MMU
MULTIMEDIA UNIVERSITY

TDS2101 - INTRODUCTION TO DATA SCIENCE

Topic: Big Data in the Retail Sector

Prepared by:

Oi Zhen Fan 1181300513

Lee Xi Jie 1161204459

Table of Contents

Part A

Part B

Part A

1.0 Introduction

Data science and big data as a whole have revolutionized several industries in this era of globalization, allowing the industries and organization to discover the patterns, understand the customers' behaviors in order to produce quality products. For example, big data acts as a vital role in the retails section, which can be used to identify a lot of patterns from the big data. For example, market basket analysis, sales forecasting and etc. In our project, It is an interesting topic as it is related to the retail sector as we can see the patterns from the big dataset that we found.

2.0 Business Use Cases

2.1 Amazon

Amazon.com is the 3rd largest retailer, and can be known as the largest e-commerce site to have existed. From our studies, Amazon has millions of users' information from daily transactions. As such, Amazon takes the opportunity that uses the data to create a personalized shopping experience. For example, it uses the data to predict the sales, recommendation products for the users based on the customer behaviors. Moreover, Amazon has offered AWS (Amazon Web Services) which contains big data to help the companies to create useful applications.

2.2 Walmart

Walmart is the world's largest retailer and company by revenue, studied by [1]. Walmart not only has the website applications, they also own the physical shop in the store. Same goes to Walmart, they collected millions of data that can be used to predict the sales and boost their sales for their company. Furthermore, Walmart studies the big data to understand the busy hours across the data and determine how many counters are needed to be opened at the particular time range.

3.0 Why is Big Data important nowadays?

Nowadays, more retail companies are growing. Hence it makes it more data-driven from day to day. Inside the data, there usually is important insights for the business strategies. As we mentioned previously, big data provides the company an opportunity to collect the customers' behaviors and the patterns from each of the markets. As a result, big data can be converted and digested in order to get a meaningful information

4.0 Potential Benefits

Companies:

1. To maximize the profits and minimize the losses of their sales.
2. To provide knowledge to the companies that products can be sold in packages to attract customers.
3. To help the companies to determine which brand of products they should add-on purchase or shouldn't purchase.

Customers:

1. To provide a list of high quality products that are highly recommended by previous buyers.
2. To help the customers to know different prices of the same products that are selling from the different merchants.

Part B

1.0 Questions to Answer

We come out with a total of 8 questions in order to let us discover some interesting patterns, which might help the companies to increase their income significantly or help the customers to buy products with the best pricing. Not only that, a butterfly effect of benefit both sides might also occur.

1.1 Descriptive and Exploratory Questions

1. How much sales can be generated based on the brands?
2. Which products' brands conducted the most promotions?
3. How is the pricing strategy of a product among the different merchants?
4. How is the pricing strategy among the top 10 brands?
5. What are the trends of the top brands among the year?
6. Do the features correlate to each other?

1.2 Predictive Questions

1. What are the factors that can be used to predict the promotion of a product ?
2. What products do the retailers need to purchase to meet customer needs?

2.0 Dataset In Use

2.1 Source

We obtained our dataset from data.world, a sampled dataset of Electronic Products and Pricing Data by nsinfiniti, which is originally obtained from Datafiniti's product database. Datafiniti is a data science team that provides access to data sourced from over thousand of websites and allows their customers to purchase or use their product API to access their E-commerce data from various retail websites. In addition, the customers are also able to integrate with the data.

2.2 Dataset Descriptions

This dataset is an E-commerce product dataset that contains a list of over 7000 electronic products from 2014 to 2018. It consists of all the pricing information of the products, for example, the products' shipping, availability, condition, and more. It also includes the products' brand, merchant, manufacturer, and other useful information.

The below table shows the data dictionary including their data types and description:

Feature Name	Type	Description
id	object	The product's ID
prices.amountMax	float	The maximum price value listed
prices.amountMin	float	The minimum price value listed
prices.availability	object	A true or false if this product is available at this price
prices.condition	object	The condition of the product when being sold at this price
prices.currency	object	The currency listed for amountMin and amountMax
prices.dateSeen	object	A list of dates when this price was seen
prices.isSale	bool	A true/false for whether or not this price is a sale/discounted price
prices.merchant	object	The merchant and/or website selling at this price
prices.shipping	object	The shipping terms associated with this price
prices.sourceURLs	object	A list of URLs where this price was seen
asins	object	The ASIN (Amazon identifier) used for this product
brand	object	The brand name of this product

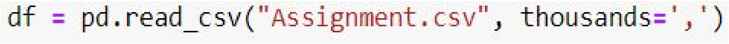
categories	object	The list of category keywords used for this product across multiple sources
dateAdded	object	The date this product was first added to the product database
dateUpdated	object	The most recent date this product
ean	float	The EAN codes for this product
imageURLs	object	A list of image URLs for this product
keys	object	A list of internal Datafiniti identifiers for this product
manufacturer	object	The manufacturer of this product
manufacturerNumber	object	The manufacturer or model number of this product
name	object	The product's name
primaryCategories	object	A list of standardized categories to which this product belong
sourceURLs	object	A list of URLs used to generate data for this product
upc	object	The UPC code for this product
weight	object	The weight of this product

3.0 Data Cleaning/Preprocessing

3.1 Data Cleaning

3.1.1 Import Dataset

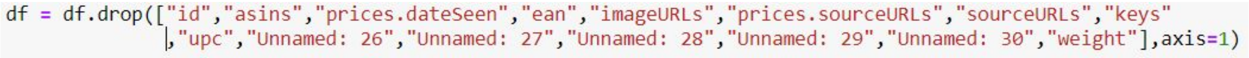
First and foremost, we read in our Electronic Products and Pricing Data dataset.



```
df = pd.read_csv("Assignment.csv", thousands=',')
```

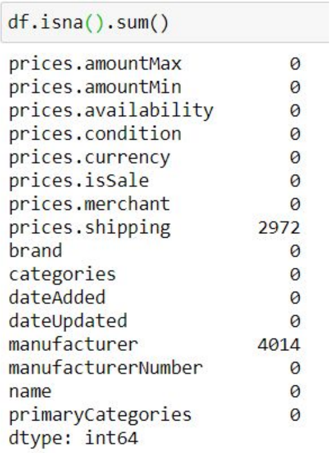
3.1.2 Drop redundant and ineffectual features

We decide to drop the features from the dataset that we think are not ineffectual and redundant to do further exploration or prediction.



```
df = df.drop(["id","asins","prices.dateSeen","ean","imageURLs","prices.sourceURLs","sourceURLs","keys"
              ,"upc","Unnamed: 26","Unnamed: 27","Unnamed: 28","Unnamed: 29","Unnamed: 30","weight"],axis=1)
```

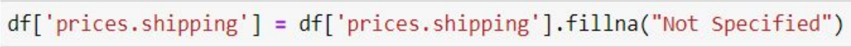
3.1.3 Handling Missing Values



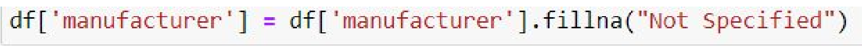
```
df.isna().sum()

prices.amountMax        0
prices.amountMin        0
prices.availability     0
prices.condition        0
prices.currency         0
prices.isSale           0
prices.merchant         0
prices.shipping      2972
brand                   0
categories              0
dateAdded               0
dateUpdated             0
manufacturer         4014
manufacturerNumber      0
name                    0
primaryCategories       0
dtype: int64
```

From the figure above we can see that prices.shipping and manufacturer have a large amount of missing values. Since both of them are categorical variables, we choose to fill those values with "Not Specified" because we cannot simply fill in categorical variables.

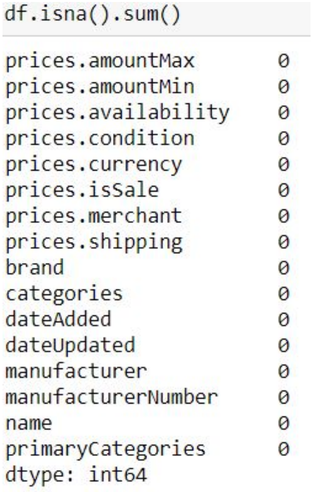


```
df['prices.shipping'] = df['prices.shipping'].fillna("Not Specified")
```



```
df['manufacturer'] = df['manufacturer'].fillna("Not Specified")
```

Below are the following result:



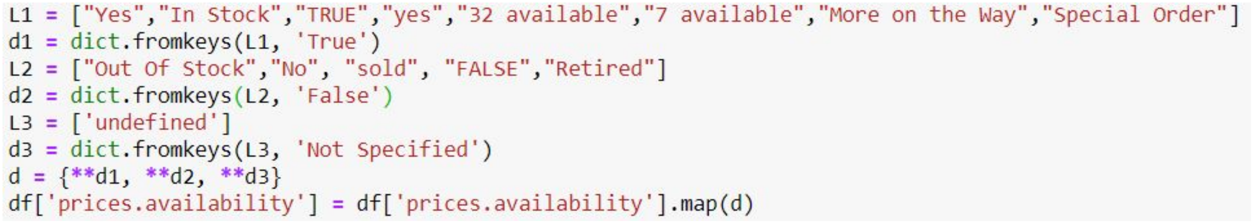
```
df.isna().sum()

prices.amountMax       0
prices.amountMin       0
prices.availability    0
prices.condition       0
prices.currency        0
prices.isSale          0
prices.merchant        0
prices.shipping        0
brand                  0
categories             0
dateAdded              0
dateUpdated            0
manufacturer           0
manufacturerNumber     0
name                   0
primaryCategories      0
dtype: int64
```

3.1.4 Mapping Values

After filling in the missing values, we found out that some values of a few columns have the same meaning or they are inconsistent data that have naming conventions. For example, in the prices.availability column, "Yes","In Stock","TRUE","yes","32 available","7 available","More on the Way","Special Order" have a very similar meaning. So, we decided to map them into the same value, which is “True”. And then we perform the same step for both “False” and “Not Specified”.

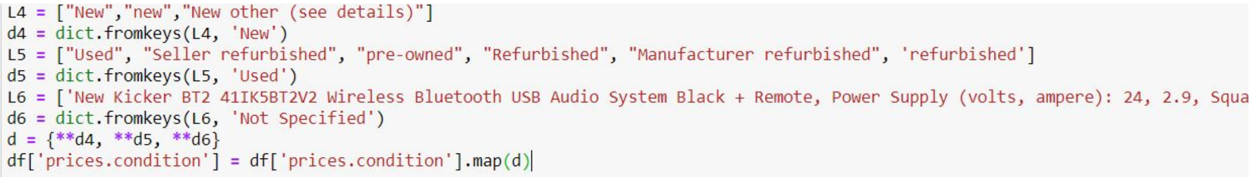
Values mapping for prices.availability:



```
L1 = ["Yes","In Stock","TRUE","yes","32 available","7 available","More on the Way","Special Order"]
d1 = dict.fromkeys(L1, 'True')
L2 = ["Out Of Stock","No", "sold", "FALSE","Retired"]
d2 = dict.fromkeys(L2, 'False')
L3 = ['undefined']
d3 = dict.fromkeys(L3, 'Not Specified')
d = {**d1, **d2, **d3}
df['prices.availability'] = df['prices.availability'].map(d)
```

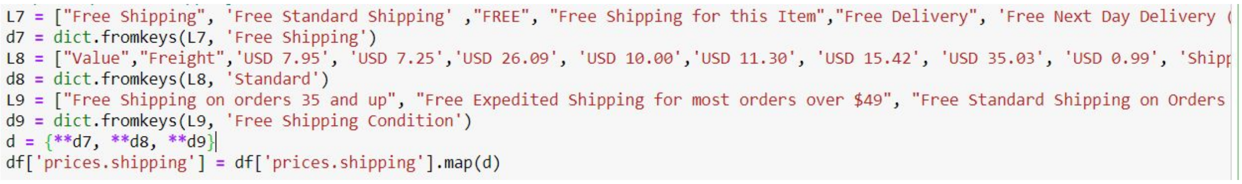
For the columns prices.condition, we discovered that there are some redundant values. Subsequently, we decided to map them as “Not Specified”.

Values mapping for prices.condition:



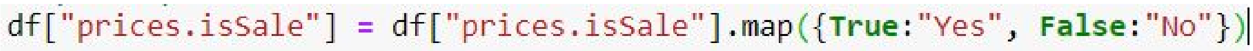
```
L4 = ["New","new","New other (see details)"]
d4 = dict.fromkeys(L4, 'New')
L5 = ["Used", "Seller refurbished", "pre-owned", "Refurbished", "Manufacturer refurbished", 'refurbished']
d5 = dict.fromkeys(L5, 'Used')
L6 = ['New Kicker BT2 41IK5BT2V2 Wireless Bluetooth USB Audio System Black + Remote, Power Supply (volts, ampere): 24, 2.9, Squa
d6 = dict.fromkeys(L6, 'Not Specified')
d = {**d4, **d5, **d6}
df['prices.condition'] = df['prices.condition'].map(d)
```

Values mapping for prices.shipping:



```
L7 = ["Free Shipping", 'Free Standard Shipping' ,"FREE", "Free Shipping for this Item","Free Delivery", 'Free Next Day Delivery (
d7 = dict.fromkeys(L7, 'Free Shipping')
L8 = ["Value","Freight",'USD 7.95', 'USD 7.25','USD 26.09', 'USD 10.00','USD 11.30', 'USD 15.42', 'USD 35.03', 'USD 0.99', 'Shipp
d8 = dict.fromkeys(L8, 'Standard')
L9 = ["Free Shipping on orders 35 and up", "Free Expedited Shipping for most orders over $49", "Free Standard Shipping on Orders
d9 = dict.fromkeys(L9, 'Free Shipping Condition')
d = {**d7, **d8, **d9}
df['prices.shipping'] = df['prices.shipping'].map(d)
```

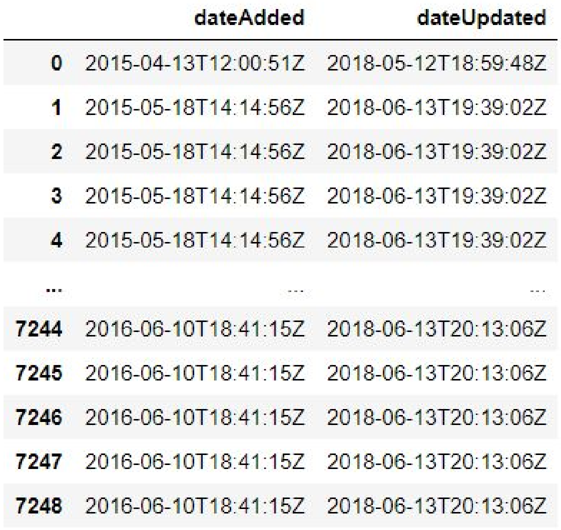
Values mapping for prices.isSale:



```
df["prices.isSale"] = df["prices.isSale"].map({True:"Yes", False:"No"})
```

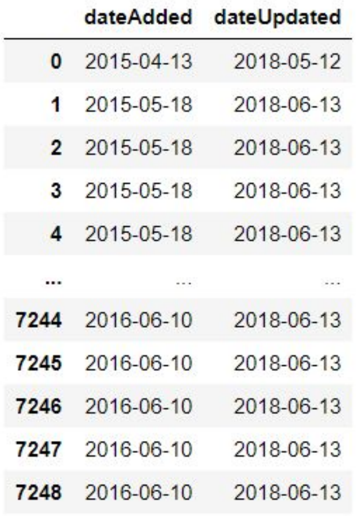
For the column dateAdded and dateUpdated, since the values of these two columns show the yy/mm/dd and also the exact time, we decided to just remain the yy/mm/dd in order to make us easier to do further exploration.

Before:



	dateAdded	dateUpdated
0	2015-04-13T12:00:51Z	2018-05-12T18:59:48Z
1	2015-05-18T14:14:56Z	2018-06-13T19:39:02Z
2	2015-05-18T14:14:56Z	2018-06-13T19:39:02Z
3	2015-05-18T14:14:56Z	2018-06-13T19:39:02Z
4	2015-05-18T14:14:56Z	2018-06-13T19:39:02Z
...	...	...
7244	2016-06-10T18:41:15Z	2018-06-13T20:13:06Z
7245	2016-06-10T18:41:15Z	2018-06-13T20:13:06Z
7246	2016-06-10T18:41:15Z	2018-06-13T20:13:06Z
7247	2016-06-10T18:41:15Z	2018-06-13T20:13:06Z
7248	2016-06-10T18:41:15Z	2018-06-13T20:13:06Z

After:

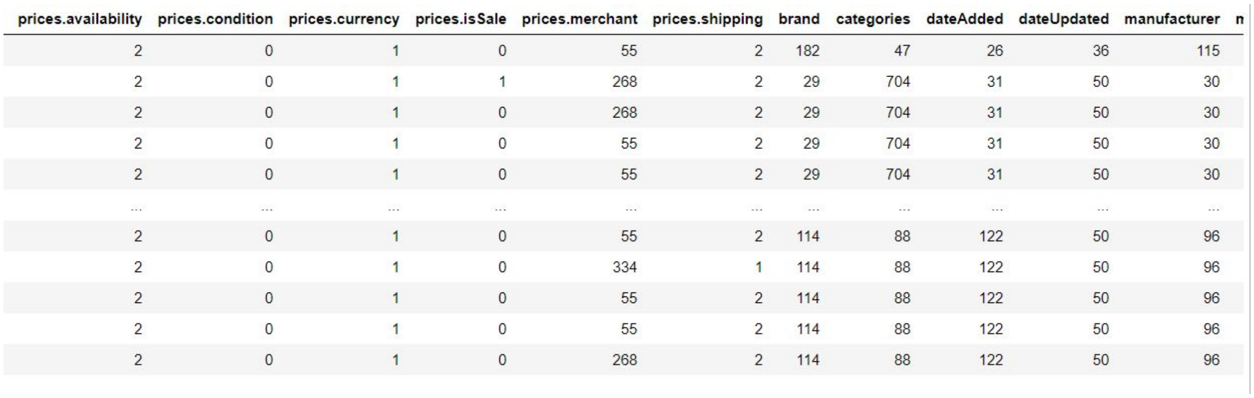


	dateAdded	dateUpdated
0	2015-04-13	2018-05-12
1	2015-05-18	2018-06-13
2	2015-05-18	2018-06-13
3	2015-05-18	2018-06-13
4	2015-05-18	2018-06-13
...	...	...
7244	2016-06-10	2018-06-13
7245	2016-06-10	2018-06-13
7246	2016-06-10	2018-06-13
7247	2016-06-10	2018-06-13
7248	2016-06-10	2018-06-13

3.2 Label Encoding

After mapping all the values, we perform label encoding by converting all the categorical features' values into numerical values. This is a vital step as we need to fit the values into machine learning algorithms in order to do prediction afterwards. In addition, since we will also do correlation checks later, label encoded dataset is also needed in this case.

Figure below shows the label encoded dataset:

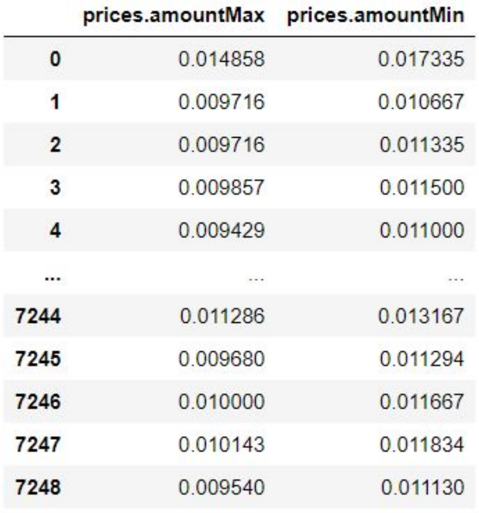


prices.availability	prices.condition	prices.currency	prices.isSale	prices.merchant	prices.shipping	brand	categories	dateAdded	dateUpdated	manufacturer	n
2	0	1	0	55	2	182	47	26	36	115	
2	0	1	1	268	2	29	704	31	50	30	
2	0	1	0	268	2	29	704	31	50	30	
2	0	1	0	55	2	29	704	31	50	30	
2	0	1	0	55	2	29	704	31	50	30	
...	...	...	...	...	...	...	...	...	...	...	
2	0	1	0	55	2	114	88	122	50	96	
2	0	1	0	334	1	114	88	122	50	96	
2	0	1	0	55	2	114	88	122	50	96	
2	0	1	0	55	2	114	88	122	50	96	
2	0	1	0	268	2	114	88	122	50	96	

3.3 Data Transformation

In this part, we perform min-max normalization to two particular numerical columns, which are prices.amountMax and prices.amountMin. The purpose of this is to eliminate the units of measurement and give an equal weight to the features selected. And most importantly, it will definitely help us to have a better understanding of our dataset as the numerical values will fall into a range between 0.0 and 1.0.

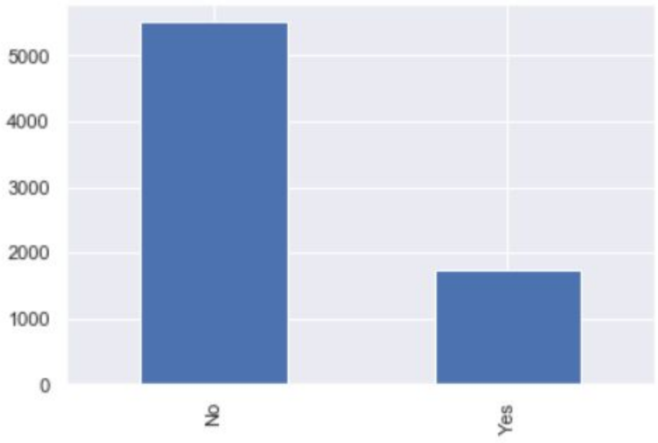
Figure below shows the values after performing the following normalization:



	prices.amountMax	prices.amountMin
0	0.014858	0.017335
1	0.009716	0.010667
2	0.009716	0.011335
3	0.009857	0.011500
4	0.009429	0.011000
...	...	...
7244	0.011286	0.013167
7245	0.009680	0.011294
7246	0.010000	0.011667
7247	0.010143	0.011834
7248	0.009540	0.011130

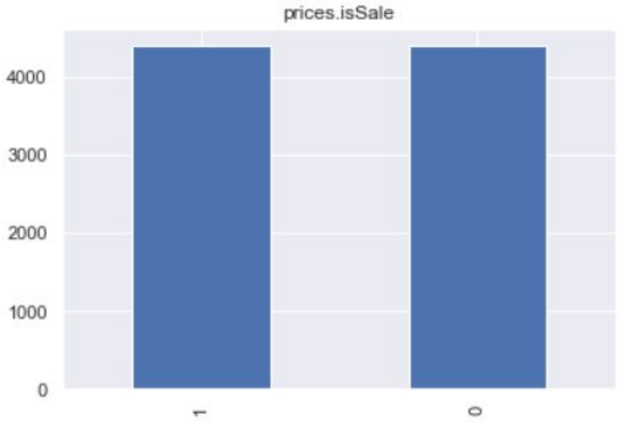
3.4 Handling Imbalanced Data

In our project, we choose price.isSale and price.availability as two of our target features with the intention of doing classification afterwards. And we found out that both of them are imbalanced. So, we decided to oversampled the target features and the oversampling technique we used is SMOTE. We use oversampling instead of undersampling because oversampling will not face the problem of losing data as this is not a large dataset as well. The purpose of dealing with imbalanced data is to increase accuracy when doing classification. In the case of price.isSale, there are two values, which are "No" and "Yes". The bar chart below displays their respective values:



We can clearly see that "No" is almost 4000 more than "Yes". Subsequently, we oversample it.

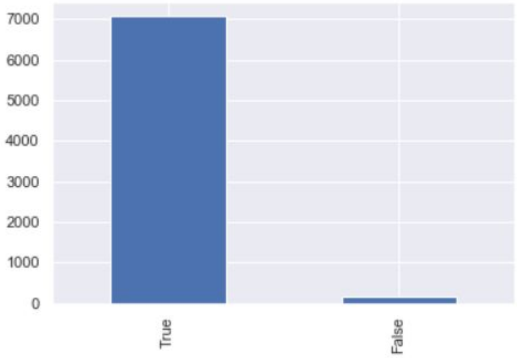
And below is the result:

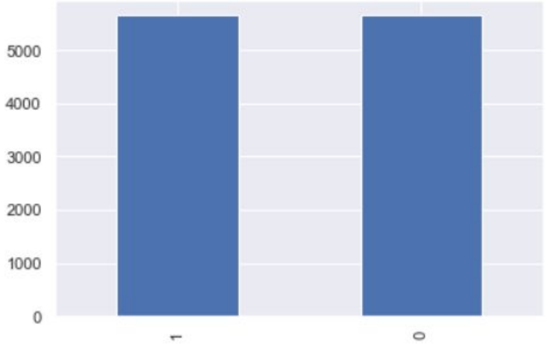


For another target feature prices.availability, there is another value “Not Specified” other than “Yes” and “No”, So, we drop it because it is pointless to predict the value as “Not Specified”

Figures below shows the before and after sampling data:

Before: After:

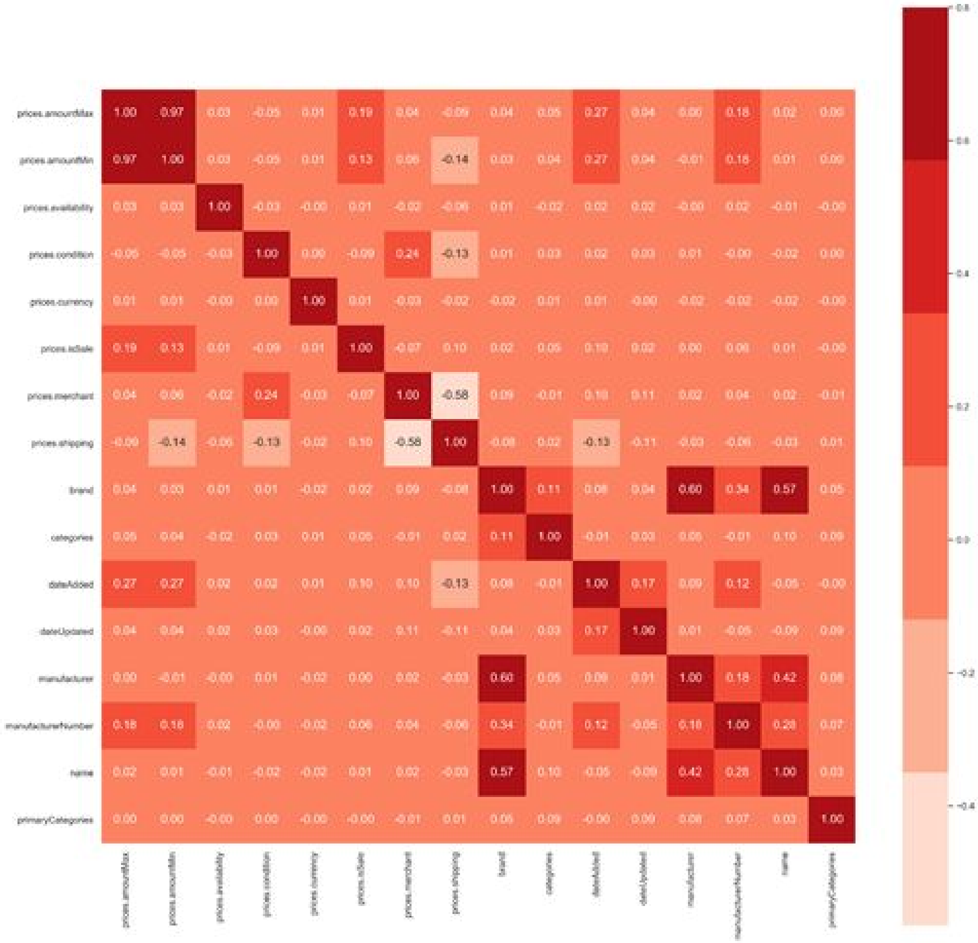




4.0 Exploratory Data Analysis

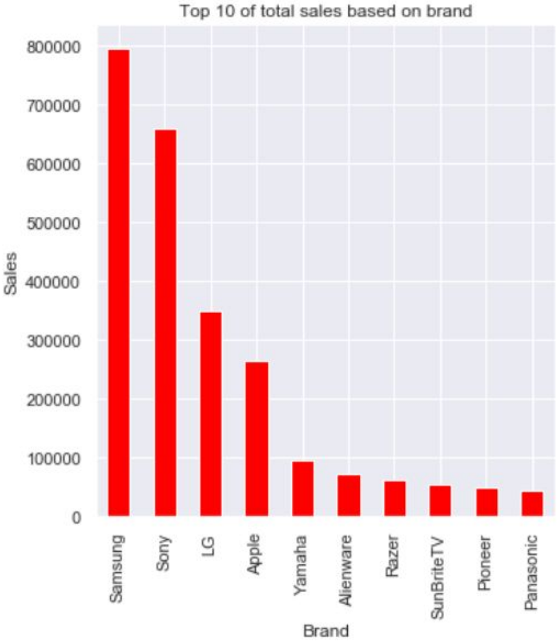
After we pre-processed and cleaned the dataset, we explored and analysed to figure out the answer which can answer the descriptive questions in the section we proposed the questions. By doing this exploratory data and analysis, we can have a better understanding of the datasets and discover the pattern by visualizing by the supporting graph.

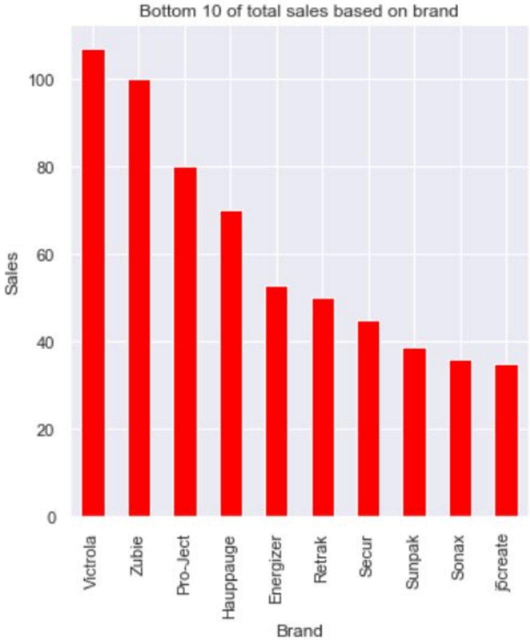
4.1 Do the features correlate to each other?



Better visualization to this figure can be seen from the path IDS Project/conf.png. Based on the heatmap above, there are more attributes in the dataset that show no correlation with each other. However, there are also some attributes that show the median-strength correlations manufacturer, brand and the name. Besides it also shows the weak correlation between the prices.merchant and the prices.shipping.

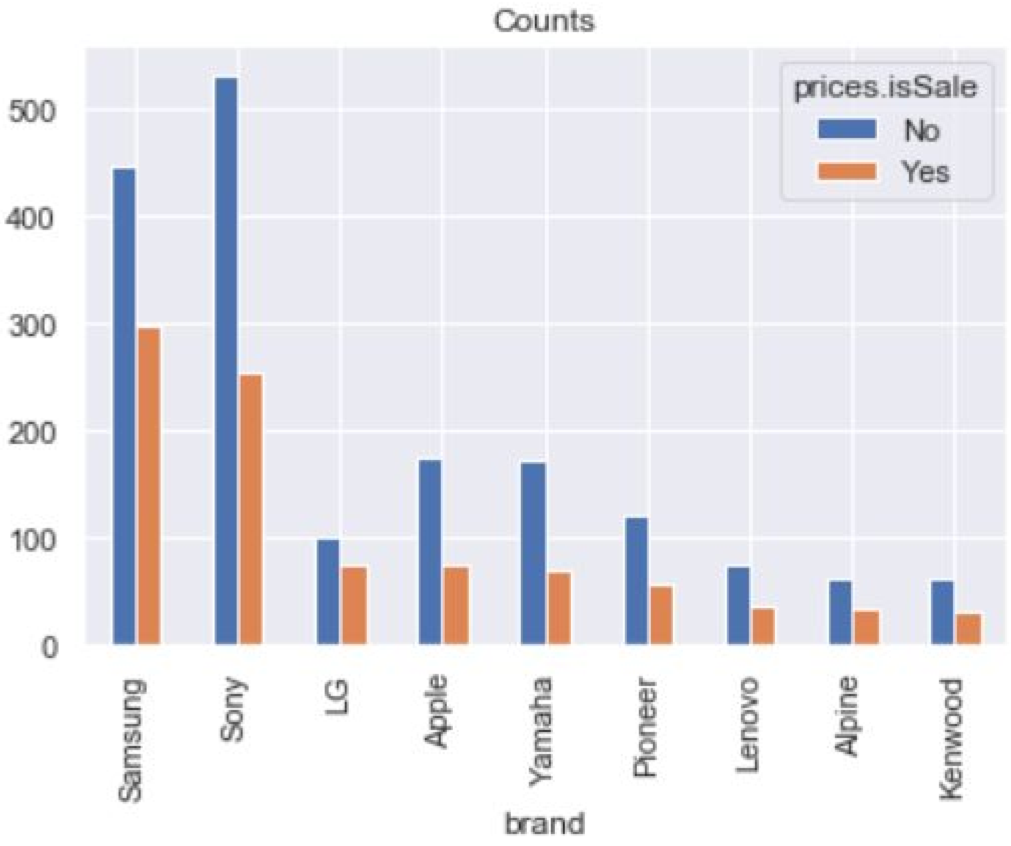
4.2 How much sales can be generated based on the brands?





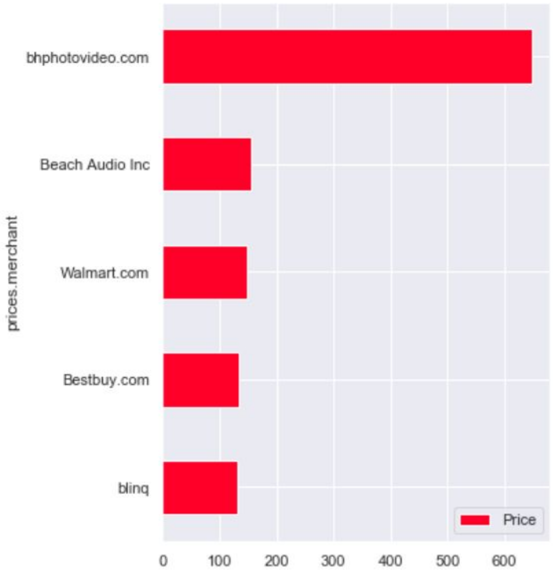
From the results we generated, we have made the brand with the top 10 and bottom 10. Hence from this histogram, we can clearly see that the Samsung brand is the top brand and the sales can reach 800000. Meanwhile the Bottom 10, the pcreate is the not really famous brand and the sales can go only 40 in the big data that we collected.

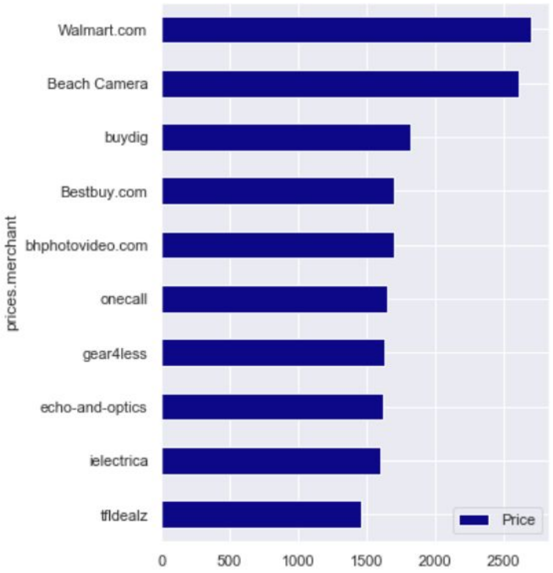
4.3 Which products' brands conducted the most promotions?

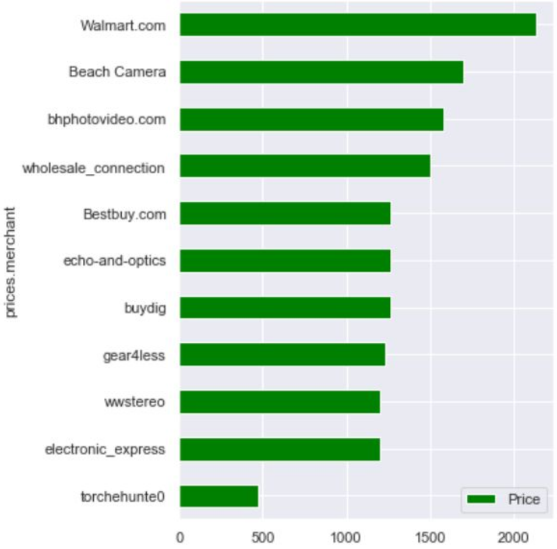


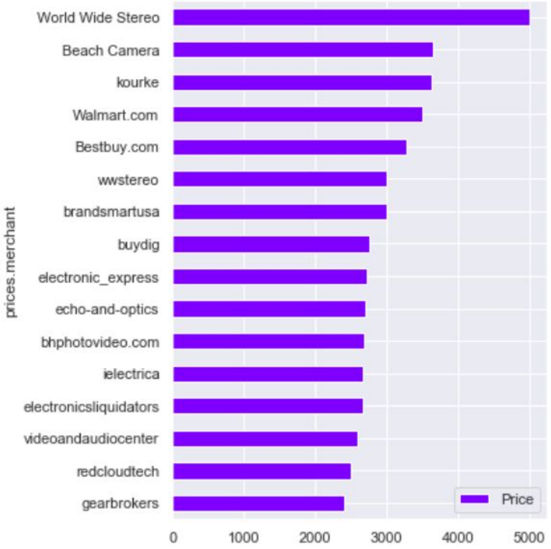
From the results, we can clearly see that the number of times the top 10 brands make the promotions is different. For example, we can see that although Samsung generated more sales than Sony, the numbers of promotions made from Sony are more than Samsung.

4.4 How is the pricing strategy of a product among the different merchants?



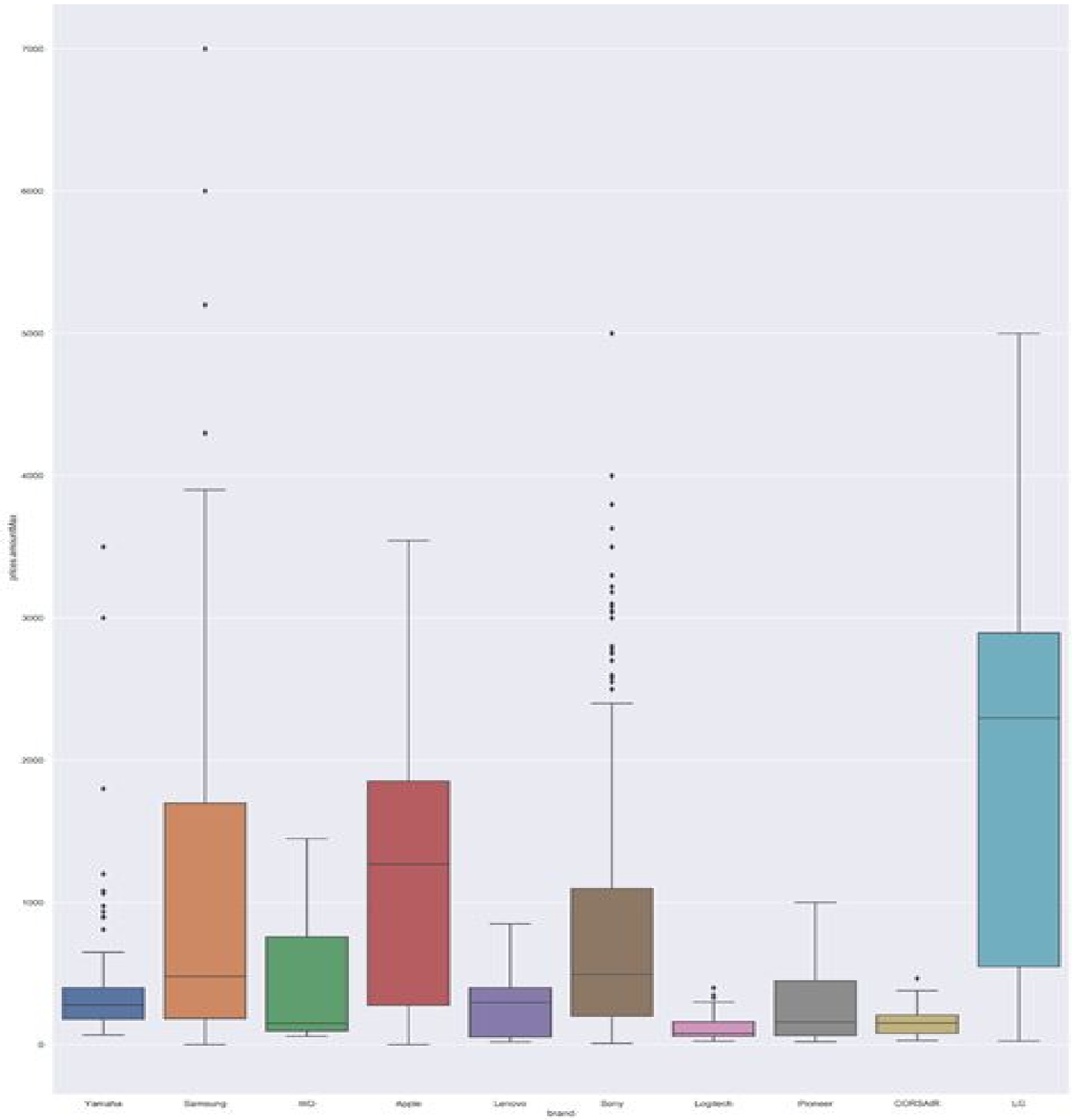






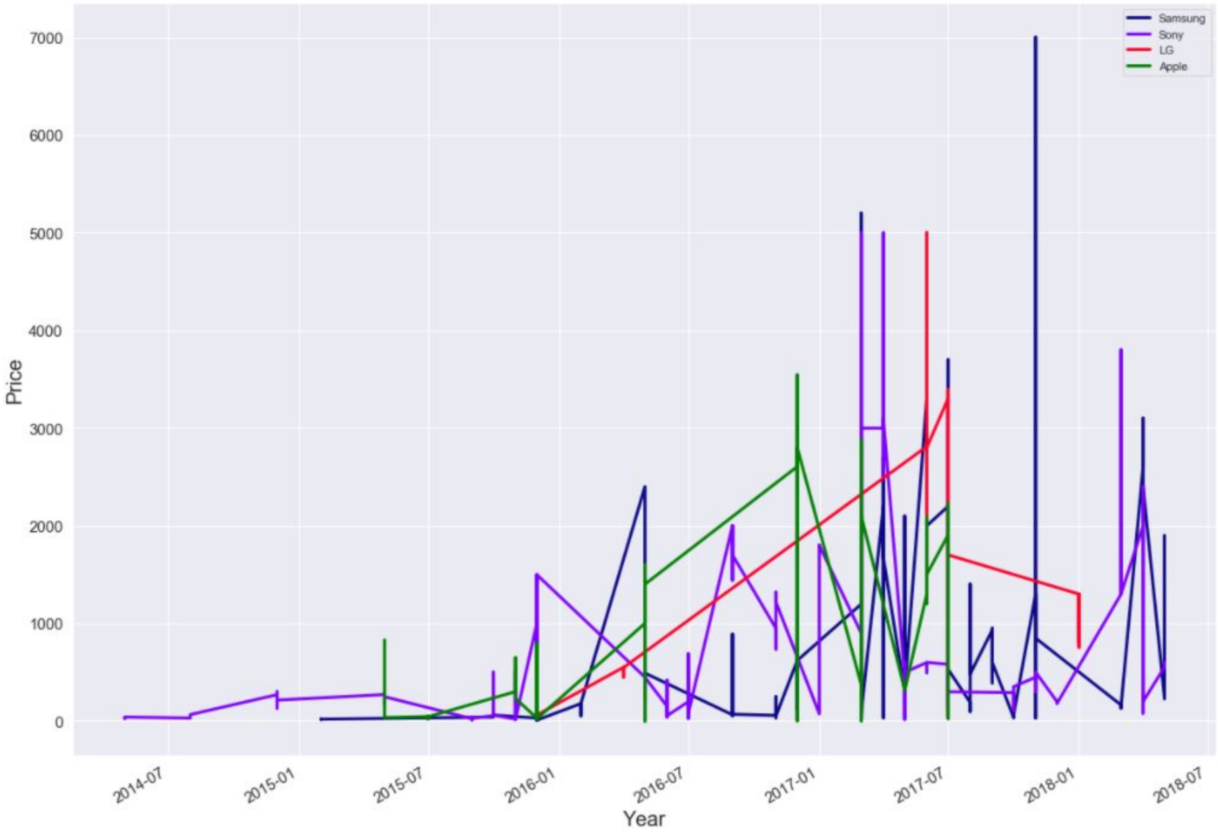
From the bar plot above, we can clearly see that the pricing strategy is a bit different from each of the merchants. For example, in the red bar plot, we use the product named “4TB Network OEM HDD Retail Kit (8-Pack, WD40EFRX, Red Drives)” to compare. Hence we can clearly see that the prices in bhphotovideo.com reach until 600 but the rest only sell the products between the range 100 - 150. Furthermore, in the blue bar plot, we used the product name “MU8000-Series 65-Class HDR UHD Smart LED TV” to compare, as we can see that the prices are more average from the merchant excluding the walmart.com and beach camera which they sell more expensive. Moreover, from the green bar plot, we used the “SAMSUNG 65 Class 4K (2160P) Ultra HD Smart QLED HDR TV QN65Q6FNAFXZA (2018 Model)"” to compare. As the bar plot shows that the torchehunte0 sells the product with only 500. As for the last product we used to compare is “XBR-X850E-Series 75-Class HDR UHD Smart LED TV”, which from the bar plot we can clearly see all the merchants selling the prices average excluded the World Wide Stereo sell the product as price 5000.

4.5 How is the pricing strategy among the top 10 brands?



In this case, we select brand and prices.amountMax as our x and y respectively in order to construct a box plot that compares the pricing among the 10 selected brands. From the box plot, we can see that LG has the highest pricing strategy as it has the largest median and Logitech has the lowest pricing strategy. And from the spreadness of the boxes, we can see that Samsung, Apple, Song, and LG are wider compared to the other brands. It might mean that there are varied products under these 4 brands or the 4 brands have more pricing strategies for their products. We also can see that the boxes of Samsung, Sony, Logitech, Lenovo, and Pioneer are skewed to the right. It can mean that they have lower pricing strategies for their products. Although there are several outliers, we decided to keep those outliers after determination because it does not really make sense to remove outliers from a price feature.

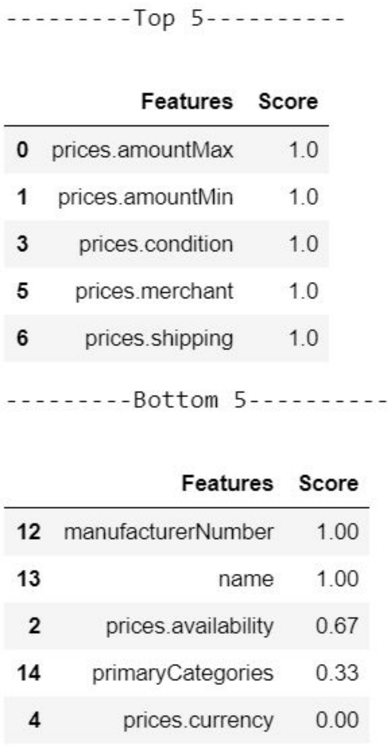
4.6 What are the trends of the top brands among the year?



In order to answer this question, we select Samsung, Sony, LG, and Apple as our target brands because they have various pricing strategies according to the box plot of section 4.5. From the line graph, we can see that the overall pricing for the 4 brands are quite consistent throughout four years except for some special cases. For example, we can see that Samsung has very high pricing for some of its particular products around January 2018. Sony and LG have a very high pricing value in the first half of 2017. Moreover, Apple’s pricing is not that high compared to the other 3 brands in these four years.

5.0 Feature Selection

In the features selecting, we perform the Boruta identify which features are the most important for us to do the classification and the machine learning techniques. It will select those features in the training dataset that are most relevant in predicting the target variable. Furthermore, boruta is a wrapper algorithm which is built around the random forest algorithm. In the boruta, it tries to capture all the significant features to the target data. In the beginning, we fit the X and y, prices.isSales and prices.availability into the feat-selector to obtain the score of the features. The figure below shows that the features score after we fit the X and y. From the figures, we can obtain what are the least significant and important features for the target data.



```
---------Top 5----------
```

	Features	Score
0	prices.amountMax	1.0
1	prices.amountMin	1.0
3	prices.condition	1.0
5	prices.merchant	1.0
6	prices.shipping	1.0

```
---------Bottom 5----------
```

	Features	Score
12	manufacturerNumber	1.00
13	name	1.00
2	prices.availability	0.67
14	primaryCategories	0.33
4	prices.currency	0.00

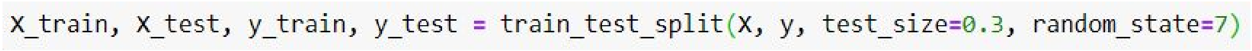
By using this method, we can obtain the most important features and drop the not important features from the result we obtained. So in our case, we drop the prices.currency which only has the score with 0.00 before doing the classification.

6.0 Machine Learning Techniques

In our project, we have constructed two predictive questions, and we decided to use classification to answer both questions. As we mentioned earlier, we have selected prices.isSale and prices.availability as our target data, which are the two y and other features in this dataset will be our X. Two classification algorithms will be applied, Random Forest Classifier and K-Nearest Neighbors respectively.

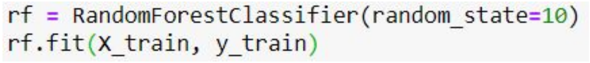
6.1 What products are suggested to be promoted?

In order to answer this question, we need to fit the data into both classifiers to do the prediction. We firstly perform the train-test split. This step is to split the dataset into train data and test data. We choose prices.isSale as the y for this question.

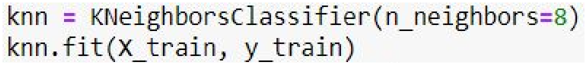


```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=7)
```

As a result, X-train, X-test, y-train, and y-test will be produced. Subsequently, we fit the training dataset into both classifiers.

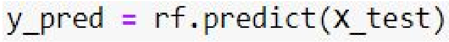


```
rf = RandomForestClassifier(random_state=10)
rf.fit(X_train, y_train)
```

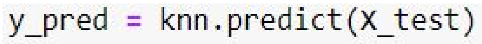


```
knn = KNeighborsClassifier(n_neighbors=8)
knn.fit(X_train, y_train)
```

After fitting in the training dataset, we can use X-test to predict the y. Additionally, the predicted y will be compared with the y-test. The purpose of the comparison is to evaluate the accuracy score of these models. A high accuracy score can indicate that if we fit in with a real world dataset, the prediction of it will be very accurate.



```
y_pred = rf.predict(X_test)
```



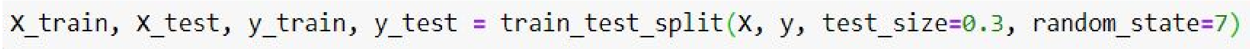
```
y_pred = knn.predict(X_test)
```

After the y-pred is produced, we then can get the accuracy score. As for the result, we get a score of 0.85 for Random Forest Classifier, which is a high accuracy score. But for K-Nearest Neighbors Classifier, it only has a score of 0.74, which we think that it's only a decent one.

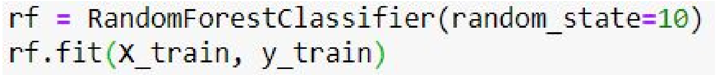
Model	Accuracy(%)
Random Forest	85
K-Nearest Neighbors	74

6.2 What products do the retailers need to purchase to meet customer needs?

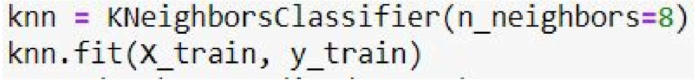
For this question, we perform the same steps as the previous question, but instead we choose prices.availability as the y.



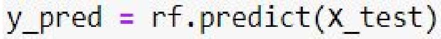
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=7)
```



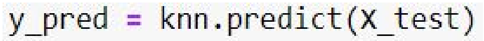
```
rf = RandomForestClassifier(random_state=10)
rf.fit(X_train, y_train)
```



```
knn = KNeighborsClassifier(n_neighbors=8)
knn.fit(X_train, y_train)
```



```
y_pred = rf.predict(X_test)
```



```
y_pred = knn.predict(X_test)
```

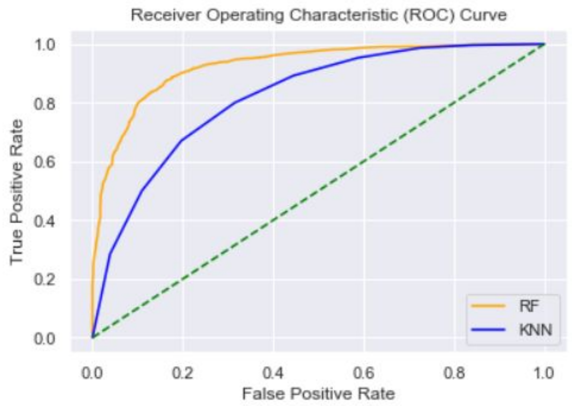
As a result for the accuracy of these two models, they have a score of 0.98 and 0.90 respectively, which are significantly high.

Model	Accuracy(%)
Random Forest	98
K-Nearest Neighbors	90

6.3 Model Validation

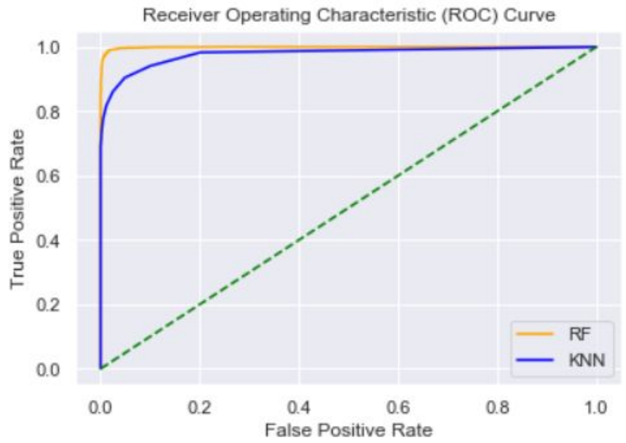
We evaluate the two models by using the ROC (Receiver Operating Characteristics) curve. ROC curve plots True Positive Rate on the y-axis against False Positive Rate on the x-axis. ROC curve indicates that the closer the curve is to 1.0 in the upper left corner, the better the model.

ROC curve for question 6.1:



In this case, we can evidently see that the curve of RF is closer to 1.0 than KNN, so we can say that RF is a better model for our case.

ROC curve for question 6.2:



The two models we used to do classification are good for this question because we can clearly see that the curve of KNN and RF are very close to 1.0, which KNN is a slightly better one.

7.0 Closing Notes

7.1 Answering the Business Case

From the plot that we generated through the exploratory data and analysis, it helped us to answer the questions that we mentioned earlier. From the plot, we have shown that the products trend, sales generator, the brand trends and the pricing strategy etc.

On the other hand, we have also implemented a predictive model that can help us to predict what are the factors that can be used to predict the promotion and the availability of the products.

7.2 Challenges we faced

Throughout the project, we have learnt more from not only the theory-based knowledge and also the hands-on. For example, in this project, we apply the data science pipeline techniques that we learnt in this subject. Overall in this project, the selecting dataset is a challenging task. This is because when we choose the dataset, we need to think of the data quality and the interesting insight that can be formed from the dataset. Furthermore, the dataset that we found is hard to be merged from any other dataset, hence we need to focus our questions based on the dataset that we chose at the end.

7.3 Way forward for future insights

Nowadays, the data is generated faster in only a second. Hence with the social media or the ecommerce applications that we use every day, it could certainly be mined by the data scientist in order to produce a product trends or getting the interest from the customers. For example, it can help the retailers to understand what they need to focus on this group of customers and what products they need to focus in order to boost their sales.

8.0 Conclusion

“Without data, you’re just another person with an opinion.”, a quote from W. Edwards Deming. This quote must have profoundly influenced a lot of data scientists nowadays. Without data, if we are knowledgeable, intelligent, talented, or even a genius, and then we give out an opinion about a question, there must be some people questioning our opinion absolutely. So, we can definitely say that the word evidence is paramount in this real world. But with data, we can explore it in depth, thus find interesting patterns and finally use it as evidence to answer those questions or provide useful insights.

In our project, as we have mentioned in part A, our main objective is to benefit both retailers and customers. To achieve it, we have used a lot of data science techniques in order to answer the questions by following the data science pipeline. During the process, we have faced many minor issues such as, we have experienced the difficulty to collect data because the two datasets we found are unable to merge together and we have to decide which dataset is more suitable for us. Of course, there are many more issues. Nevertheless however, our project is progressing well as we complete preprocessing, mining, predictive modeling and visualization of the data.

Although we come out with a satisfied result for us as beginners, we think it's still far away from perfect. We believe that if we keep absorbing the knowledge of data science, it will absolutely improve our skills and efficiencies on future projects.

Eventually, this project is very interesting for us as novices to try out our python and data analytic skills that we learnt from classes. It definitely gives us a deep hands-on experience and also enriches our knowledge in the data science area.

References

[1] 5 Ways Walmart Uses Big Data to Help Customers. (2017, August 7). Retrieved from https://blog.walmart.com/innovation/20170807/5-ways-walmart-uses-big-data-to-help-customers

[2] The Power of Big Data in Retail, Retrieved from https://www.yodlee.com/retail-merchants/the-power-of-big-data-in-retail

[3] How Amazon uses Big Data, Retrieved from https://www.bernardmarr.com/default.asp?contentID=712

[4] Top 10 Retailer in the World, Retrieved from https://www.investopedia.com/articles/markets/122415/worlds-top-10-retailers-wmt-cost.asp

[5] Electronic Products and Pricing Data https://data.world/datafiniti/electronic-products-and-pricing-data