E-COMMERCE DATA ANALYSIS FOR CUSTOMER INSIGHTS AND BUSINESS GROWTH

ProjectGroupF - E-Commerce:
CHONG KIN HOW 17202112
LIM JOU HUI 17201197
MOHAMED FATHI MOHAMED HAMAD 17198285
RAJIB KANTI PAL 17199062
LIU HONG YANG 17201091
KUAN NAM SUNG 17029014
SRITHARAN SIVAGURU 17198431 | GROUP LEADER

FACULTY OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY
UNIVERSITY OF MALAYA
KUALA LUMPUR
2020

CHAPTER 1: INTRODUCTION

1.1 Introduction

The Internet has become common infrastructure. Where it is available, 8 out of 10 people are using it. Big data that is mainly generated through the Internet via services such as E-commerce and social networking sites, with a variety of characteristics. Traditional information infrastructure such as RDBMs are not sufficient to handle the vast amount of data. Thus, new solutions specially designed for big data are needed. These solutions are available in the form of specialized tools such as Hadoop, Hbase, Hive, Pig, etc. (Soni, 2017). Big data enables a massive amount of e-Commerce data to be processed through these new solutions to generate useful business insights. For instance, the behavior of the online shopping users can be analyzed to establish product recommendation solutions. (G. et al., 2019).

1.2 Problem Statement

E-commerce platforms produce a high volume of data from online transactions between their sellers and customers. Generating useful insights about their customers, products, and services from the captured data could help to optimize their seller supply chain, understand customer behavior better with the hope of generating higher revenue. The volume of data generated from e-commerce platforms, poses a challenge to carry out data analytics, hampering the ability of sellers to determine the trend of product demand and customer satisfaction.

As a result, customer retention could decline and hamper sales opportunities, resulting in lost revenue and lost business. For our project, we attempt to use Hadoop technologies to store and analyze gathered e-commerce data to address some of the above problems. Hence, the following objectives have been formulated.

1.3 Objectives

This section describes the objectives of this project. The main purpose of this project is to generate useful insights for sellers on e-Commerce platforms. The scope of our project, covers the following research objectives:

1. To develop a product recommendation system from e-Commerce transaction data to help sellers to improve their revenues.
2. To create an exploration tool for sellers to compare their statistics with other sellers.
3. To create a rating recommendation tool for sellers to improve the sale of their items on an e-Commerce platform.

The code to the above solutions are available in Appendix A.

CHAPTER 2: METHODOLOGY

2.1 Data Collection

For the product recommendation system, we utilized an online retail dataset from Kaggle. The URL to the dataset is provided in Appendix A. The chosen dataset contains 541910 rows of data with 8 attributes. The attributes are invoice number, stock code, description, quantity, invoice date, unit price, customer, and country of customer. We chose this dataset for the product recommendation system as it contains all customer transaction records for developing a good product recommendation system. We had to use a different dataset from the rest of the solutions delivered by this project, because the scrapped Shopee item dataset does not contain transaction records. The dataset scrapped from Shopee only contains item catalog data. As a result, there are two different datasets for this project.

For the rest of the solutions delivered by this project, the Shopee item catalog is used through data scraped from the Shopee Malaysia website (www.shopee.com.my). The item data scraped from Shopee contains 15 attributes with 114269 rows of data. The data set contains the following attributes: item label, item stars, number of item ratings, number of items sold, minimum item pricing, maximum item pricing, available item stock, name of seller shop, number of ratings received by seller, number of products the seller has in their catalog, seller response rate to customer enquiry, seller response times to customer enquiries, how long the seller has been on Shopee, the number of followers and the URL to the item on Shopee.

2.2 Methodology

This section describes the methodologies used to develop the tools needed to meet our project objectives.

Recommender System

The tools used to develop the product recommendation system for Shopee are Hadoop, Hive, Hive mall, and Tableau. Firstly, the Kaggle datasets with 8 attributes were imported into HDFS. Next, a Hive table was created to store the datasets into Hive. The data pre-processing has been carried out in Hive before continuing to the subsequent process. Then, another table was created to compute and display the top five most recent purchased items for each user. The co-occurrence-based recommendation table was created, after that, combined with a recently purchased products table to select maximum top five products as the recommended products for each user. The collaborative filtering recommender was selected because this method is easier and more direct to get the recommended products based on item-based recommendation. Lastly, the hive was connected with Tableau in order to display the product recommendation result in a proper manner and format. We chose Tableau because we would like to explore connecting Hive to a third party BI tool.

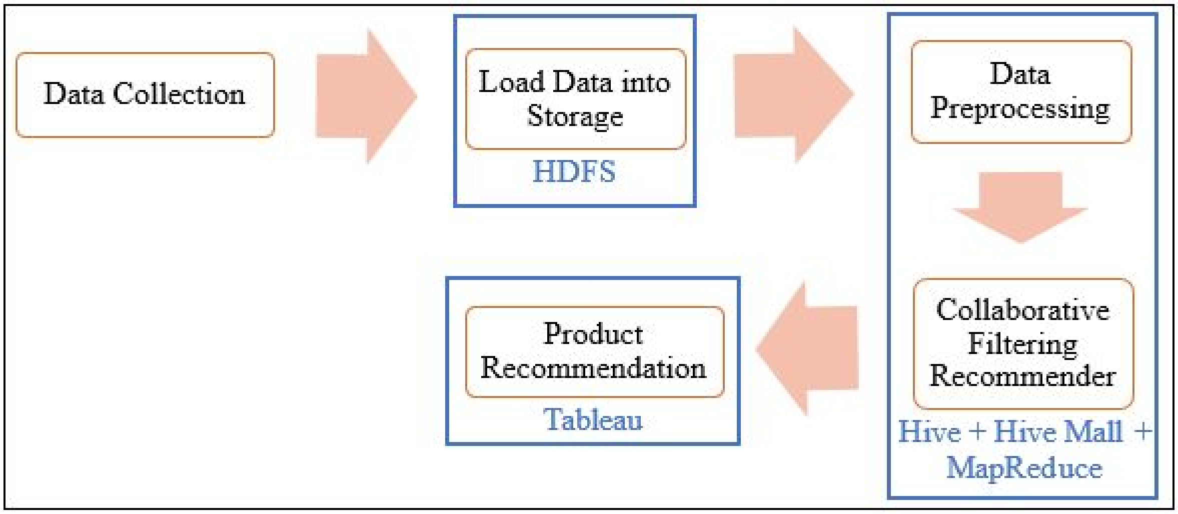


Figure 2.1: Data pipeline

Exploration and Visualization Tool

For the data exploration and visualization tool, we used the scrapped Shopee dataset, Hive and Python. We first load the data we have crawled from Shopee to Hive. Then we use the pyhive package in python to extract the data from hive to python for further processing. Pandas has been used in Python for processing purposes as it can use the data frame for processing and it makes data manipulation easy, which becomes unwieldy when the data is huge. The processing steps for this tool includes grouping the attributes within the dataset, to remove potential duplicates and removing NULL values from the data set. The data was then used to produce visualizations such as graphs and various visual indicators that consist of the number of seller ratings, number of products, number of followers and responses per hour. This would enable the seller to compare the graphs to discern their current position or average ranking against other sellers on the Shopee platform.

Rating Prediction Tool

For the rating prediction tool we used the Shopee item catalog data that was scrapped from the Shopee website to perform the rating prediction for each item. This rating attribute is referred to as 'Stars' on the item listing page. We do provide simple visual statistics of the item category to show the seller how the particular category is performing in terms of ratings, stocks, items sold and the number of followers.

Modeling

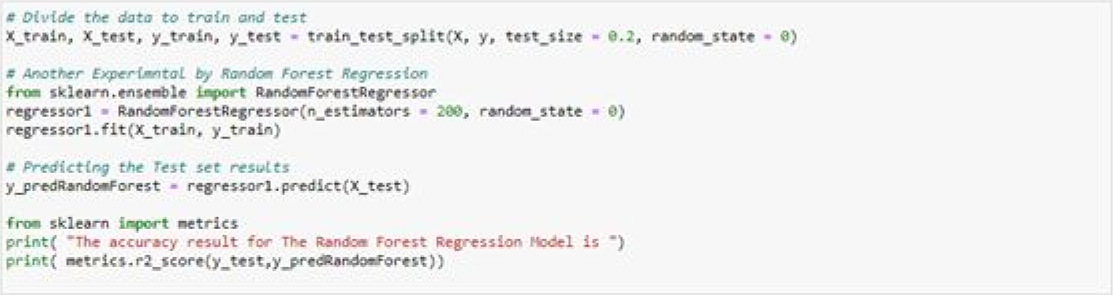
In this stage of the tool, the machine-learning model is built using the most suitable algorithm. In this phase, the model is trained using the training set produced in the previous stages. . This will be done by dividing the training set into two parts, one for training and the other for validation. Random forest regression will be used to predict the target variable Ratings by using the suitable factors from the given dataset.

Used feature for Prediction

For this prediction process, seven features have been selected from the dataset as predictors variables are Ratings (Number of people rate the item), Sold (Number of items sold), PriceMin, PriceMax, SellerRatings, ResponseRate, Followers (Number of followers). These independent variables will be used to predict the target variable Stars.

Machine Learning Model Selection

In this section the model selection will be explained. In this project, many regression models have been applied and tested before reaching the conclusion of why Random Forest Regression is used as the main predictor to build the Ratings predictor model. For this tool, Multi Linear Regression (MLR), Support Vector Regression (SVP) and Random Forest Regression (RFR), were tested. The code snippets used to test the three regression models is depicted in figures, Figure 2.2, Figure 2.3, Figure 2.4.



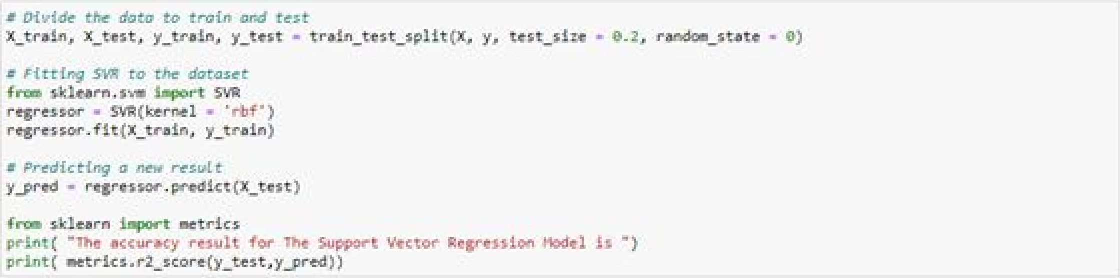
```
# Divide the data to train and test
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = 0)

# Another Experimntal by Random Forest Regression
from sklearn.ensemble import RandomForestRegressor
regressor1 = RandomForestRegressor(n_estimators = 200, random_state = 0)
regressor1.fit(X_train, y_train)

# Predicting the Test set results
y_predRandomForest = regressor1.predict(X_test)

from sklearn import metrics
print( "The accuracy result for The Random Forest Regression Model is ")
print( metrics.r2_score(y_test,y_predRandomForest))
```

Figure 2.2: Code to build Random Forest Regression



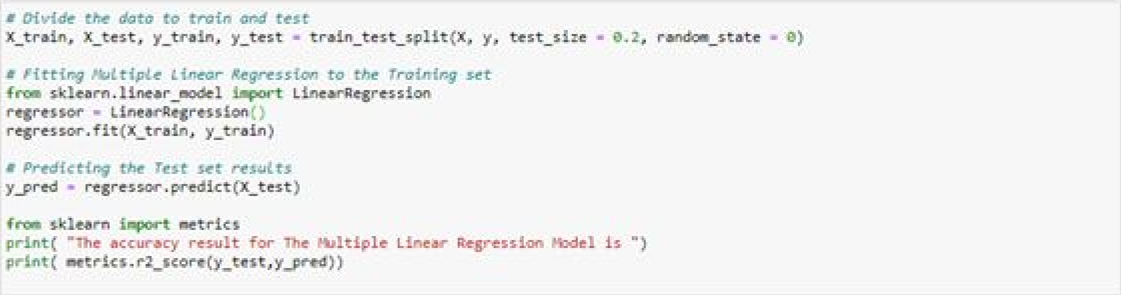
```
# Divide the data to train and test
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = 0)

# Fitting SVR to the dataset
from sklearn.svm import SVR
regressor = SVR(kernel = 'rbf')
regressor.fit(X_train, y_train)

# Predicting a new result
y_pred = regressor.predict(X_test)

from sklearn import metrics
print( "The accuracy result for The Support Vector Regression Model is ")
print( metrics.r2_score(y_test,y_pred))
```

Figure 2.3: Code to build Support Vector Regression



```
# Divide the data to train and test
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = 0)

# Fitting Multiple Linear Regression to the Training set
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(X_train, y_train)

# Predicting the Test set results
y_pred = regressor.predict(X_test)

from sklearn import metrics
print( "The accuracy result for The Multiple Linear Regression Model is ")
print( metrics.r2_score(y_test,y_pred))
```

Figure 2.4: Code to build Multiple Linear Regression

The reasoning behind not selecting the MLR model is that it is very sensitive to numeric values, due to the linear equation used in the algorithm. Another reason for this result is the sensitivity towards outliers (Knaub, 2017). This means that the features in the collected data would affect the performance of the model.

SVR did provide good performance. However, there are some data size related limitations that prevent us from selecting it for our tool. The drawback with SVR is with the number of the dimensions in the dataset (the number of variables), especially if the variables have a wide range which is the case in our data. In the case of our dataset, SVR is required to build many hyper plains based on the number of variables (Awad, & Khanna, 2015).

RFR, performed well for the following reasons. Firstly, Random forests use ensemble learning that take many weak learners and use them to get better accuracy by applying averaging or voting to select the best model results. Another reason is, that by default, using ensemble could easily avoid the model from overfitting (Segal, 2004).

Based on the previous comparison and reasons mentioned, the Random Forest Regression (RFR) was selected by us to develop the Ratings predictor model. After the features and model are selected, the final step is to fit the features into the model to build the Ratings predictor. The Figure 2.5 shows the process.

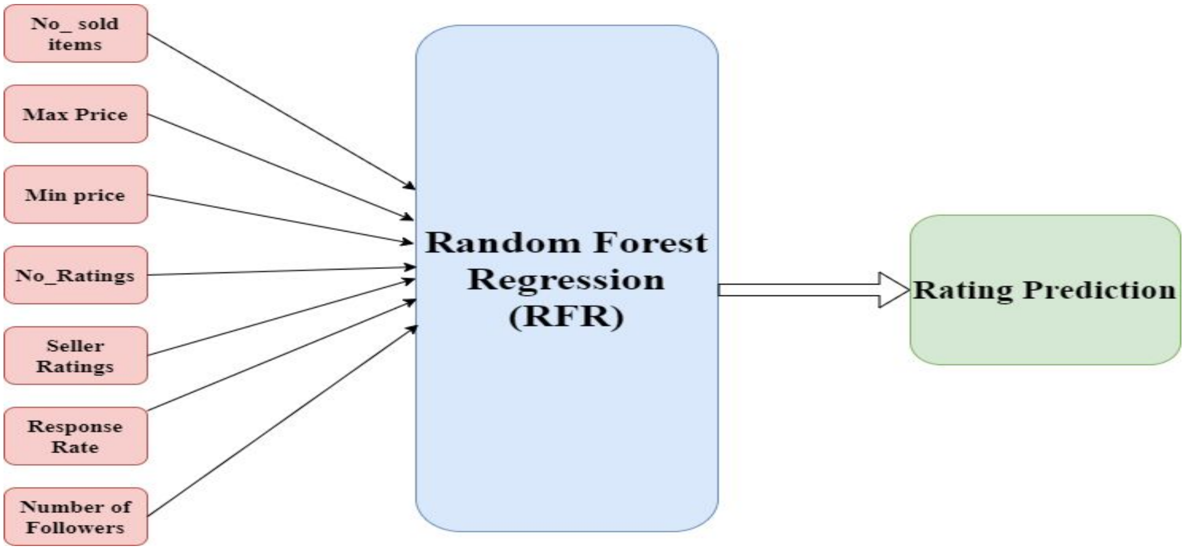


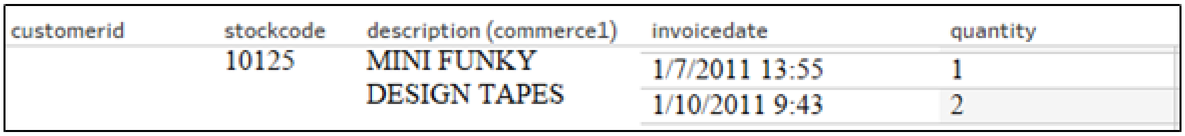
Figure 2.5: Fitting the Features in to the Model

CHAPTER 3: OUTCOME

For this project, we do not use any real time data. Hence, we chose HBase and Hive as our primary big data tool to generate our results.

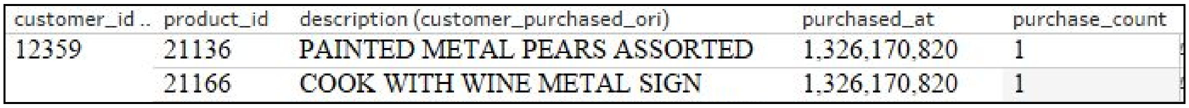
Product Recommender System Results

Hive is a SQL engine built on top of Hadoop to run MapReduce jobs through SQL like queries instead of Java programs. Hive mall was chosen to run our recommender system algorithms internally. Table 3.1 displays the online shopping details for each customer. They total 5 out of 8 attributes that were displayed in Table 3.1 as those 5 attributes are important attributes for the product recommendation system. The invoice date in Table 3.1 has been processed from original format of dd-mm-yyyy hh:mm using UNIX timestamp to number of seconds from some arbitrary point in the past and named it as purchased_at as shown in Table 3.2. Furthermore, the purchase_count column is defined as the total count of purchased products for each customer.



customerid	stockcode	description (commerce1)	invoicedate	quantity
	10125	MINI FUNKY DESIGN TAPES	1/7/2011 13:55	1
			1/10/2011 9:43	2

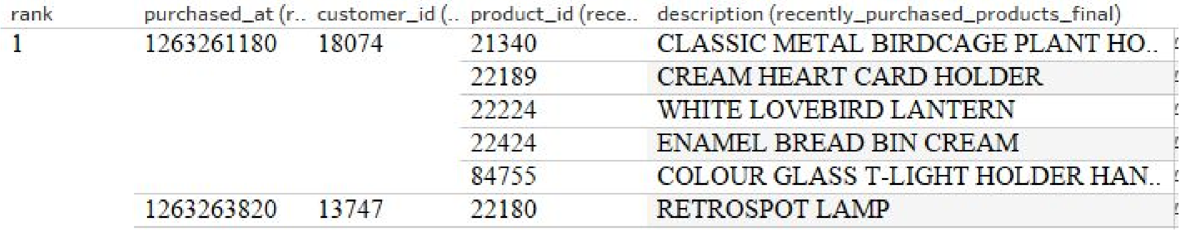
Table 3.1: Customer Purchased Original (First two rows)



customer_id ..	product_id	description (customer_purchased_ori)	purchased_at	purchase_count
12359	21136	PAINTED METAL PEARS ASSORTED	1,326,170,820	1
	21166	COOK WITH WINE METAL SIGN	1,326,170,820	1

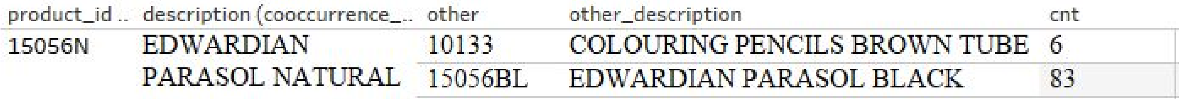
Table 3.2: Customer Purchased Final (First two rows)

Next the top five recently purchased items from each customer have been listed using hivemall tools as shown in Table 3.3. After that, we used co occurrence count as a relevance score to establish the recommended Shopee products. The co occurrence table can be found in Table 3.4. For instance, when customers buy Edwardian parasol natural as listed in the description column of Table 3.4, they will also buy other products as shown in the column of other description. The table 3.4 also showed the total count of that particular purchased item in the cnt column. Lastly, the final product recommendation can be seen in the Table 3.5. It is recommended based on both top five recently purchased items (Table 3.3) and co occurrence items (Table 3.4) with the additional information of the maximum top five count (cnt) number as shown in Table 3.4.



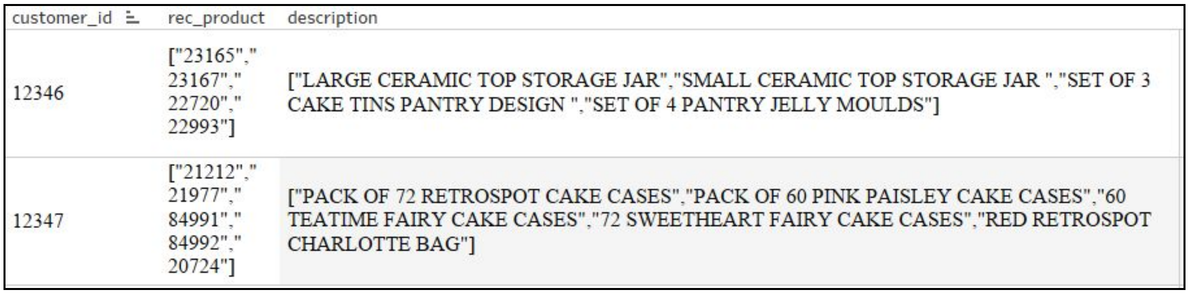
rank	purchased_at (r..	customer_id (..	product_id (rece..	description (recently_purchased_products_final)
1	1263261180	18074	21340	CLASSIC METAL BIRDCAGE PLANT HO..
			22189	CREAM HEART CARD HOLDER
			22224	WHITE LOVEBIRD LANTERN
			22424	ENAMEL BREAD BIN CREAM
			84755	COLOUR GLASS T-LIGHT HOLDER HAN..
	1263263820	13747	22180	RETROSPOT LAMP

Table 3.3: Recently Purchased (First six rows)



product_id ..	description (cooccurrence_..	other	other_description	cnt
15056N	EDWARDIAN PARASOL NATURAL	10133	COLOURING PENCILS BROWN TUBE	6
		15056BL	EDWARDIAN PARASOL BLACK	83

Table 3.4: Cooccurrence (First two rows)



customer_id	rec_product	description
12346	["23165"," 23167"," 22720"," 22993"]	["LARGE CERAMIC TOP STORAGE JAR","SMALL CERAMIC TOP STORAGE JAR ","SET OF 3 CAKE TINS PANTRY DESIGN ","SET OF 4 PANTRY JELLY MOULDS"]
12347	["21212"," 21977"," 84991"," 84992"," 20724"]	["PACK OF 72 RETROSPOT CAKE CASES","PACK OF 60 PINK PAISLEY CAKE CASES","60 TEATIME FAIRY CAKE CASES","72 SWEETHEART FAIRY CAKE CASES","RED RETROSPOT CHARLOTTE BAG"]

Table 3.5: Final product recommendation (First two rows)

Visualization Results

In this part, we aim to find which seller has the best performance in the Shopee based on different measurements. Customers could utilize the visualization dashboard to explore the sellers' behaviors and choose the best seller to offer them online shopping services. The visualization Dashboard has three main interactive components.

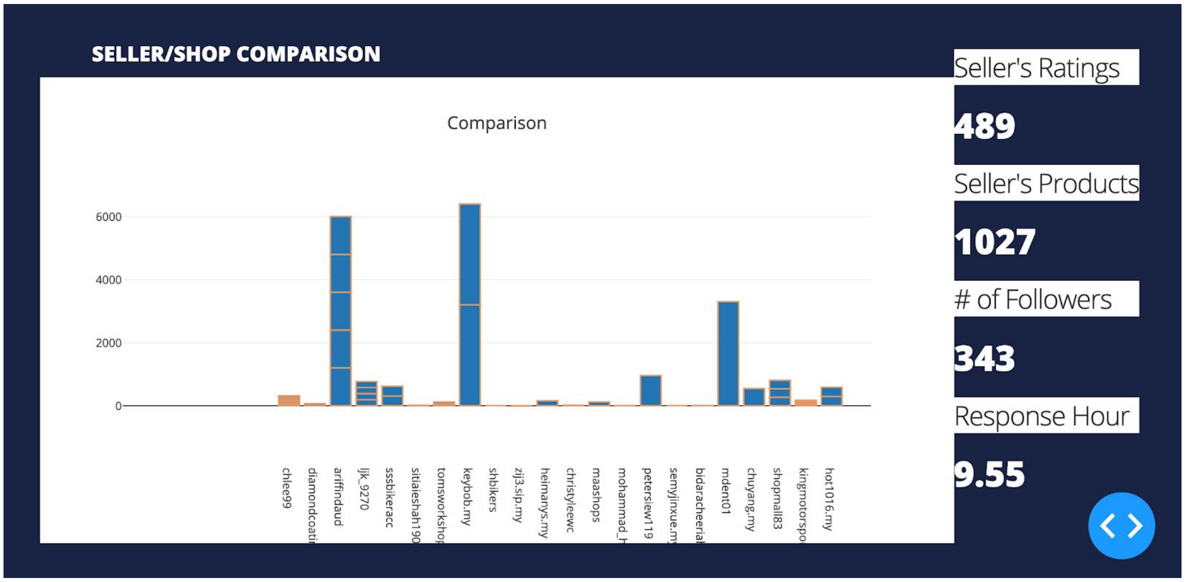
- Text description

- Dropdown
- Bar Charts

In the text description part, there includes a specific description of the Dashboard. Users could understand the functionality of the interface.

In the Dropdown component, users could select one value from a series of options based on their preferences

Finally, the interactive results would be plotted in the bar charts.


SELLER/SHOP COMPARISON

Seller's Ratings
489

Seller's Products
1027

# of Followers
343

Response Hour
9.55

Figure 3.6 Seller comparison results from Dashboard

On the left side of Figure3.6, the bar chart shows which seller has the best performance in the chosen metrics. From the bar charts, we can see that shop “keybob.my” has the highest scores than any other seller. However, in the right side of Figure3.6, the four values represent the average of values in four measurements individually.   Overall, users could manipulate the Dashboard to compare the sellers and find out which one is the most favorite seller evaluated by other customers. Besides, the users could also make decision making to gain better servers based on their comparison results.

CHAPTER 4: CONCLUSION/FUTURE WORK

We created a recommendation system for Shopee successfully using historical transaction records. Hive took quite a long time (around 20 minutes) to process the transaction records and create the final product recommendation. Besides that, the process of connecting Hive to Tableau is quite challenging. We faced a few hiccups while setting up the connection. The network adapter in Oracle VM has to be attached to Bridged Adapter in order to get a proper and working IP address to set up the connection in Tableau. For future work, we would like to explore other big data tools such as Hbase and MongoDB to compare the performance with Hive.

For the prediction tool, we initially decided to use a cloud-based environment to enable all team members that are working on the item catalog data to use the same HBase resource. We set up clusters in Azure HDInsight with HBase enabled. Then we tried various different Python packages to connect with HBase in Azure HDInsight.

These packages were:

- CDATA – We initially could not install the required CDATA package in Python v3.7.4 on Windows 10.
- Starbase – The connection was successful, and we can read data of any single row but could not read the whole data altogether and convert it to a dataframe for further processing. If we wanted to read all of our dataset of 114269 rows by executing a query for each row, then it takes a long time to load the data.
- SqlAlchemy – We could not connect with HBase with a different combination of parameters.
- HappyBase – we just could not install the package in our Python environment.

Eventually, after a week, our azure cluster environment was disabled after running out of trial credit. Then, we had to set up another environment in Azure HDInsight. Surprisingly, this time we could not connect with the Starbase package with a different set of parameters. We have downloaded the CDATA package from their website and installed it manually but had no luck. We could not connect with HBase using the CDATA library. After struggling with this connectivity, we have decided to use our own Hadoop setup installed in our VM on Ubuntu Linux.

We then repeated the same above process with our VM setup. Using the CData package we were successfully able to connect and read all the data from HBase using the CDATA package, but the problem is converting the result set to pandas dataframe returns all columns as a single column. We have changed the method and finally got the expected output using the "read_sql" method. Further integrating with Streamlit we were able to execute our prediction tool and everything was working as expected.

In future we would like to consider the option of using either Java or .Net to overcome the above issues and be able to collaborate with our project team better.

APPENDIX A

Here is the GitHub location for all the source code and data used by this project:

https://github.com/sritharans/ProjectGroupF

Here is the link for the video that was presented for this project:

https://mega.nz/file/gAVywYDA#1aKeB63QZuscq2M2O1BTqgSn2V2eOHOOCkinILQALeg

REFERENCES

Soni J., (2017). A Hybride Product Recommendation Model Using Hadoop Server for Amazon Dataset. *ISSN 0973-6107 Volume 10, Number 6 (2017) pp. 1691-1705.*

G . C. Selvi., Singh. I. (2019). Optimized Recommendation System for E-Commerce on Product Features and User Behavior. *International Journal of Recent Technology and Engineering 2, 8*(2), 748-759. doi:10.35940/ijrte.b2401.078219.

Knaub, James. (2017). Essential Heteroscedasticity. 10.13140/RG.2.2.20928.64005.

Awad, M., & Khanna, R. (2015). Support vector regression. In Efficient Learning Machines (pp. 67-80). Apress, Berkeley, CA.

Segal, M. R. (2004). Machine learning benchmarks and random forest regression.