Female Clothing E-commerce Review and Rating

Hsin-Yu Chen
Huang-Chin Yen
Tian Tian

Introduction

The increasing online access and smartphone penetration contribute to the rise of e-commerce. Women's clothing is one of which has huge potential. According to the Fashion eCommerce Report 2020, from Statista, "Fashion is the largest B2C eCommerce market segment and its global size is 9. The market is expected to grow further at 12.2% per year and reach a total market size of US$991.64billion by the end of 2024."

On top of that, the user recommendation and rating are useful KPIs when running an e-commerce business. These KPIs help e-commerce store owners understand their customers' purchasing experiences so that owners can make decisions for the store and improve both browsing and purchasing flows. On the other hand, customers cannot physically try on or touch the items, so other customers' reviews turn a critical factor to influence the willingness of purchase.

Data Description and Preprocessing

The dataset, Women's E-Commerce Clothing Reviews, from Kaggle contains over 230,000 reviews and ratings written by customers.

Its nine variables listed below offer us an opportunity to parse out the rating and evaluate customers sentiment:

1. **Clothing ID**: Unique ID of the product
2. **Age**: Age of the reviewer
3. **Title**: Title of the review
4. **Review**: Text review
5. **Rating:** Product rating by reviewer
6. **Recommended IND**: Whether the product is recommended or not by the reviewer
7. **Positive Feedback**: Count Number of positive feedback on the review
8. **Division Name**: Name of the division product is in
9. **Department Name**: Name of the department product is in
10. **Class Name**: Type of product

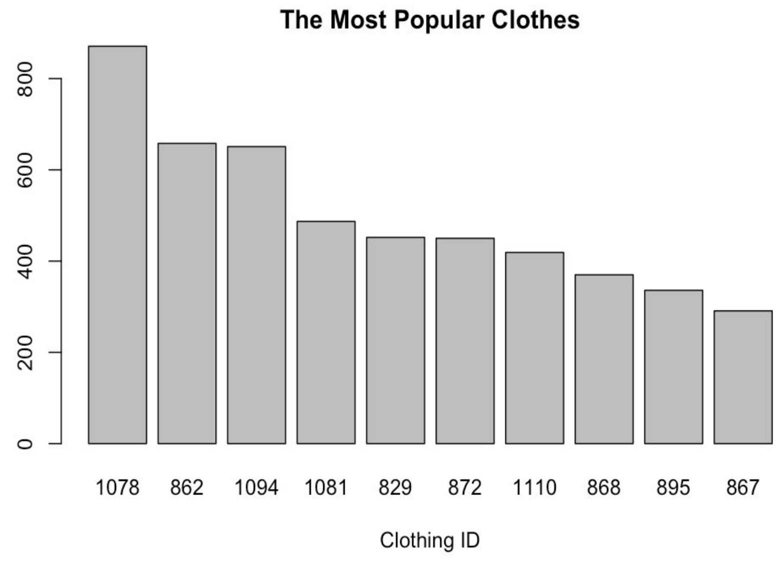


Figure1: The Most Popular Clothes

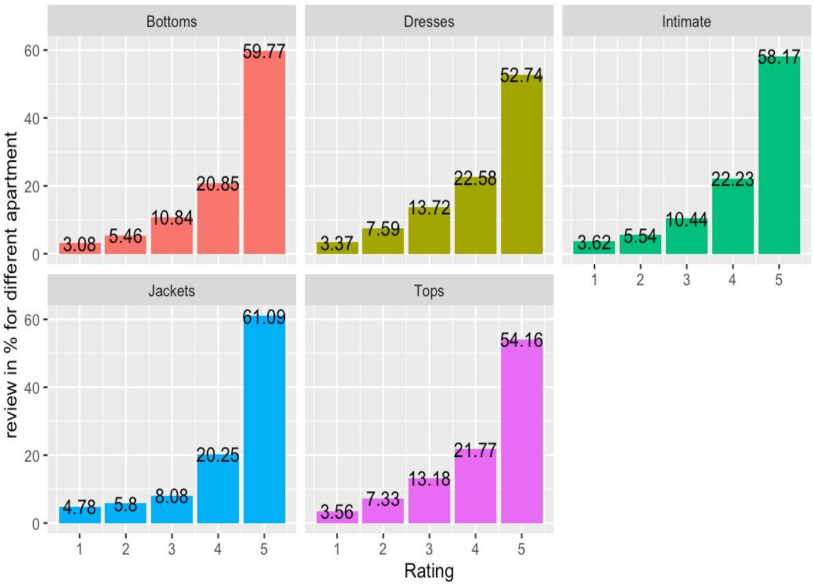


Figure2: Rating Distribution among Different Department

Roughly going through the dataset, we found that the dataset has 1095 different types of clothes. Among them, the clothes with ID 1078, 862, 1094, 1081, 829, 872, 1110, 868, 895, 867 are the most popular 10 items in this e-store because they received the most comments and reviews. On top of that, Jacket has the highest number of ratings, 61.09%, compared to other clothing types. It also has the highest number of one-star ratings, which is about 4.78%. We believe this is because of the more number of clothes selling out, the higher chance of receiving both negative and positive reviews. The second and third departments are bottoms and intimate.

Expectations, Purpose, and Methods

As the dataset is relatively clean, we removed some missing values from the dataset and received a new dataset of 23,486 complete observations. Here are some possible relationships we would like to explore:

1. **Age and Rating:** Is there any relationship between age and rating?
2. **Recommended and Rating:** Is there any difference in the possibility of recommendation among different ratings?
3. **Rating & Review Texts**: figure out the word choice among different ratings to get an insight into customer sentiment.

The methods used in this project include ordinal logistic regression, word cloud, Quadratic Discriminant Analysis (QDA), Support Vector Machine (SVM), Naive Bayes, tree based methods like Classification Tree, Random Forest, XGBoost.

Logistic regression

Logistic regression is a statistical method for analyzing a dataset in which the outcome is measured with a dichotomous variable (only two possible outcomes). One of the limitations of the Logistic regression is that it is highly reliant on a proper presentation of the dataset, all the important independent variables should be identified clearly. Another limitation is that it can only predict a categorical outcome.

Word Cloud

Word Cloud is used to visualize and highlight popular or trending terms/ words based on frequency of use and prominence.

Quadratic Discriminant Analysis (QDA)

As a more general version of the linear classifier, QDA separates measurements of two or more classes of objects or events. The disadvantage of QDA is that it cannot be used as a dimensionality reduction technique.

Support Vector Machine (SVM)

SVM sorts data into categories by drawing hyperplanes to separate the groups of data according to patterns.The algorism is often applied to text categorization, image classification, handwriting recognition and in the sciences. However, when the dataset is large and has more noise, it is not suitable to use SVM.

Naive Bayes

As a probabilistic machine learning model , Naive Bayes model helps the users find the probability of A happening, given that B has occurred

Tree based method

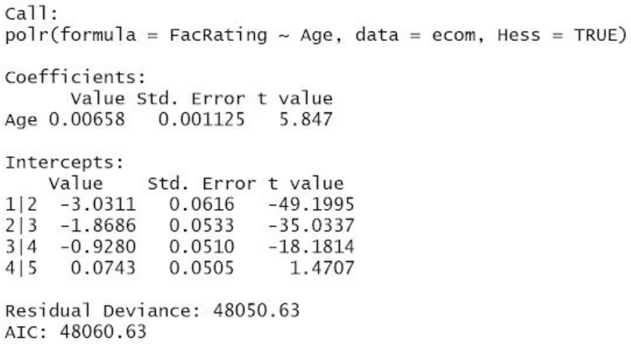
Classification tree - A tool used to limit the class of classifiers. Its tree-like structure makes it easy to interpret and shows the user clearly which classification indeed produces the correct result.

Random Forest - It consists of a large number of individual decision trees. Each individual tree in the random forest spits out a class prediction. We use the class with the most votes to do further analysis. Since for Random Forest, the range of predictions is bound by the highest and lowest labels in the training data, there might be a covariate shift problem when the training and prediction inputs differ in their ranges.

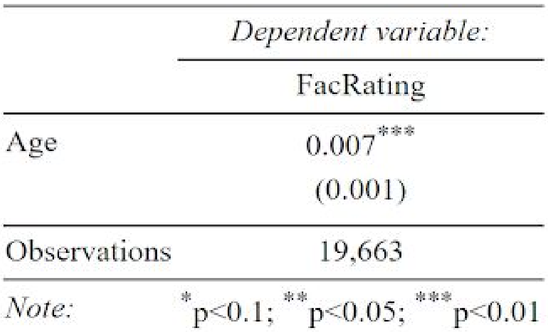
XGBoost - XGBoost stands for Extreme Gradient Boosting. XGBoost computes the second-order gradients to find more information about the direction of gradients as well as to get to the minimum of loss function. On top of that, it advances the regularization so that the model generalization is improved,too.

Age and Rating

In this section, we are interested in potential relationships between age and rating. Is it possible that elder or younger customer groups tend to give higher rates to products they purchased? From 1 to 5, the distance between different rates is not known or not meaningful, only order matters, so rating belongs to ordinal data. With age being the only predictor, we will build an ordinal logit regression model with rating being the response.

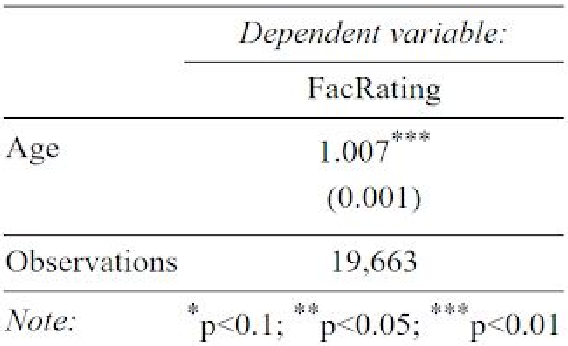


```
Call:
polr(formula = FacRating ~ Age, data = ecom, Hess = TRUE)

Coefficients:
      Value Std. Error t value
Age 0.00658   0.001125   5.847

Intercepts:
    Value    Std. Error t value
1|2  -3.0311   0.0616   -49.1995
2|3  -1.8686   0.0533   -35.0337
3|4  -0.9280   0.0510   -18.1814
4|5   0.0743   0.0505     1.4707

Residual Deviance: 48050.63
AIC: 48060.63
```



	Dependent variable:
	FacRating
Age	0.007***
	(0.001)
Observations	19,663
Note:	*p<0.1; **p<0.05; ***p<0.01

The summary of the model shows that the coefficient of Age has a t-statistic of 5.8, which is larger than 2. It also has a p-value of less than 1%. We can conclude that it is a significant predictor, age has a positive impact on rating. Senior buyers tend to give products higher rates.



	Dependent variable:
	FacRating
Age	1.007^{***} (0.001)
Observations	19,663
Note:	*p<0.1; **p<0.05; ***p<0.01

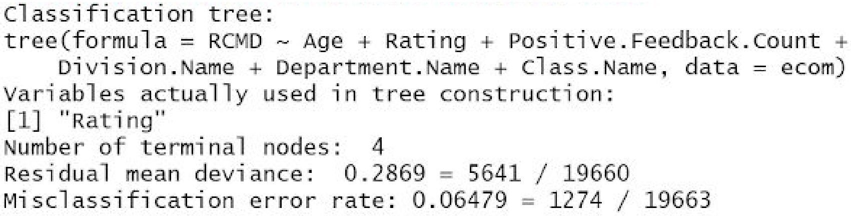
To be more specific, we take the exponential of the coefficient, that is 1.007.
Keeping all other variables constant, when Age increases by one year, the customer is 1.007 times more likely to give a random product a higher rate of one point. In other words, the odds of a random product getting one point of higher rate is 7% when the buyer is one year older.

Recommended or Not?

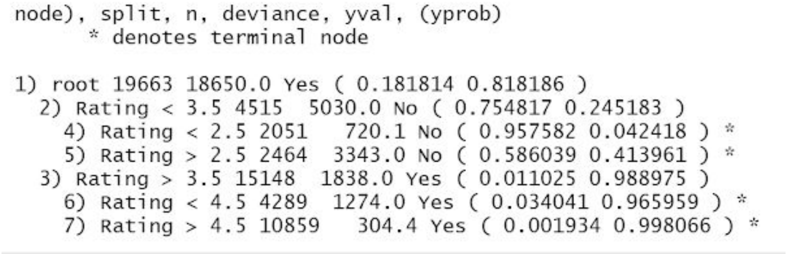
In this section, we are interested in finding key factors that determine whether a product is likely to be recommended or not. This is a classification problem, a classification tree is appropriate to generate the result. After the original dataset being randomly split into a training and a testing set, a classification tree is constructed using all reasonable variables.

Possible predictors are age, rating, the number of possible feedbacks, name of division, department and class of the product. For example, it is very possible that shoes are less likely to be recommended than loungewear, because of higher standards or more dimensions of evaluation.

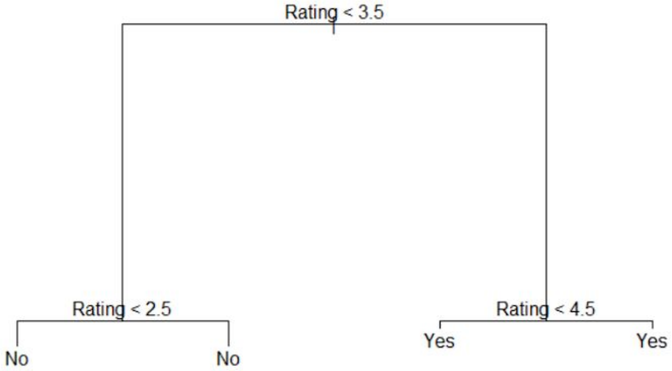
Columns that are not meaningful, such as product ID, and those that are difficult to be categorized, such as review title and contents, are not included in this tree.



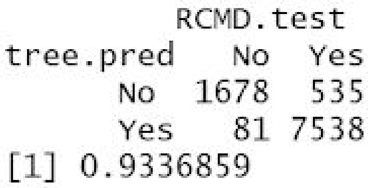
```
Classification tree:
tree(formula = RCMD ~ Age + Rating + Positive.Feedback.Count +
    Division.Name + Department.Name + Class.Name, data = ecom)
Variables actually used in tree construction:
[1] "Rating"
Number of terminal nodes:  4
Residual mean deviance:  0.2869 = 5641 / 19660
Misclassification error rate: 0.06479 = 1274 / 19663
```



```
node), split, n, deviance, yval, (yprob)
      * denotes terminal node

1) root 19663 18650.0 Yes ( 0.181814 0.818186 )
  2) Rating < 3.5 4515  5030.0 No ( 0.754817 0.245183 )
    4) Rating < 2.5 2051   720.1 No ( 0.957582 0.042418 ) *
    5) Rating > 2.5 2464  3343.0 No ( 0.586039 0.413961 ) *
  3) Rating > 3.5 15148  1838.0 Yes ( 0.011025 0.988975 )
    6) Rating < 4.5 4289  1274.0 Yes ( 0.034041 0.965959 ) *
    7) Rating > 4.5 10859   304.4 Yes ( 0.001934 0.998066 ) *
```

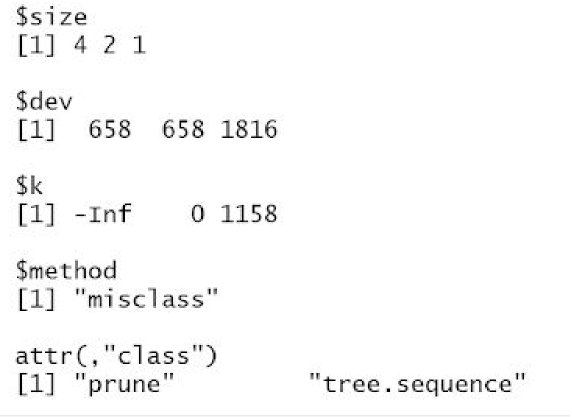


Here comes a tree with 4 terminal nodes, only rating is actually used in this tree. The training error rate is 0.06479.



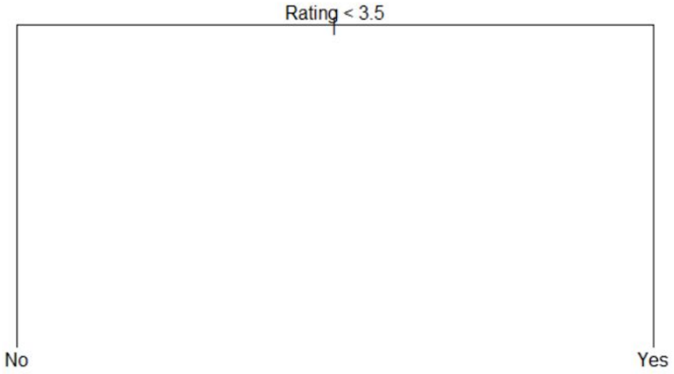
```
              RCMD.test
tree.pred   No  Yes
      No  1678  535
      Yes   81 7538
[1] 0.9336859
```

Fitting the tree in the testing dataset turns out an accuracy rate of 93.4%. But there is an unreasonable shortcoming of this tree. After the first branch differentiates ratings by 3.5, there is only one prediction for each branch. We can guess that criterions 2.5 and 4.5 are not actually functioning.



```
$size
[1] 4 2 1

$dev
[1]  658  658 1816

$k
[1] -Inf    0 1158

$method
[1] "misclass"

attr(,"class")
[1] "prune"         "tree.sequence"
```

To confirm the conjecture, cross-validation is performed to check if the tree can be further pruned. The result shows that trees with 4 and 2 terminal nodes correspond to the same cross-validation error rate, that is 622.



Application of the prune misclassification function suggests a simple two-node tree. The conclusion is: a product with rating lower 3.5 is not likely to be recommended.

Rating and Review Texts

In this section, we are interested in exploring what keywords in the review text represent that a product is satisfying and whether we can predict the rating score of the product based on content in the review.

Text Processing

1. Clean the Text Content

   Before applying machine learning models to our data, we have to deal with the text data first. Therefore, it is essential to clean the content of review texts for model preparation by two steps. We first split the review into lower-case words and utilize *nltk* to remove noise, such as punctuations, numbers, links, and stopwords(commonly used words of a language – is, am, the, of, in, etc). The next step is to normalize words, which is a pivot step for feature modeling with text as it converts the high dimensional features(N different features) to the low dimensional space(1 feature). For example – “look”, “looked”, “looks”, and “looking” are the different variations of the word – “look”. Though they mean different things, contextually they all are similar. There are two methods of lexicon normalization; Stemming or Lemmatization. In this report, we apply Lemmatization, as it will return the root form of each word, instead of just stripping suffixes. The below figure shows the comparison of review text before and after the cleaning process.

Before Cleaning Review Text	After Cleaning Review Text
I love, love, love this jumpsuit. it's fun, flirty, and fabulous! every time i wear it, i get nothing but great compliments!	love love love jumpsuit fun flirty fabulous every time wear get nothing great compliment
Very comfortable fabric and fits nicely. i am 5'10" like it states the model is, and it falls shorter on me that viewed, but still looks nice. it will be an easy dress to use for different looks.	comfortable fabric fit nicely like state model fall shorter viewed still look nice easy dress use different look

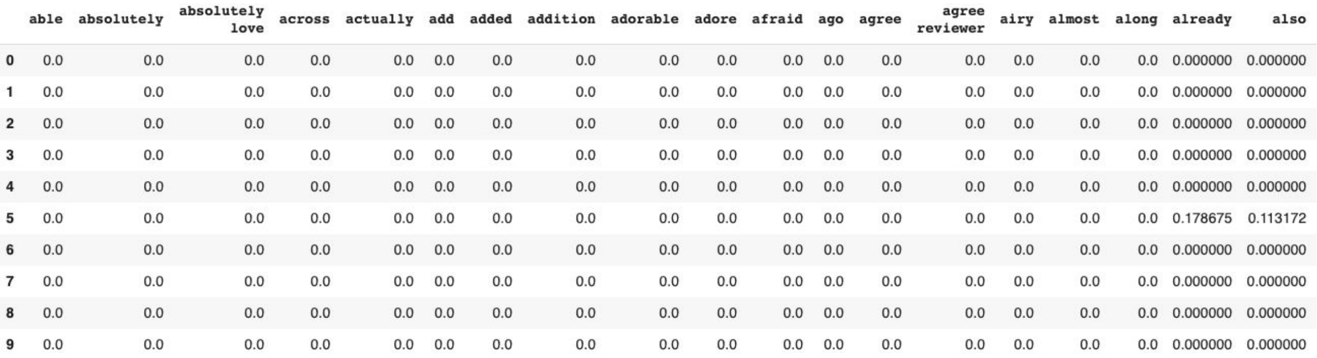
Figure3: Comparison of Cleaned Review Text

2. Vectorize Text Data

In order to analyze text data and fit into models, we need to turn text data into numerical features. There are various models for word embedding, including bag-of-words(BOW), Term Frequency-Inverse Document Frequency (TF-IDF), and Word2Vec, etc. In this report, we apply TF-IDF since it is the most common methods and also it is easy to conduct. TF-IDF is a weighted model commonly used for information retrieval problems. It aims to convert the text documents into vector models on the basis of the occurrence of words in the documents without taking considering the exact ordering.

- TF: Stands for Term Frequency, which measures how frequently a word occurs in a review and divided by the total number of words in that document.Since it is possible that a word would appear much more times in long reviews than shorter ones. Therefore, we need to divide by the review length for the word frequency, which can be thought of as similar to normalization.
- IDF: Stands for Inverse Document Frequency, which measures how important a word is. While computing TF, all words are considered equally important. However, for some certain words, such as "is", "of", and "that", may appear a lot of times but have little importance. Therefore, we need to weigh down the frequent words while scaling up the rare ones.

Scikit-learn provides two ways to vectorize text and get the TF-IDF weight matrix. One way is a two-part process of using the *CountVectorizer* to count how many times each word shows up in each review, followed by the *TfidfTransformer* to calculate the weights for each word and generate the weight matrix. The other does both steps in a single *TfidfVectorizer*. In this report, we choose to proceed with the first method. This method enables us to associate each word in a review with a number that represents how important each word is in that review. The below table is the TF-IDF score of the word in each review.



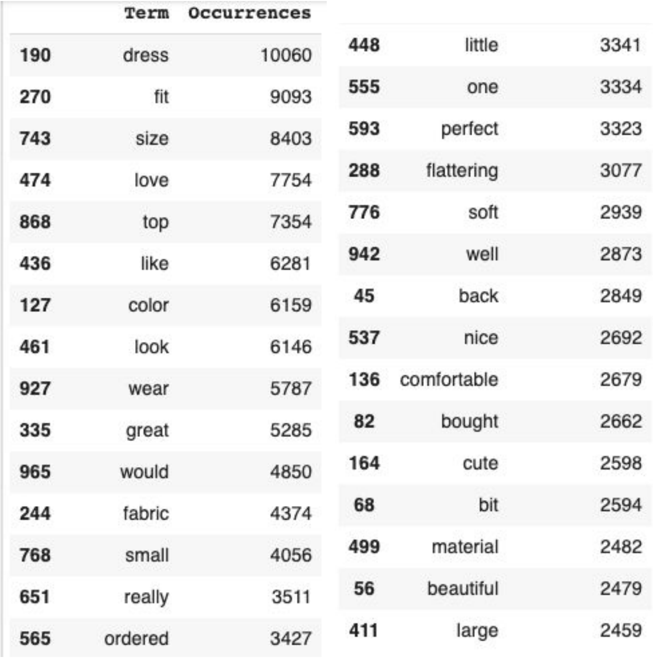
	able	absolutely	absolutely love	across	actually	add	added	addition	adorable	adore	afraid	ago	agree	agree reviewer	airy	almost	along	already	also
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.000000
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.000000
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.000000
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.000000
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.000000
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.178675	0.113172
6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.000000
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.000000
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.000000
9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.000000

Figure4: TF-IDF Weight Matrix

In conclusion, the higher the score, the more relevant that word is in that particular review. Also, reviews with similar, relevant words will have similar vectors, which is what we are looking for in a machine learning algorithm.

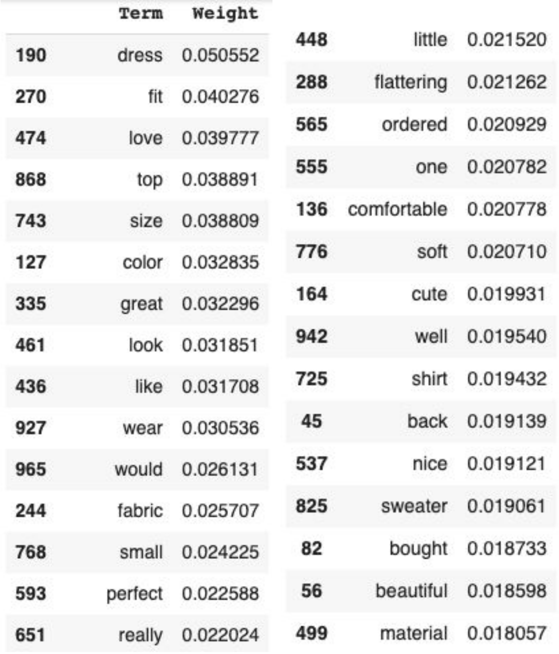
3. Review Text Exploration

From the below Figure5 and Figure6, We can see that most of the top 30 frequently occurred words are positive, such as love, great, perfect. And the top 30 weighted words are mostly similar to top frequently occurred words, only slightly different.



	Term	Occurrences
190	dress	10060
270	fit	9093
743	size	8403
474	love	7754
868	top	7354
436	like	6281
127	color	6159
461	look	6146
927	wear	5787
335	great	5285
965	would	4850
244	fabric	4374
768	small	4056
651	really	3511
565	ordered	3427
448	little	3341
555	one	3334
593	perfect	3323
288	flattering	3077
776	soft	2939
942	well	2873
45	back	2849
537	nice	2692
136	comfortable	2679
82	bought	2662
164	cute	2598
68	bit	2594
499	material	2482
56	beautiful	2479
411	large	2459

Figure5: Top 30 Frequently Occurred Words in the Review



	Term	Weight
190	dress	0.050552
270	fit	0.040276
474	love	0.039777
868	top	0.038891
743	size	0.038809
127	color	0.032835
335	great	0.032296
461	look	0.031851
436	like	0.031708
927	wear	0.030536
965	would	0.026131
244	fabric	0.025707
768	small	0.024225
593	perfect	0.022588
651	really	0.022024
448	little	0.021520
288	flattering	0.021262
565	ordered	0.020929
555	one	0.020782
136	comfortable	0.020778
776	soft	0.020710
164	cute	0.019931
942	well	0.019540
725	shirt	0.019432
45	back	0.019139
537	nice	0.019121
825	sweater	0.019061
82	bought	0.018733
56	beautiful	0.018598
499	material	0.018057

Figure6: Top 30 Frequently Weighted Occurred Words in the Review

Machine Learning

After vectorizing the text data, we can input the weight matrix into our machine learning models. In our dataset, there are five classes of rating scores, which would be a multiclass problem to classify those rating scores. We apply mainly seven different machine learning algorithm models to predict the rating based on the review, including Logistic Regression, Quadratic Discriminant Analysis (QDA), K-Nearest Neighbors(KNN), Support Vector Machine(SVM), Naive Bayes, Random Forest, and Boosted Decision Tree. Our main goal here is to compare the prediction accuracy of the above models and find out which model performances the best. Before applying models, we implement the one-hot encoding on the "Rating" variable, and we split the dataset into 70% training and 30% testing set. Since there are more numbers of higher ratings, we also deal with the imbalanced class in our models.

Below Figure7 shows the model accuracy by using testing data for each method. Comparing all models together, we can see that Gradient Boosted Decision Tree performances the best with model accuracy of about 61.5%.

- Comparing K-Nearest Neighbors and Radius-Based Nearest Neighbors, we find that Radius-Based Nearest Neighbors have a higher model accuracy with about 55.6% than KNN.
- Among Support Vector Machine models, we can see that the radial kernel performances the best with model accuracy of about 60%.
- Among Naive Bayes models, the accuracy of Multinomial Naive Bayes is far better than Gaussian Naive Bayes. The assumption of Gaussain Naive Bayes is that only simple assumptions could be used to specify the generative distribution for each label. On the other hand, Multinomial Naive Bayes assumes the features are generated from a simple multinomial distribution. Since the multinomial distribution describes the probability of observing counts among a number of categories, therefore, Multinomial Naive Bayes is more appropriate for features that represent counts or count rates, meaning it is more suitable for classification with word counts or frequencies for text classification.
- For Boosted Decision Tree, Gradient Boosted Decision Tree and Adaptive Boosted Decision Tree both performance better than Random Forest. Although the two boosted decision tree model accuracies are close, Gradient Boosting performs better than Adaptive Boosting. Since AdaBoost is sensitive to noisy data and outliers, resulting in some problems that it can be less susceptible to the overfitting problem than other learning algorithms.

Model		Model Accuracy
Name	Parameter Used	Testing Set
Logistic Regression	Inverse of regularization strength=0.01	57.3% (0.5734)
QDA		58.8% (0.5884)
KNN	Number of neighbors=5	50.4% (0.5036)
RadiusNeighbors	Range of parameter space=10	55.6% (0.5558)
SVM	Linear Kernel	55.1% (0.5511)
	Polynomial Kernel	59.5% (0.5953)
	Radial Kernel	60.0% (0.6004)
Naive Bayes	Gaussian Naive Bayes	31.9% (0.3198)
	Multinomial Naive Bayes	60.5% (0.6048)
Random Forest	Number of trees=500	46.7% (0.4671)
XGBoost	Number of trees=500 Maximum tree depth=5 Boosting learning rate=0.5	**61.5% (0.6150)**
AdaBoost	Number of trees=500 Boosting learning rate=0.5	60.5% (0.6050)

Figure7: Comparison of Model Accuracy(5 Classes)

However, the highest model accuracy is only 61.5% for using five classes. We would like to obtain a higher prediction accuracy, therefore we would like to further discover how to improve our model.

To better understand the frequently occurred words, we plot the word cloud for each rating.

- Rating = 4 and Rating = 5
  We can see that for rating equals to 4 and 5, most of the frequently used words in the review are positive, such as love, great, perfect, comfortable, beautiful.

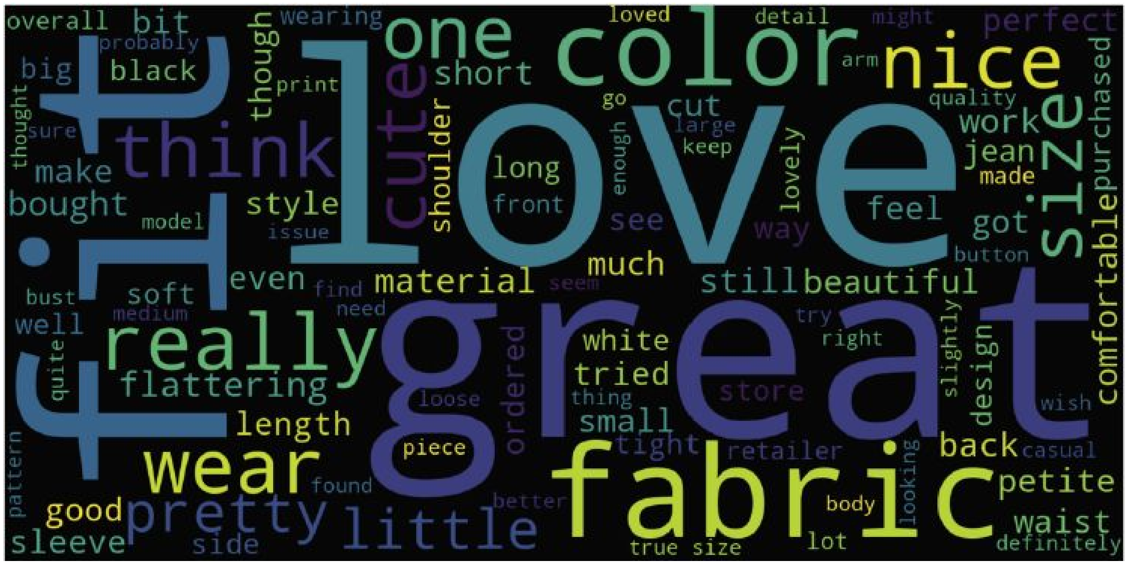


Figure8: Word Cloud for Rating = 4

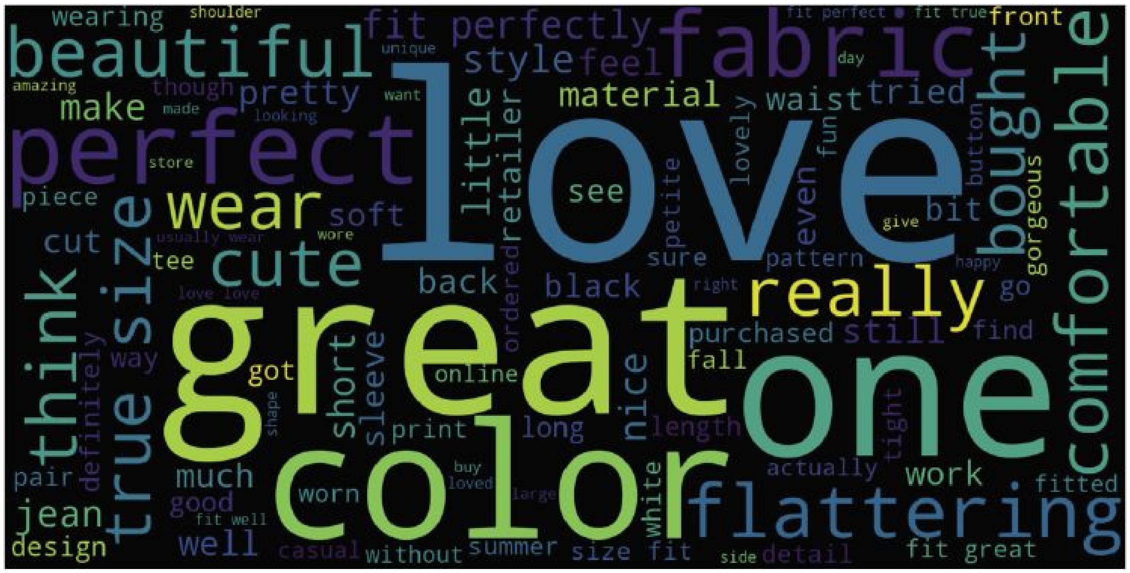


Figure9: Word Cloud for Rating = 5

- Rating = 3
  When the Rating Score is 3, top words are love, fabric, color, really, fit. However, there are not just positive words used in the review texts, but also negative words appear, such as disappointed, unfortunately.

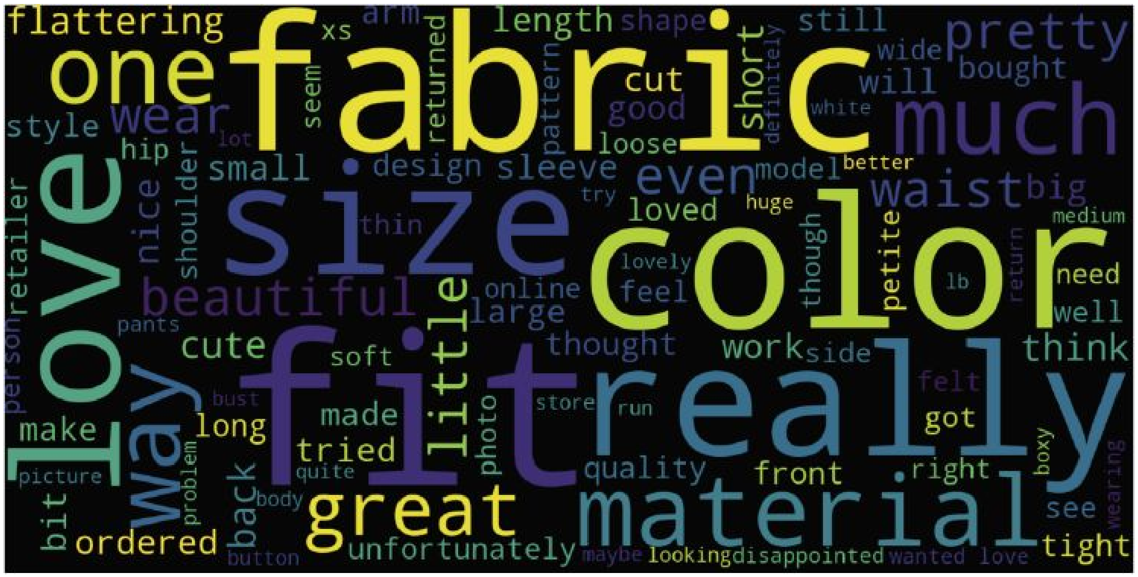


Figure10: Word Cloud for Rating = 3

- Rating = 1 and Rating = 2
  When the Rating Score is 1 or 2, there are more negative words than positive words used in the review texts, such as returned, cheap, unflattering, bad, disappointed, unfortunately.

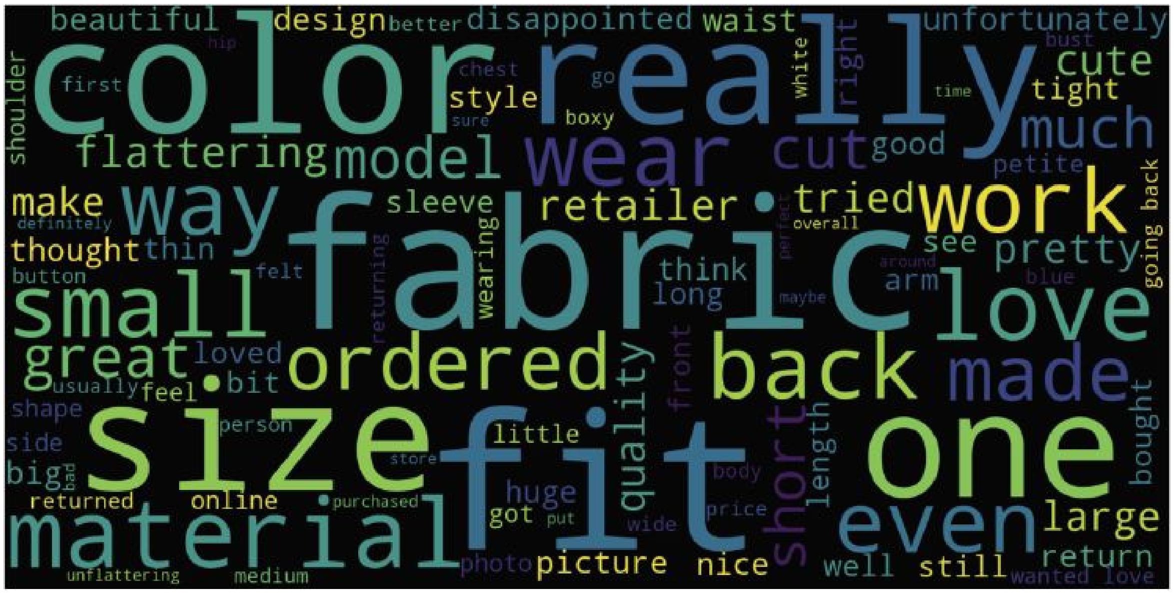


Figure11: Word Cloud for Rating = 2

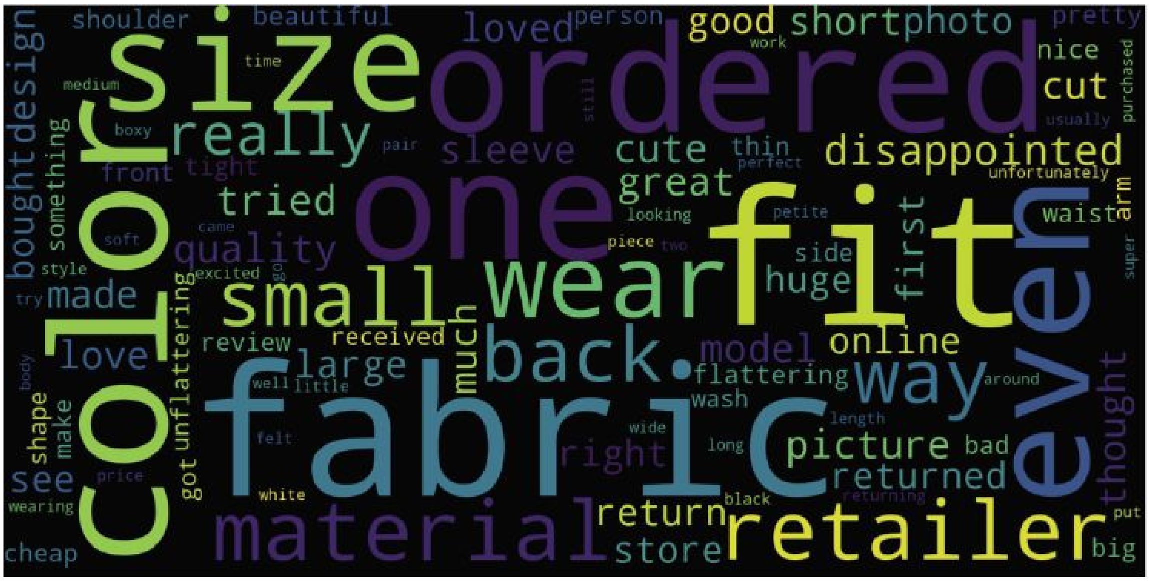


Figure12: Word Cloud for Rating = 1

From the word clouds, we can see that Rating 1 and 2 have similar frequent words in the review and Rating 4 and 5 are similar, which causes the difficulty to classify between Rating 1 and 2, also Rating 4 and 5. We think this may be the main reason to cause our previous model accuracy to be low.

Therefore, instead of using 5 classes, in the following section we only use 3 classes for prediction. Also, from the previous model results, we know that Radius-Based Nearest Neighbors, Support Vector Machine with radial kernel and Multinomial Naive Bayes perform better when comparing with its own other similar algorithms; thus, in this part, we exclude other similar models. We split the dataset into 70% training and 30% testing sets. Below Figure13 shows the model accuracy by using testing data for each method. When we use 3 classes for prediction, all model accuracy is enhanced and all are approximately higher than 75%. Comparing all models together, we can see that Random Forest performs the best with model accuracy of about 80.6%.

Model		Model Accuracy	
Name	Parameter Used	Training Set	Testing Set
Logistic Regression	Inverse of regularization strength=0.01	77.1% (0.7711)	74.9% (0.7492)
QDA		85.2% (0.8523)	78.6% (0.7862)
RadiusNeighbors	Range of parameter space=10	77.0% (0.7704)	77.0% (0.7701)

SVM	Radial Kernel	85.9% (0.8595)	80.2% (0.8021)
Naive Bayes	Multinomial Naive Bayes	79.7% (0.7970)	79.1% (0.7914)
Random Forest	Number of trees=1000	**88.1% (0.8813)**	**80.6% (0.8062)**
XGBoost	Number of trees=1000 Maximum tree depth=4 Learning rate=0.01	82.6% (0.8257)	79.5% (0.7954)
AdaBoost	Number of trees=1000 Learning rate=0.01	77.5% (0.7745)	77.3% (0.7727)

Figure13: Comparison of Model Accuracy(3 Classes)

Since the Random Forest model performs the best, we then plot the top 25 most important words for predicting rating from the Random Forest model. From Figure14, we can see that there are positive words and also negative words. Words such as size and fabric, suggest that customer care about the product quality and the product needs to fit well. Also, words such as comfortable reveal how the clothes feel from customers. In conclusion, those words stand an important part in predicting rating.

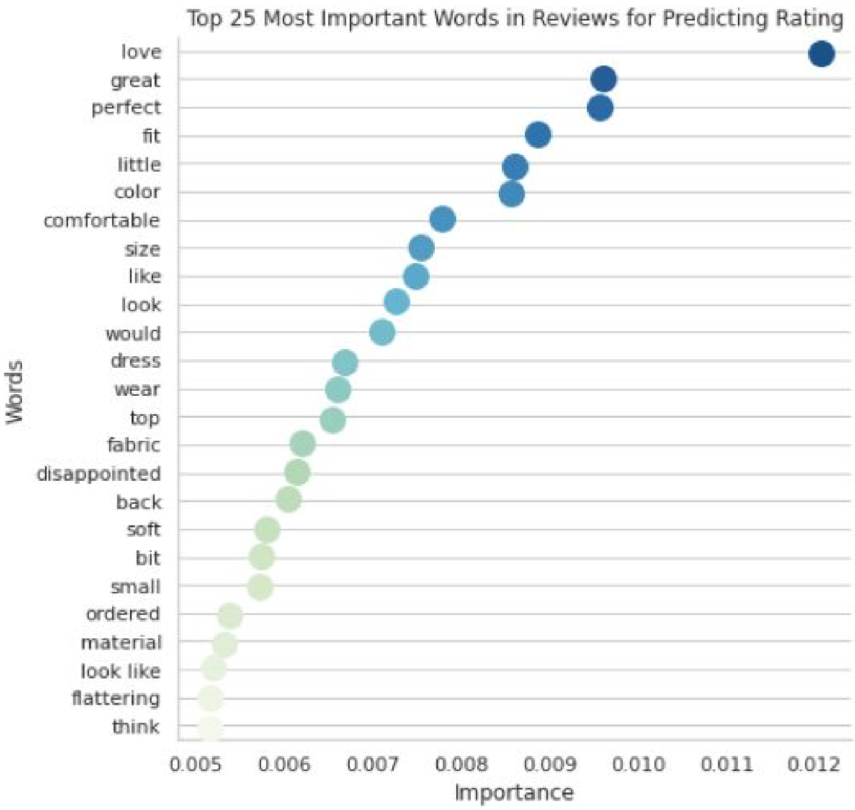


Figure13: Top 25 Most Important Words in Reviews for Rating Prediction

Conclusion and Remarks

In conclusion, the e-store we are analyzing has customers between age 30 to 50, they tend to give both positive and negative feedback more often than other age group customers(Figure14 and Figure15).

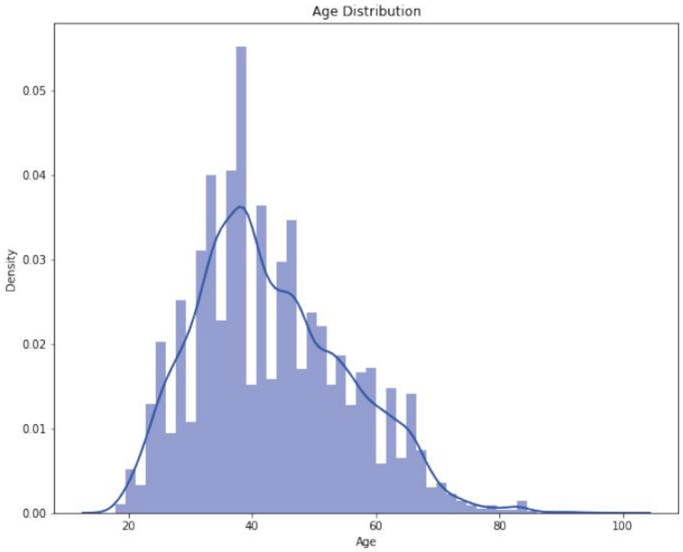


Figure14: Age Distribution

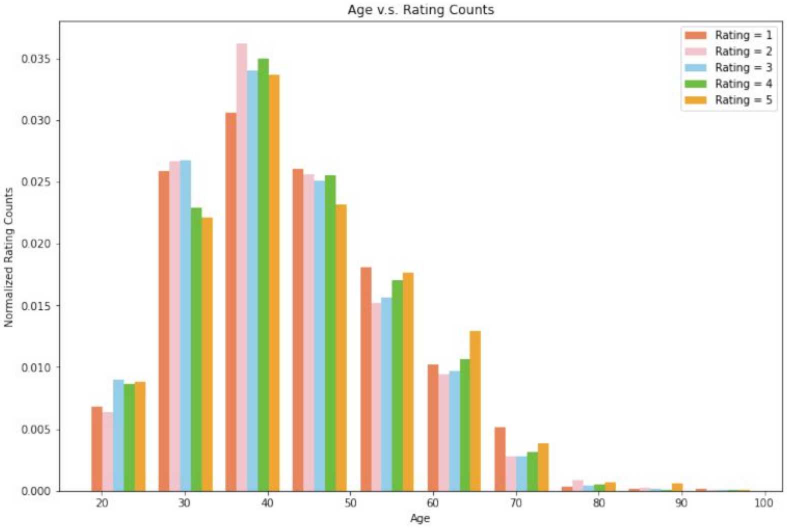


Figure15: Age and Rating Counts

From the Rating, Recommended and Review perspectives, the e-store receives more positive reviews than negative reviews, as well as there are more higher ratings and higher proportion of customers would choose to recommend products to others(Figure16).

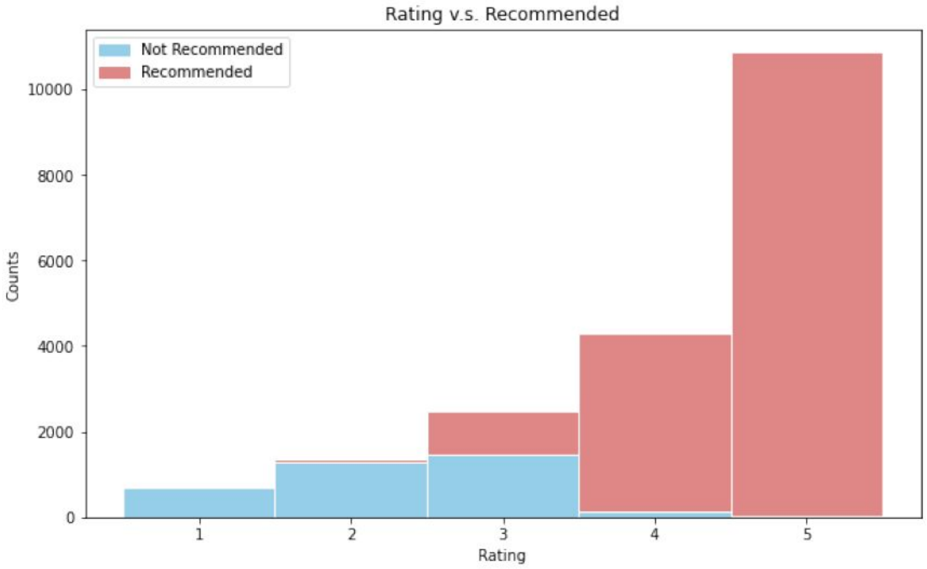


Figure16: Rating and Recommended

In order to boost the sales, we recommend the e-store owner to invest more on jackets, bottoms and intimate. Also, based on our prediction model of rating by reviews as well as keywords in reviews, the e-store owner should pay more attention to the clothing quality and size fit, which we believe customers care most about.

Contribution

Hsin-Yu Chen: Rating & Review Texts/ Conclusion and Remarks
Huang-Chin Yen: Introduction/Data Description and Preprocessing/ Expectations, Purpose, and Methods
Tian Tian: Age and Rating / Recommended or not?

References

https://www.statista.com/study/38340/ecommerce-report-fashion

https://scikit-learn.org/stable/modules/multiclass.html?fbclid=IwAR1G4Etrb66cZTk0IQsXwXAoLFUMuOCjfumPKQSwYt3oRVmV2htPONsng5A/

Packages used in Python:
Pandas / Numpy - For Data Manipulation
Seaborn / Matplotlib - For Data Visualization
Nltk - For Natural Language Processing
Sklearn - For Machine Learning