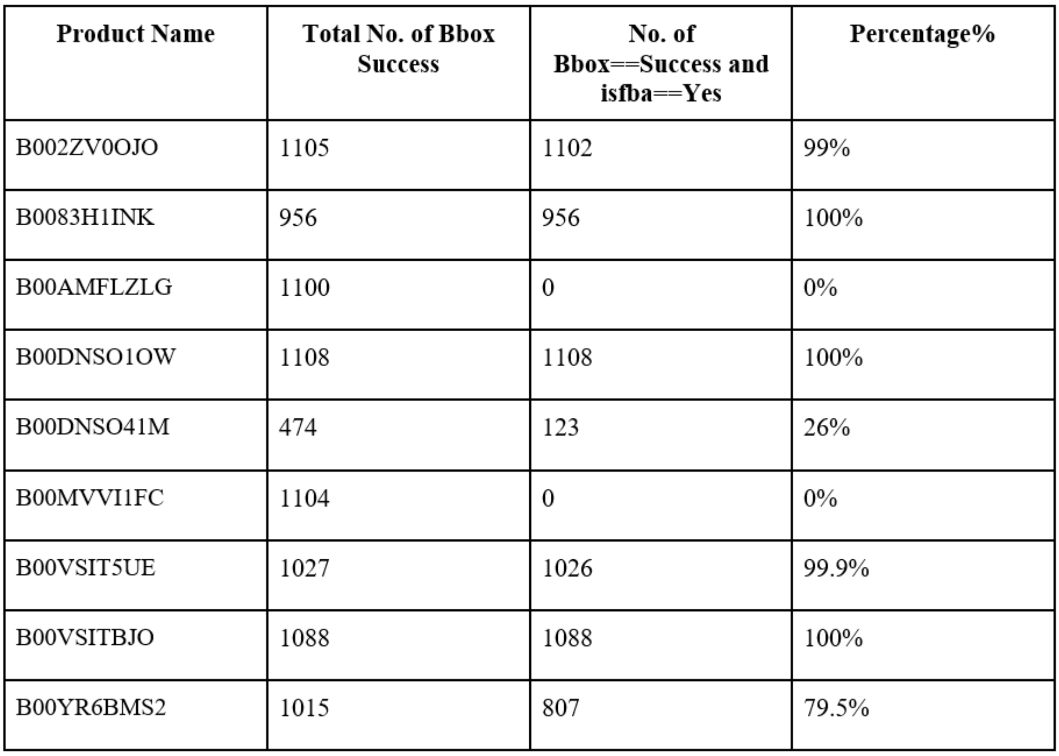
Product Name	Page	No. of Bbox Success
B002ZV0OJO	1	1105
B0083H1INK	1	956
B00AMFLZLG	1	1100
B00DNSO1OW	1	1108
B00DNSO41M	1	474
B00MVVI1FC	1	1104
B00VSIT5UE	1	1026
B00VSIT5UE	2	1
B00VSITBJO	1	987
B00VSITBJO	2	101
B00YR6BMS2	1	1015

As can be seen, for the products "B00VSIT5UE" and "B00VSITBJO" the sellers appearing in the buy box are on the first page. However, we cannot say anything about the other products since we do not have their page 2 versions on the training data. Page information by seller can be found in the "Appendices" section again.

To analyze the effect of Prime and Fba on buy bux successes for each product, the tables below are constructed:



Product Name	Total No. of Bbox Success	No. of Bbox==Success and isPrime==Yes	Percentage%
B002ZV0OJO	1105	1102	99%
B0083H1INK	956	956	100%
B00AMFLZLG	1100	0	0%
B00DNSO1OW	1108	1108	100%
B00DNSO41M	474	123	26%
B00MVVI1FC	1104	0	0%
B00VSIT5UE	1027	1024	99.7%
B00VSITBJO	1088	1088	100%
B00YR6BMS2	1015	804	79.2%



Product Name	Total No. of Bbox Success	No. of Bbox==Success and isfba==Yes	Percentage%
B002ZV0OJO	1105	1102	99%
B0083H1INK	956	956	100%
B00AMFLZLG	1100	0	0%
B00DNSO1OW	1108	1108	100%
B00DNSO41M	474	123	26%
B00MVVI1FC	1104	0	0%
B00VSIT5UE	1027	1026	99.9%
B00VSITBJO	1088	1088	100%
B00YR6BMS2	1015	807	79.5%

From these tables it can be concluded that for most of the products except “B00AMFLZLG”, “B00MVVI1FC” and “B00DNSO41M” Fba and Prime has a strong influence on appearing in the buy box.

Using the same logic, the effect of Prime and Fba on buy bux successes for sellers are shown as follows:

Seller Id	# Bbox Success	# Bbox==Success and isPrime==Yes	%	# Bbox Success	# Bbox==Success and isfba==Yes	%
A1CPWE4BUHBCWE	15	15	100%	15	15	100%
A1L9D0A8YQ4I2J	52	52	100%	52	52	100%
A1LLDJOGLVSHKP	17	17	100%	17	17	100%
A1NNAMFIGL0WW	1	1	100%	1	1	100%
A1P24ZYMPUKXCC	18	18	100%	18	18	100%
A22RMSXAK08YRI	8	8	100%	8	8	100%
A2BTS1X5W6FOE2	29	29	100%	29	29	100%
A2TC87EJKQMY9O	31	19	61.3%	31	22	71%
A33ZYGKLC0HPSW	7	7	100%	7	7	100%
A3HZNJ7XC4ABL8	52	52	100%	52	52	100%
A3SQ9YFE6CSCS0	551	551	100%	551	551	100%
A7Q70VOZ5QTKB	6	6	100%	6	6	100%
A9X5UX7R2WF5U	7	7	100%	7	7	100%
amazon	5423	5423	100%	5423	5423	100%
A1ZI2ITZC97U95	NA	NA	NA	2	2	100%

From these tables we understand that the sellers having Prime and Fba appear in the buy box mostly.

f. In part b, we see that the price information is an important factor for appearing in the buy box. For example, the seller who offers less price has a better chance to win. Therefore, in order to give this information to our data, the price column in the training data is changed as follows:

$$\text{Price Gap Ratio} = \frac{\text{price of a product for a given time} - \text{min price of the product}}{\text{min price of the product}}$$

We know that buy bux prices follow the min price for most of the products. Therefore, with this formulation we want to measure how far a product’s price is from the product’s min for a given time. So it can be seen that the larger the price gap ratio the less chance to win the buy box. In this way we make the price scales smaller which is usually necessary in the problems containing data. Otherwise, it can dominate other features just because its values while it is not an important feature, for example.

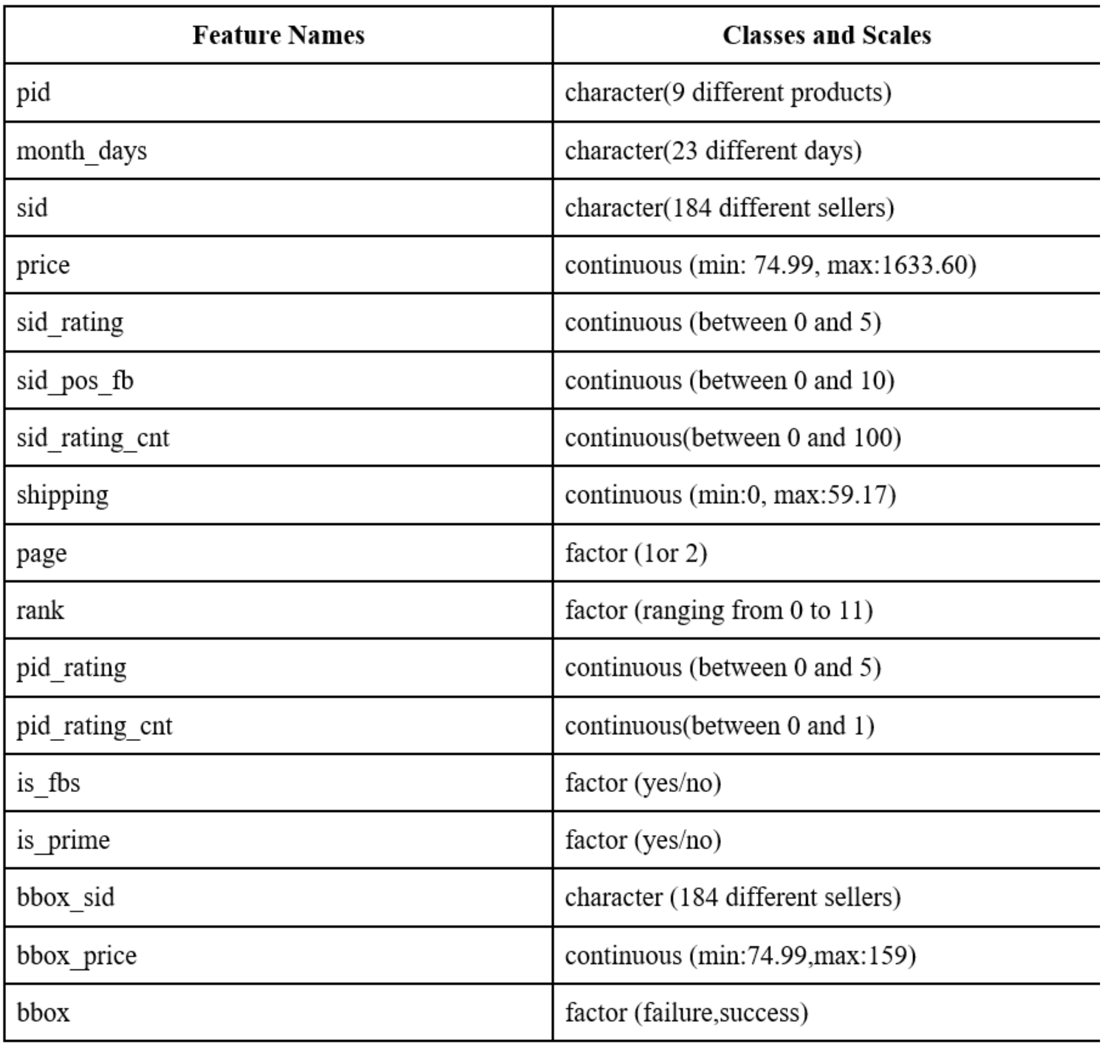
In addition, shipping costs are usually playing a key role in our decision process. When we want to buy a product and pick a seller we consider not only the seller's price but the price containing the seller's shipping cost. In order to put this information to our data and model we create another feature called "total weighted cost" using the formula below:

$$Total\ Weighted\ Cost = 0.8 * Price + 0.2 * Shipping\ Cost$$

The weights are selected by our personal experiences. Since we give more importance to the price in the online shopping process we select 0.8 for this feature. We think that this feature will be beneficial in creating different models for each product since the price values are product dependent. In this project, we create only 1 model. Therefore, we did not include this feature but we put this idea for future work.

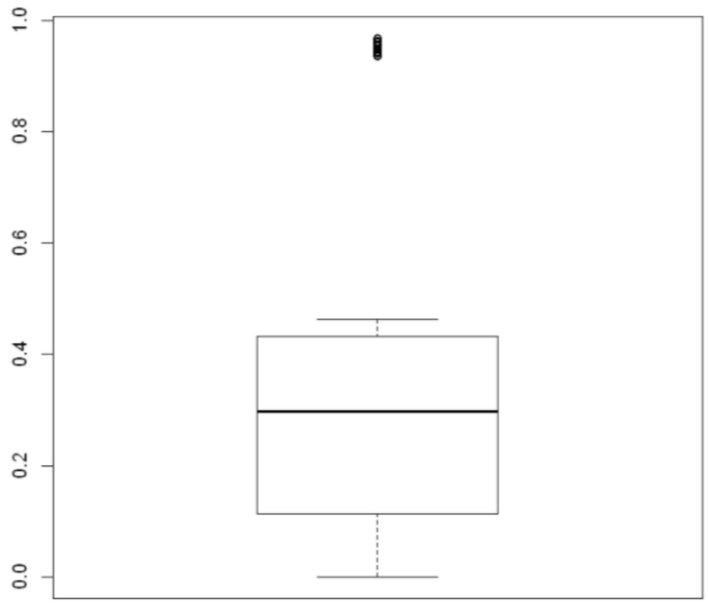
g.

We have 17 columns in our training data. Some of them such as is_prime, is_fb, bbox,etc are factor variables and some of them are continuous variables. The table that summarizes these properties are given below:



Feature Names	Classes and Scales
pid	character(9 different products)
month_days	character(23 different days)
sid	character(184 different sellers)
price	continuous (min: 74.99, max:1633.60)
sid_rating	continuous (between 0 and 5)
sid_pos_fb	continuous (between 0 and 10)
sid_rating_cnt	continuous(between 0 and 100)
shipping	continuous (min:0, max:59.17)
page	factor (1or 2)
rank	factor (ranging from 0 to 11)
pid_rating	continuous (between 0 and 5)
pid_rating_cnt	continuous(between 0 and 1)
is_fbs	factor (yes/no)
is_prime	factor (yes/no)
bbox_sid	character (184 different sellers)
bbox_price	continuous (min:74.99,max:159)
bbox	factor (failure,success)

h. When we do some analysis to detect outliers, we realize that there are many outliers in the training data. For example, pid_pos_fb column has outliers:

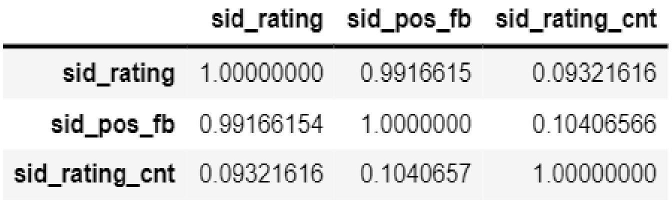


The outliers are product 'B00DNSO1OW' and product 'B00DNSO41M'. If we remove them from the training data it may cause loss of information for these products. Therefore, we decided to keep the outliers in the data.

Question 2.

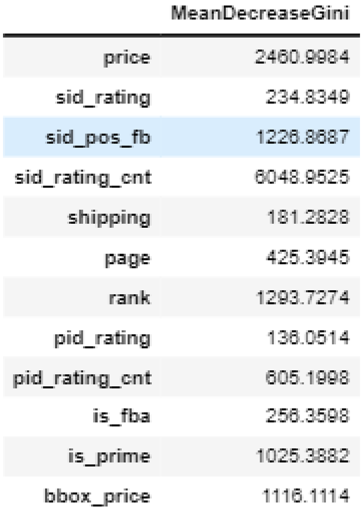
In part d, It is found that amazon wins the buy box with approximately 92% percent against the other sellers. So, we decided to convert the “sid” column to a binary feature called “is_Amazon” to specify whether the seller is amazon or not. In addition, from our experience on the training data, the boy bux price should be close to the minimum price. Therefore, we want to add “price gap ratio” feature to our data. Also, in part c, we found that sid_rating, sid_rating_cnt, and sid_pos_fb are the important factors on appearing in the buy box. From our personal experience, we know that sid_rating and sid_pos_fb are highly correlated with each other.

When we construct a correlation table among them, the result occurs exactly the way we expected:



	sid_rating	sid_pos_fb	sid_rating_cnt
sid_rating	1.00000000	0.9916615	0.09321616
sid_pos_fb	0.99166154	1.0000000	0.10406566
sid_rating_cnt	0.09321616	0.1040657	1.00000000

Therefore, we want to choose one of them to construct a model. We construct a basic Random Forest model to understand which one of them is a more important feature. The results are the following:



	MeanDecreaseGini
price	2460.9984
sid_rating	234.8349
sid_pos_fb	1226.8687
sid_rating_cnt	6048.9525
shipping	181.2828
page	425.3945
rank	1293.7274
pid_rating	136.0514
pid_rating_cnt	605.1998
is_fba	256.3598
is_prime	1025.3882
bbox_price	1116.1114

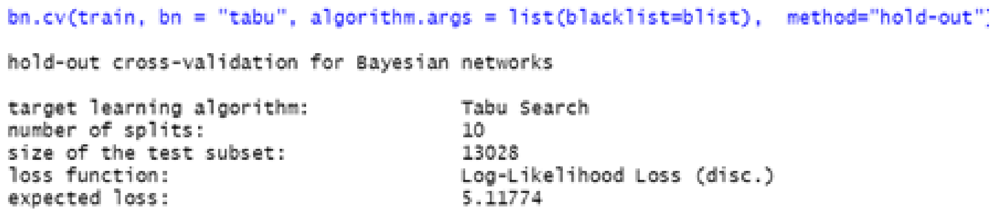
In Random Forest models, the larger the gini index the more important the feature. As can be seen, the gini index of sid_pos_fb is higher than the index of sid_rating. Therefore, we decided to select sid_pos_fb as one the nodes in our model.

Then, we also found that is_fba and is_prime features are also important features for 5 of the 9 different products. In addition, we realized that a small number of sellers other than amazon have is_prime = yes and is_fba = yes property. Therefore, we thought that it is better to put "is_Amazon" feature since it mostly contains the is_fba and is_prime information. Therefore, in this part we decided not to include these factors to keep the model as small as possible. Also, in part e, we realized that rank is important and the chance of winning increases as we approach rank 0. Therefore, this will be included in our model.

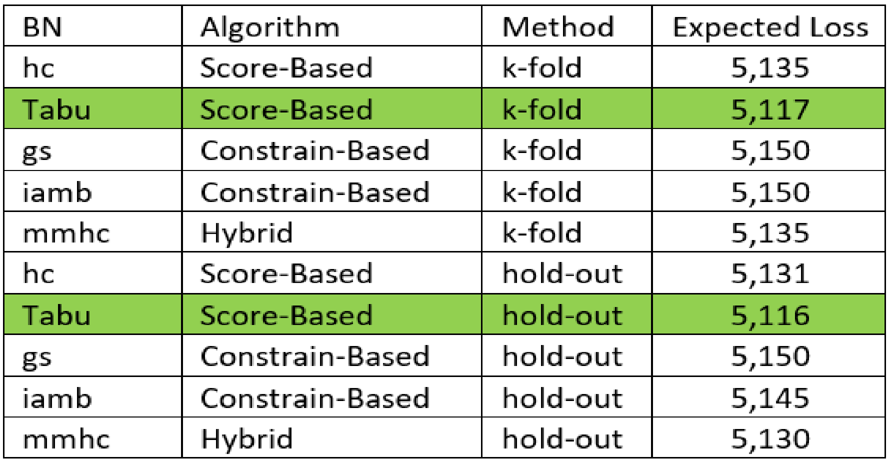
Since other columns such as product rating or product rating count are dependent on the product type we discard these informations from our model. We also remove the informations of "epoc" and "pid" since they have nothing to do with the bbox column.
As a result our single final model will contain is_Amazon, price gap ratio, rank, sid_pos_fb and the output variable bbox.

Question 3.

Five different methods (Hill Climbing, Tabu Search, Grow Shrink, Incremental Association and Max-Min Hill Climbing) which are scored-based, scored-based, constraint-based, constraint-based and hybrid algorithms respectively, are used to learn structure of BN with a given train data.
After whitelist and blacklist created, all algorithms make the same DAG and score. Then, cross-validation is made with two different methods (k-fold and hold-out). Tabu search algorithm expects minimum loss on both methods against other methods. Boosttrapping is also give the same DAG with tabu search. Thus, tabu search algorithm is used for the rest of the study.



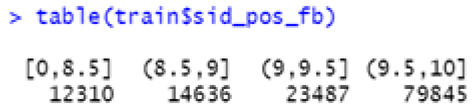
```
bn.cv(train, bn = "tabu", algorithm.args = list(blacklist=blist),  method="hold-out"

hold-out cross-validation for Bayesian networks

target learning algorithm:              Tabu Search
number of splits:                       10
size of the test subset:                13028
loss function:                          Log-Likelihood Loss (disc.)
expected loss:                          5.11774
```



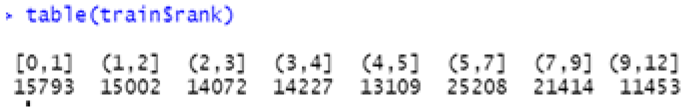
BN	Algorithm	Method	Expected Loss
hc	Score-Based	k-fold	5,135
Tabu	Score-Based	k-fold	5,117
gs	Constrain-Based	k-fold	5,150
iamb	Constrain-Based	k-fold	5,150
mmhc	Hybrid	k-fold	5,135
hc	Score-Based	hold-out	5,131
Tabu	Score-Based	hold-out	5,116
gs	Constrain-Based	hold-out	5,150
iamb	Constrain-Based	hold-out	5,145
mmhc	Hybrid	hold-out	5,130

"is_Amazon", "price_gap_ratio", "sid_pos_fb", "rank" and "bbox" parameters are used to learn DAG. "price_gap_ratio", and "sid_pos_fb" parameters does not discrete and it is known that continues nodes cannot be parent of discrete nodes. Therefore, our model tends to be misinterpreted. To prevent this problem these two parameters transformed to discrete by making intervals so that it can be treated as factor.

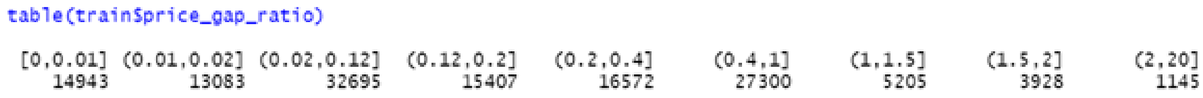
While converting continues parameter to discrete intervals, two factor is considered. The first one is expected importance of value. For instance, customer’s decision may not change if the “sid_pos_rate” is 9.9 or 9.8. Therefore, we start with 0.5 interval but end with 0:7 interval since it does not make importance once it goes below 7. The second factor is almost equally and large enough sized interval. This part is important. If there is no data on train corresponding to test observation, algorithm cannot predict that test observation. To solve that interval length should be increased. However, to increase the effect of parameter interval length should be decreased. Here, it is tried to find optimum interval values which maximize the prediction accuracy.



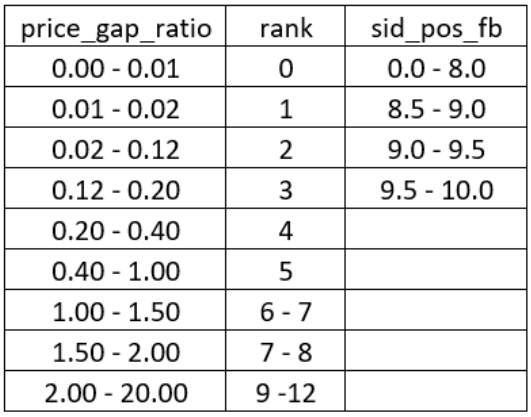
```
> table(train$sid_pos_fb)

 [0,8.5]  (8.5,9]  (9,9.5] (9.5,10] 
   12310    14636    23487    79845 
```



```
> table(train$rank)

[0,1]  (1,2]  (2,3]  (3,4]  (4,5]  (5,7]  (7,9] (9,12] 
15793  15002  14072  14227  13109  25208  21414  11453 
```



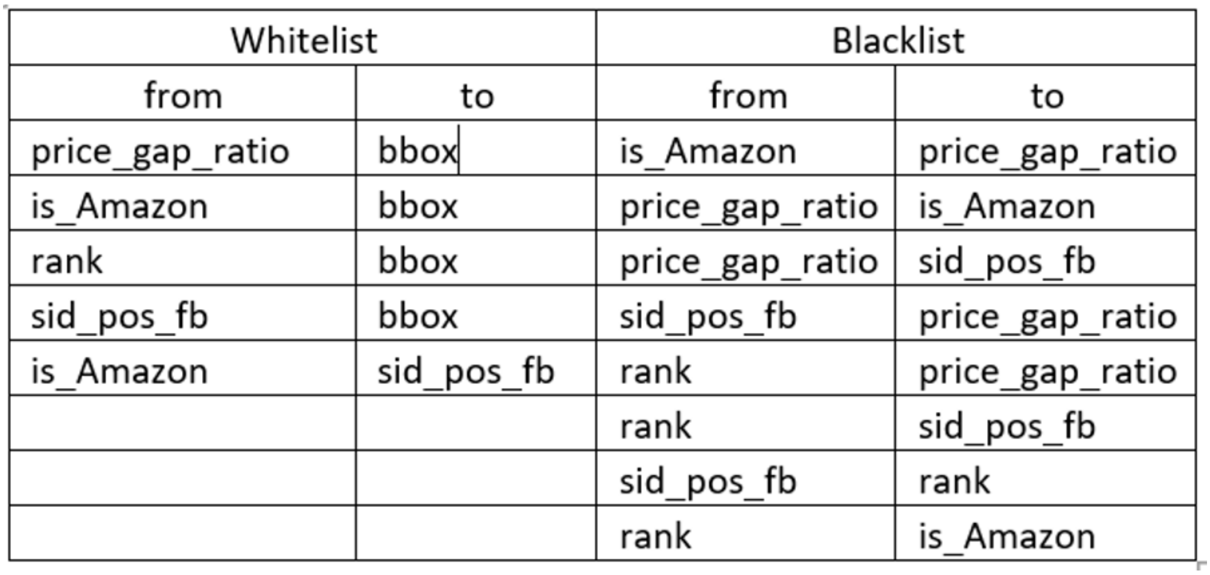
```
table(train$price_gap_ratio)

 [0,0.01] (0.01,0.02] (0.02,0.12]  (0.12,0.2]   (0.2,0.4]     (0.4,1]     (1,1.5]     (1.5,2]      (2,20] 
    14943       13083       32695       15407       16572       27300        5205        3928        1145 
```



price_gap_ratio	rank	sid_pos_fb
0.00 - 0.01	0	0.0 - 8.0
0.01 - 0.02	1	8.5 - 9.0
0.02 - 0.12	2	9.0 - 9.5
0.12 - 0.20	3	9.5 - 10.0
0.20 - 0.40	4	
0.40 - 1.00	5	
1.00 - 1.50	6 - 7	
1.50 - 2.00	7 - 8	
2.00 - 20.00	9 -12	

First DAG does not seem good model since we know some information between parameters. Therefore, whitelist and blacklist are used to force wanted arc and prevent unwanted arcs between parameters.

We are making whitelist and blacklist with the highest possible and logical information between nodes since learnt DAG is heuristic depends on whitelist and blacklist approval order. We start with whitelist since it is superior of blacklist. We expect all of chosen parameters should affect bbox. Therefore, we make whitelist for these and then continue to for the rest. The final whitelist and blacklist can be shown in below table.



Whitelist		Blacklist	
from	to	from	to
price_gap_ratio	bbox	is_Amazon	price_gap_ratio
is_Amazon	bbox	price_gap_ratio	is_Amazon
rank	bbox	price_gap_ratio	sid_pos_fb
sid_pos_fb	bbox	sid_pos_fb	price_gap_ratio
is_Amazon	sid_pos_fb	rank	price_gap_ratio
		rank	sid_pos_fb
		sid_pos_fb	rank
		rank	is_Amazon

Question 4.

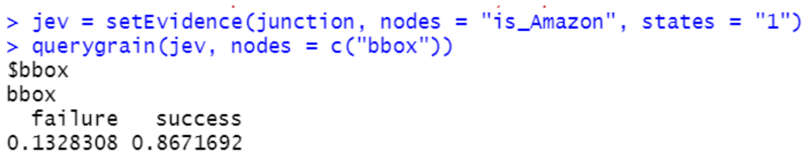
a.

There are 2*2*9*9*4 parameters of the model that are learnt by using training data. Since there are many parameters, only few of them are shown in the below section.

```
                    is_Amazon
bbox                0 1
  failure 0.9959866221
  success 0.0040133779

, , price_gap_ratio = (0.4,1], rank = (2,3], sid_pos_fb = (9,9.5]

          is_Amazon
bbox                0 1
  failure 0.9969040248
  success 0.0030959752

, , price_gap_ratio = (1,1.5], rank = (2,3], sid_pos_fb = (9,9.5]

Parameters of node price_gap_ratio (multinomial distribution)
Conditional probability table:

, , price_gap_ratio = [0,0.01]
          is_Amazon
rank                0           1
  [0,1]  0.365258462 0.935327635
  (1,2]  0.188664393 0.006837607
  (2,3]  0.121315490 0.023361823
  (3,4]  0.116767253 0.001709402
  (4,5]  0.059302020 0.015099715
  (5,7]  0.057990029 0.017663818
  (7,9]  0.060788944 0.000000000
  (9,12] 0.029913409 0.000000000

, , price_gap_ratio = (0.01,0.02]
          is_Amazon
rank                0           1
  [0,1]  0.077611940 0.005665722
  (1,2]  0.101256874 0.056657224
  (2,3]  0.107541241 0.314447592
  (3,4]  0.097486253 0.235127479
  (4,5]  0.103534957 0.198300283
  (5,7]  0.207227023 0.184135977
  (7,9]  0.207069914 0.005665722
  (9,12] 0.098271799 0.000000000


Parameters of node sid_pos_fb (multinomial distribution)
Conditional probability table:

          is_Amazon
sid_pos_fb          0           1
  [0,8.5]  0.0989359 0.0000000
  (8.5,9]  0.1176300 0.0000000
  (9,9.5]  0.1887658 0.0000000
  (9.5,10] 0.5946682 1.0000000
```

b.

The success probability of winning bbox of amazon as a saler P(bbox=success | sid=amazon) is calculated as 86.7% with below function:



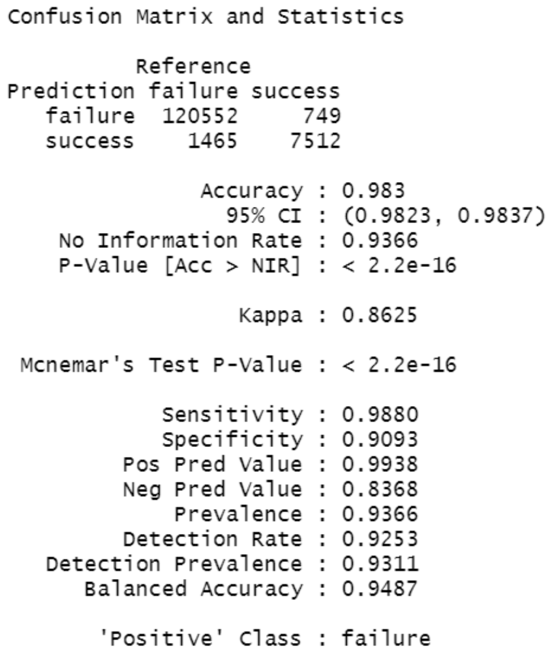
```
> jev = setEvidence(junction, nodes = "is_Amazon", states = "1")
> querygrain(jev, nodes = c("bbox"))
$bbox
bbox
  failure   success 
0.1328308 0.8671692 
```

c.

Firstly, prediction model with the chosen dag and parameters ("is_Amazon", "price_gap_ratio", "sid_pos_fb" and "rank") is created. Then, prediction made with training data to see the accuracy level of model and it is calculated as 98.3%. Once test result are uploaded, prediction made with test data on the same prediction model similarly. The accuracy of test model predictions calculated as 93.85%. Confusion matrix of both train and test data prediction results are shown in the below:

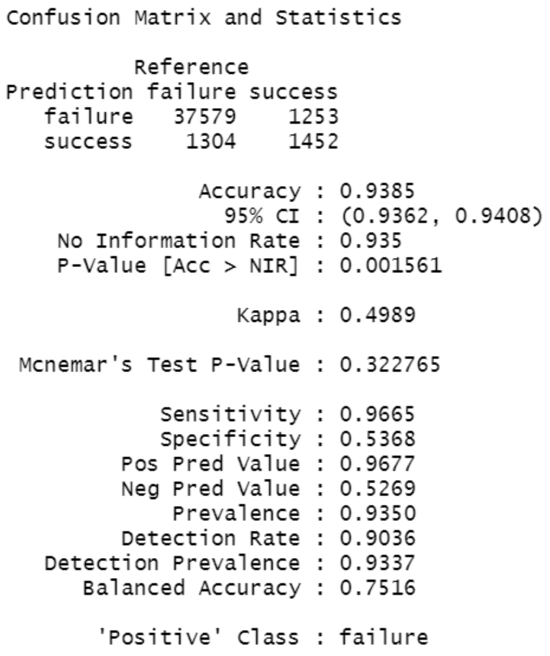
d.

Prediction on train data learning from train data



```
Confusion Matrix and Statistics

          Reference
Prediction failure success
   failure  120552     749
   success    1465    7512
                                          
               Accuracy : 0.983           
                 95% CI : (0.9823, 0.9837)
    No Information Rate : 0.9366          
    P-Value [Acc > NIR] : < 2.2e-16       
                                          
                  Kappa : 0.8625          
                                          
 Mcnemar's Test P-Value : < 2.2e-16       
                                          
            Sensitivity : 0.9880          
            Specificity : 0.9093          
         Pos Pred Value : 0.9938          
         Neg Pred Value : 0.8368          
             Prevalence : 0.9366          
         Detection Rate : 0.9253          
   Detection Prevalence : 0.9311          
      Balanced Accuracy : 0.9487          
                                          
       'Positive' Class : failure         
```

Prediction on test data learning from train data



```
Confusion Matrix and Statistics

          Reference
Prediction failure success
   failure   37579    1253
   success    1304    1452
                                          
               Accuracy : 0.9385          
                 95% CI : (0.9362, 0.9408)
    No Information Rate : 0.935           
    P-Value [Acc > NIR] : 0.001561        
                                          
                  Kappa : 0.4989          
                                          
 Mcnemar's Test P-Value : 0.322765        
                                          
            Sensitivity : 0.9665          
            Specificity : 0.5368          
         Pos Pred Value : 0.9677          
         Neg Pred Value : 0.5269          
             Prevalence : 0.9350          
         Detection Rate : 0.9036          
   Detection Prevalence : 0.9337          
      Balanced Accuracy : 0.7516          
                                          
       'Positive' Class : failure         
```

Question 5.

In this project, our main goal is to understand the decision behind amazon's algorithm for choosing the winner of buy box. In the first parts of the report, it is explained that price has a great effect on bbox decision. Therefore, we created a new feature called "price gap ratio". Since we are not allowed to have continuous variables in our model, we decided to consider discretizing "price gap ratio". We tried to do this process consciously as explained in part 3 and 4. However, we realized that we could have some price gap ratios in the test data which are out of the training ranges. Therefore, without knowing the test data ranges we should not divide the continuous variables into intervals. However, it is not possible to know test data in real life. In addition, although our accuracy on the test data looks good (93.85%) we should not focus on the accuracy values only but the other measures such as balanced accuracy since in our training data, we have more failures than successes. In our case, balanced accuracy results are in the acceptable range. As a result, if we want to use continuous variables to learn the model, we should convert these into factors carefully not to have a problem in the prediction process.

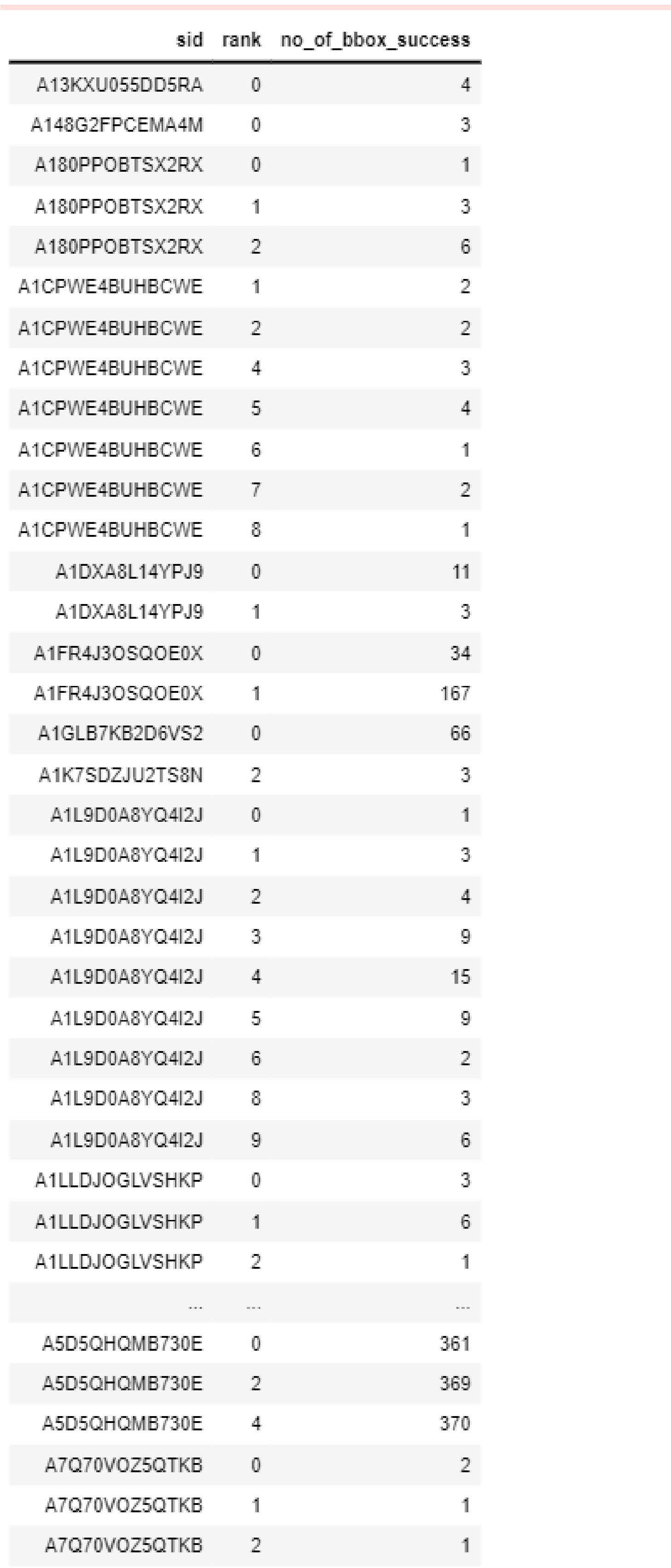
Shipment costs might affect customers' opinions. Therefore, it might be included in the model. We try to apply this parameter to the model; however, its effect change according to the product price. Since our model is general for any model, including this factor in the model may mislead the algorithm. It can be useful if one model created per product. However, it is not an efficient way for amazon since there is a billions type of product. This approach might combine with price_gap_ratio descriptive statistics as below:

$$price\_ship\_ratio = \frac{Current(Price * 0.8 + Shipping * 0.2) - Min(Price * 0.8 + Shipping * 0.2)}{Min(Price * 0.8 + Shipping * 0.2)}$$

In conclusion, Model has 4 input parameters and lower price_gap_ratio (lower price), lower rank, higher x positive feedback rating seller has a high chance to win buy box. Additionally, amazon as a seller has huge effects on winning buy box. We obtain %93.85 as a model accuracy. It might be increased if one model created per product as it is mentioned before due to different value intervals of parameters.

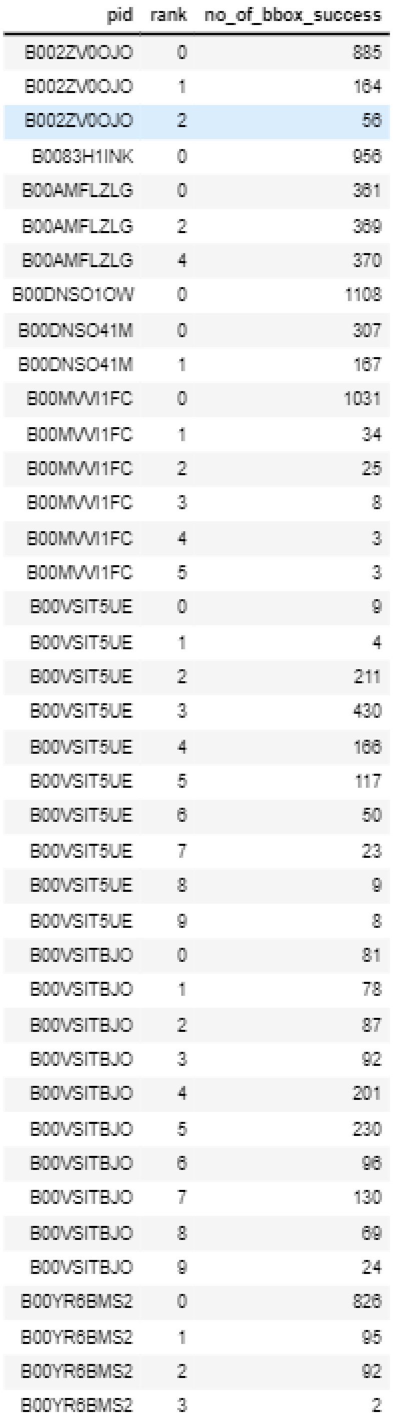
Appendices:

Rank information by seller:



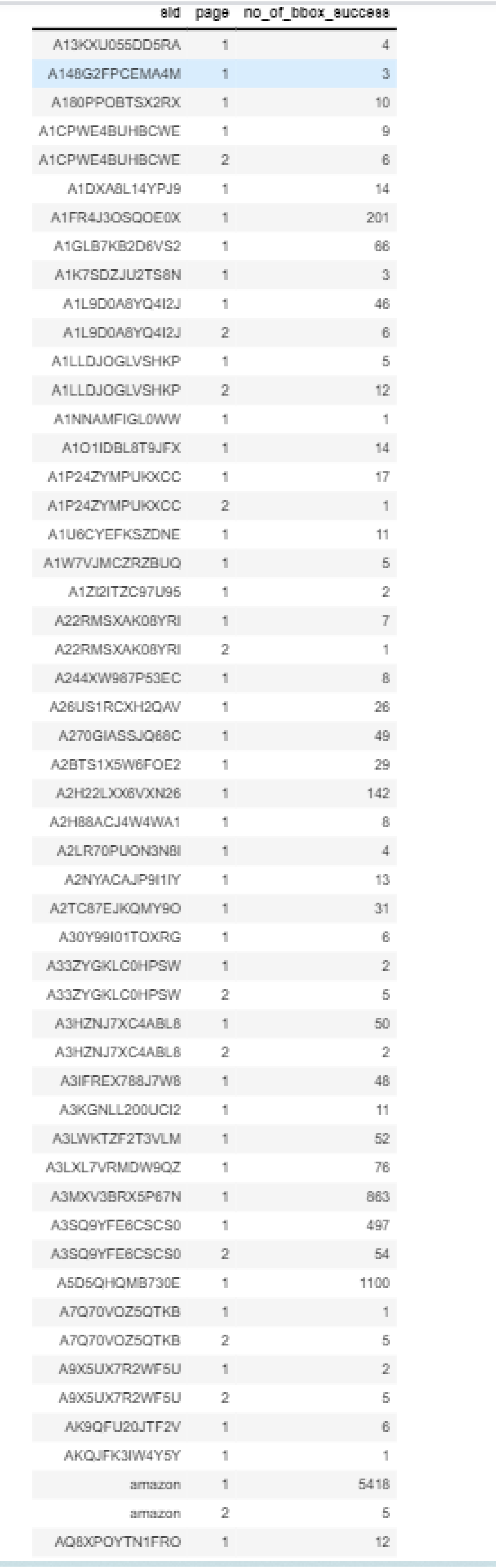
sid	rank	no_of_bbox_success
A13KXU055DD5RA	0	4
A148G2FPCEMA4M	0	3
A180PPOBTSX2RX	0	1
A180PPOBTSX2RX	1	3
A180PPOBTSX2RX	2	6
A1CPWE4BUHBCWE	1	2
A1CPWE4BUHBCWE	2	2
A1CPWE4BUHBCWE	4	3
A1CPWE4BUHBCWE	5	4
A1CPWE4BUHBCWE	6	1
A1CPWE4BUHBCWE	7	2
A1CPWE4BUHBCWE	8	1
A1DXA8L14YPJ9	0	11
A1DXA8L14YPJ9	1	3
A1FR4J3OSQOE0X	0	34
A1FR4J3OSQOE0X	1	167
A1GLB7KB2D6VS2	0	66
A1K7SDZJU2TS8N	2	3
A1L9D0A8YQ4I2J	0	1
A1L9D0A8YQ4I2J	1	3
A1L9D0A8YQ4I2J	2	4
A1L9D0A8YQ4I2J	3	9
A1L9D0A8YQ4I2J	4	15
A1L9D0A8YQ4I2J	5	9
A1L9D0A8YQ4I2J	6	2
A1L9D0A8YQ4I2J	8	3
A1L9D0A8YQ4I2J	9	6
A1LLDJOGLVSHKP	0	3
A1LLDJOGLVSHKP	1	6
A1LLDJOGLVSHKP	2	1
...	...	...
A5D5QHQMB730E	0	361
A5D5QHQMB730E	2	369
A5D5QHQMB730E	4	370
A7Q70VOZ5QTKB	0	2
A7Q70VOZ5QTKB	1	1
A7Q70VOZ5QTKB	2	1

Rank information by product:



pid	rank	no_of_bbox_success
B002ZV0OJO	0	885
B002ZV0OJO	1	164
B002ZV0OJO	2	56
B0083H1INK	0	956
B00AMFLZLG	0	361
B00AMFLZLG	2	369
B00AMFLZLG	4	370
B00DNSO1OW	0	1108
B00DNSO41M	0	307
B00DNSO41M	1	167
B00MVVI1FC	0	1031
B00MVVI1FC	1	34
B00MVVI1FC	2	25
B00MVVI1FC	3	8
B00MVVI1FC	4	3
B00MVVI1FC	5	3
B00VSIT5UE	0	9
B00VSIT5UE	1	4
B00VSIT5UE	2	211
B00VSIT5UE	3	430
B00VSIT5UE	4	166
B00VSIT5UE	5	117
B00VSIT5UE	6	50
B00VSIT5UE	7	23
B00VSIT5UE	8	9
B00VSIT5UE	9	8
B00VSITBJO	0	81
B00VSITBJO	1	78
B00VSITBJO	2	87
B00VSITBJO	3	92
B00VSITBJO	4	201
B00VSITBJO	5	230
B00VSITBJO	6	96
B00VSITBJO	7	130
B00VSITBJO	8	69
B00VSITBJO	9	24
B00YR6BMS2	0	826
B00YR6BMS2	1	95
B00YR6BMS2	2	92
B00YR6BMS2	3	2

Page information by seller:



sid	page	no_of_bbox_success
A13KXU055DD5RA	1	4
A148G2FPCEMA4M	1	3
A180PPOBTSX2RX	1	10
A1CPWE4BUHBCWE	1	9
A1CPWE4BUHBCWE	2	6
A1DXA8L14YPJ9	1	14
A1FR4J3OSQOE0X	1	201
A1GLB7KB2D6VS2	1	66
A1K7SDZJU2TS8N	1	3
A1L9D0A8YQ4I2J	1	46
A1L9D0A8YQ4I2J	2	6
A1LLDJOGLVSHKP	1	5
A1LLDJOGLVSHKP	2	12
A1NNAMFIGL0WW	1	1
A1O1IDBL8T9JFX	1	14
A1P24ZYMPUKXCC	1	17
A1P24ZYMPUKXCC	2	1
A1U6CYEFKSZDNE	1	11
A1W7VJMCZRZBUQ	1	5
A1ZI2ITZC97U95	1	2
A22RMSXAK08YRI	1	7
A22RMSXAK08YRI	2	1
A244XW987P53EC	1	8
A26US1RCXH2QAV	1	26
A270GIASSJQ68C	1	49
A2BTS1X5W6FOE2	1	29
A2H22LXX6VXN26	1	142
A2H88ACJ4W4WA1	1	8
A2LR70PUON3N8I	1	4
A2NYACAJP9I1IY	1	13
A2TC87EJKQMY9O	1	31
A30Y99I01TOXRG	1	6
A33ZYGKLC0HPSW	1	2
A33ZYGKLC0HPSW	2	5
A3HZNJ7XC4ABL8	1	50
A3HZNJ7XC4ABL8	2	2
A3IFREX788J7W8	1	48
A3KGNLL200UCI2	1	11
A3LWKTZF2T3VLM	1	52
A3LXL7VRMDW9QZ	1	76
A3MXV3BRX5P67N	1	863
A3SQ9YFE6CSCS0	1	497
A3SQ9YFE6CSCS0	2	54
A5D5QHQMB730E	1	1100
A7Q70VOZ5QTKB	1	1
A7Q70VOZ5QTKB	2	5
A9X5UX7R2WF5U	1	2
A9X5UX7R2WF5U	2	5
AK9QFU20JTF2V	1	6
AKQJFK3IW4Y5Y	1	1
amazon	1	5418
amazon	2	5
AQ8XPOYTN1FRO	1	12